Informatics

Proper Abstractions for Digital Electronic Circuits: A Physically Guided Approach

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Dipl.-Ing. Dipl.-Ing. Jürgen Maier, BSc.
Matrikelnummer 00825749

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Ulrich Schmid

Diese Dissertation haben begutachtet:

Jorge Juan-Chico	Hussam Amrouch

Wien, 12. November 2021

Jürgen Maier

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

Proper Abstractions for Digital Electronic Circuits: A Physically Guided Approach

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Dipl.-Ing. Jürgen Maier, BSc.
Registration Number 00825749

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Ulrich Schmid

The dissertation has been reviewed by:

Jorge Juan-Chico

Hussam Amrouch

Vienna, 12th November, 2021

Jürgen Maier

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Dipl.-Ing. Jürgen Maier, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 12. November 2021

Jürgen Maier

Danksagung

In erster Linie möchte ich mich bei meinem Betreuer Ulrich Schmid recht herzlich für seine ausgezeichnete Führung und seine aktive Unterstützung im Laufe dieser Dissertation bedanken. Seine Tür war jederzeit geöffnet um meine verrückten und gewagten Thesen zu diskutieren. Ein ähnlicher Dank geht an Andreas Steininger, mit dem ich so manche spaßige Diskussion über unsere verrückten "Badewannenideen" geführt habe. Die aktive Mitarbeit und besonders die kritischen Kommentare von beiden haben mir erlaubt, meine wissenschaftlichen Fähigkeiten zu verbessern sowie als Person zu wachsen.

Mein Dank geht weiters an meine Co-Autoren und Arbeitskollegen an der Embedded Computing System Group, für sehr interessante und motivierende Diskussionen und Zusammenarbeiten, sei es in der Forschung oder der Lehre. Stellvertretend möchte ich hier Matthias Függer, Robert Najvirt, Thomas Nowak, Florian Huemer und Kyrill Winkler nennen. Ranus forever!

Zum Schluss möchte ich meiner Freundin Sandra von Herzen für ihre Unterstützung und ihr Verständnis, besonders in Zeiten großer Frustration und geringer Freizeit, danken.

Acknowledgements

First an foremost I would like to thank my advisor Ulrich Schmid for his guidance and active support throughout this thesis. His door was always open to discuss my crazy and adventurous ideas. Similarly, I want to thank Andreas Steiniger, with whom I had several joyful discussions about our lunatic “bathtub ideas”. The participation and critical comments of both enabled me to mature my scientific skills and to grow as a person.

My thank also goes to all my co-authors and co-workers at the Embedded Computing Systems group, for many fruitful discussions and collaborations, being it in research or in teaching. Exemplarily I would like to mention Matthias Függer, Robert Najvirt, Thomas Nowak, Florian Huemer and Kyrill Winkler. Ranus forever!

Last but not least I would like to thank my girlfriend Sandra for her support, especially during periods of big frustration or limited spare time.

Kurzfassung

Immense Fortschritte in der Verarbeitung von Halbleiter-Materialien in den letzten Jahrzehnten haben es ermöglicht, Transistoren immer weiter zu verkleinern und dadurch immer größere und schnellere digitale Schaltungen zu produzieren. Der daraus resultierende Anstieg der Komplexität hatte allerdings auch negative Auswirkungen, zum Beispiel auf die Verifikation: Obwohl es heutzutage möglich ist, die physikalischen Prozesse, die das Verhalten der Schaltung bestimmen, mittels hoch-präziser Modelle zu beschreiben, macht gerade die Größe und Komplexität dieser Modelle eine Simulation/Berechnung in vernünftiger Zeit undurchführbar. Eine mögliche Lösung besteht darin, Abstraktionen einzuführen. Deren Aufgabe ist es, den Verifikationsaufwand durch das Ausblenden gewisser Details zu verringern, ohne aber die Genauigkeit signifikant zu beeinträchtigen. Selbstredend ist die Entwicklung solcher Abstraktionen eine sehr anspruchsvolle Aufgabe: Zu wenig oder die falschen Informationen stellen ein unvollständiges Abbild der Realität dar, wohingegen exzessive Modelle sehr langsam sind.

Aus diesem Grund untersuchen und entwickeln wir in dieser Arbeit passende Abstraktionen für digitale elektrische Schaltungen. Wir sind der Meinung, dass optimale Resultate nur dann erzielt werden, wenn (i) das zugrundeliegende physikalische Verhalten verstanden und (ii) basierend auf den gewonnenen Einsichten passende abstrakte Modelle und Parameter ausgewählt werden. Dementsprechend wird die Entwicklung einer Abstraktion auf eine Beobachtung und eine nachfolgende Schlussfolgerung reduziert, sodass weder Annahmen noch Mutmaßungen notwendig sind. Anzumerken ist, dass die in dieser Arbeit präsentierten Modelle nicht dazu gedacht sind, existierende zu ersetzen, sondern lediglich eine Alternative zu hoch entwickelten (z.B. Differentialgleichungssystemen in analogen Simulationen) oder übermäßig vereinfachten Ansätzen (z.B. pure oder inertiale Verzögerungsmodelle in digitalen Simulationen) darstellen. Insgesamt zielen wir darauf ab, verlässliche Modelle, die einen großen Gültigkeitsbereich und eine hohe Genauigkeit haben, zu entwickeln, wobei der Simulationsaufwand gering gehalten werden soll.

Die Erreichung dieser Ziele erforderte das Studium folgender Modellierungsebenen:

1) Analoge Abstraktionen: Um das analoge Verhalten verschiedenster logischer Gatter in einer einfachen Art und Weise zu beschreiben, entwickeln wir neue Modelle basierend auf physikalisch inspirierten Transistormodellen. Obwohl diese vernünftige Resultate liefern, ist der zu investierende Aufwand zu hoch, um eine Evaluierung von großen Schaltungen zu erlauben. Daher sind weiter Vereinfachungen notwendig. Unter Zuhilfenahme von

analytischen Berechnungen und Fittings zielen wir auf mathematischen Funktionen ab, die eine Approximation der realen analogen Signale erlauben. Wir zeigen, dass eine steigende und eine fallende Trajektorie, die jeweils den gesamten erlaubten Spannungsbereich durchläuft, dafür ausreichen, da durch eine sorgfältige Kombination dieser beiden (im Detail die Addition von in der Zeit verschobenen Versionen) jede beobachtet Signalform approximiert werden kann. Wir sind überzeugt, dass unser Ansatz die Entwicklung eines analogen Simulationstools mit hoher Genauigkeit aber nur einem Bruchteil der Verifikationszeit (verglichen mit etablierten Methoden) ermöglicht.

2) Digitale Abstraktionen: Wir führen eine sorgfältige Analyse des Involution Delay Modells durch, das zur Zeit den einzigen Kandidaten für eine wirklichkeitsgetreue Abschätzung von Signalverzögerungen in digitalen Schaltungen darstellt, und erweitern es beträchtlich: Basierend auf physikalisch motivierten Überlegungen sind wir in der Lage, (i) etliche Mängel zu identifizieren, (ii) vernünftige Erklärungen für diese zu liefern sowie (iii) passende Erweiterungen zu entwickeln, die die aufgezeigten Probleme beheben. Im Detail sind das eine analytische Berechnung der verwendeten Verzögerungsfunktion, die Aufweichung verschiedenster Einschränkungen (führt zu einfacherer Anwendbarkeit), und die Erweiterung in Richtung Indeterminismus (resultiert in größerem Einsatzbereich). Mittels formaler Beweise und mathematischer Analysen zeigen wir, dass diese Änderungen die wesentlichen Eigenschaften des Modells nicht beeinträchtigen. Simulationen einfacher Schaltungen liefern erstmalig quantitative Resultate, die die höhere Genauigkeit verglichen mit anderen Methoden belegen. Damit wird ein fairer Vergleich zwischen etablierten Methoden und dem Involution Delay Modell möglich, der auch einen nicht zu vernachlässigenden, aber trotzdem vernünftigen, Mehraufwand gezeigt hat.

3) Unsere Ausführungen über analoge und digitale Abstraktionen werden durch eine detaillierte Analyse des Schmitt-Triggers vervollständigt, insbesondere in Hinblick auf dessen Anfälligkeit für Metastabilität (Spannungswerte in der Mitte des möglichen Spannungsbereiches bzw. verzögerte Transitionen). Wir stellen neue Methoden zur Charakterisierung des metastabilen Verhaltens vor und wenden diese auf drei modernen Implementierungen an. Die gewonnenen Erkenntnissen nutzen wir, um mit Hilfe analoger Simulationen zu zeigen, dass nahezu beliebige Trajektorien am Ausgang des Schmitt-Triggers durch eine präzise Steuerung des Einganges erzeugt werden können. Eine Kaskadierung, wie das z.B. auch mit Flip-Flops in Synchronizern gemacht wird, verbessert das Verhalten nur teilweise, da auch einige ungewollte Effekte hinzukommen. Insgesamt zeigen unsere Resultate allerdings, dass eine äußerst präzise Steuerung des Eingangssignals notwendig ist, um einen Schmitt-Trigger metastabil zu halten, was daher dementsprechend unwahrscheinlich in praktischen Anwendungen ist.

Aus den Antworten, die wir durch die Beantwortung dieser äußerst interessanten Forschungsfragen erhalten haben, können wir schlussfolgern, dass es keine “Zaubermethode” im Hinblick auf die abstrakte Modellierung gibt. Jede einzelne Abstraktion ist in gewisser Hinsicht einzigartig und bedarf einer sorgfältigen Analyse der dominanten physikalischen Prozesse. Nur so kann eine optimale Leistung, eine hohe Genauigkeit und eine breiter Anwendungsbereich erzielt werden.

Abstract

Over the last decades, major improvements in handling semiconductor materials led to a massive shrinkage of transistor sizes that, in turn, enabled engineers to realize larger and faster digital circuits. The resulting increase in complexity had, however, negative effects on verification: Although nowadays highly accurate models of the main physical processes, which govern the behavior of a circuit, are available, the size and complexity of these models makes it impossible to finish simulations/computations in reasonable time. One possible solution is to introduce abstractions, which have the goal to reduce the verification effort by hiding certain details while preserving accuracy. Naturally, developing proper abstractions is a very challenging task: Too little or the wrong information provide an incomplete picture while excessive models tend to be slow.

In this thesis, we, thus, study proper abstractions for digital electronic circuits. In our opinion, the best results are achieved by (i) understanding the underlying physical behaviors and (ii) picking appropriate abstract models and parameters based on the gained insights. This effectively reduces the task to observation and conclusion, so no assumptions or even guessing is required. Whereas the abstractions and models presented in this thesis are not meant to replace existing approaches, they provide an alternative in between highly sophisticated methods (e.g., ordinary differential equations in analog simulations) and overly simplified ones (e.g., digital models utilizing pure and inertial delays). Overall, we aim at achieving reliable models, which provide high coverage and accuracy at low verification efforts compared to existing approaches.

To achieve this goal, we thoroughly studied the following model domains:

1) Analog abstractions: To describe the analog behavior of various logic gates in a simplified fashion, we develop new models based on physically inspired basic transistor equations. Although these provide reasonably accurate results, the required effort is still too high for large-scale verification. Consequently, we employ further abstractions. Using analytic calculations and fittings, we aim at mathematical functions that allow an approximation of the analog waveforms. We show that unique rising and falling full-range switching waveforms provide a very good basis, since their proper combination (more specifically, the addition of time-shifted versions) is able to closely approximate every observed shape. We are convinced that our approach will enable the development of an analog simulation suite with high accuracy, which only needs a fraction of the verification time required for established analog simulation methods.

2) Digital abstractions: We thoroughly analyze and extend the Involution Delay Model, the only candidate for a faithful delay estimation method known so far. Based on physically guided considerations, we (i) identify several shortcomings, (ii) provide a proper explanation and (iii) develop improvements that remove the observed problems. More specifically, we show how to calculate delay functions analytically, relax certain restrictions that impaired easy applicability, and even introduce non-determinism to improve the model coverage. Formal proofs and deductions are used to show the correctness of our new abstractions. Simulations of simple circuits allow, for the first time, a quantitative evaluation of the superior accuracy and the not insignificant, but quite reasonable, overhead. This enables a fair comparison of the Involution Delay Model and state-of-the-art digital delay models.

3) We complement our efforts on analog and digital abstractions by an in-depth investigation of the Schmitt Trigger, in particular, its susceptibility to metastability (intermediary output values, late transitions). By introducing and using various novel methods, we are able to characterize the metastable behavior of this gate, i.e., when to expect which effects. Exploiting this knowledge, we show, based on analog simulations, how to generate an arbitrary output waveform in a common implementation by controlling the input accordingly. We also argue that cascading Schmitt Triggers, as it is done with Flip-Flops in a synchronizer, only improves the situation partially, as new undesired effects are added. Overall, our results, however, show that a very fine-grained control of the input is demanded to exploit metastable behavior in the Schmitt Trigger, making it very unlikely in normal operation.

From the answers we obtained by investigating these interesting research questions, we can conclude that there is no “silver bullet” w.r.t. modeling abstractions. Every approach is unique in some respect and thus requires a careful analysis of the governing physical behavior to achieve the optimal performance, accuracy and coverage.

Contents

CHAPTER 1

Introduction

The most basic components in modern circuit designs are *transistors*, as by careful arrangement any desired logic functionality can be realized [98]. To manufacture a transistor, proper processing of semiconductor materials, most and foremost Silicon, is indispensable. Over the last decades the respective methods were continuously refined which enabled massive improvements, in particular, miniaturization [80]. In consequence more functionality at higher speed and lower power consumption can be built on a single chip. The downside of this overall very encouraging development is the tremendous increase in complexity and hence in design and verification effort. This contradicts, however, the demands from customers and industry, which expect product updates in regular, if not decreasing, intervals. To close this ever increasing gap it is essential to develop methods that hide non-relevant details and thus allow the designer to focus on the problem at hand, i.e., to introduce abstractions.

Abstractions are able to improve both the design and/or the verification of a circuit. For the former, highly sophisticated tools aim to support the engineer by simplifying the design process. Some enable an abstract, high level description of the desired circuit while taking care of the synthesis steps. Others provide graphical support for low level tasks such as place & route or even run these autonomously. In the case of verification, abstractions are supposed to enable a proper, but simplified, description of the underlying hardware. These are essential for modern circuit designs, as measurements on fabricated devices are hardly possible due to the very limited time-to-market [47]. Consequently simulations have to predict the final behavior such that errors can be detected and corrected as early as possible in the design process.

The most accurate methods for analyzing a circuit are currently simulations in the analog domain, where the electric voltages and currents are described continuously both in time and value. For this purpose, analytic device models derived from the underlying physical processes are evaluated numerically in software. A very prominent example for the latter is the tool HSPICE by Synopsys. Not surprisingly with improving, i.e., shrinking, technologies, the transistor models had to be extended multiple times, as new physical effects surfaced. Supplementary new device layouts, such as Buried Oxide (BOX) [141], Silicon-on-Insulator (SOI) [139] or Silicon-on-Sapphire (SOS) [148], just to name a few, also triggered necessary adaptions. For this reason various transistor models are available nowadays for HSPICE.

With feature sizes approaching the atomic lattice constant of Silicon (≈ 0.5431 nm) [15], modeling a transistor at the physical level gets increasingly challenging. Nevertheless, quantum physics enabled researchers to explain the underlying phenomenons, such that the behavior, even on those small scales, is already very well understood and accurately modeled. The (often very large number of) parameters for a particular technology are in general provided by the manufacturer and therefore enable highly accurate simulations prior to fabrication. Naturally this comes at a cost: Solving the often hundreds of (ordinary differential) equations per transistor[1] is expensive and thus quickly becomes prohibitive with growing circuit size.

Larger digital circuits are hence predominantly described in the digital domain, which is still continuous in time but only features two logic values, LO and HI. If the analog waveform drops below the lower discretization voltage value V_{LO}, the digital signal becomes LO, if it exceeds the upper value V_{HI}, it turns HI. This leads to the infamous *digital logic*, with its zeros and ones, that everyone is familiar with.

Representing circuits in the digital domain not only simplifies their design, since *Boolean Logic* can be used, but also enhances their verification. For example, to trace the temporal evolution of a signal throughout a circuit, i.e., to run a *timing analysis*, it is sufficient to propagate the occurrence times of the jumps in the value domain, the *transitions*. In this fashion, the number of evaluation steps can be reduced significantly, such that the resulting *event-triggered simulations* are finally able to cover large circuits and long simulation times. The main challenge here is to provide reasonable propagation delay values, which are commonly extracted from extensive analog simulations carried out in advance. Prominent examples are the *Extended Current Source Model* (ECSM) by Cadence [33] or the *Complex Current Source Model* (CCSM) by Synopsys [25].

Different kinds of timing analyses are possible. While *static timing analysis* [59] solely considers the static delay of each gate to calculate path delays, *timing simulations* apply inputs and are thus able to identify more advanced effects like signal degradation or interference. Timing simulation hence needs to predict time and direction of a gate's output transitions based on the incoming ones. Multiple delay models have been developed for this purpose, with *pure* and *inertial* delay [150] being the most basic examples. Their rather simple mode of operation unfortunately leads to significant mispredictions, especially for short pulses. An improvement can be achieved by utilizing a delay function instead of a constant delay value, as is the case for, e.g., the Degradation Delay Model (DDM) [82] and the Involution Delay Model (IDM) [17]. Note that the authors of the IDM were able to show in [27], that all other existing delay models, including DDM, are not able to faithfully model certain kinds of circuits. Thus, the IDM represents currently the only candidate for a faithful delay model. Unfortunately, it is still at a very early development stage, meaning that, at the moment, it is properly

[1] State-of-the-art chip use several billion of these in a single design

defined for minimalistic single input-single output gates (buffers and inverters) only.

Picking suitable values for the discretization threshold voltages V_{LO} and V_{HI} is by no means trivial and actually has a big impact, since they determine if an analog trace is visible in the digital domain or not. In the reverse direction, digital transitions are often implicitly associated with steep rail-to-rail waveforms. Considering pulses that only reach an intermediate level, this is obviously not the case. In rare situations it is actually possible that arbitrary voltage values, even in between V_{LO} and V_{HI}, manifest themselves. Although affected circuits strive heavily to escape this *metastability* [53], it still may take an infinite amount of time to do so. This is even more severe, as metastable states are intrinsically invisible in the digital domain. Solely upon metastability resolution, late transitions may be detected.

One major cause for metastability is the violation of a memory element's setup and hold-time [53]. In these cases, the input changes at approximately the same time the memory tries to capture it, which leads to undefined behavior. Metastable behavior of the Flip-Flop has hence been well researched in the last decades. Nevertheless, also other, less investigated and more complex, circuits show undesired behavior. One remarkable example is the Schmitt Trigger, which is often believed to be immune to metastability, and thus used as filter in multiple occasions.

Although the digital abstraction clearly provides big benefits, the examples presented so far also show that they have to be deployed with great care, especially concerning information propagation. Masking potentially important data may cause an incorrect or incomplete signal description, such as the invisibility of metastability in the digital domain. On the other hand only absolutely necessary information should be added, as otherwise the abstraction looses its effectiveness. Thus identifying the crucial underlying properties that need to be preserved is one of the most challenging tasks. Luckily there exists, in our opinion, a simple and straightforward approach for deriving suitable abstractions: thorough investigation of the physical processes. Properly understanding the mechanisms that dominate the property that shall be modeled, i.e., which parts contribute in which fashion, provides detailed insights and thus eases the identification of both crucial and potentially neglectable parts.

1.1 Contributions

In this thesis, we will therefore utilize detailed physical knowledge to develop new abstractions and enhance existing ones. In a first step, the physical behavior, especially the interactions among components, is analyzed. The achieved results are then condensed such that the abstraction represents all necessary information while it still retains low complexity. Compared to approaches that solely use HSPICE data for fitting, methods developed in this fashion have the advantage that all parameters can be retraced directly to physics. In the best case, this allows generic adaptions in the case the underlying hardware changes.

In detail we ask the following research questions:

Q1) Is it possible to use simplified models to simulate and verify analog voltage values throughout large digital circuits?

Q2) Can we use dedicated knowledge of the circuit structure to enhance reliable delay estimation methods on a digital level regarding accuracy and applicability?

Q3) Can we reasonably capture metastable behavior of elaborate gates in simplified models and can the latter be easily characterized?

In the course of this thesis we will show that every question can be answered with *Yes*! The arguably most important theoretical result is the extension of the Involution Delay Model that (i) allows to use arbitrary discretization thresholds and (ii) provides a uniform digital representation for unique analog trajectories while faithfulness is fully retained. The most relevant practical results are the methods we developed to characterize the metastable behavior of a Schmitt Trigger. We not only determined the possible metastable input-output values (in fact infinitely many) but also derive a measure for resolution speed and time.

In order to guarantee simple reproducibility of the data gathered for this thesis, which is, in our opinion, a very important property, we published all the developed tools/frameworks under open source licenses. The links to the corresponding online repositories can be found within the description of the single tools.

1.2 Outline

This thesis is organized in the following fashion: In Chapter 2, we provide basic information about semiconductor materials, their physical properties and how they can be used to build modern transistors. This quite in-depth knowledge is required in the succeeding chapters to develop abstractions that match the behavior of real circuits. Be advised that due to the broad scope of the thesis a detailed state-of-the-art analysis of each research question will be presented separately in the respective chapter.

Chapter 3 focuses on research question Q1), i.e., modeling and describing the behavior in the analog domain. In sharp contrast to modern approaches, we present methods with reasonable accuracy but largely reduced complexity. In more detail, we use very simple non-linear transistor models to gain analytic descriptions of logic gates.

We also investigate approaches to approximate the switching waveform either by using analytic considerations or by fitting mathematical functions. For the latter, we try to answer the question how the fittings have to be combined to also cover non-optimal waveforms. The overall goal of this research is to predict the analog behavior based on very few parameters, which can be propagated within a circuit.

Parts of this work have been published in

[1] Chuchu Fan, Yu Meng, Jürgen Maier, Ezio Bartocci, Sayan Mitra, and Ulrich Schmid. „Verifying nonlinear analog and mixed-signal circuits with inputs“. In: *IFAC-PapersOnLine* 51.16 (2018). 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018. ISSN: 2405-8963

[2] Jürgen Maier. *Modeling the CMOS Inverter using Hybrid Systems.* Tech. rep. TUW-259633. E182 - Institut für Technische Informatik; Technische Universität Wien, 2017

Despite the difficulty for collaborative work to distinguish the individual contributions, it is fair to say that I was responsible for the development of the analog gate models and the integration in the simulation and verification tools. To support our simulations, we even created our own tool called MACS, which is publicly available on GitHub[2] under the *MIT*-license and allows to design circuits and evaluate the corresponding analog waveforms in MATLAB. An automatic export to multiple verification tools prevents errors and assures that exactly the circuit, that was simulated, is verified. The tool is based on simulations done in close cooperation with Amin Ben Sassi, albeit, I realized the implementation.

In Chapter 4, we address research question Q2) by conducting a thorough analysis and improvement of the IDM. Our research allows us to derive general statements about the characterization and the shape of the delay functions, to predict the impact of changes in the circuit and to come up with a first analytical description. Running simulations of more elaborate circuits reveals its easy applicability. The possibilities to choose arbitrary discretization thresholds and to add non-determinism further enhances the capabilities of the approach, since variations, due to for example aging, can be easily covered. The latter is in our opinion a very important ingredient for future circuit verifications as those uncertainties allow to verify a unit for longer operation times.

To simplify the application of the IDM we developed the InvTool, which is capable to simulate a circuit fully automatically in a popular digital simulation tool. The InvTool is actually less a distinct program but more a general suite that allows to analyze a given circuit: it reads user defined input specifications, performs simulations (analog and digital), evaluates the results using different metrics, and exports the results in a human-readable form.

This chapter is based mainly on the content published in

[3] J. Maier, M. Függer, T. Nowak, and U. Schmid. „Transistor-Level Analysis of Dynamic Delay Models“. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC).* May 2019

[4] Jürgen Maier, Daniel Öhlinger, Ulrich Schmid, Matthias Függer, and Thomas Nowak. „A Composable Glitch-Aware Delay Model“. In: *Proceedings of the 2021 on Great Lakes Symposium on VLSI.* GLSVLSI '21. Virtual Event, USA: Association for Computing Machinery, 2021. ISBN: 9781450383936

[5] M. Függer, J. Maier, R. Najvirt, T. Nowak, and U. Schmid. „A faithful binary circuit model with adversarial noise“. In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE).* Mar. 2018

[2] https://github.com/jmaier0/macs

[6] Jürgen Maier. „Gain and Pain of a Reliable Delay Model“. In: *2021 24th Euromicro Conference on Digital System Design (DSD)*. 2021

[7] Daniel Öhlinger, Jürgen Maier, Matthias Függer, and Ulrich Schmid. „The Involution Tool for Accurate Digital Timing and Power Analysis“. In: *Integration* 76 (2021). ISSN: 0167-9260

[8] D. Öhlinger, J. Maier, M. Függer, and U. Schmid. „The Involution Tool for Accurate Digital Timing and Power Analysis“. In: *2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. July 2019

For these publications, it is even harder to distribute the individual contributions. In [3] I did the main parts of the analytic considerations and experimental evaluations while in [4] I was mainly responsible to build the presented extensions. In [5] I developed, jointly with Ulrich Schmid, the fundamental basic concept and all the proofs. In the case of the InvTool I was mainly responsible for a first, primitive version, which was later largely extended by our student Daniel Öhlinger in the course of his bachelor thesis [19]. The final suite is now publicly available on GitHub[3] under GPL3 license. Besides being the main supervisor of the thesis I supported the proper characterization of the InvTool leading to the results published in [7] and [8].

The focus of Chapter 5 is to answer research question Q3), more specifically to investigate metastability in Schmitt Trigger (S/T) circuits, which are often said to be immune to metastability and thus used to clean signals. We are, however, able to show that even this gate can be driven in such undesired states, although it takes considerable more effort. Even worse; every possible analog intermediate value can be kept for an infinite amount of time. To characterize an implementation, which is important to (i) make comparisons and (ii) evaluate the resilience against metastability, we develop multiple methods based on extensive analog simulations.

This chapter is mainly based on the publications

[9] J. Maier and A. Steininger. „Efficient Metastability Characterization for Schmitt-Triggers“. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019

[10] J. Maier, C. Hartl-Nesic, and A. Steininger. *Comprehensive Characterization of Schmitt-Triggers*. submitted to TCAS I June’21

[11] A. Steininger, J. Maier, and R. Najvirt. „The Metastable Behavior of a Schmitt-Trigger“. In: *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2016

[12] A. Steininger, R. Najvirt, and J. Maier. „Does Cascading Schmitt-Trigger Stages Improve the Metastable Behavior?“ In: *2016 Euromicro Conference on Digital System Design (DSD)*. Aug. 2016

[3] https://github.com/oehlinscher/InvolutionTool

For [9] and [10] it is fair to say that I was the main designer of the different characterization methods and responsible for the implementation and evaluation. To speed up the characterization we created the fully automatic tool MEAT which executes the different approaches. It is publicly available on GitHub[4] under the *MIT*-license. For the publications [11] and [12], I was primary involved in general considerations regarding the behavior, experimental exploration and parts of the evaluation.

Chapter 6 shortly reviews some ideas we were not able to finish within this thesis, which are, however, interesting avenues for further research. Finally, the thesis is concluded in Chapter 7.

[4] https://github.com/jmaier0/meat

CHAPTER 2

Background

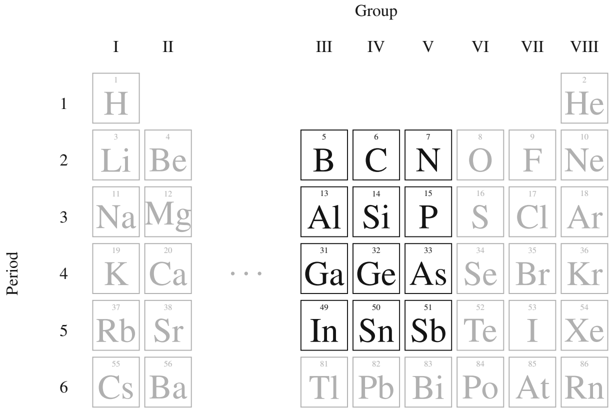
The goal of this thesis is to develop respectively enhance circuit abstractions based on physical considerations. For this task, a proper understanding of the underlying physical processes is mandatory. In this chapter, we will thus very briefly introduce semiconductors, their charge transport properties and the effects of introducing impurity atoms. Afterwards we investigate the internal structure and the resulting characteristics of different transistor implementations. These information will be later used to introduce transistor models, which allow us in turn to describe arbitrary logic gates.

Please note that all the content presented in this chapter has been gathered from various fantastic textbooks, e.g., by Kittel [86], Sze and Ng [73], Howe and Sodini [114], Hodges, Jackson, and Saleh [89], Streetman and Banerjee [30] or Tsividis and McAndrew [31]. Naturally our selection, regarding references and especially content, is very limited and represents just a tiny fraction of the available material. Interested readers are thus referred to the excellent literature for further information.

2.1 Semiconductors

In modern electronic circuits all devices are, in general, built on top of a single chunk of semiconductor. In this case the circuit is denoted as Integrated Circuit (IC). The main benefit, compared to separated components that are finally assembled, is the higher integration density, which enables a reduced power consumption and increased operation speed[1]. The bare semiconductor chunk the IC is built on is called *die*, whereat their combination, i.e., after the circuit has been implemented, is called *chip*. A single die, in turn, is cut out of a bigger piece, a round disc called *wafer*, which is structured at once during manufacturing. Thus in a single of the many steps required to build an IC, e.g., photolitograhpy, etching and implantation, multiple dies are structured in parallel.

[1] In the last years, actually a trend in the opposite direction has developed, with the goal to make circuits more flexible.



Group

Period	I	II		III	IV	V	VI	VII	VIII
1	1 H								2 He
2	3 Li	4 Be		5 B	6 C	7 N	8 O	9 F	10 Ne
3	11 Na	12 Mg		13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
4	19 K	20 Ca	. . .	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
5	37 Rb	38 Sr		49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
6	55 Cs	56 Ba		81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn

Figure 2.1: Part of the periodic table highlighting the most important groups III, IV and V for electronic circuits. Most common is Si, however also III-V compounds such as GaAs or InSb are utilized.

For the wafer, also called *bulk* in circuit design, semiconductor materials are used, for example the very popular Silicon (Si). It has several advantages, first and foremost its broad availability, as it can be gathered from quartz sand. Actually, around 30 % of the earths crust is made of Silicon, making it, behind Oxygen, the second most common element [81]. The question we want to address in the sequel is: What are the unique properties of a semiconductor material that are beneficial for ICs? Its name already indicates some relationship to conductors, but what is the actual difference? To answer these questions, we have to exploit the material properties regarding charge transport, and, in order to achieve that, we have to investigate the internal organization of solids.

2.1.1 Lattice Structure

Despite being called *atom*, which is derived from the Greek word *atomos* meaning “not cuttable” [55], it is possible to split the previously believed smallest particle into even smaller components. The atomic core contains protons and neutrons, while the much smaller electrons rotate in so-called *shells* around it. Protons and electrons carry an electric charge of opposite sign such that their count has to be equal in a neutral atom.

To form solid matter lots of atoms, not necessarily of the same kind, assemble by forming bonds among each other. The driving force behind this process is the reduction of the atomic energy, since lower values are preferred. For this reason, atoms are thriving to fill vacant spots in the outermost electron shell, e.g., by sharing electrons with the neighboring atoms. This bonding, called covalent bonding [152], can be used to explain the lattice structure of semiconductors: Assume that atom A has four out of eight spots in its outermost shell filled by electrons. In the periodic table shown in Figure 2.1 such elements are located in group IV. Atom A can share each of these four electrons with another atom. In return each of these “bonded” atoms also shares an electron with atom A such that it overall ends up with eight outer electrons, filling the shell completely.

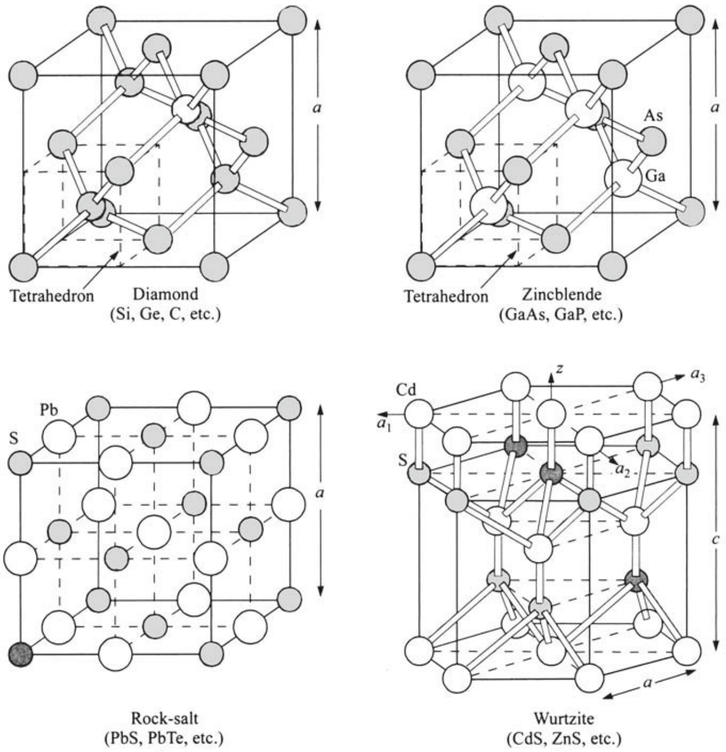


Figure 2.2: Primitive cells of various lattice structures. a denotes the lattice constant. For the Wurtzite structure the lattice constants differ between spatial directions. Taken from [73, p. 9].

Covalent bonding can be observed for group IV elements, such as Silicon (Si) or Germanium (Ge). It is, however, also encountered in materials consisting of elements from group III and V in the periodic table (also called III-V semiconductors). Common examples for electronic devices are GaAs, InSb, AlSb, or even more complicated ones, like $\mathrm{Al_xGa_{1\text{-}x}As}$ and $\mathrm{InAs_xSb_{1\text{-}x}}$ with $x \in [0,1]$ denoting the material ratio. Thus, for electronic circuits, group III, IV and V elements (cp. Figure 2.1) are the most important ones. Please note that in all cases each single atoms forms in total four bonds; for III-V semiconductors group III atoms attach to four group V ones and vice versa. A very important aspect, especially when analyzing the condensed matter, is the position of the bond partners in space, as this influences its characteristics a lot. Different structures can be observed in nature such as rock-salt, wurtzite and zincblende/diamond (see Figure 2.2), whereat the latest are the ones that can be found in the semiconductors we are considering. A common figure of merit, in this regard, is the lattice constant a, which represents the size of a unit cell, i.e., a three dimensional unit that, if shifted by multiples of a in all spatial directions, recreates the lattice. Just to give you an intuition: $a_{Si} \approx 0.5431\,\mathrm{nm}$ while $a_{GaSb} \approx 0.609\,593\,\mathrm{nm}$ [15].

2.1.2 Charge Transport

The lattice structure has a big impact on the charge transport mechanisms in a solid, which determine how well electrons can propagate through the material. This becomes obvious when the crystal lattice is rotated in different spatial directions. For certain constellations, one can see right through the whole material, while for others vision is completely blocked. Similar to the light, also the electrons experience differing resistance, with some directions being more favorable than others.

To derive analytic results, simple quantum electronic considerations are required. Using this formalism, combined with the periodic nature of the lattice, it is possible to determine the achievable electron energies as a function of the wave vector k, which represents the wavelength and its direction. The achieved results differ significantly among materials. Conductors impose no constraints, which enables electrons to change their energy gradually. Semiconductors, on the contrary, define distinct energy bands (the Conduction Band (CB) at higher energies and the Valence Band (VB) at lower energies), which are separated by the *band gap*. Electrons can only achieve energy values inside the distinct bands but not in between, whereat only those in the CB can contribute to charge transport. This has severe impacts on the material's electrical characteristics: A certain amount of energy, either provided by light or by heating the material, is required to lift electrons to the upper band. As a result, at very low temperatures no charge carrier transport is possible. With increasing temperature, the conductivity increases, whereat the degree of improvement depends on the size of the band gap E_g. The latter is actually what distinguishes a semiconductor and an insulator, like SiO_2. Figure 2.3 shows the band diagram for two semiconductor materials whereat Table 2.1 lists some band gaps.

2.2 Doping

On a small scale, electric current is achieved by moving charge carriers, in detail electrons. The macroscopic current can thus be defined as the charge transfer per time unit, i.e., $I = \frac{Q}{t}$, where t denotes the time frame, and Q the respective charge crossing a surface (for example the cross section of a wire) during that time. By counting the number of electrons, whereat each has a specific charge q, Q, and in consequence the current, can be determined. Note that electrons, which are detached from an atom, can move with much higher mobility through the material, which in turn also increases the conductivity. In conductors, e.g., metals, this is easily done, since the initial bounding of electrons to a core is very weak. In semiconductors, however, the electrons are required to literally bond the atoms together (cp. Section 2.1). Thus considerable effort is required to detach those electrons.

2.2.1 Electrons and Holes

When an electron is freed from its respective core it creates a vacancy that weakens the bond. In consequence other electrons, for example ones bound at neighboring atoms, are attracted to fill the vacancy. In this case the transferred electron leaves a vacancy at

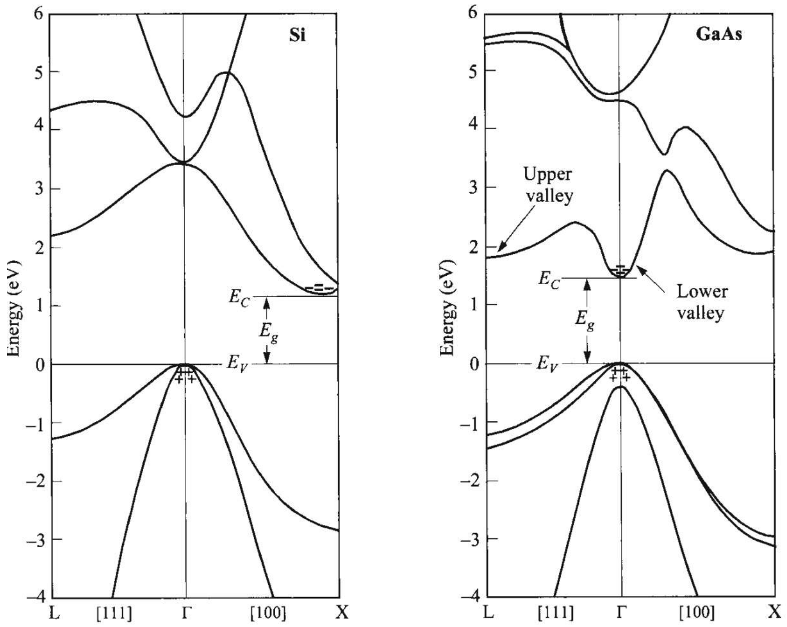


Figure 2.3: Band diagram for Silicon (left) and GaAs (right) showing the CB (top) and VB (bottom). [100] *and* [111] *correspond to two different directions of the wave vector k. Taken from [73, p. 14]*

its original position, which can again be filled by an electron. Overall, this results also in a charge carrier movement, as electrons hop from one atom to the next. Describing this movement is, however, hard, since multiple particles have to be traced. Actually, it is much easier to model the moving vacancy, the so-called *hole*. Although being just a vacancy, the hole can be interpreted as a moving particle, comparable to the electron. The main differences are, that holes have lower mobility and propagate in an electric field in the opposite direction, forming a separate way of charge carrying. Please note that in general both methods are available, whereat in most cases one is the dominant one. Its respective charge carriers are called the majority, while the other ones are denoted as the minority charge carriers. Whether electrons or holes represent the majority depends on the material and especially the doping, which we will consider in a minute.

In semiconductors, freeing an electron from its bonds is actually equivalent to raising it from the Valence Band to the Conduction Band. Energy and impulse can be in-/decreased but have to be preserved, e.g., by interaction with light particles (photons) or lattice vibrations (phonons). Transferring an electron to the CB creates a hole in the VB, thus this procedure is called electron-hole pair generation, while the reverse case is called electron-hole pair recombination. In this regard, please keep in mind that electrons always tend to reach the lowest possible energy while holes thrive for the highest one.

One of the mechanisms that we mentioned to increase the electron energy are interactions with phonons, i.e., collisions with the vibrating atomic cores. An abstract

Material	band gap E_g [eV]
Silicon (Si)	1.12
Germanium (Ge)	0.661
Gallium Arsenide (GaAs)	1.424
Indium Antimonide (InSb)	0.17
Silicon dioxide (SiO_2)	7.5 - 11.15 [140]

Table 2.1: Band gap of various materials at temperature $T = 300\,\mathrm{K}$. Semiconductors have much lower band gaps compared to an insulator like SiO_2. For the latter the internal structure has a huge impact, explaining the large variations. If not stated otherwise, data taken from [15].

description of these vibrations is actually the temperature T, whereat higher values represent stronger vibrations and thus higher energy. Therefore, the temperature can be used to determine the amount of electron/holes for a specific energy value E. For this purpose, we first need a probability distribution, which states how probable it is to find a particle at a specific energy. One possibility is the Fermi-Dirac statistic in the form

$$f(E) = \left(e^{\frac{E-E_F}{k_B \cdot T}} + 1 \right)^{-1},$$

whereat E_F denotes the *Fermi Energy*. Its physical meaning is that, at temperature $T = 0$, causing $f(E)$ to become a Heaviside jump at $E = E_F$, all allowed energy levels up to E_F are filled, while those above are vacant. With increasing temperature $f(E)$ gets shallower such that probabilities of $f(E)|_{E>E_F} > 0$ are achieved. Please note that the higher the temperature the more probable higher energies can be reached. The overall amount of electrons available for carrier transport is finally achieved by multiplying the density of states $d(E)$, denoting how many states per energy level are possible, with $f(E)$ and integrating over all energies. Holes are calculated in the same fashion with the difference that $1 - f(E)$ is used as multiplicant. In an *intrinsic* material, which solely contains atoms of the desired elements (e.g., Si in a Silicon bulk, Gallium and Arsenic in GaAs)[2], the amounts of electrons and holes have to be equal, since they are always created in pairs. In the model this is achieved by properly adjusting E_F. Finally note that $d(E)$ is not the same in the valence and conduction band, such that the Fermi Energy E_F actually depends on T.

2.2.2 Breaking the Symmetry

For useful electronic devices, we need to break the symmetry between electrons and holes in a material. This has lots of useful implications that we will investigate in the remainder

[2] This perfect situation can never be achieved in reality. Therefore, also materials with a negligible amount of foreign atoms are called intrinsic.

of this chapter. For now, let us focus on the process used to create such an imbalance, *doping*. In detail, a certain amount of impurity atoms, belonging to an element not part of the material, are implanted on purpose. These then replace the original atoms in the lattice and thus change the material properties substantially. For Silicon, member of group IV in the periodic system and thus having four outer electrons, two forms of doping are possible: replacing an Si atom by an element of group III (with one less outer electron) or of group V (with one additional electron)[3]. The former is called positive doping (p-type) and is often done using Boron (B), while the latter is called negative doping (n-type) and often nitrogen (N) or phosphorus (P) is used for this purpose.

What impact does doping have? Obviously, the impurity atoms do not have the correct number of electrons to form the original bonds. Either there is an electron missing, leading effectively to a hole, or the bond got an excessive electron, which is only loosely bound. In the band diagram this leads to additional energy levels, which are inside the band gap, either very close to the VB (p-type) or CB (n-type), as shown in Figure 2.4. Already at very low temperatures a lot of electrons in a p-type material transition from the Valence Band to the intermediate energy level, creating holes that are in this case the majority charge carriers. For n-type material the excessive electrons in the intermediate energy levels quickly transition to the Conduction Band, while no holes in the VB are created. In this case the holes are thus the minority charge carriers.

Since atoms are heavily displaced during doping, a lot of disturbances are created in the crystal throughout the process. In general, the introduced atoms have a differing lattice constant resulting in stressed, i.e., either compressed or stretched, atomic bonds. Also the implantation itself is a very disruptive process. One possibility is to highly accelerate the dopants and shoot them into the material, which partly destroys the lattice structure. To heal at least some of the induced damage, the now doped material is heated in a succeeding step.

2.3 p-n Junction

Directly connecting p- and n-type material, i.e., creating a p-n junction, results in one of the most basic devices, the diode. It allows current flow only in one direction, whereat the strength increases exponentially when a certain voltage value is exceeded. We are going investigate the p-n junction in greater detail in the following, as we will encounter it also when analyzing the transistor.

Let us first investigate the consequences of connecting n- and p-type material. As we have seen, the former has lots of electrons in the CB and the latter lots of holes in the VB. Upon contact, one might expect that electrons and holes recombine until an intrinsic semiconductor is achieved. Despite sounding very reasonable at a first glance, this happens only for a very limited amount of time. At closer observation, we see that electrons, which propagate from the n- to the p-type material and then recombine, leave a positively charged atom behind. Conversely, the atoms originally missing an electron,

[3] There are actually more possibilities, like replacing it with another group IV element. These are, however, out of scope for this thesis.

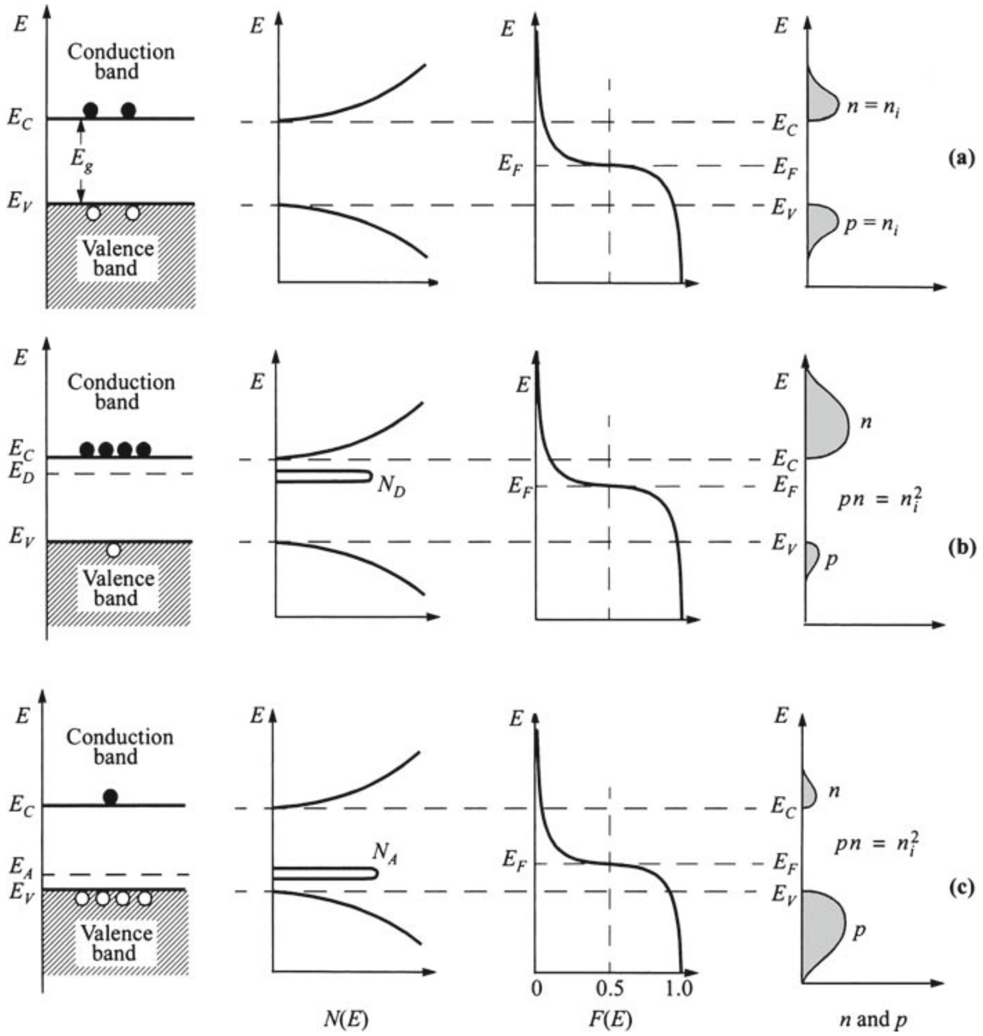


Figure 2.4: Band diagram for an (a) intrinsic, (b) n-type doped and (c) p-type doped semiconductor. $N(E)$ shows the density of states and $F(E)$ the distribution function. Please note that the product of electrons and holes is constant in all cases, i.e, $n \cdot p = n_i^2$. Taken from [73, p. 24]

i.e., which posses a hole, are finally negatively charged. If no outer potential is present no fresh majority carriers are provided to compensate the loss. This way an electric field builds up near the interface which prevents further charge transfer when the Fermi levels of both materials are aligned, as is shown in Figure 2.5 (d).

The width of the region of positively and negatively charged atoms, called the Space Charge Region (SCR) [cp. Figure 2.5 (a)], depends heavily on the doping concentration. As stated before, electrons and holes combine until the Fermi levels are aligned, i.e., the potential difference is compensated. The latter can be calculated as the integral over the electric field, which in turn is simply the integral over all charged atoms. If there are many dopants, the electric field builds up quickly resulting in a short SCR, but high electric fields. For low doping wider depletion regions and lower field strengths are observed. Please note that the materials influence each other in this regard, as a poorly doped material on one side also reduces the required field strength in the other. This can be clearly observed in Figure 2.5 (b), where the majority of the potential difference Ψ_{bi} is achieved in the n-type material [cp. Figure 2.5 (c)]. In this case, the SCR reaches a long distance in the poorly doped material and only little into the strongly doped one.

Investigating the impact on the band diagrams shown in Figure 2.5 (d) provides further insight. Recall, that electrons always tend towards the lowest energy, comparable to stones in the water. By bending the energy band upwards, like seen by the electrons in the n-type material, a barrier is formed that blocks them from reaching the p-type material. Conversely, holes, which are like bubbles in a liquid, tend towards higher energy levels. For them, the downwards bending from p- to n-type material forms a barrier as well. Therefore no current flow is possible. By applying a voltage to the p-type material, the energy bands are shifted against each other, such that the Fermi levels are no longer aligned. If the voltage is applied in forward direction (positive value), the p-type end of the diode is pushed towards lower energies, which reduces the barrier height for electrons and holes, and thus leads to an exponential increase in current.

If the voltage is applied in the reverse direction, the barrier grows steadily. In this state, only very little current is conducted, which actually results from stochastic electron hole pair generation processes. With growing potential difference also the SCR extends, until the terminals are hit. Naively one could suspect this behavior to continue infinitely. However, above the *breakthrough voltage* a vast increase in current can be observed. To understand this phenomena quantum mechanical considerations are necessary. In a nutshell, the decreasing barrier thickness increases the chance of electrons tunneling right through the barrier, something that is not possible in the macroscopic world.

2.4 Bipolar Transistor

Finally, we have everything at our disposal to describe the behavior of the transistor, a device that can, in the simplest case, be considered as a voltage respectively current controlled adjustable conductor. Over the past decades different implementations have been developed, whereat we will start with the one that was historically used first in an actual circuit, the bipolar transistor.

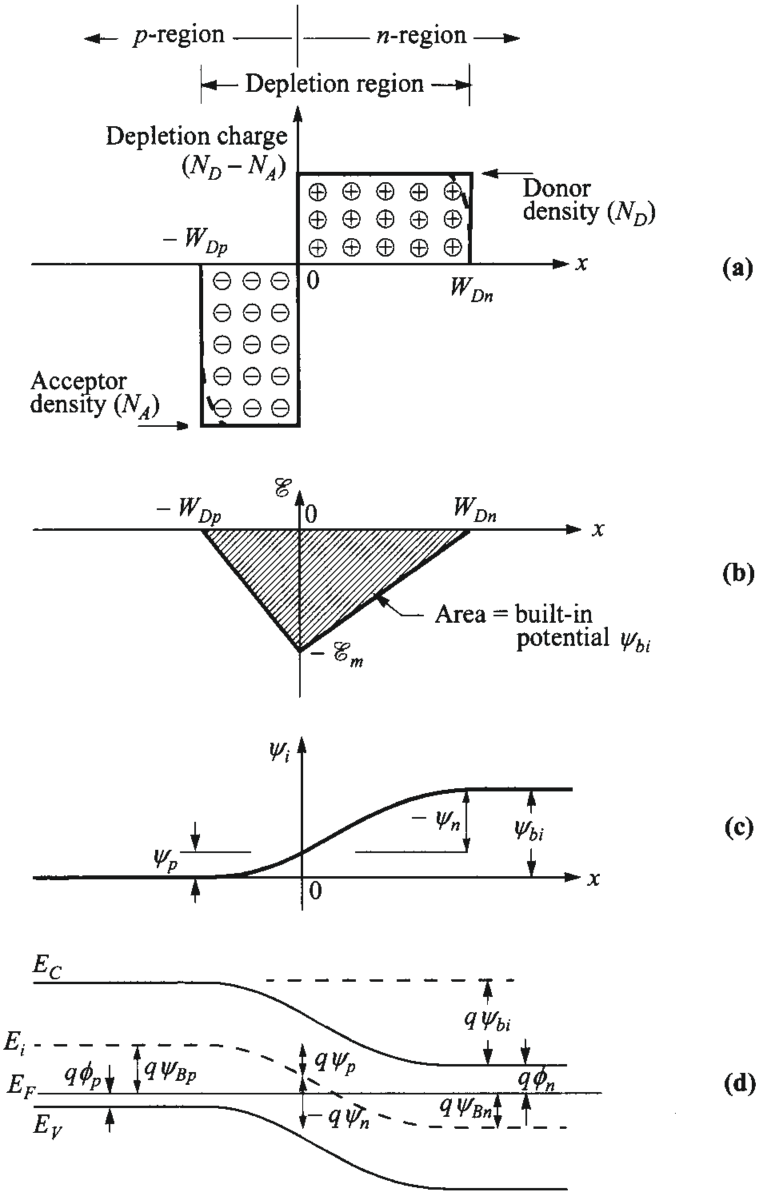


Figure 2.5: (a) Charge distribution, (b) electric field, (c) potential distribution and (d) band diagram of a p-n junction. Please note the asymmetries due to differing doping of p- and n-material. In this case the former is higher doped. Taken from [73, p. 80]

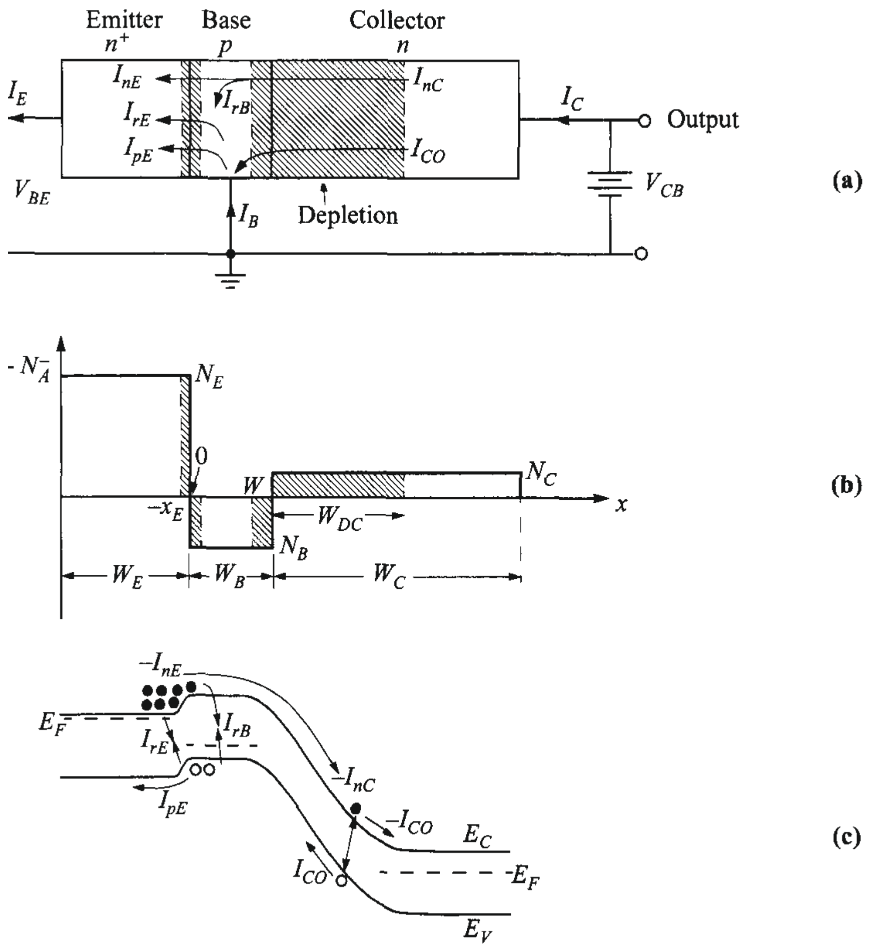


Figure 2.6: (a) Internal Structure, (b) doping profile and (c) band diagram of a bipolar transistor. The base-collector diode used in reverse direction collects (most of) the charge carriers injected by the emitter-base diode, which is driven in forward direction. Taken from [73, p. 246]

Internally, the bipolar transistor consists of nothing more than two diodes in series, as is shown in Figure 2.6. Both an NPN or a PNP stack, where N and P denote the doping type of the single layers, are possible. Independent of the chosen stack structure, the outer layers are called *emitter* (E) and *collector* (C), while the inner one is called *base* (B). Depending on the potential differences between base and emitter, respectively base and collector, the corresponding diodes are driven in forward or backward direction.

Only if the two diodes operate in differing directions charge transport between emitter and collector is possible. Figure 2.6 shows an NPN implementation with the emitter-base diode in forward and the collector-base diode in backward direction. In this setup, electrons are inserted into the base by the emitter. Due to the strong electric field, which is created by the collector-base diode in backward direction, these are immediately sucked to the collector.

The mode of operation of a bipolar transistor has several undesired side effects. Firstly, some of the electrons injected by the emitter recombine before they reach the collector. Secondly, holes are able to propagate from base to emitter since the emitter-base diode is used in forward direction, resulting in a leakage current that increases the power consumption. Considering this base current I_B, it is actually possible to describe the emitter-collector current as a multiple of I_B, i.e., that the transistor is a current amplifier. Note that various countermeasures have been developed to reduce these parasitic effects. One of them includes the utilization of more sophisticated doping profiles, as is shown in Figure 2.6 (b). Due to this unfavorable properties, bipolar transistors are currently mainly used in analog circuits where a high speed of operation is desired.

2.5 Field Effect Transistor

The majority of transistors that are used in modern circuit designs are a Field Effect Transistor (FET). Actually, such a device has already been proposed in the 1920s, however, the technology to implement it was not yet available at that time. In contrast to the bipolar transistor, where the base current is amplified, the FET uses an electric field to change the conductivity of the device, and thus is more power efficient.

2.5.1 Mode of Operation

The internal structure of the FET, shown in Figure 2.7, is very similar to the bipolar transistor. Again a PNP resp. NPN stack is used, with the difference that the outer contacts are now denoted by *source* (S) and *drain* (D), and the middle layer is controlled via the *bulk* (B) terminal. In addition, a fourth contact, the *gate* (G), is added, which sits on top of the middle layer but is separated from the whole device by a thin insulator.

In contrast to the bipolar transistor, the bulk terminal is used to ensure that both p-n junctions operate in reverse direction at all times. The question, which immediately arises, is, how charge carrier exchange is possible in such a setup. In a nutshell, applying a sufficiently large potential at the gate results in a very thin conducting *channel* right below the insulator, which connects source and drain.

Let us investigate this channel in detail. Applying a gate voltage induces an electric field across the insulator. This causes the energy bands near the surface to bend (cp. Figure 2.8), such that decreasing majority charge carriers are available. For moderate amounts this is called *depletion*. By further increasing the gate voltage, first *weak*, then *moderate* and finally *strong inversion* is reached. Inversion indicates that the minority charge carriers actually exceed the majority ones in number. In the end, a thin *channel* between source and drain is created, which enables current flow through the device. The NPN based device is called n-channel Metal Oxide Semiconductor Transistor (nMOS), whereat the *n* denotes the n-type channel and MOS refers to the structure of the gate-

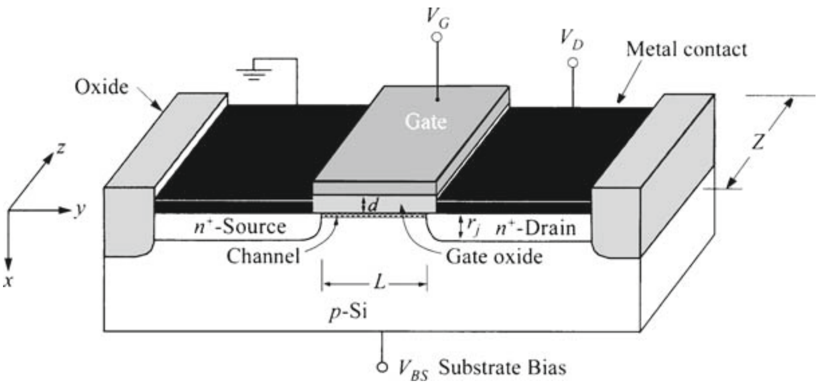


Figure 2.7: Internal structure of a FET. On top of the inner layer of an NPN structure, separated by a thin insulator, resides the gate, which controls the conductivity of the underlying channel. From [73, p. 298]

bulk capacitance composition (Metal-Oxide-Semiconductor). In this context this type of transistor is called Metal Oxide Semiconductor-Field Effect Transistor (MOS-FET).

In the following, we want to investigate the behavior of the nMOS more closely. The PNP based p-channel Metal Oxide Semiconductor Transistor (pMOS) can be analyzed analogously. For a start assume that gate, drain and source are connected to GND (please note that the bulk is pinned to GND as well). The band diagram in three dimensions for this case is shown in Figure 2.8 (b). By changing the potential difference across the gate-bulk capacitance formed by the insulator, i.e., by applying a voltage V_G to the gate terminal, the conductivity inside the channel changes. There are actually two ways to explain this behavior:

1. Gate and bulk form, together with the insulator in between, a capacitance. Applying a positive voltage causes the majority carriers, i.e., the positively charged holes, to be pushed away from the interface and electrons to be attracted. At a certain point, the latter exceed the former, thus creating a connection consisting of electrons from drain to source. If the voltage at the gate is removed, the holes return into the channel and the conductivity decreases.

2. When observing the band diagram in Figure 2.8 (b) a downward bending at the interface for a positive voltage at the gate can be identified, meaning that the CB is pushed closer and closer to the Fermi Energy E_F, while the distance to the VB increases. Thus, according to the the Fermi distribution, the amount of electrons increases while the amount of holes declines. Releasing the voltage bends the bands back in their original position, also resetting the carrier distribution.

Even though a channel is created in inversion, no current is flowing when drain and source are at the same potential. Therefore, we will, in the sequel, assume $V_D \neq V_S$.

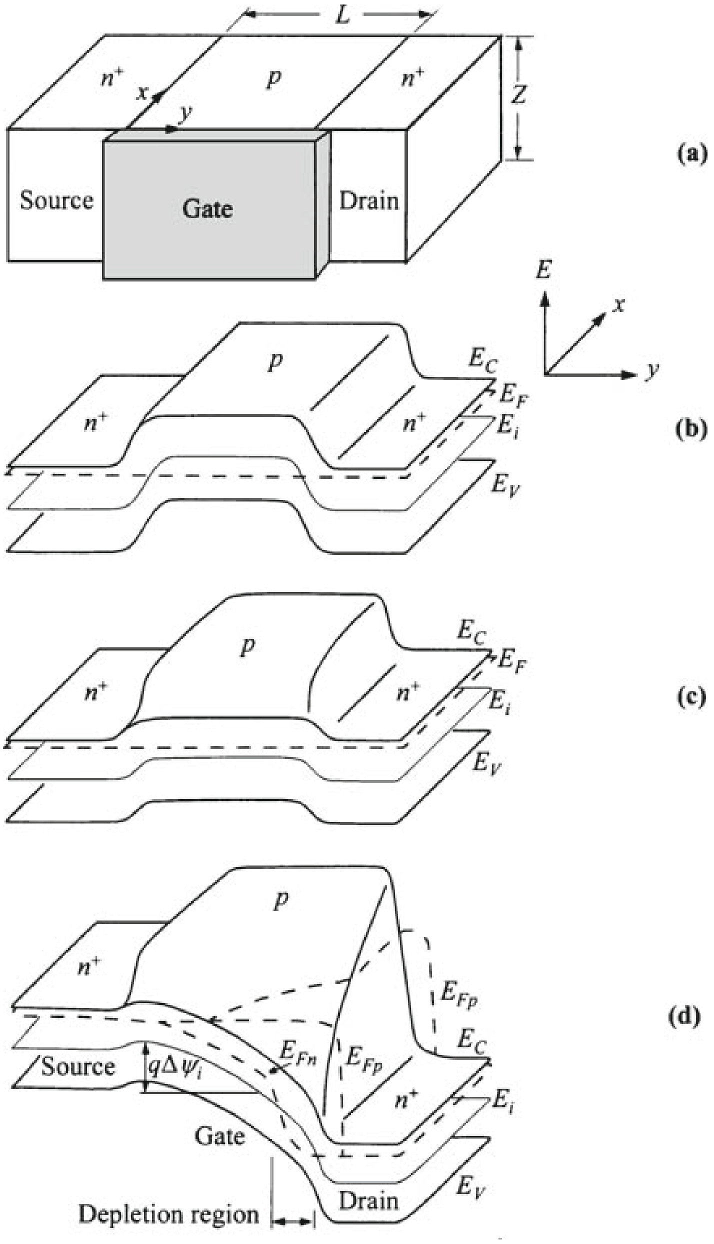


Figure 2.8: Band diagrams of the FET for (b) no gate voltage, (c) considerable gate voltage and (d) additional drain-source voltage. Taken from [73, p. 299]

Please note that FETs, in the fashion we have presented them so far, are symmetric devices regarding drain and source. To tackle various effects, real implementation are optimized to make the current flow in one direction more favorable than in the other. In this thesis, we define the current flow for nMOS from drain to source, i.e., $V_D > V_S$, while for the pMOS the reverse case ($V_S > V_D$) is assumed.

A potential difference between source and drain of an nMOS in the described fashion causes electrons to propagate from source to drain[4] through the channel. Initially the current increases linearly with $V_D - V_S$ as the electric field in the channel also grows linearly. At the same time, however, a decreasing amount of minority carriers is available at the drain end of the channel, since $V_G - V_D$ decreases. This rather complex, three dimensional problem is depicted in Figure 2.8 (d). As $V_G - V_D$ declines, the difference between Fermi level (E_{Fn} in the figure) and CB increases. Thus, the channel at that location turns from strong to moderate inversion, weak inversion and finally depletion. When more majority than minority charge carriers are available we say that the channel is *pinched off*. This does not mean that the current flow is interrupted, since there is still no energy barrier for the electrons. Instead, further increasing V_D has diminishing effect on the current strength.

In the beginning, we already mentioned that the bulk forces the internal diodes in reverse direction. In this fashion, wasteful leakage currents between drain and bulk, respectively source and bulk, are prevented. In an optimal FET the only charge carrier transport is thus between source and drain. Since in pMOS devices the bulk is an n-type semiconductor its voltage value V_B should always be at least as high as those on drain and source. We therefore attach it to the supply voltage V_{DD}. Similarly, the potential of the bulk in the nMOS must always be below every other, such that we connect it to GND. Note that the bulk also has an impact on the onset of inversion, which depends on the voltage drop across the insulator. Thus, the bulk terminal is sometimes used in real circuits to adapt the transistor behavior.

2.5.2 Operation Regions

The transistor is an analog device, i.e., everything, in particular the current through the device, is a continuous function of the gate (V_G), source (V_S) and drain (V_D) voltages (assuming that bulk is pinned to the appropriate value). Calculations of respective analog waveforms are very challenging, especially for small scale devices. To simplify the evaluation analog simulation suites, e.g., the very popular HSPICE, have been developed. They are able to predict the behavior with very high accuracy, which makes it possible to use the results as golden reference. This is also done in this thesis.

The rate of change of the drain current I_D, forced by a constant change on one of the terminals, is not constant but depends heavily on the *operation point*, i.e., the values of V_G, V_S and V_D at that time. Therefore, the input space is divided into operation regions whereat in each region a different formalism is used to describe the behavior

[4] Note the difference to the technical current direction, which is directed from the higher to the lower voltage.

of the transistor. In this thesis we are going to distinguish the following cases: The *sub-threshold* region (ST), where I_D is very small, the *ohmic* region (OHM) where I_D is proportional to the input voltage and the *saturation* region (SAT) where the current only changes moderately.

We already stated, that the behavior of the transistor depends on the voltage values at its terminals. However, it is more accurate to say that not the absolute values are of interest but solely the difference among them. For this purpose, recall the analyses we conducted earlier on the nMOS. A positive gate voltage caused a bending of the energy bands underneath the gate, such that electrons can enter the channel from source and propagate to drain. Assume for a start that $V_S = 0$ and $V_G > 0$ large enough that current flow is possible. By increasing V_S, the energy bands in the source contact are pushed downwards, enabling declining number of electrons to enter the channel. One can picture this scenario as a water level that drops and thus exceeds a barrier less, leading to less water (current) flow over the barrier. To reach the same conductivity as before, the barrier has to be declined as well, which is achieved in the nMOS by increasing V_G. This shows, that the conductance at the source side is effectively controlled by the voltage difference between gate and source $V_{GS} = V_G - V_S$ for nMOS respectively source and gate $V_{SG} = V_S - V_G$ for the pMOS. Reverting the measurement direction has the advantage that we can use the same formalism for n- and pMOS. The differing directions of I_D, which is a result of our earlier stated demand that $V_D > V_S$ (nMOS) and $V_S > V_D$ (pMOS), also imply a dependence on $V_{DS} = V_D - V_S$ for the earlier while the latter is governed by $V_{SD} = V_S - V_D$. To ease explanations V_{Gy} will be used throughout this thesis to denote both V_{GS} and V_{SG} while V_{Dy} is used simultaneously for V_{DS} and V_{SD}. In the following the behavior in the different operation regions is further evaluated.

Region (ST)

We start our analysis with $V_{Gy} = 0$, meaning that almost no minority charge carriers are in the channel beneath the gate, and therefore no current can be conducted. As V_{Gy} is increased, the amount of minority charge carriers and thus the conductivity gradually increases too, as can be seen in Figure 2.9. In circuit design this an undesired property, as we would like to divide the input space into single distinct regions. For this purpose the *threshold voltage* V_{th} is defined, which marks the value that has to be exceeded to achieve considerable levels of conduction. Note that this value experiences fluctuations in real circuits, e.g., by random dopant placement (see [110]), transistor scaling for achieving optimal tradeoffs (see [58]) or simply varying V_B.

Extracting the threshold voltage from analog simulations is not an easy task, as can be retraced in Figure 2.9. Several approaches have been proposed, whereat we will use the ESR method presented by Ortiz-Conde et al. [95] in this thesis. The authors propose to determine V_{th} by setting the bulk (V_B) and source voltage (V_S) to GND/V_{DD} (nMOS/pMOS) while $V_D = V_G$ is sweeped between GND and V_{DD}. After the simulation, the squared root of the drain current $\sqrt{I_D}$ is linearly fitted in the point with the highest derivative. The value of V_{GS} where the linear fit crosses the x-axis then determines the threshold voltage (see Figure 2.9).

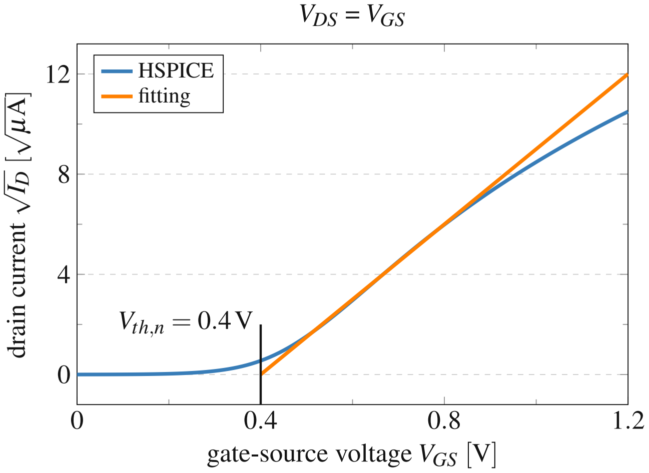


Figure 2.9: Square root of the drain current $\sqrt{I_D}$ through a 65 nm nMOS over V_{GS} ($V_{DS} = V_{GS}$). The threshold voltage $V_{th,n}$ was determined by a linear extension in the point with the highest derivative.

Region (OHM)

As soon as charge transport is possible current can be conducted, whereat the value of I_D depends naturally on V_{Dy}, the potential difference between source and drain. The induced electric field along the channel accelerates the charge carriers and thus movement. Naturally, the higher the field strength the bigger the current[5]. The increase of I_D is here in linear dependence to the increase in voltage, i.e., $\Delta I_D \cdot R = \Delta V_{DS}$, which is the reason why this is called the ohmic region (OHM). Simulations on modern devices (see Figure 2.10) show very good agreement.

Region (SAT)

The ever increasing potential at the drain has the additional effect that the energy bands are pushed to lower levels. Thus, the bending, induced by the gate voltage, diminishes. If V_{Dy} reaches a certain point, in the following called the saturation voltage V_{Dsat}, the inversion at the drain end of the channel starts to fade, which reduces conductivity drastically. This is actually the opposite case of our discussion regarding the threshold voltage when increasing V_{GS}. As a result, the channel is said to be *pinched off* and the effective length of the channel decreases. Since there is still an electric field, the transport mode in this area changes from drift to ballistic, meaning that the electrons can be imagined to pass right through. This works well for distances below the *mean free path* length, which indicates how far an electron can travel in average before being scattered. From here onwards the current only increases moderately, as can be seen in Figure 2.10. This is the reason why this operation region is called saturation region (SAT). Table 2.2 summarizes the single operation regions and their boundary conditions.

[5] At least for low field strength. This topic is covered in greater detail in Section 2.5.3

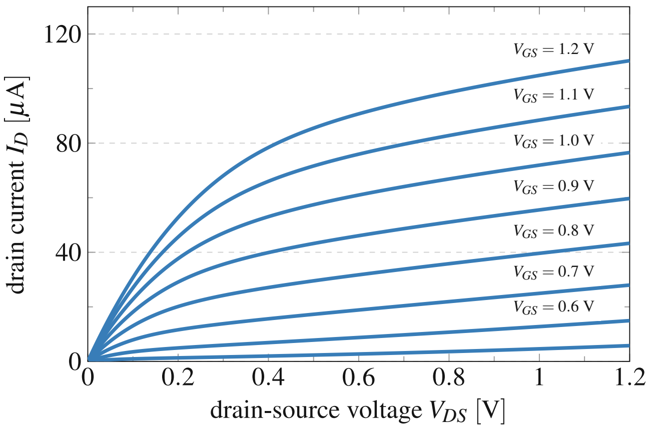


Figure 2.10: HSPICE simulations of the drain current I_D through a 65 nm nMOS over V_{DS} for different values of V_{GS}. The deviating behavior in (OHM) and (SAT) is clearly visible, however, a unique separation is not easily possible.

2.5.3 Short Channel Effects

The analyses conducted so far actually assumed a rather ideal transistor and neglected a lot of phenomena that became dominant with decreasing feature size. While early implementations delivered nearly constant current in (SAT), modern devices show significant variations (cp. Figure 2.10). One of the main causes is the decreasing channel length, i.e., the distance between source and drain. Newly discovered mechanisms that can be retraced to the declining length of the channel are thus summarized as Short Channel Effect (SCE). The most prominent ones are:

Drain Induced Barrier Lowering (DIBL)

The energy bands in the bulk can only change gradually, meaning that the CB in an nMOS along the channel has to first rise and then again drop towards the end. If the latter is very steep and the channel is very short the maximum is not reached any more, as the energy band has to start bending down early. This, however, effectively reduces the barrier height, which is the only thing that prevents electrons from entering the channel. While this has the maybe beneficial effect that the threshold voltage decreases it has the definitely negative effect, that more charge carriers can pass the barrier when the transistor is shut off. Thus the leakage current, i.e., the charge carriers that are transmitted but do not contribute to the calculation and in consequence the power consumption, increases. Actually, nowadays such static leakage currents, which appear when the terminal voltages are not changing, exceed the dynamic currents during switching, resulting in lots of wasted energy, which is one of the key challenges for the future.

region	condition
(ST)	$V_G < V_{th}$
(OHM)	$V_{Gy} > V_{th}$ and $V_{Dy} < V_{Dsat}$
(SAT)	$V_{Gy} > V_{th}$ and $V_{Dy} > V_{Dsat}$

Table 2.2: Operation regions of n- and pMOS

Velocity Saturation

With decreasing channel length, the electric field in the channel increases. This leads to an improved acceleration of the charge carriers and thus a higher drain current. However, at some point saturation can be observed, meaning that the charge carriers can only move with a maximum velocity. This results from the fact that scattering, e.g., collisions with atomic cores, becomes dominant. Note that saturation is only achieved for very high field strengths in the range of tens of kV/cm. This value is, however, easily reached in modern devices, since $V_{DS} = 0.8\,\mathrm{V}$ along a channel of 10 nm length already results in 800 kV/cm.

Hot Carrier Degradation

Since the kinetic energy of the electron depends quadratically on its speed, stronger electric fields also increase the energy of the charge carriers. For field strength as computed in the previous paragraph, some of them can turn into *hot carriers*. This means that they accumulate so much energy to be able to tunnel through, or even worse into, the insulator. The latter are called *traps* and have the effect, that the overall device characteristics is altered significantly, especially for very low device geometries. Examples are varying channel conductivity or threshold voltage as well as increased leakage through the gate.

2.5.4 Small Signal Analysis

So far, we have investigated the behavior of the transistor for every possible input constellation, which is called *big signal analysis*. Sometimes it is, however, only important to determine the behavior in a small region around the operation point. In this *small signal analysis* the behavior of complex devices is linearly interpolated by ideal components such as resistors, capacitances and current sources. This enables an analytic description and analysis of also more elaborate gates. Small signal models for the transistor, as we will present them in the following, have been provided for example by Howe and Sodini [114] and Tietze, Schenk, and Gamm [16]. To better distinguish small signal analyzes, small letters are used to denote currents and voltages, for example v_{DS} instead of V_{DS}.

Static Small Signal Circuit

In the following we will develop the static small signal circuit for an nMOS shown in Figure 2.11 step by step. Based on the operation point $(V_{GS}, V_{BS}, V_{DS}) = (A, B, C)$ we investigate each input separately. Let us begin with the gate. In Figure 2.9 we have

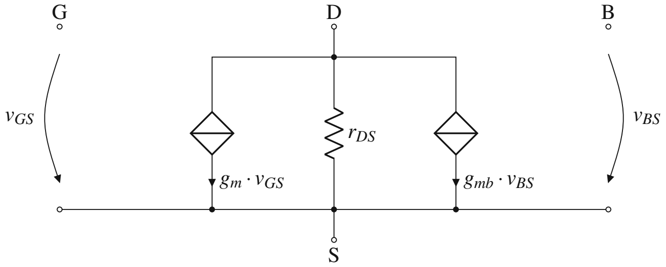


Figure 2.11: Static small signal circuit for an nMOS. The contributions of v_{GS} and v_{BS} are modeled by controlled current sources. Heavily inspired by [114] and [16].

already shown that increasing V_{GS} leads to more current for constant V_{DS}. To estimate the variations around the operation point we linearize I_D and model the current by a voltage controlled current source with a value of $g_m \cdot v_{GS}$, whereat

$$g_m = \left.\frac{\partial I_D}{\partial V_{GS}}\right|_{V_{GS}=A}.$$

Similarly the influence of the bulk can be represented. At $V_{BS} = 0$ the band diagram is the one we observed for a diode. For values of $V_{BS} > 0$ the bands are shifted against each other, making it easier to pass the barrier. In return, this means that the threshold voltages decreases or in other words that the current for a constant gate-source and drain-source voltage increases. Thus, this influence is again modeled using a controlled current source of the magnitude $g_{mb} \cdot v_{BS}$ with

$$g_{mb} = \left.\frac{\partial I_D}{\partial V_{BS}}\right|_{V_{BS}=B}$$

Please note that it is often possible to neglect the influence of the bulk, as source and bulk are shorted wherever possible, resulting in $v_{BS} = 0$.

Finally varying V_{DS} leads to a corresponding increase or decrease of I_D (cf. Figure 2.10). This can simply be depicted by adding a resistor r_{DS} whose value is the linear approximation in the operation point, i.e.,

$$r_{DS} = \left.\frac{\partial I_D}{\partial V_{DS}}\right|_{V_{DS}=C}$$

Dynamic Small Signal Circuit

In the static small signal circuit all changes have an immediate effect. In reality, however, everything happens with limited speed. The main causes are capacitances. Adding them to the model results in the dynamic small signal circuit shown in Figure 2.12[6]. The

[6] Please note that this is still a rather simplistic model, neglecting, for example, the resistors in the terminals.

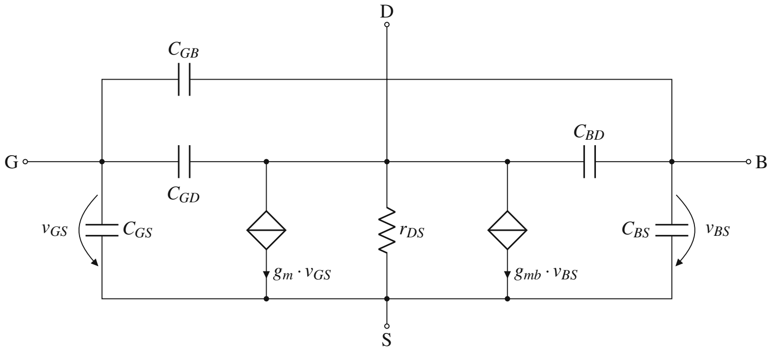


Figure 2.12: Simplified dynamic small signal circuit for an nMOS. Compared to the static circuit capacitative couplings between the terminals is added. Heavily inspired by [114] and [16].

gate is coupled to the other terminals by three separate capacitances. C_{GS} and C_{GD} are caused by spatial overlaps between gate and source respectively gate and drain. C_{GB} denotes the coupling via the oxide with the channel, thus representing the time it takes to attract minority charge carriers and push away the majority ones. In addition, drain and source are coupled to the bulk, which essentially represents the SCR of the diodes.

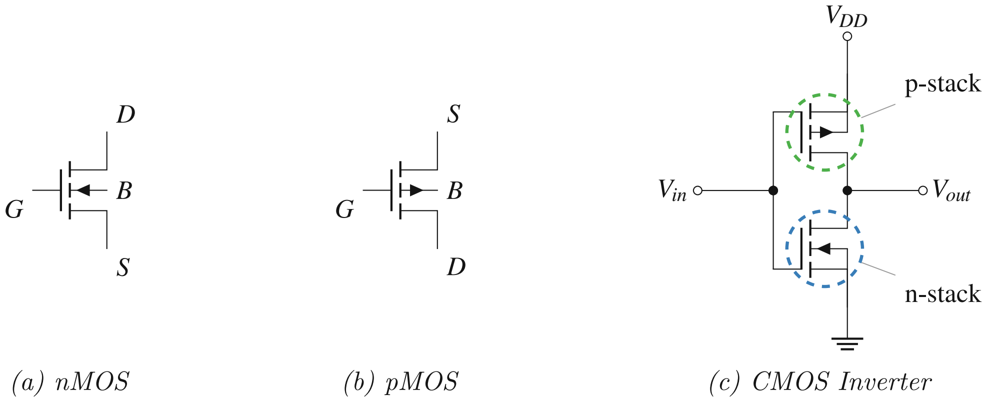
2.6 Junction Field Effect Transistor

For sake of completeness, we want to briefly discuss the Junction Field Effect Transistor (JFET), which is situated somewhere between FET and bipolar transistor both regarding its internal structure and overall functionality. Starting from a bipolar transistor structure (NPN respectively PNP), a fourth gate terminal is added once again, however, this time not separated by a an insulator but directly implanted into the base material. Important is, that (i) the gate has to be differently doped than the base and that (ii) it surrounds the channel that shall be controlled. The first condition leads to a Space Charge Region that is basically used to control the current flow. In detail, please recall that the SCR increases when a diode is driven in reverse direction. By surrounding the channel, as required in the second condition, the SCRs of opposite gate contacts at some point will touch and grow together. Since there are no free charge carriers available in the channel in this case, current flow is effectively prevented.

Please note that a JFET is a *normally on* device, i.e., it conducts when no voltage is applied to its gate and blocks current if it is. In contrast, the earlier presented FET is a *normally off* device, behaving just opposite.

2.7 CMOS Technology

In the previous section we have shown different implementations of a transistor. Similarly, logic gates can also be realized in multiple fashions, varying in the used transistor



(a) nMOS (b) pMOS (c) CMOS Inverter

Figure 2.13: Transistor symbols used in circuit schematics for nMOS (a) and pMOS (b). Combining both results in an Inverter (c).

types, their location and connections and the accompanying components. Such an implementation style is called *logic family*. Historically there have been various approaches such as Resistor-Transistor Logic (RTL), Transistor-Transistor Logic (TTL), Emitter-Coupled Logic (ECL), pMOS, nMOS or BiCMOS [22]. The most prominent for electronic devices is, however, Complementary Metal Oxide Semiconductor Technology (CMOS) [98] which we will therefore also use in this thesis. It utilizes both types of MOS-FETs (n- and pMOS), whereat for the logic family pMOS and nMOS only one of them is used (the other is replaced by a resistor). Using both has the advantage that, neglecting static leakage current, current only flows during switching, which reduces the overall power consumption. The respective symbols in circuit schematics are shown in Figure 2.13.

Recall that the phrase *Metal Oxid Semiconductor* (MOS) actually refers to the used materials, in detail to the metal gate, the oxid (which serves as insulator) and the semiconductor bulk. One oxide often used in this regard is SiO_2, which can easily be grown and is very stable. This was originally one of the main arguments for using Silicon. For modern technologies the Silicon oxide, however, reaches its limits, which made it necessary to switch to *high-k* materials. These have a higher dielectric constant and are thus able to reduce the electric field in the insulator, allowing thicker layers and thus higher resilience against tunneling. For that reason nowadays also the more general term Metal Insulator Semiconductor-Field Effect Transistor (MIS-FET) is used.

We mentioned shortly, that, at least in theory, current is only flowing during circuit switching. In detail a CMOS logic consists of (dual) p- and n-stacks which connect the output to V_{DD} (p) or GND (n). In the static case only one of them is conducting, such that the output is either charged or discharged. During switching, however, both are conducting at the same time for a short period, leading to direct current from V_{DD} to GND and thus leakage. A very simple example is the inverter shown in Figure 2.13c. It consists of two transistors, one n- and one pMOS. More elaborate gates can be realized by placing transistors in parallel (equivalent to a disjunction) or in series (equivalent to conjunction). In Chapter 3 some of these will be explained in more details.

2.8 State-of-the-Art

In this chapter, we were only able to introduce the most basic concepts regarding semiconductors in general and transistors in detail. However, state-of-the-art devices nowadays are already advanced version of what was presented here. Therefore, we shortly highlight the main differences and what to expect for future devices.

Reducing the feature size of electronic circuits has yielded tremendous improvements in circuit design: Latencies decline with parasitic capacitances, resulting in less time for (dis-)charge processes. Thinner insulators and shorter channels allow lower supply voltages, leading to a reduced current strength and thus power consumption.

Currently technologies with a feature size in the low nm range are available. Although we can expect even smaller technologies in the future, physical limitations will soon make any further improvements impossible. This becomes evident when comparing the dimensions of a transistor to the lattice constant of silicon, which is approximately $5.431\,\text{Å} \triangleq 0.5431\,\text{nm}$ [15]. This means, that already today only a handful of atoms are used to build a transistor. As one can imagine this makes uniform doping a very challenging task, whereat the actual position of the dopants in such devices also has a big impact (cf. the discussion of traps in the insulator or the threshold voltage).

Since scaling is not a viable option, new layouts using novel materials are investigated. In the MOS-FETs presented so far the gate is only on top of the channel, which is due to the *planar* technology build up. It is, however, more beneficial to apply a gate on multiple sides, similar to the JFET, leading to more minority charge carriers and thus higher conductivity. This is already achieved in *Fin-FETs* where the PNP respectively NPN structure (drain-bulk-source) is manufactured as an upstanding fin, and the gate is wrapped around three sides of the inner layer. Scientists are currently searching for ways to also apply the gate to the fourth side utilizing nanowires, thus achieving *gate-all-around FETs*. Furthermore, completely new structures for charge transportation, such as carbon nano tubes formed from graphene, are investigated but not yet ready for productive systems. Overall very interesting times with a lot of novel ideas lie ahead.

CHAPTER 3

Analog Circuit Modeling

A proper understanding of the physical processes governing the behavior of semiconductors in general, and Field Effect Transistors in particular, enables the development of accurate analytic models, like the BSIM family[1]. Such models are mandatory to realistically predict voltages and currents prior to fabrication, and thus to verify the correctness of a design at an early stage. Numerical evaluations in simulation suites, like HSPICE, then lead to very precise results, which involve, however, a computationally very expensive task. In consequence, such fine-grained analyses can only be executed in reasonable time for circuits of rather limited size.

In this chapter we will therefore investigate, if and how an analog waveform can be approximated with considerable less effort. We want to emphasize at this point, that our goal is not to compete with highly accurate analog simulation suites, like HSPICE. Instead, we see our approaches, which can be situated in between digital and fully fledged analog simulations, as an enrichment to available simulation methods. The main area of applications, in our opinion, is to provide a rough and quick approximation to identify nodes, showing a potentially malicious behavior, within a circuit. In a succeeding step, analog simulations can be used to either confirm or reject the predictions.

For a start we introduce three simplified transistor models, which will then be used to describe various logic gates. A detailed behavioral analysis is executed for each of them, which is important for predicting delay and metastability in Chapter 4 respectively Chapter 5. Due to the simplicity of these transistor models, general purpose computation and even formal verification tools can be utilized, which increases the range of possible applications significantly. Nevertheless, the achieved improvements turn out to insufficient for determining analog traces for larger circuits. Thus we finally investigate possible approaches to approximate analog waveforms by mathematical functions which will, hopefully, result in an event-driven analog simulation suite.

[1] http://bsim.berkeley.edu/models/

3.1 Experimental Setup

For the research conducted within this thesis, measurements on fabricated devices is not feasible (too expensive, too slow, desired resolution of results not possible). Instead, we resorted to HSPICE as golden reference, which is reasonable due to its high accuracy. Note that in some rare cases, which will be explicitly noted in the text, numerical issues forced us to switch to the tool Spectre by Cadence.

For our analysis we further utilized multiple technology libraries, to get an intuition of changes across technology borders. The details are stated below, whereat we determined the threshold voltage as described in Section 2.5.2. If not stated otherwise technology (T65) was used. Despite being rather old, it is the sole one that contains also layout information, which leads to even more accurate simulation results.

(T65) 65 nm UMC library ($V_{DD} = 1.2\,\mathrm{V}, V_{th,n} \approx 0.4\,\mathrm{V}$, $V_{th,p} \approx 0.47\,\mathrm{V}$) using a Synopsys LEVEL 54 BSIM4 MOSFET model

(T28) 28 nm UMC high-performance-computing technology ($V_{DD} = 0.9\,\mathrm{V}, V_{th,n} \approx 0.47\,\mathrm{V}$, $V_{th,p} \approx 0.44\,\mathrm{V}$) using a Synopsys LEVEL 54 BSIM4 MOSFET model

(T15) 15 nm Nangate Open Cell Library with FreePDK15$^{\mathrm{TM}}$ FinFET models [37] ($V_{DD} = 0.8\,\mathrm{V}, V_{th,n} \approx 0.17\,\mathrm{V}$, $V_{th,p} \approx 0.17\,\mathrm{V}$) using a Synopsys LEVEL 72 BSIM-CMG MOSFET model

3.2 Transistor Models

Changing technologies and layouts[2] demand regular updates of the transistor model to cover new physical phenomena or structural changes. Over the last decades, a lot of behavioral descriptions have thus been accumulated. The manual of HSPICE alone lists 38 supported transistor models, which include, among others, EPFL-EKV [118] (Synopsys LEVEL 55), Advanced Compact MOSFET (ACM) [70] based on the Unified Charged Control Model [106], PSP [76] (Synopsys LEVEL 69) and the Hiroshima Starc IGFET Model (HiSIM) [35, 78] (Synopsys LEVEL 64). Very popular is also the BSIM model family by UC Berkeley[3], e.g., BSIM (Synopsys LEVEL 13), BSIM 2 (Synopsys LEVEL 39), BSIM3 [124] (Synopsys LEVEL 49 and 53), BSIM4 (Synopsys LEVEL 54), BSIM5 [71], BSIM-BULK (Synopsys LEVEL 77) and BSIM-CMG [32] (Synopsys LEVEL 72).

As all models are freely available, one can easily investigate how the analog waveforms are calculated. For this thesis we searched for approaches whose parameters can be directly mapped to physical parameters, to enable a simple characterization. In the best case, following this philosophy allows a straightforward transfer of results to future technologies, as was pointed out by Miura-Mattausch et al. [97]. Appropriate models have been proposed by Klös and Kostka [103], Hauser [83] and Khakifirooz, Nayfeh, and

[2] There are actually a lot more possibilities than the ones we discussed in Chapter 2.

[3] http://bsim.berkeley.edu/models/

Antoniadis [63]. In [75, 90, 127] these results have been extended by further investigating quantum mechanical effects. A very popular model is the Alpha Power Law by Sakurai and Newton [128], whose mapping to physical quantities was shown by Bowman et al. [104]. Due to increasing inaccuracies it was later improved to the Extended Alpha Power Law by Chandra, Kumar Yati, and Bhattacharyya [60].

Although having these models at our disposal, we decided, after careful considerations, to utilize three simplified approaches that will be introduced in the sequel: The Basic Model, which was one of the earliest transistor descriptions and is accordingly simple, the Elaborate Model which already partly covers short channel effects and the Uniform Model, which manages to describe the behavior in all operation regions by a single expression. Although these approaches have a rather limited accuracy they still show reasonable results even for modern technologies. Their trump card, however, is simplicity, which enables the usage of general purpose simulation and verification tools.

Since the descriptions for n- and pMOS differ only minorly, in detail only in the direction the voltage values are measured (compare Section 2.5.2), we will use in the following a unified notation that addresses both transistors at the same time. Consistently with Section 2.5.2, V_{Gy} will be used to simultaneously denote V_{GS} for the nMOS and V_{SG} for the pMOS. Accordingly V_{Dy} has to be replaced by V_{DS}/V_{SD} for nMOS/pMOS. Please keep in mind that also the parameters, e.g., the threshold voltage V_{th}, in the equations differ between n- and pMOS, and have to be replaced accordingly..

3.2.1 Basic Model

The Shichman-Hodges transistor model [151] (Synopsys LEVEL 1) was developed in 1968 and provides, from a modern viewpoint, only a rudimentary description. For this thesis, we even apply further simplifications (proposed by Sze and Ng [73]), which lead to a system of equations that can be handled in analytic considerations.

Sub-threshold (ST)

For $V_{Gy} < V_{th}$ this model completely neglects the current through the transistor, i.e.,

$$I_D^{ST} = 0 \ .$$

For $V_{Gy} > V_{th}$ the transistor either operates in the ohmic or saturation region, whereat the saturation voltage

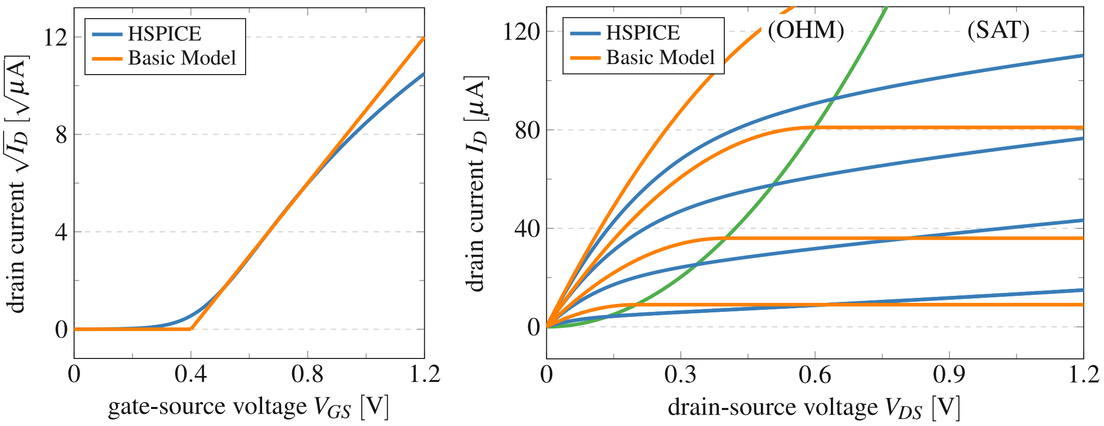
$$V_{Dsat} = V_{Gy} - V_{th}$$

is used. More specifically (SAT) is entered when the voltage difference between gate and drain drops below V_{th}. Compared to simulations (see Figure 3.1b) the model seems to overestimate V_{Dsat} for modern technologies.

Saturation (SAT)

In saturation the current solely depends on the gate voltage and is described by

$$I_D^{SAT}(V_{Gy}, V_{Dy}) = \frac{S}{2} \cdot (V_{Gy} - V_{th})^2$$



(a) $I_D(V_{GS}, V_{GS})$

(b) $I_D(V_{GS}, V_{DS})$

Figure 3.1: Basic Model approximations of the drain current I_D through an nMOS in technology (T65) (a) for $V_{GS} = V_{DS}$ and (b) over V_{DS} for $V_{GS} \in \{1.2, 1, 0.8, 0.6\}$V. This simple model shows quite some deviations, especially for large gate-source voltages.

with $S = 2 \cdot k^2$. It can be easily retraced that k represents the slope of the linear fitting in Figure 3.1a that was also used to determine the threshold voltage. In a physical interpretation it corresponds to the driving strength of the transistor and is, in general, directly proportional to the transistor width.

Note that this definition of I_D^{SAT} neglects V_{Dy} completely, leading to constant values with varying drain-source voltage (see Figure 3.1b). This clearly contradicts the behavior of modern devices.

Ohmic region (OHM)

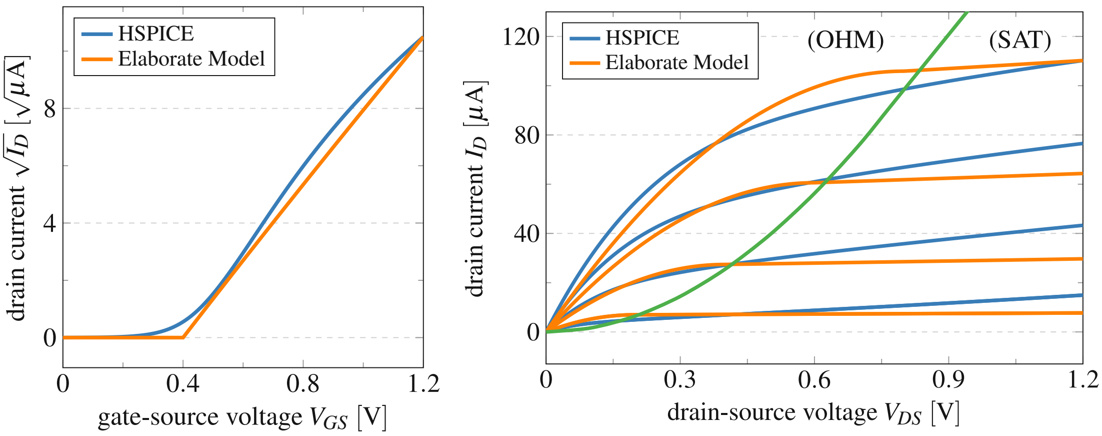
In the ohmic region ($V_{Dy} < V_{Dsat}$, $V_{Gy} > V_{th}$) the drain current is modeled by

$$I_D^{OHM}(V_{Gy}, V_{Dy}) = S \cdot V_{Dy} \cdot (V_{Gy} - V_{th} - V_{Dy}/2)$$

The parabolic shape has its maximum at the transition point to (SAT). The fitting to HSPICE simulations (see Figure 3.1b) shows substantial inaccuracies. The model especially falsely predicts the quite fast transition from the linear shape ($I_D \approx A \cdot V_{Dy}$) to the near constant current in (SAT), which might be caused by a poor estimation of V_{Dsat}.

3.2.2 Elaborate Model

The second approach, which is comparable to the Alpha Power Law by Sakurai and Newton [128], is based on the analysis from Arora [123] and includes more elaborate short channel effects.



(a) $I_D(V_{GS}, V_{GS})$ *(b)* $I_D(V_{GS}, V_{DS})$

Figure 3.2: Elaborate Model approximations of the drain current I_D through an nMOS in technology (T65) (a) for $V_{GS} = V_{DS}$ and (b) over V_{DS} for $V_{GS} \in \{1.2, 1, 0.8, 0.6\}$V. The current increase in (SAT) is still underestimated.

Sub-threshold (ST)

As before the current in the sub-threshold case is not considered at all, i.e.,

$$I_D^{ST} = 0 \ .$$

The saturation voltage, however, is extended significantly, leading to

$$V_{Dsat} = \left(\frac{L\, v_{sat}}{\mu_s}\right) \left(\left[1 + 2\frac{\mu_s}{\alpha\, L\, v_{sat}}(V_{Gy} - V_{th})\right]^{1/2} - 1\right) \tag{3.1}$$

which is based on various physical and fitting parameters. Beside the channel length L (distance between source and drain contact) the gate oxide capacitance per unit area C_{ox}, the saturation velocity v_{sat} and the low field mobility

$$\mu_s = \frac{\mu_0}{1 + \theta(V_{Gy} - V_{th})}$$

are used. α and θ are empirical parameters.

Ohmic region (OHM)

For $V_{Dy} < V_{Dsat}$ the current is approximated by

$$I_D^{OHM}(V_{Gy}, V_{Dy}) = \left(1 + \frac{\mu_s}{L\, v_{sat}} V_{Dy}\right)^{-1} \cdot S \cdot V_{Dy} \cdot (V_{Gy} - V_{th} - \alpha V_{Dy}/2) \tag{3.2}$$

with

$$S \approx \frac{\mu_s\, C_{ox}\, W}{L}$$

	$\mu_0\,[\mathrm{cm^2/Vs}]$	$C_{ox}\,[\mathrm{F/cm^2}]$	$W\,[\mathrm{cm}]$	$L\,[\mathrm{cm}]$	$N_b\,[\mathrm{cm^{-3}}]$
nMOS	349.85	1.26×10^{-6}	4.5×10^{-5}	6×10^{-6}	1.68×10^{17}
pMOS	104.45	1.33×10^{-6}	6.3×10^{-5}	6×10^{-6}	3.99×10^{17}

Table 3.1: Elaborate Model values directly determined from HSPICE parameters.

and W being the channel width. In comparison to the Basic Model, solely the first term was added[4], which causes the current to decrease stronger with increasing V_{Dy}. Comparisons to HSPICE simulation results, shown in Figure 3.2b, reveal a very good agreement in the ohmic region. Please note, that the values presented in the figure have already been scaled by a factor of 10, meaning that the approach in its original form overestimates the current significantly.

Saturation (SAT)

The single process that is modeled in the saturation region is the shortening of the channel, which leads to a slight but steady increase in I_D. This is realized by multiplying the value at the boundary between ohmic and saturation region with a length based scaling factor, i.e.,

$$I_D^{SAT}(V_{Gy}, V_{Dy}) = I_D^{OHM}(V_{Gy}, V_{Dsat})\,\frac{L}{L - l_d}\ . \tag{3.3}$$

Unfortunately, calculating l_d by

$$l_d = \sqrt{\frac{V_{Dy} - V_{Dsat}}{a}} \qquad \text{and} \qquad a = \frac{qN_b}{2\epsilon_0\epsilon_{si}},$$

as proposed in the original publication, using the electron charge $q = 1.602 \times 10^{-19}$ J, the bulk doping N_b (see Table 3.1) as well as the permittivity values $\epsilon_{si} = 11.68$ and $\epsilon_0 = 8.854 \times 10^{-14}$ F/cm leads to unreasonable results. In particular, a short steep increase in I_D at the border of (OHM) and (SAT) was observed. Sometimes l_d even exceeded L, leading to $I_D^{SAT} < 0$. For this reason we utilized

$$l_d = L \cdot \ln\left[1 + \left(\frac{V_{Dy} - V_{Dsat}}{V_P}\right)\right] \tag{3.4}$$

instead (proposed by Arora [123]), whereat V_P represents a fitting parameter.

Parameters

Several physical and fitting parameters are used in the definition of the Elaborate Model. Please note, that it is not within the scope of this thesis to find the best possible fitting, resulting in maybe non-ideal values. To derive reasonable results for technology (T65) we use different sources:

[4] Since $\alpha = 1$ its addition to the last term has no impact.

	v_{sat} [cm/s]	α	θ	a [V/cm^2]	V_P [V]	V_{th} [V]
nMOS	8×10^6	1	0.6	1×10^{11}	1	0.4
pMOS	6×10^6	1	0.8	1×10^{11}	1	0.47

Table 3.2: Parameters for Elaborate Model determined from literature and simple fitting.

1. Literature: the saturation velocity v_{sat}, the electron charge q and the permittivity values ϵ_{si} and ϵ_0

2. Synopsys LEVEL 54 BSIM 4 model parameters: In detail the channel width W and length L (HSPICE: W and L), the low field mobility μ_0 (HSPICE: $u0$), the bulk doping N_b (HSPICE: $ndep$) and the gate capacitance per unit area $C_{ox} = \epsilon_{ox}\epsilon_0/t_{ox}$ (HSPICE: $epsrox \cdot \epsilon_0/toxm$)

3. Simple fitting and estimation: Some of the remaining parameters, such as the threshold voltages, were easily obtained using simulations. Others, for example α and θ (range of $0.03\,\mathrm{V}^{-1}$ to $0.1\,\mathrm{V}^{-1}$), had to be estimated.

The achieved values for n- and pMOS can be found in Table 3.1 and Table 3.2.

3.2.3 Uniform Model

The transistor models, introduced so far, utilize a distinct expressions for I_D in each operation region. This is, actually, one of the major challenges during evaluation, since it is mandatory to continuously check whether the operation region has switched. Much easier to handle is a single equation that remains valid over the whole input and output range. The Uniform Model, introduced by Arora [123], thus merges the distinct equations from the Elaborate Model into a single expression by introducing smooth transitions among the operation regions. Overall the familiar looking expression

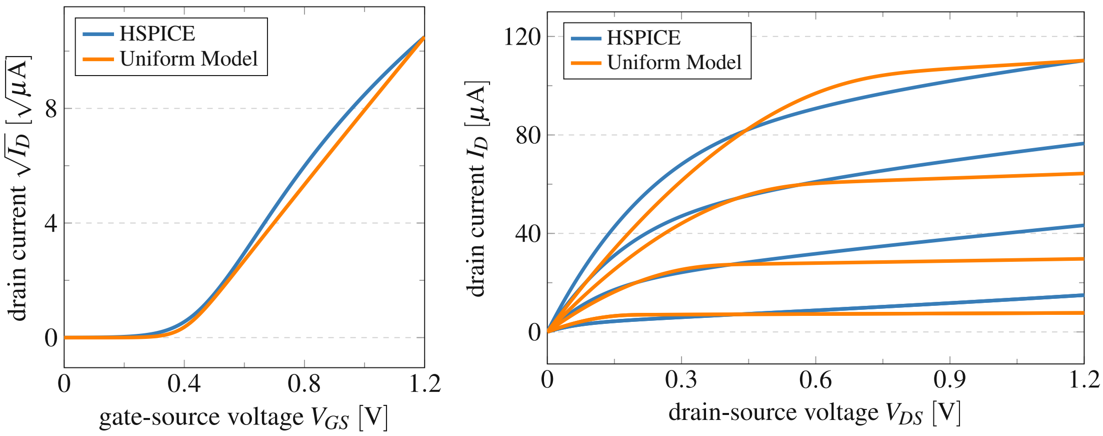
$$I_D(V_{Gy}, V_{Dy}) = \left(1 + \frac{V_{Dy}\,\mu_s}{(L-l_d)v_{sat}}\right)^{-1} \cdot \frac{W\,\mu_s\,C_{ox}}{(L-l_d)} \cdot V_{DU} \cdot (V_{GU} - V_{th} - \alpha\,V_{DU}/2)$$

is achieved, which is very similar to (3.2). The main difference is, that in this case $V_{GU} = V_{GU}(V_{Gy})$ and $V_{DU} = V_{DU}(V_{Dy})$ denote functions that take care of the smooth transitioning and are defined as

$$V_{GU}(V_{Gy}) = \eta\,V_t \cdot \ln\left[1 + \exp\left(\frac{V_{Gy} - V_{th}}{\eta\,V_t}\right)\right] + V_{th} \tag{3.5}$$

$$V_{DU}(V_{Dy}) = V_{Dsat} \cdot \left[1 - \frac{1}{B} \cdot \ln\left(1 + e^{A(1 - V_{Dy}/V_{Dsat})}\right)\right] . \tag{3.6}$$

Many new parameters have been introduced here: In (3.5), $\eta = 1 + \frac{C_d}{C_{ox}}$ (typical range 1 to 3) is used with C_d being the depletion region capacitance. V_t represents the thermal



(a) $I_D(V_{GS}, V_{GS})$

(b) $I_D(V_{GS}, V_{DS})$

Figure 3.3: Uniform Model approximations of the drain current I_D through an nMOS in technology (T65) (a) for $V_{GS} = V_{DS}$ and (b) over V_{DS} for $V_{GS} \in \{1.2, 1, 0.8, 0.6\}$V. The achieved results are very much comparable to the Elaborate Model.

	V_t [V]	A	B	V_P [V]	η
n- and pMOS	26×10^{-3}	10	$\ln(1 + e^A)$	1	1.5

Table 3.3: Additional parameters for the Uniform Model.

voltage $\frac{k_B T}{q}$ ($k_B = 1.380$ J/K the Boltzmann constant, and T the temperature in Kelvin), which results to approximately 26 mV at room temperature (300 K). In (3.6) the constant A and $B = \ln(1 + e^A)$ are utilized. Note that V_{Dsat} is computed according to (3.1) with V_{Gy} replaced by $V_{GU}(V_{Gy})$, while for l_d (3.4) is reused.

Table 3.3 provides an overview of all parameters used in the Uniform Model. The fitting to HSPICE simulations can be observed in Figure 3.3, whereat, once again, already the scaled values are presented (scaling factor ≈ 3).

In the sequel, we will investigate the smoothing functions V_{DU} and V_{GU} more closely. For $V_{Dy} = 0$, (3.6) evaluates to

$$V_{DU}(0) = V_{Dsat} \cdot \left[1 - \frac{1}{B} \cdot \ln\left(1 + e^A\right)\right] = V_{Dsat} \cdot \left[1 - 1\right] = 0$$

due to the definition of B. Since V_{DU} is used as a multiplicative factor, the overall result $I_D = 0$, as expected, is achieved. For $V_{Dy} \gg 0$ the logarithmic term approaches zero leading to

$$\lim_{V_{Dy} \to \infty} V_{DU} = V_{Dsat} \ ,$$

which corresponds to $I_D(V_{Gy}, V_{Dsat})$. Due to (3.4) this implies $l_d = 0$, which finally results in (3.2), i.e., the current expression of the Elaborate Model. Note that the rate

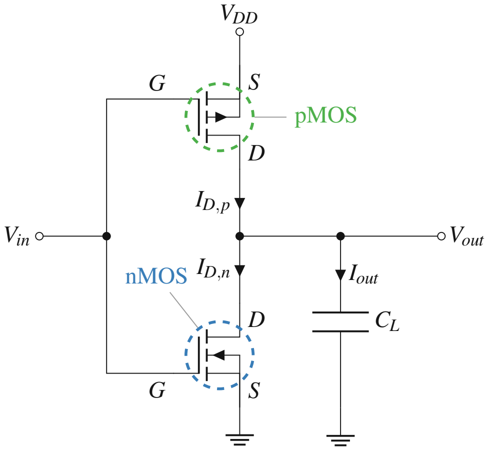


Figure 3.4: CMOS Inverter implementation on the transistor level, consisting of a pMOS (top transistor) and nMOS (bottom transistor). Their respective drain currents $I_{D,n}$ and $I_{D,p}$ are used to determine I_{out}, which (dis)charges the load capacitance C_L and thus determines V_{out}.

of change between the limits observed above is controlled by the factor A, with higher values leading to quicker changes.

Similar analyses can be carried out for V_{GU} and (3.5). For $V_{Gy} = 0$ the exponent becomes negative and the logarithmic term tends towards zero, leading to $V_{GU} = V_{th}$. This does not seem very accurate at a first glance, as this would lead to $I_D^{SAT} \propto -V_{Dy}^2$. However, when plugging $V_{Gy} = V_{th}$ into (3.1), V_{Dsat} becomes zero and, in consequence, also V_{DU}. Again, this is in accordance with the behavior of an actual transistor. For $V_{Gy} \gg 0$ the exponential term becomes dominant. So we can approximate

$$\eta\, V_t \cdot \ln\left[1 + \exp\left(\frac{V_{Gy} - V_{th}}{\eta\, V_t}\right)\right] \approx \eta\, V_t \cdot \frac{V_{Gy} - V_{th}}{\eta\, V_t} = V_{Gy} - V_{th}$$

and in consequence

$$V_{GU}(V_{Gy}) = V_{Gy} - V_{th} + V_{th} = V_{Gy} \ ,$$

which again leads to the expression observed in the Elaborate Model. Comparable to A in (3.6), η determines the transition period whereat here, in contrast, lower values lead to higher speed.

3.3 Inverter Models

Based on the transistor models defined in the previous section, we are finally able to describe logic gates. We will start our analysis with the simplest CMOS circuit, the Inverter, which is shown in Figure 3.4.

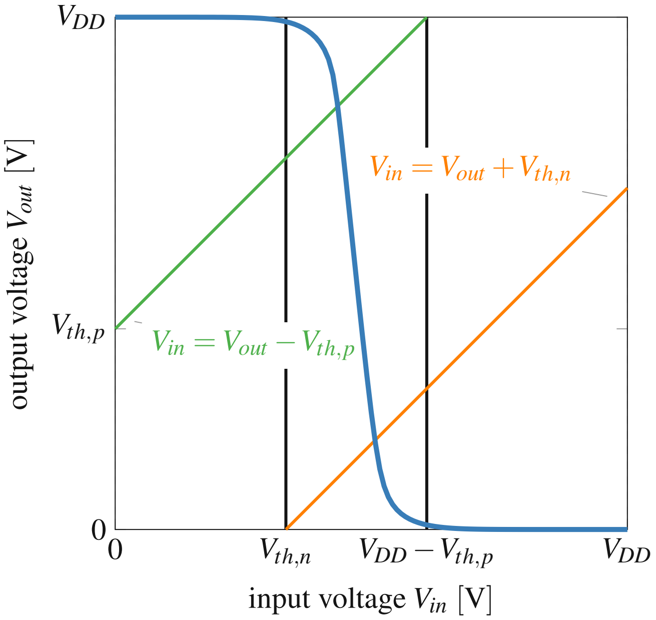


Figure 3.5: Transfer characteristic $f_s(V_{in}) = V_{out}$ for a CMOS Inverter in technology (T65). Operation regions are determined using the Basic Model. In between the lines $V_{out} - V_{th,p}$, $V_{out} + V_{th,n}$ and $V_{th,n} < V_{in} < V_{DD} - V_{th,p}$ both transistors are operating in (SAT) such that the Inverter operates as a linear amplifier.

3.3.1 Behavioral Analysis

Despite the simple internal structure of the Inverter, the task to model the analog output behavior must not be underestimated. Responsible are primarily the coupling capacitances between input and output, in detail between the gate and source/drain contacts of the transistors (not shown in the figure; cf. Section 2.5.4).

For a start, we will thus focus on deriving the static transfer characteristic $f_s(V_{in}) = V_{out}$ shown in Figure 3.5. Assume that initially $V_{in} = \texttt{GND}$. In this case the pMOS is conducting ($V_{SG} = V_S - V_G = V_{DD} - \texttt{GND} > V_{th,p}$) while the nMOS is not ($V_{GS} = V_G - V_S = \texttt{GND} - \texttt{GND} < V_{th,n}$), i.e., ($I_{D,n} = 0$). Consequently the load capacitance will be fully charged, leading to $f_s(\texttt{GND}) = V_{DD}$. The same result is actually obtained for all input values smaller than the nMOS threshold value $V_{th,n}$. Please note, that a real Inverter already starts to change slightly before, which is a consequence of the utilized method to determine the threshold value (cf. Section 2.5.2).

Analogously $f_s(V_{in} > V_{DD} - V_{th,p}) = \texttt{GND}$ can be determined. In between, i.e., for $V_{th,n} < V_{in} < V_{DD} - V_{th,p}$, both transistors are conducting and thus form a simple voltage divider. In this case a direct connection between GND and V_{DD} is established, which results in high power consumption. The concrete output voltage is thereby determined by the conductivity ratio of the transistors. If both are in (SAT), $f_s(V_{in})$ is almost linear, i.e, the Inverter can be interpreted as an amplifier.

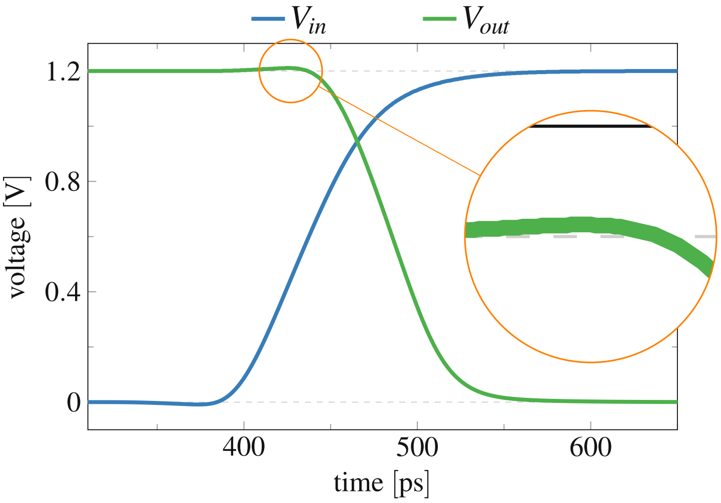


Figure 3.6: Dynamic HSPICE switching behavior of a CMOS Inverter in technology (T65). For this simulation an Inverter chain with capacitive load was used to properly shape the pulses. The figure shows input and output of the fifth Inverter. Please note the over/undershoot before the actual transition.

More interesting for our research is the transient behavior. A corresponding HSPICE simulation is shown in Figure 3.6. The significant delay between V_{in} and V_{out} has two major causes: (1) The output value is only able to change after V_{in} crossed the threshold of the nMOS. (2) The output capacitance has to be discharged, which is limited by the current conducted by the nMOS.

The simulation also reveals a voltage over- respectively undershoot on both signals right at the beginning of the transition. This behavior can be explained by a capacitive coupling between in- and output, also called feedthrough (Hodges, Jackson, and Saleh [89]) or feed forward (Shoji [134]), which has been analyzed, among others, by Huang et al. [72] and Kabbani, AlKhalili, and Al-Khalili [84]. To cover this effect, the transistor level Inverter model has to be extended by a coupling capacitance C_M (see Figure 3.7). Its value in our considerations is approximately the sum of the gate-drain capacitances of n- and pMOS, i.e., $C_M = C_{GD,n} + C_{GD,p}$. The current through C_M can be described by

$$C_M \frac{\mathrm{d}}{\mathrm{d}t}(V_{in} - V_{out}) = I_{in}$$

A changing input value consequently induces a current, which has to be matched at the output node (see Figure 3.8). In our simulation only the paths to the output capacitance and towards the supply voltage are conducting. For the latter a negative current $I_{D,p} < 0$ is achieved and thus a falling potential from V_{out} to V_{DD}, i.e., $V_{out} > V_{DD}$ (the overshoot on the output). Only after the nMOS starts to conduct, an additional path towards GND is opened, which allows V_{out} to drop. Please note that this does not cause I_{in} to vanish. Actually there is an input current as long as either V_{in} or V_{out} change. However, due to the much larger currents through the transistors, it has only minor impact at this point.

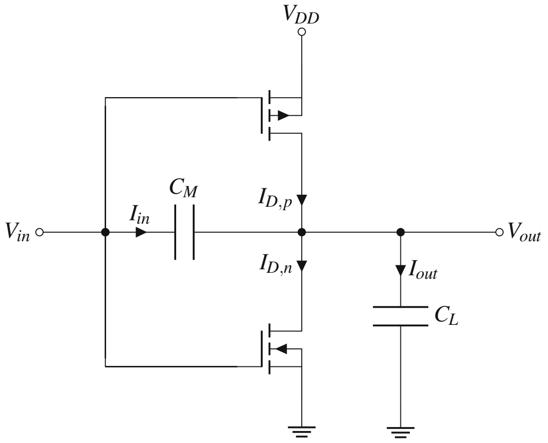


Figure 3.7: Transistor level Inverter implementation extended by coupling capacitance C_M between in- and output, which is responsible for the observed over-/undershoot effects.

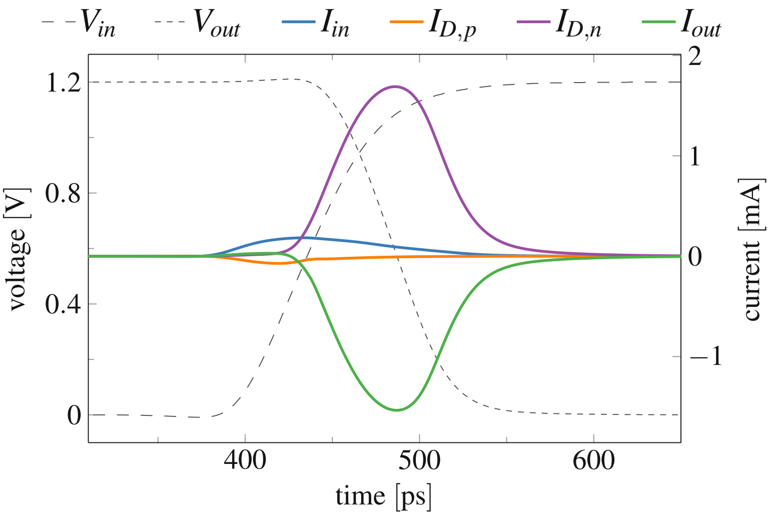


Figure 3.8: Currents inside a CMOS Inverter in technology (T65) during an in-/output transition (thin lines). The input transition induces a current I_{in} leading to $I_{D,p} < 0$. This behavior stops approximately when I_{in} and $I_{D,n}$ cross.

From this analysis it should already be clear, that the input slope has a large impact on the overshoot. Steeper transitions lead to a higher input current and thus cause a larger over-/undershoot, while for smoother transitions it can hardly be recognized. Nevertheless, it is important to keep in mind that this effect exists, especially when analyzing more complex gates later on.

3.3.2 Hybrid Inverter Model (`InvHy`)

Now that we are aware of the physical behavior of an Inverter, we will investigate, how the Basic Model and Elaborate Model have to be used to achieve a reasonable description. Recall, that these models use a separate expression for each operation region, which makes it necessary to keep track of the operation conditions at all times. One possible way is to define distinct states with a specific system of equations, guards which indicate if a state has to be left and a set of transitions to other states. Such a description is called a hybrid model, which we will develop in the sequel.

States, Guard and Invariants

The first task is to identify all possible states of the CMOS Inverter in the hybrid model. A similar analysis has already been carried out by Hodges, Jackson, and Saleh [89], who ended up with five different states (for a single full range input transition only). Shoji [134] used different operation regions, which renders the results hardly comparable.

Please recall, that each transistor can be in one of three operation regions (cf. 2.5), so at most $3 \times 3 = 9$ different states are possible. Luckily two of them are physically unreasonable for general implementations, so only 7 different states are required. Each of them will be explained in detail on the next pages, while a summary is presented at the end of this section.

We already mentioned the crucial task of guards issuing necessary state transitions, however neglected so far, that these transition have to be carried out instantaneously. Otherwise, it might be possible to activate several guards simultaneously, causing the scheduler to decide indeterministically on the succeeding state and the switching time. Since this behavior does not represent a valid physical behavior, it has to be prevented. For this purpose we implement each guard also as an invariant, i.e., the guard $X > Y$ is complemented by the invariant $X \leqslant Y$ and $X < Y$ by $X \geqslant Y$. This has the effect, that the simulation is immediately interrupted as soon as an invariant is violated and the corresponding guard gets activated. For better readability, the invariants will not be explicitly shown in the sequel, as they can be easily derived anyway.

The graphical representation in the $V_{in} - V_{out}$ plane (see Figure 3.9) helps to retrace the executed evaluation steps. We develop the hybrid model state by state by following a rising input transition (start in the top left corner and move to the right). The achieved states are named by the scheme

“State” <name> <operation region pMOS>, <operation region nMOS> .

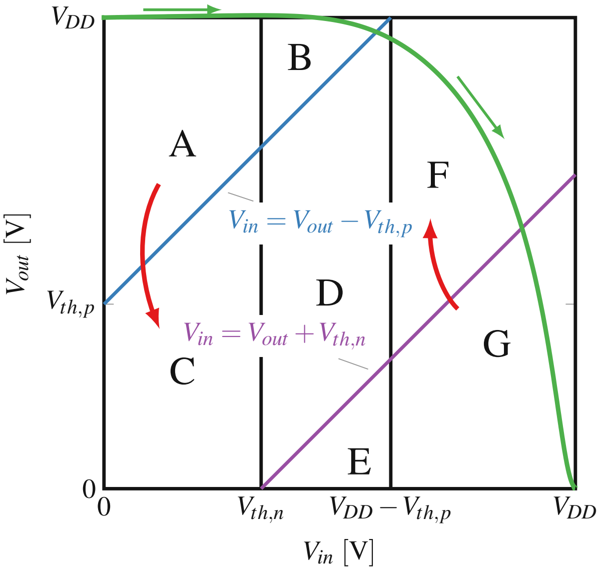


Figure 3.9: States of `InvHy` *in the $V_{in} - V_{out}$ plane. The green line denotes a typical rising input transition. The Basic Model was used to calculate the region boundaries. Red arrows indicate physically unreasonable transitions.*

Note that the output voltage derivative is determined according to the simple Inverter structure shown in Figure 3.4 as

$$C_L \frac{\mathrm{d}}{\mathrm{d}t} V_{out} = I_{out} = I_{D,p} - I_{D,n}.$$

State A (OHM), (ST)

Initially the input is at GND and the output thus at V_{DD}, implying that the conditions $V_{DD} - V_{Dsat,p} < V_{out} < V_{DD}$ and $0 < V_{in} < V_{th,n}$ are satisfied. More specifically the pMOS is in its (OHM) and the nMOS in its (ST) operation region leading to the overall output current

$$I_{out} = I_{D,p}^{OHM}(V_{DD} - V_{in}, V_{DD} - V_{out})$$

The first condition — stating that V_{out} stays above $V_{DD} - V_{Dsat,p}$ while $V_{in} < V_{th,n}$ — cannot be violated in physical circuits. Due to the fact that, according to our simplifications, the nMOS is not conducting in this case, only a charging current to the output capacitance is possible. Consequently the single way to leave this state is by violating the second condition, i.e., by increasing V_{in} above $V_{th,n}$. The single guard[5] for this state can hence be written as

[5] The notation for the guards is the region name followed by an increasing number starting with 1.

name	condition	goal state
A1	$V_{in} > V_{th,n}$	B

Please note that a state with $V_{out} < V_{DD} - V_{Dsat,p}$ and $0 < V_{in} < V_{th,n}$ — as will be defined later as state C — does indeed exist. It is just not reachable from this state.

State B (OHM), (SAT)

As V_{in} exceeds $V_{th,n}$ the nMOS starts to conduct. Note that it immediately operates in its (SAT) region, since $V_{DS} = V_{DD} > V_{Dsat,n} = V_{in} - V_{th,n}$. Due to the fact that the pMOS stays in its (OHM) region the output current I_{out} results to

$$I_{out} = I_{D,p}^{OHM}(V_{DD} - V_{in}, V_{DD} - V_{out}) - I_{D,n}^{SAT}(V_{in}, V_{out}) \ .$$

This state can be left in two different directions: If (i) V_{in} drops again below $V_{th,n}$, causing the nMOS to switch back to (ST), or (ii) if $V_{DD} - V_{Dsat,p} > V_{out}$. In the latter case the output voltage drops to the point where the pMOS enters (SAT) (move vertically in Figure 3.9). Note that for our circuits $V_{DD} - V_{Dsat,p} > V_{Dsat,n}$, which implies that the nMOS can only reach its (OHM) region after the pMOS has entered its (SAT) (cf. Figure 3.5). Therefore the state where both transistors are in (OHM) is unreasonable and is one of the two that can be removed. Overall, the guards for leaving this state are

name	condition	goal state
B1	$V_{in} < V_{th,n}$	A
B2	$V_{DD} - V_{Dsat,p} > V_{out}$	D

Please recall that $V_{Dsat,p}$ depends on V_{Gy} and thus on V_{in}, meaning that for guard B2 in- and output have to be considered.

State D (SAT), (SAT)

Finally, both transistors operate in their (SAT) region, i.e., the Inverter behaves as a linear amplifier (cf. Section 3.3.1). This leads to an output current of

$$I_{out} = I_{D,p}^{SAT}(V_{DD} - V_{in}, V_{DD} - V_{out}) - I_{D,n}^{SAT}(V_{in}, V_{out})$$

Depending on the input signal different successor states are possible (cf. Figure 3.9; move left/right or up/down): For slow trajectories, the output sticks close to the static characteristic $f_s(V_{in})$, which eventually causes one of the transistors to switch to (OHM). Fast input transitions, however, give the output no time to react and cross $V_{DD} - V_{th,p}$ resp. $V_{th,n}$ before the condition mentioned before is met, i.e., one transistor is then operating in (ST). Overall the guards for leaving state D are therefore

name	condition	goal state
D1	$V_{DD} - V_{Dsat,p} < V_{out}$	B
D2	$V_{in} < V_{th,n}$	C
D3	$V_{out} < V_{Dsat,n}$	E
D4	$V_{in} > V_{DD} - V_{th,p}$	F

State C (SAT), (ST)

This state can only reached from state D by a steep falling input transition such that V_{in} drops below $V_{th,n}$, while retaining $V_{DD} - V_{Dsat,p} > V_{out}$. In particular, the nMOS enters (ST) while the pMOS stays in (SAT). Consequently, the output current results to

$$I_{out} = I_{D,p}^{SAT}(V_{DD} - V_{in}, V_{DD} - V_{out}) \ .$$

If V_{in} stays below $V_{th,n}$ then V_{out} will eventually reach the point where the pMOS enters its (OHM) region (i.e., $V_{DD} - V_{Dsat,p} < V_{out}$). If V_{in} starts rising fast enough, however, it is possible that the nMOS enters its (SAT) operation region. Thus the guards result to

name	condition	goal state
C1	$V_{DD} - V_{Dsat,p} < V_{out}$	A
C2	$V_{in} > V_{th,n}$	D

State E (SAT), (OHM)

Increasing V_{in} slowly in state D causes the output to drop rapidly. Eventually $V_{out} < V_{Dsat,n}$ is satisfied and the nMOS enters its (OHM) region, while the pMOS stays in (SAT). The current then results to

$$I_{out} = I_{D,p}^{SAT}(V_{DD} - V_{in}, V_{DD} - V_{out}) - I_{D,n}^{OHM}(V_{in}, V_{out}) \ .$$

Further increasing V_{in}, more specifically when $V_{DD} - V_{th,p}$ is crossed, causes the pMOS to stop conducting (enter its (ST) operation region) while decreasing V_{in} returns nMOS to (SAT). The guards for this state are thus

name	condition	goal state
E1	$V_{out} > V_{Dsat,n}$	D
E2	$V_{in} > V_{DD} - V_{Dsat,p}$	G

State F (ST), (SAT)

Comparable to C this state is solely reachable from D by a steep rising input transition, which causes the pMOS to stop conducting. The current in this case results to

$$I_{out} = -I_{D,n}^{SAT}(V_{in}, V_{out}) \ .$$

Eventually, the output voltage drops below $V_{Dsat,n}$ causing the nMOS to enter its (OHM) region, even for constant V_{in}. If the latter drops, however, faster and becomes smaller than $V_{DD} - V_{th,p}$, this causes the pMOS to enter (SAT) again. Therefore the guards for leaving state F are

name	condition	goal state
F1	$V_{in} < V_{DD} - V_{th,p}$	D
F2	$V_{out} < V_{Dsat,n}$	G

State G (ST), (OHM)

Finally, when the transition is nearly finished, this state is entered where the nMOS operates in its (OHM) and the pMOS in its (ST) region. The current then results to

$$I_{out} = -I_{D,n}^{OHM}(V_{in}, V_{out}) \ .$$

The only way to leave this state is to drive V_{in} below $V_{DD} - V_{th,p}$, such that the pMOS starts conducting again. It is not possible to drive the nMOS back into its (SAT) operation region, since this is equivalent to increasing V_{out} for $V_{in} > V_{DD} - V_{th,p}$. For this input voltage, however, solely the nMOS is conducting, making it only possible to discharge the capacitance at the output. Considering this, the single guard is

name	condition	goal state
G1	$V_{in} < V_{DD} - V_{Dsat,p}$	E

This concludes the hybrid Inverter model shown in Figure 3.10. Guards, invariants and drain current for each state are summarized in Tables 3.4, 3.5 and 3.6 respectively. Please note that the same set of states could be achieved by investigating a falling input transition, with the difference, that the states are encountered in the reverse order.

The above described set of states (A-G) is complete, meaning that only seven of the possible $3^2 = 9$ states are valid. We already discussed why the state with both n- and pMOS in (OHM) is unreasonable. In addition the state where both are in (ST) is only reachable if $V_{th,n} > V_{DD} - V_{th,p}$ which is for example the case in technology (T28). We will not discuss the appropriate model in detail but just show a graphical representation in Figure 3.11. The number of possible states reduces to five as it is now mandatory that at least one transistor is in (ST). This is in accordance to previous, far more

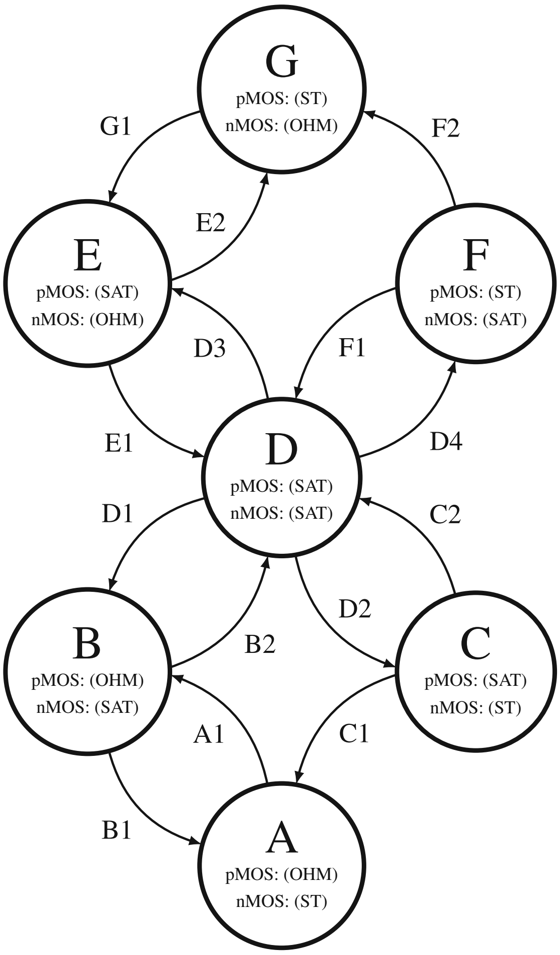


Figure 3.10: Graphical representation of the hybrid Inverter model. Each node represents a single state which is identified by the operation regions of the n- and pMOS.

elaborate Inverter models, which have been developed by Consoli, Giustolisi, and Palumbo [50] (distinguish between constant and varying output), Chaourani and Nikolaidis [40] (considers sub-threshold voltage) and Chaourani et al. [39] (far more states used).

from state	guard name	condition	to state
A	A1	$V_{in} > V_{th,n}$	B
B	B1	$V_{in} < V_{th,n}$	A
	B2	$V_{DD} - V_{Dsat,p} > V_{out}$	D
C	C1	$V_{DD} - V_{Dsat,p} < V_{out}$	A
	C2	$V_{in} > V_{th,n}$	D
D	D1	$V_{DD} - V_{Dsat,p} < V_{out}$	B
	D2	$V_{in} < V_{th,n}$	C
	D3	$V_{out} < V_{Dsat,n}$	E
	D4	$V_{in} > V_{DD} - V_{th,p}$	F
E	E1	$V_{out} > V_{Dsat,n}$	D
	E2	$V_{in} > V_{DD} - V_{th,p}$	G
F	F1	$V_{in} < V_{DD} - V_{th,p}$	D
	F2	$V_{out} < V_{Dsat,n}$	G
G	G1	$V_{in} < V_{DD} - V_{th,p}$	E

Table 3.4: Guards of the single states in the `InvHy` *model. If the condition is fulfilled the guard is triggered and the system changes from the state in the first column to the state in the last column.*

state	invariant name	invariant
A	AI1	$V_{in} \leqslant V_{th,n}$
B	BI1	$V_{in} \geqslant V_{th,n}$
	BI2	$V_{DD} - V_{Dsat,p} \leqslant V_{out}$
C	CI1	$V_{DD} - V_{Dsat,p} \geqslant V_{out}$
	CI2	$V_{in} \leqslant V_{th,n}$
D	DI1	$V_{DD} - V_{Dsat,p} \leqslant V_{out}$
	DI2	$V_{in} \geqslant V_{th,n}$
	DI3	$V_{out} \geqslant V_{Dsat,n}$
	DI4	$V_{in} \leqslant V_{DD} - V_{th,p}$
E	EI1	$V_{out} \leqslant V_{Dsat,n}$
	EI2	$V_{in} \leqslant V_{DD} - V_{th,p}$
F	FI1	$V_{in} \geqslant V_{DD} - V_{th,p}$
	FI2	$V_{out} \geqslant V_{Dsat,n}$
G	GI1	$V_{in} \geqslant V_{DD} - V_{th,p}$

Table 3.5: Invariants of single states in the `InvHy` *model. These are the negation of the guards shown in Table 3.4. If the invariant gets invalid a transition is triggered.*

state	I_{out}
A	$I_{D,p}^{OHM}(V_{DD}-V_{in},V_{DD}-V_{out})$
B	$I_{D,p}^{OHM}(V_{DD}-V_{in},V_{DD}-V_{out})-I_{D,n}^{SAT}(V_{in},V_{out})$
C	$I_{D,p}^{SAT}(V_{DD}-V_{in},V_{DD}-V_{out})$
D	$I_{D,p}^{SAT}(V_{DD}-V_{in},V_{DD}-V_{out})-I_{D,n}^{SAT}(V_{in},V_{out})$
E	$I_{D,p}^{SAT}(V_{DD}-V_{in},V_{DD}-V_{out})-I_{D,n}^{OHM}(V_{in},V_{out})$
F	$-I_{D,n}^{SAT}(V_{in},V_{out})$
G	$-I_{D,n}^{OHM}(V_{in},V_{out})$

Table 3.6: Expression for the output current I_{out} in each state of the hybrid Inverter model.

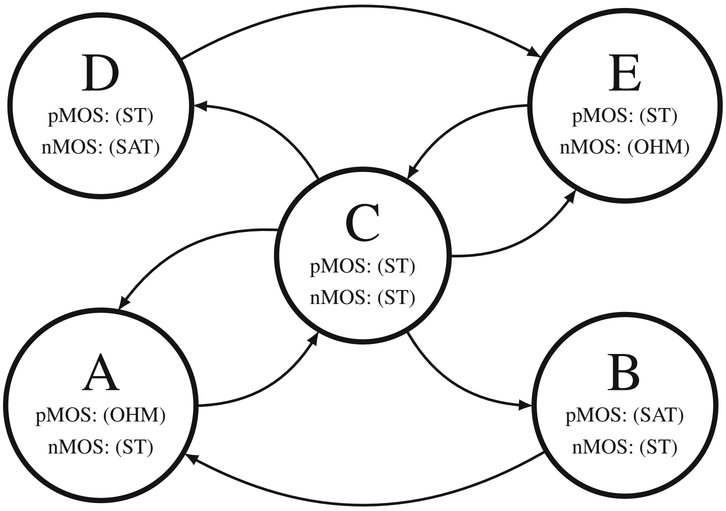
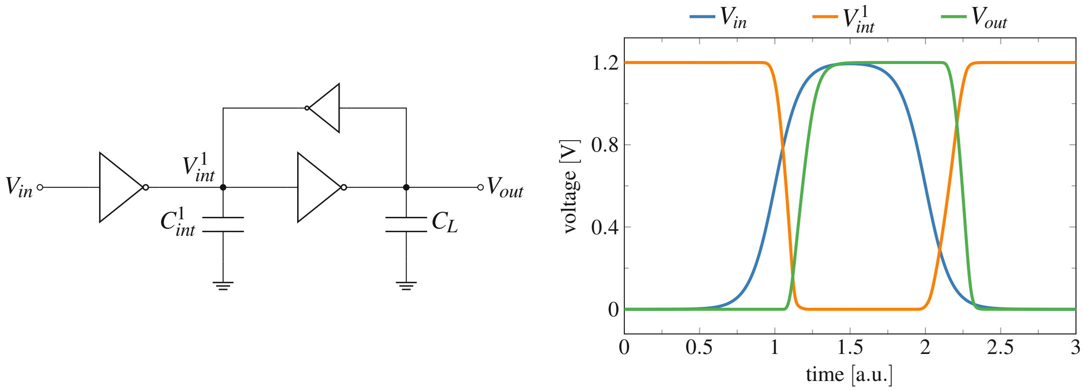


Figure 3.11: Graphical representation of the hybrid Inverter model for the case that $V_{th,n} > V_{DD} - V_{th,p}$. In each state, represented by a node showing the operation regions of the transistors, one of them has to be in (ST). Thus the number of possible states is reduced.



(a) Implementation *(b) Simulation results in MACS*

Figure 3.12: Memory element (D-latch) modeled using the Uniform Model. The internal capacitance C_{int}^1 is required to evaluate the voltage value at this point, whereat we used $C_L/C_{int}^1 = 2$.

3.3.3 Uniform Inverter Model (`InvUni`)

The hybrid model introduced in the previous section is already quite complicated, although it describes the simplest of all gates. This is a direct consequence of the differing expressions for I_D among the transistor operation regions, which quickly increase the number of states. Recall that the Uniform Model, in contrast, utilizes a single equation, which will be exploited in the sequel to derive circuit models in a simple and straightforward fashion. More specifically the transistor currents are summed up at crossing points according to Ohm's laws, leading to a set of n Ordinary Differential Equations (ODE), where n is the number of (internal) nodes. To evaluate these equations, we developed the MACS tool, which will be described in detail in Section 3.6.2.

For the Inverter the complete description collapses to

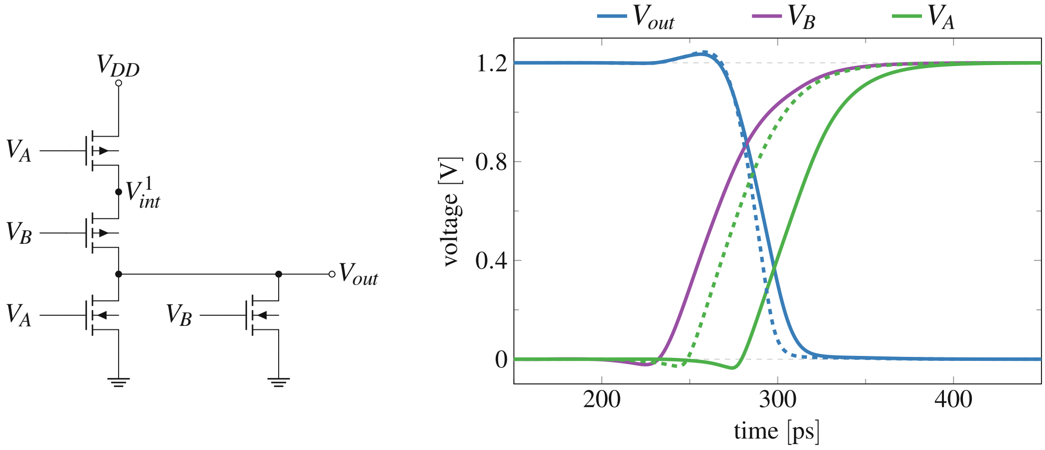
$$C_L \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{out} = I_{D,p}(V_{DD} - V_{in}, V_{DD} - V_{out}) - I_{D,n}(V_{in}, V_{out}) = I_{inv}(V_{in}, V_{out}) \; .$$

Consider the huge improvement compared to the hybrid model. This single equation can be easily evaluated in general purpose tools and thus leads quickly to results.

Reusing the Inverter model enables immediately the description of more complicated circuits, like the D-latch shown in Figure 3.12a. A simple system of two ODEs, in detail

$$\begin{aligned} C_L \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{out} &= I_{inv}(V_{int}^1, V_{out}) \\ C_1 \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{int}^1 &= I_{inv}(V_{out}, V_{int}^1)/A + I_{inv}(V_{in}, V_{int}^1) \end{aligned} \tag{3.7}$$

is sufficient. Please note that the feedback Inverter in the memory loop has to be weaker than the one connected to V_{in}, to assure reliable (re)set of the value in the loop. For this purpose the respective current is divided by a factor $A > 1$. Figure 3.12b shows MACS simulations for $A = 5$.



(a) Implementation *(b) Simulation results in HSPICE*

Figure 3.13: The CMOS NOR gate. Compared to the Inverter one n- and pMOS were added, whereat the former are in parallel and the latter in sequence. Simulations for Two values of Δ show a much steeper output transition for the smaller one. Please note also the over-/undershoot, which we already analyzed for the Inverter.

3.4 NOR-Gate Model

The fact, that the Inverter model, based upon then Uniform Model, could be developed and evaluated very easily, raises the question how well the approach performs for more complex gates. We start with the NOR gate, which realizes the boolean equation

$$O = \neg(A \vee B)$$

We picked this gate as it consists of four transistors only (compared to the six for the OR) and is thus the most natural extension of the Inverter. Nevertheless, as we will see in the sequel, this minor change has a significant impact on the overall behavior, as nearly-simultaneous input transitions can interfere with each other. This will become important for the delay estimations in Chapter 4.

3.4.1 Behavioral Analysis

The transistor level implementation of the NOR gate is shown in Figure 3.13a. The structural differences to the Inverter are quite significant: While the Inverter has unique, comparable paths connecting the output to V_{DD}/GND, the NOR gate has two transistors in parallel (n-stack) and two transistors in sequence (p-stack). In addition, the relative arrival time difference Δ between the two inputs has a big impact.

In general, Single Input Switching (SIS) ($|\Delta| \gg 0$) and Multi Input Switching (MIS) ($|\Delta|$ small) are distinguished, whereat the latter can cause input-to-output delay variations. This behavior was described, among others, by Shoji [134] or Chandramouli and Sakallah [116]. Melcher, Röthig, and Dana [125] further observed that, depending on the output

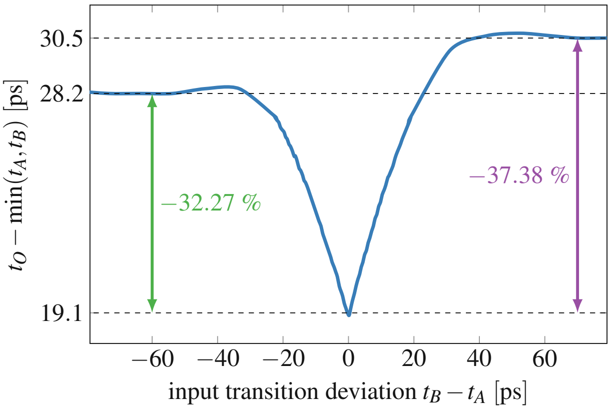


Figure 3.14: Delay to cross $V_{DD}/2$ at the NOR output in technology (T65) in relation to the relative input arrival times. Remarkable are the differing final values for large deviations and the local maxima for intermediate deviations (explanation in the text).

direction, slow down and speed up collisions have to be distinguished. In the sequel we are going to provide a detailed physical explanation for the phenomenons.

Falling Output Transition

We first investigate rising input transitions, starting from $V_A(0) = V_B(0) =$ GND. In response to each transition, one of the parallel nMOS transistors, connecting V_{out} to GND, starts to conduct, while one of the two pMOS transistors in series stops conducting. In consequence, the load capacitance (not shown in the figure) is drained, and V_{out} starts to drop. Clearly this process can be accelerated when both nMOS are conducting, which can also be seen in HSPICE simulation results shown in Figure 3.13b. The closer the input transitions the steeper the output trajectory.

To quantify the decrease in the overall delay, let t_A, t_B and t_O denote the points in time when V_A, V_B respectively V_{out} cross $V_{DD}/2$. As the first rising input starts the output transition, Figure 3.14 shows $t_O - \min(t_A, t_B)$ in relation to $t_B - t_A$. If both input transitions are far apart, the delay for the SIS case can be observed. Interestingly the values differ between the two inputs. This can be easily explained by the fact, that switching only input A keeps the output connected to V_{int}^1, which thus has to be discharged as well. Consequently, the specific delay actually depends on the value of V_{int}^1, which has to be considered when conducting simulations. We considered the worst case and set $V_{int}^1 = V_{DD}$. In contrast, for a switch on input B only the load capacitance has to be discharged, which explains the lower delay.

With decreasing $|t_B - t_A| = |\Delta|$ the output delay decreases. Although, theoretically, an improvement of 50 % should be possible, we obtain only 32 % respectively 37 % from our simulations. This is, however, well in accordance with previous observations, e.g., by Shin et al. [64], Sridharan and Chen [79] and Fukuoka, Tsuchiya, and Onodera [68].

At last we focus on the local maxima around $t_B - t_A = -40$ ps respectively 50 ps,

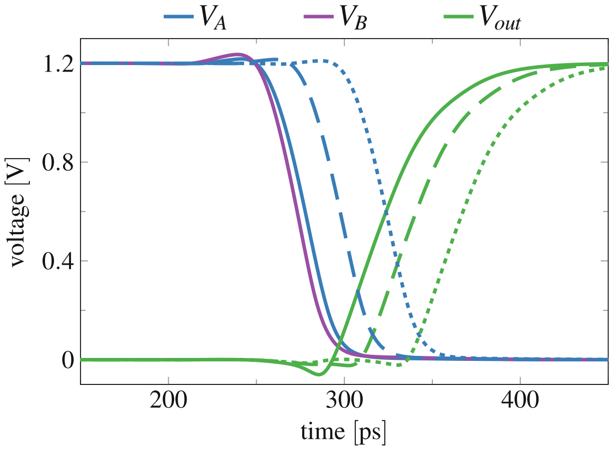


Figure 3.15: HSPICE simulations for a rising output transition of a NOR gate in technology (T65). Two values of Δ are used. The shape of the output waveform stays constant, it solely shifts in time.

which seem very counter-intuitive. This effect looks just like the overshoot in temporal simulations (cf. Section 3.3.1) and can actually be explained in a very similar fashion, i.e., by considering a capacitative input-output coupling. Crucial for this is to recall that for the first part of an input transition the transistors do not react, since the respective threshold voltages have not yet been crossed. In this time frame, the transition thus induces a parasitic current that slows down the output but does not contribute to its compensation. Consequently the delay increases, which can be seen in the figure. Moving the transitions closer together yields longer phases of parallel current conductance and thus finally a decrease in delay.

Rising Output Transition

The behavior for rising output transitions differs significantly compared to falling ones, since the corresponding pMOS transistors are aligned in series. This way, an internal node, which may store arbitrary values, is created and has to potentially be charged during switching, which naturally increases the delay of the output signal. An in-depth analysis of this circumstance, which becomes especially interesting for multiple transistors in a row, has been presented, e.g., by Kabbani, AlKhalili, and Al-Khalili [84] and Chatzigeorgiou, Nikolaidis, and Tsoukalas [105].

Obviously both transistors have to conduct to initiate an output transition. A single falling input thus only cuts the connection to GND through one of the nMOS while simultaneously one of the pMOS starts conducting. Only after the second input switch the output starts to change (see Figure 3.15). Remarkably, the shape of the output transition stays approximately the same, solely its point in time changes. This is in stark contrast to the falling output case.

The thoughtful reader might have already inferred from the previous explanations,

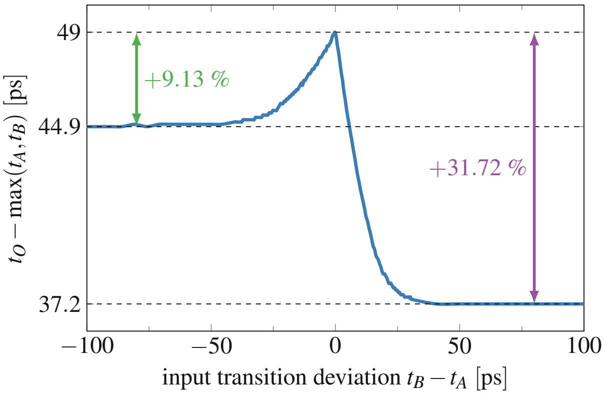


Figure 3.16: Time for the NOR output to reach $V_{DD}/2$ in technology (T65) after the second input reached that value, in relation to Δ. Remarkable are the differing final values for large deviations, which depend on the initial value of V^1_{int} (GND in these simulations).

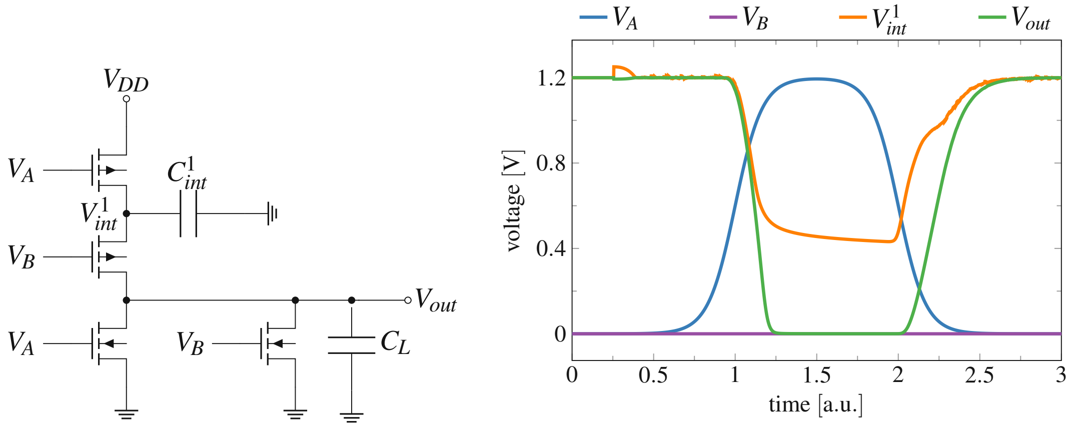
that the order of the input transitions matters. For $V_A = \texttt{GND}$ and $V_B = V_{DD}$, V^1_{int} will be charged to V_{DD}, which will later speed up the rising output transition. For $V_A = V_{DD}$ and $V_B = \texttt{GND}$, it gets, however, discharged to GND. In this case, an additional delay, resulting from charging V^1_{int} to V_{DD} is introduced. This explains the severely differing delays for $|t_A - t_B| \gg 0$ in Figure 3.16. In general, the delay increases with the distance of the last switching transistor to the output node (see also Shoji [134]).

Interestingly, for rising output transitions the delay increases for $|t_A - t_B| \to 0$. An explanation for this behavior can again be found in a capacitive coupling between the inputs and the node V^1_{int}. When both inputs switch at the same time, there is a current induced over the coupling capacitances into the internal node. However, since none of the adjacent transistors is yet conducting, the charge carriers accumulate at V^1_{int}. Only after conductance is established, which is initially a very slow procedure, they can be removed. For input transitions further apart, this charge was already, at least partly, removed and thus the delay decreases. In our simulations, we see an increase of 9 respectively 31 % which is in accordance to previous results by Shin et al. [64], Sridharan and Chen [79] and Fukuoka, Tsuchiya, and Onodera [68].

Please note that, similar to falling output transitions, the initial value of the internal node impacts the absolute delay. In our simulations we assumed the worst case, i.e., $V^1_{int} = \texttt{GND}$.

3.4.2 Hybrid NOR Model

The development of a hybrid Inverter model (cf. Section 3.3.2) was already a very big challenge. Considering the four transistors of the NOR gate, and thus $3^4 = 81$ potential states, the complexity explodes. For that reason we skipped the hybrid NOR model (also for all subsequent circuits) and immediately turned to the Uniform Model.



(a) Implementation *(b) Simulation results in MACS*

Figure 3.17: Implementation of a NOR gate in the Uniform Model and MACS simulation results for an SIS on input A. For the simulations a ratio of $C_L/C^1_{int} = 4$ was used. V^1_{int} shows some artifacts which are not visible in HSPICE simulations.

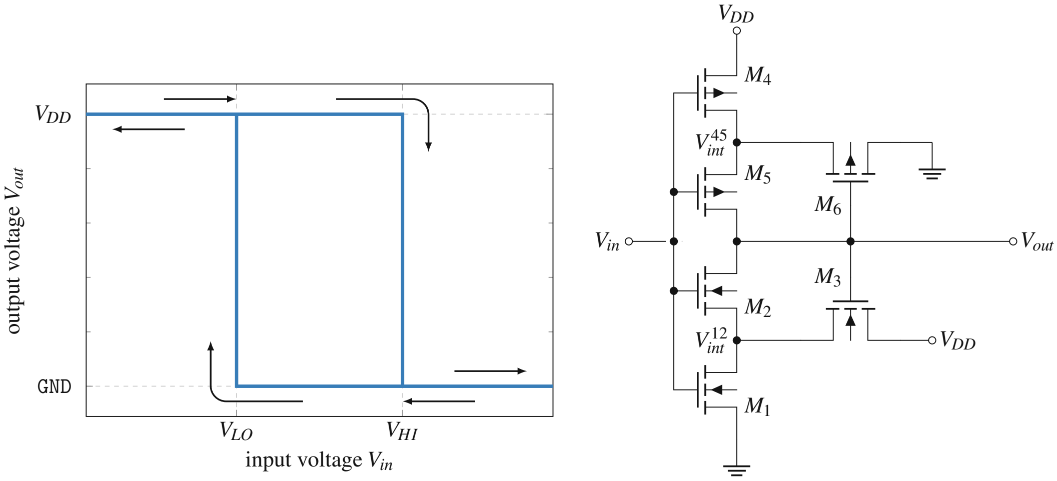
3.4.3 Uniform NOR Model

The uniform model for the NOR gate is very much comparable to previous examples (cf. Section 3.3.3). Similarly to the Inverter, we have to add a capacitance at the internal node, as shown in Figure 3.17a, to keep track of the respective voltage value. For the NOR gate, the overall system of equations is

$$\begin{aligned} C^1_{int} \cdot \frac{\mathrm{d}}{\mathrm{d}t} V^1_{int} &= I_{D,p}(V_{DD} - V_A, V_{DD} - V^1_{int}) - I_{D,p}(V^1_{int} - V_B, V^1_{int} - V_{out}) \\ C_L \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{out} &= I_{D,p}(V^1_{int} - V_B, V^1_{int} - V_{out}) - I_{D,n}(V_A, V_{out}) - I_{D,n}(V_B, V_{out}) \ . \end{aligned}$$

Note that the second input signal increases the complexity only marginally. Corresponding SIS simulation results in MACS are shown in Figure 3.17b, whereat we used $C_L/C^1_{int} = 4$. Although the trajectories, overall, match HSPICE results very well, some signals (especially the internal voltage V^1_{int}) show artifacts, which we suspect to be caused by numerical inaccuracies.

To conclude this section, we want to note that we also investigated other standard 2-input gates, e.g., OR and Muller-C. Since the application of the Uniform Model is according to the circuits already presented in the previous sections, we do not explicitly show the result here. In summary, writing down the respective equation is very simple, even with growing circuit sizes. Nevertheless, solving them becomes increasingly complicated, leading to improved simulation times. The achieved trajectories, nonetheless, show a physically reasonable behavior.



(a) Output hysteresis *(b) Implementation*

Figure 3.18: The six transistor S/T implementation.

3.5 Schmitt Trigger Model

The third and last circuit we analyze in detail is the single-input S/T. This gate has an inherent state, meaning that its behavior depends not only on the current input value but also on the complete input history. We will exploit this property in Chapter 5, where we investigate metastability.

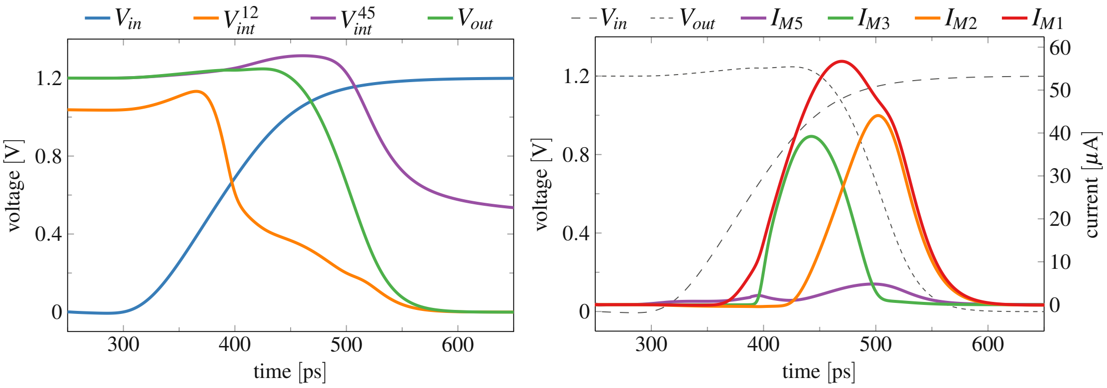
3.5.1 Introduction

The key property of an S/T is the fact, that the output depends on V_{in} via some hysteresis, as is shown in Figure 3.18a. More specifically, the value of V_{out} stays constant until a certain upper (V_{HI}) respectively lower (V_{LO}) threshold has been crossed.

The basic idea behind the S/T is to propagate an input transition only after it reaches a value that is reasonable close to V_{DD} respectively GND and thus filter smaller pulses automatically. This is in most cases desired, as such short pulses tend to drive digital circuits into undesired metastable states, which may not be resolved in bounded time. More on this topic follows in Chapter 5.

3.5.2 Behavioral Analysis

Many different implementations of an S/T are available in literature. A nice overview and review of some of these is presented by Dokic [51]. Basically two categories can be distinguished: inverting ($V_{out} = V_{DD}$ for low input values, cf. Figure 3.18a) and non-inverting ($V_{out} =$ GND for low input values). In the following, we will shortly discuss an inverting six transistor design shown in Figure 3.18b. A more thorough analytical investigation is provided by Filanovsky and Baltes [120].



(a) Voltage *(b) Current*

Figure 3.19: HSPICE simulation results of the S/T.

Assume the initial state V_{in} = GND and $V_{out} = V_{DD}$ where the transistors M_1, M_2 and M_6 are not conducting, M_4 and M_5 are fully conducting and M_3 is almost conducting. Note that the latter charges V_{int}^{12} until V_{GS} drops below $V_{th,n}$ and prevents any further charge carrier flow. In the sequel we investigate the effect of a rising input transition. HSPICE simulation results (voltages and currents) for this situation are shown in Figure 3.19a resp. Figure 3.19b.

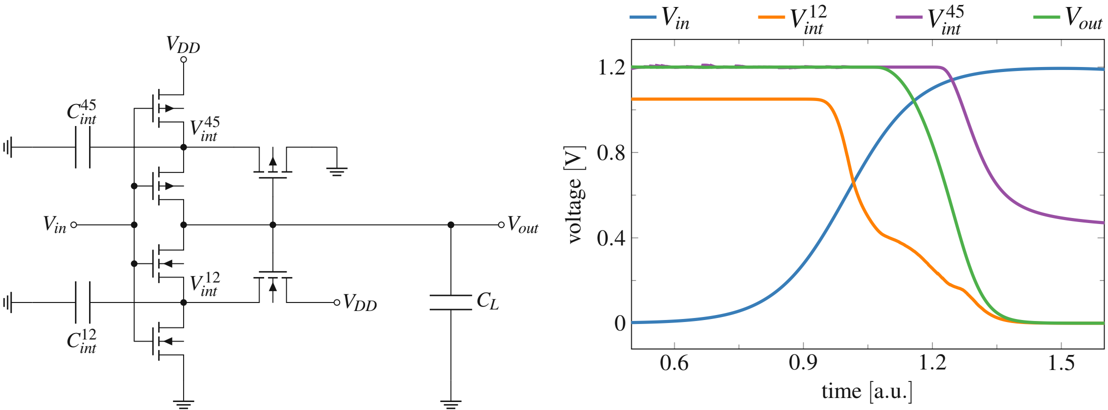
After V_{in} exceeds $V_{th,n}$ transistor M_1 slowly starts to conduct and thus creates a direct connection over M_3 between GND and V_{DD}, i.e., the voltage drop is split across these two transistors. The higher V_{in}, the more current M_1 is able to conduct for the same V_{DS} and thus the lower V_{int}^{12} gets. The exact values actually depend on the relative sizing of M_1 and M_3, so an appropriate layout is key in this implementation.

Eventually $V_{in} - V_{int}^{12} > V_{th,n}$ is satisfied, which causes also transistor M_2 to conduct. Although often interpreted as the upper threshold voltage V_{HI}, the corresponding value of V_{in} only marks the onset of direct current flow (see also Filanovsky and Baltes [120]). In Chapter 5 we will see that the difference to V_{HI} is actually quite significant.

In this situation all transistors are either in (OHM) or (SAT) such that the whole circuit serves as voltage divider. Consequently, V_{out} starts to drop slightly. To retain current equilibrium the voltage drop across the p-stack, whose conductivity decreases, has to be increased, which causes V_{out} and in consequence also V_{int}^{45} to decrease. At the tipping point, i.e., the threshold voltage V_{HI}, the falling V_{int}^{45} pushes V_{SG} of transistor M_5 below the threshold, dramatically reducing the conductivity and thus forcing V_{out} to drop even lower. This causes M_3 to conduct less such that M_1 can handle more current from M_2 and thus discharge V_{out} even faster. These processes amplify each other until the p-stack is completely cut off and V_{out} is pulled to GND.

When everything has settled, V_{in} is at V_{DD}, transistors M_1, M_2 are fully conducting, M_6 is almost conducting, while M_3, M_4 and M_5 are blocking. The analysis for the falling input transition can be carried out analogously and will thus not be shown explicitly.

Earlier we stated that the behavior of the S/T depends on the history of the input



(a) Implementation

(b) Simulation in MACS

Figure 3.20: Uniform Model representation of the S/T with $C_L/C^{12}_{int} = C_L/C^{45}_{int} = 4$. The results are in very good agreement to HSPICE simulations. V^{45}_{int} shows some simulation numerical artifacts.

value and thus the circuit has a state. In this implementation the state is visible on the values V^{12}_{int} and V^{45}_{int}, which are enforced by V_{out} over transistors M_3 and M_6. Recall that V_{in} has to exceed these values to initiate a transition.

3.5.3 Uniform Schmitt Trigger Model

For the Uniform Model of the S/T two capacitance at the two internal nodes (between M_1 and M_2 as well as M_4 and M_5) have to be added (see Figure 3.20a). To describe the respective voltage values V^{45}_{int}, V^{12}_{int} and V_{out} a system of three equations is sufficient:

$$\begin{aligned}
C^{45}_{int} \cdot \frac{\mathrm{d}}{\mathrm{d}t} V^{45}_{int} &= I_{D,p}(V_{DD} - V_{in}, V_{DD} - V^{45}_{int}) - I_{D,p}(V^{45}_{int} - V_{in}, V^{45}_{int} - V_{out}) - \\
&\quad I_{D,p}(V^{1}_{int} - V_{out}, V^{1}_{int}) \\
C^{12}_{int} \cdot \frac{\mathrm{d}}{\mathrm{d}t} V^{12}_{int} &= I_{D,n}(V_{in} - V^{12}_{int}, V_{out} - V^{12}_{int}) - I_{D,n}(V_{in}, V^{12}_{int}) + \\
&\quad I_{D,n}(V_{out} - V^{12}_{int}, V_{DD} - V^{12}_{int}) \\
C_L \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{out} &= I_{D,p}(V^{45}_{int} - V_{in}, V^{45}_{int} - V_{out}) - I_{D,n}(V_{in} - V^{12}_{int}, V_{out} - V^{12}_{int})
\end{aligned}$$

Once again C^{45}_{int} and C^{12}_{int} only describe internal capacitances and can be chosen much smaller than C_L. MACS results shown in Figure 3.20b utilize $C_L/C^{12}_{int} = C_L/C^{45}_{int} = 4$ and fit qualitatively very well to the HSPICE simulations presented earlier.

3.6 Simulation and Verification

A major advantage of the simple transistor models we presented at the beginning of this chapter is their relatively simple evaluation. Although the various states introduce significant challenges, simulations can be carried out using general purpose computation tools. Complementary, even verification is enabled, which can be utilized to prove that certain boundaries are not violated by the analog waveform. In this section we will present the different solutions we employed for this purpose.

3.6.1 MATLAB/Simulink

In MATLAB/Simulink it is possible to implement our hybrid models, either based upon the Basic Model or Elaborate Model (cf. Section 3.3.2), using state automata. The nice graphical editor allows to design the state graph just as it was presented earlier, while guards and invariants are automatically monitored and executed. The tool is even able to handle the complexity of the Elaborate Model such that rather accurate results close to HSPICE simulations can be achieved.

3.6.2 MAtlab Circuit Simulations (MACS)

Throughout this chapter multiple simulation results from our tool MACS, which is publicly available[6], have been presented. It is, as indicated by its name, a simulation framework based on MATLAB and is currently able to simulate simple circuits, output the results and create input files for a verification tool. The latter is especially important for the Compare Execute Check Engine (C2E2) [26] (that will be presented in the next section), which demands that the state specifications are written (i) in one line and (ii) without the possibility to define abbreviations. Recalling the Uniform Model, this means that the expressions for V_{GU} and V_{DU} have to be expanded and inserted at every single location. This may seem trivial, leads, however, to hundreds of exponential and logarithmic terms.

Creating the input files automatically ensures, that exactly the same circuit that is simulated in MATLAB is later verified, as possible errors during the implementation of the verification model are prohibited. At the moment, only C2E2 is supported but we are aiming to also include Flow* [45], dReach [36] or CORA [23]. Although MACS is still under construction it already contains:

- the Uniform Model current equations
- suitable parameters for technology (T65)
- reference implementations of basic gates
- scripts for running an evaluation

The gate library contains an Uniform Model description for various logic operations. Missing gates can be easily added, as they are built according to their transistor level

[6] https://github.com/jmaier0/macs

```
1  function [I_m,I_out] = NOR(V_a, V_b, V_m, V_out)
2
3      parameters;
4
5      I_N1 = uniform_model(V_a, V_out, 'N');
6      I_N2 = uniform_model(V_b, V_out, 'N');
7
8      I_P1 = uniform_model(V_DD - V_a, V_DD - V_m, 'P');
9      I_P2 = uniform_model(V_m - V_b, V_m - V_out, 'P');
10
11     I_m = vpa(I_P1 - I_P2);
12     I_out = vpa(I_P2 - I_N1 - I_N2);
13
14 end
```

Listing 3.1: NOR gate implementation in MACS. Note the strong connection between description and transistor level implementation, which simplifies the design.

implementation. Listing 3.1 shows exemplarily the NOR gate. The currents through every transistor are determined by the function $uniform\_model(V_{Gy}, V_{Dy}, tType)$ (lines 5-9). The parameter $tType$ specifies, whether an nMOS ('N') or pMOS ('P') shall be implemented. At last the currents are appropriately connected at each node (I_m and I_{out}), using the function *vpa* (lines 11-12), which does a symbolic evaluation.

In a separate script (shown in Listing 3.2) the gate is instantiated (line 9) and augmented by capacitances (lines 10-11). Together with the currents determined earlier a system of ODEs is specified (lines 16-22) and then numerically evaluated (line 23). In the shown code snippet a pulse on input A, which is created by adding shifted sigmoids (more on this follows in Section 3.7.4), and a constant value on input B is simulated.

For the future we are planning to merge MACS with already existing tools and further extend its capabilities by built-in mechanisms of MATLAB.

3.6.3 Compare Execute Check Engine (C2E2)

For additional simulation and potential verification, we resorted to the C2E2. This tool was designed by Fan et al. [26] to verify hybrid automata using so-called discrepancy functions. For this thesis, we ran a hybrid Inverter model utilizing the Elaborate Model and models of elaborate gates based on the Uniform Model in C2E2.

Input Generation

C2E2 solely operates on hybrid automata using behavioral descriptions by differential equations. This also includes input signals, forcing us to develop suitable descriptions for the desired shapes: a linear input slope (`Ramp`) and a sigmoidal shape (`Sig`) have

```
%             V_m    V_out
V_init = [V_DD V_DD];
vars=[V_m V_out];
C = [C_m C_L];

% -----------------------------------------------------------------
% MODEL SECTION

[I_m, I_out] = NOR(V_a, V_b, V_m, V_out);
I_m = I_m / C_m;
I_out = I_out / C_L;

% -----------------------------------------------------------------
% SIMULATION SECTION

baseODE = odeFunction([I_m;I_out], vars, [V_a(t) V_b(t)]);

V_a = @(t) ( V_DD./(1+exp(-(a(1)*(t-c(1))))) ) + ...
      ( V_DD./(1+exp(-(a(2)*(t-c(2))))) ) - V_DD;
V_b = @(t) min(1*t,0); % constant value

F = @(t,vars) baseODE(t,vars,[V_a(t),V_b(t)]);
[t,V] = ode113(F, tspan, V_init);

% -----------------------------------------------------------------
```

Listing 3.2: NOR gate simulation in MACS.

been realized. To generate the former, a 4-state implementation is required, which is shown in Figure 3.21a. Starting in state *Up* the input derivative is set to 1, resulting in a linear increase with time. The guard $V_{in} \geq V_{DD}$ assures, that a transition to state *High* is triggered as soon as V_{DD} is reached. In this state the input is kept constant for 2 time units before dropping down again, and staying low for the same amount of time.

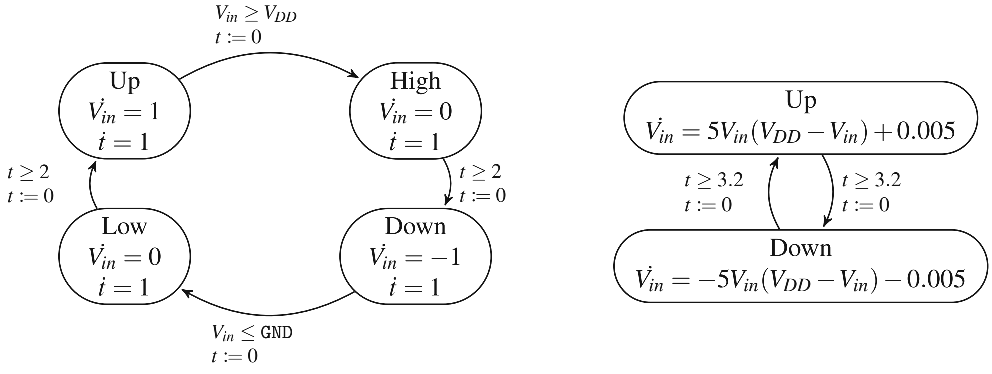
The automata for the `Sig` input (shown in Figure 3.21b) has a more elaborate description. Due to the fact that sigmoids never reach their final value (see also Section 3.7.3), it is possible to reduce the state count to 2. Beneficial for our purposes is, that the time derivative of a sigmoidal function

$$u(t) = V_{DD} \cdot (e^{-k \cdot t + d} + 1)^{-1}$$

can be written as

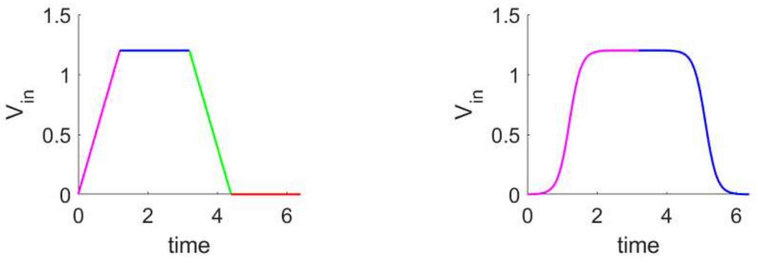
$$\frac{\mathrm{d}}{\mathrm{d}t} u(t) = k \cdot u(t) \cdot [V_{DD} - u(t)] \ ,$$

which is implemented in the figure for $k = \pm 5$. Note that an additive term ± 0.005 was used to (i) ensure $\dot{V_{in}} \neq 0$ for $V_{in} = 0$ and thus leave the initial state at all, and (ii) to reach the final value faster. Figure 3.22 shows the resulting analog waveforms for both inputs.



(a) Ramp *(b) Sigmoid*

Figure 3.21: Input generation automata for C2E2.



(a) Ramp *(b) Sigmoid*

Figure 3.22: C2E2 input traces. The colors mark the different states of the automata. Note that for `Sig` only symmetric pulses are possible, i.e., where up and down transition have the same slope.

Verification

We utilized C2E2 to successfully verify `InvHy` using the Elaborate Model as well as Inverter (`InvUni`), NOR-gate and OR-gate models based on the Uniform Model. In all cases the unsafe set was defined as $V_{out} > 1.32\,\mathrm{V}$, the time horizon as 6.4 s and an uncertainty in the initial value of V_{in} was introduced. Unfortunately, only very limited values were possible for the latter, as the tool quickly experienced numerical issues otherwise. Actually, the uncertainty rapidly declines at the beginning of the simulation, leading to almost deterministic simulations comparable to MACS.

Figure 3.23 shows the simulation results for the hybrid Inverter model `InvHy`. Overall we end up, considering the 7 states of the Inverter model and the 4 respectively 2 states of the `Ramp` respectively `Sig` automata, with $7 \times 4 = 28$ modes in the `Ramp` case and $7 \times 2 = 14$ in the `Sig` case. For the circuits based on the Uniform Model the number of states is determined solely by the input automata alone. This is clearly visible in the Inverter simulation results shown in Figure 3.24.

To investigate the scaling capabilities of C2E2, NOR- and OR-gate have been imple-

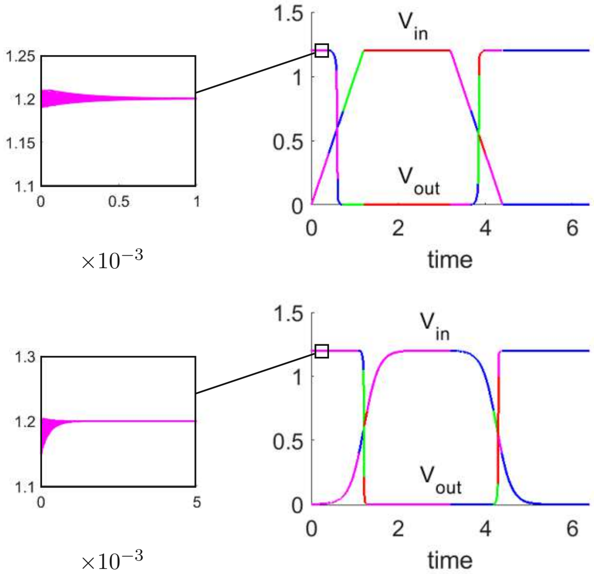


Figure 3.23: `InvHy` output voltage over-approximation set for V_{in} = `Ramp` (top) and V_{in} = `Sig` (bottom). Clearly visible are the states in V_{out} encoded by different colors.

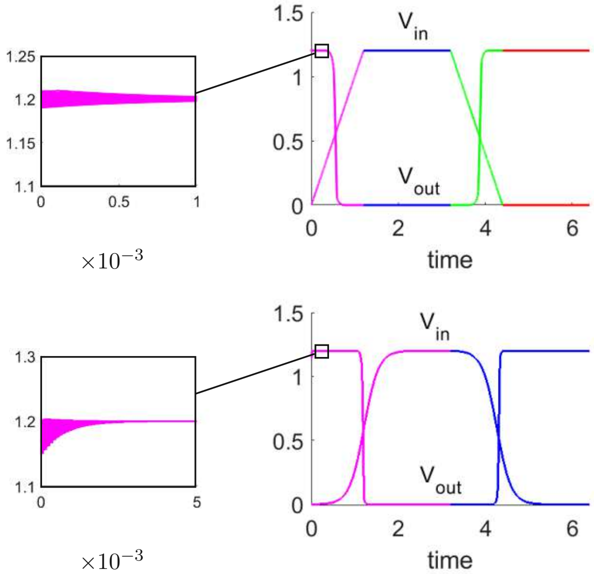


Figure 3.24: `InvUni` output voltage over-approximation set for V_{in} = `Ramp` (top) and V_{in} = `Sig` (bottom). Due to the fact that Uniform Model describes the behavior with a single equation the amount of states is significantly less.

Model	Verification parameters		Timing split [s]			time [s]
	Steps	Initial Set	Sim.	Discr.	I/O	
`InvHy`	128k	$V_{out} \in [1.15, 1.2]$	111	33	79	223
`InvUni`	64k	$V_{out} \in [1.15, 1.2]$	58	124	29	211
NOR	320k	$V_{out} \in [1.15, 1.2]$	396	1750	179	2325
OR	320k	$V_{nor} \in [1.199, 1.201]$ $V_{out} \in [0, 0.002]$	943	1722	148	2813
`InvHy`	128k	$V_{out} \in [1.15, 1.2]$	118	39	78	235
`InvUni`	64k	$V_{out} \in [1.15, 1.2]$	30	127	20	177
NOR	320k	$V_{out} \in [1.15, 1.2]$	168	1698	101	1967
OR	320k	$V_{nor} \in [1.199, 1.201]$ $V_{out} \in [0, 0.002]$	443	1778	89	2310
`InvLoop`	64k	$V_1 \in [1.0, 1.2]$ $V_2 \in [0.5, 0.6]$	27	224	5	256

Table 3.7: Verification time of `InvHy`, `InvUni`, NOR-gate and OR-gate with `Ramp` (top) and `Sig` (bottom) input and `InvLoop` without input on a laptop with standard configuration (8G RAM, Intel Core i5 CPU). All verification results are safe.

mented as well, whereat the latter is simply obtained by appending an Inverter to the former. Thus, both circuits can be evaluated at the same time. The simulation results for V_{nor} (after the NOR gate) and V_{out} (after the Inverter) are shown in Figure 3.25.

We also investigated an Inverter loop (`InvLoop`; similar to Figure 3.12a without the driving Inverter). In contrast to the other evaluated circuits, this one does not have an external input. Consequently, we just set the initial voltage values at the nodes and started the simulation. The results are as expected and thus not explicitly shown.

In general, all the simulation results are qualitatively comparable to HSPICE and show smooth output transitions even when being activated by a ramp. Verification shows that, despite initial state uncertainty, the traces quickly converge to a deterministic signal trace. The total verification time, split between simulation (Sim.) and discrepancy computation (Discr.), is shown in Table 3.7.

3.6.4 Evaluation

At this point, we recall our initial goal, i.e., to model the analog waveform in an abstract fashion, in order to handle larger circuits as well. Is this possible with the approaches presented so far? Unfortunately the answer is *No*! Although the complexity of the analog models compared to modern HSPICE models, for example, was significantly decreased, the evaluation effort is still too high to be extended to thousands of gates. Thus we have to conclude, that way more drastic abstractions are required, in order to possibly achieve our goal.

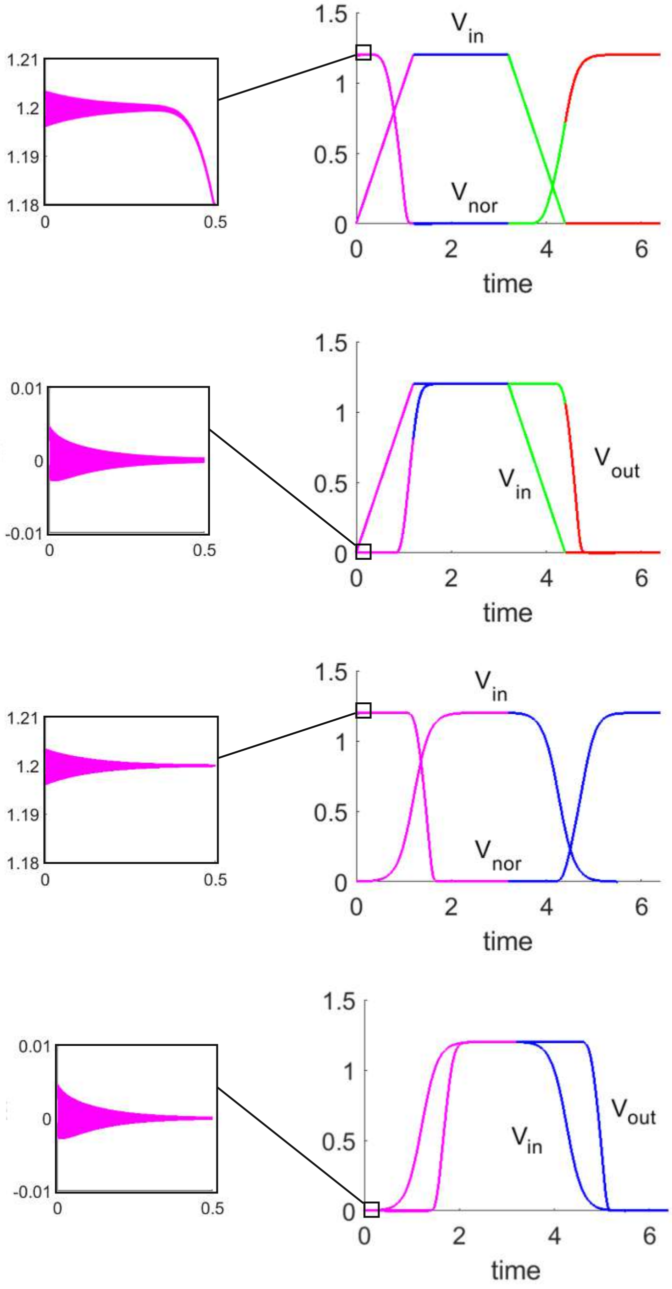


Figure 3.25: OR/NOR gate V_{nor} and V_{out} over-approximation set for V_{in} = `Ramp` *(top half) and V_{in} =* `Sig` *(bottom half). Shown are the trajectories for V_{nor} (after the NOR gate) and V_{out} (after the Inverter).*

Nevertheless, our analyses are not in vain. The insights gained upon the physical processes governing the behavior of logic gates will be used in Chapter 4 and Chapter 5 to optimize delay and metastability estimations. In addition, in the remainder of this chapter, the simulation results of these simple models provide the basis for further analog abstractions.

3.7 Analog Trace Abstraction

The conclusion of the previous chapter is devastating: MATLAB, and thus most probably also tools with comparable performance, are by far not able to execute fully-fledged analog simulations in reasonable time, even for very simplistic models. The question that naturally arises is thus, whether it is possible to introduce higher level abstraction that enable high accuracy descriptions with far less effort. In the sequel several approaches towards that goal are introduced and evaluated.

Although there are infinitely many ways for a signal to bridge the gap between GND and V_{DD}, detailed analog simulations reveal only a very small number of waveforms at any specific node in the circuit. This has several reasons: (i) The shape of the input signal has no impact on gates deep inside the circuit since each gate slightly alters the analog waveform until an equilibrium is reached. (ii) The waveforms are largely influenced by the preceding and succeeding logic, as these determine, for example, the input slope and the output capacitance. The combination of these values is unique within a circuit and, consequently, so are the analog traces.

Our approach for describing the small set of possible waveforms at a specific circuit node is to utilize the unique rising ($f_\uparrow$) and falling ($f_\downarrow$) Full-Range Switching Waveform (FRSW), i.e., a transition that starts at one extreme value (V_{DD} or GND) and ends at the other. Obviously, regular rail-to-rail switchings can be modeled very accurately, however, pulses are much harder to handle. Nevertheless, we developed good approximations even for these challenging cases by carefully combining $f_\uparrow$ and $f_\downarrow$.

3.7.1 Full-Range Switching Waveform from Simulation

The first task is naturally to characterize the FRSWs. The straightforward approach, i.e, running analog simulations and extracting the waveforms is very cumbersome as:

1. we only thrive to investigate a single gate, which may be hidden deep inside the circuit. Thus identifying a proper input constellation that leads to a single transition at the input of that gate (in both directions) is potentially a very hard task.

2. the computational effort may be very high for the simulation of the whole circuit.

These problems can be circumvented by utilizing the logic in close proximity to the gate of interest. Simulations revealed, that extracting a rather limited area around the gate is sufficient to achieve the same results as if the whole circuit was evaluated.

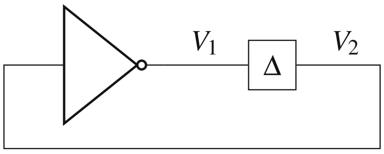


Figure 3.26: Simulation setup for determining $f_\uparrow$ and $f_\downarrow$ of the Inverter. The pure delay component Δ is required to achieve a reasonable time distance between succeeding transitions, such that interference (pulse creation) can be avoided.

In regular structures, like Inverter chains, characterization becomes even easier. In detail one can feed the output of a single Inverter back to its input until unique $f_\uparrow$ and $f_\downarrow$ are repeatedly observed. It is guaranteed that such waveforms exist, since slow input transitions lead to a faster output signal while fast inputs are slowed down at the output. The setup of an according experiment is shown in Figure 3.26.

3.7.2 Eigenfunctions

Complementary to extraction from simulations, we are also looking for an analytic approach to calculate $f_\uparrow$ and $f_\downarrow$. For this purpose imagine an infinite chain of the same Inverter: A single transition at the input of the first Inverter leads, eventually, to an output transition at each single Inverter. From a certain position onwards, one can expect to observe the same $f_\uparrow$ and $f_\downarrow$ down the chain, i.e., they recreate each other. In the sequel we will hence focus on such *Eigenfunctions*.

Eigenfunction from Static Characteristics

To reduce complexity, we first aim at determining the Eigenfunction solely based on static considerations. Obviously, this does not deliver the most accurate results, but hopefully provides a good intuition of a proper abstraction. For an Inverter, for example, the main challenge is to determine V_{in}, such that $V_{out}(V_{in})$, based on the static transfer characteristics (cf. Figure 3.5), is equal to $V_{DD} - V_{in}$.

The relationship $V_{out} = V_{DD} - V_{in}$ implies that each voltage range $[V, V + \Delta V]$ at the input has to be passed at the same rate as $[V_{DD} - V, V_{DD} - (V + \Delta V)]$ at the output. Recall for this purpose, that the Inverter can be seen as an amplifier when both transistors are in (SAT). Assuming an amplification factor $k > 1$, an input change of ΔV thus leads to an output change of $k \cdot \Delta V$. The fact, that this statement must be also valid for the equilibrium point $V_{out}(V_m) = V_m$, shows the different rates of change at in- and output. Consequently, static considerations are, unfortunately, not sufficient to derive Eigenfunctions.

Eigenfunction with Zero Delay

The static considerations have shown that, it is mandatory to consider the dynamics of the gate. The simplest approach is to utilize the `InvHy` model and start with the assumption of a zero-delayed output signal, i.e., $V_{out}(t) = V_{DD} - V_{in}(t)$. Unfortunately this quickly leads to severe problems:

(i) In this model the output has to follow the input immediately. In a real circuit, however, there is always some time shift between in- and output as the threshold voltage has to be crossed before the transistors start to conduct (cf. Section 3.3).

(ii) Recall the two-dimensional representation of the states shown in Figure 3.9. The condition $V_{out} = V_{DD} - V_{in}$ results in a straight line between the top left and lower right corner and thus forces the state evolution $A \to C \to D \to F \to G$. This path contains, however, the state transition from A to C which was shown to be physically unreasonable and thus inappropriate.

(iii) Calculating the Eigenfunction for a particular state leads to unreasonable results for certain value ranges. For example, using the Basic Model in state G the time derivative of V_{out} for $V_{out} > \frac{2}{3}(V_{DD} - V_{th,n})$ is positive and increases with increasing V_{out}. This is not physically reasonable despite input values larger than $\frac{2}{3}(V_{DD} - V_{th,n})$ are well possible in state G.

(iv) There exists a value $V_M \in [\mathtt{GND}, V_{DD}]$ such that for each initial condition $V_{in} < V_M$, and consequently $V_{out} > V_{DD} - V_M$, the stable point $(V_{in}, V_{out}) = (\mathtt{GND}, V_{DD})$ is approached. For initial conditions $V_{in} > V_M$ the final value results to $(V_{in}, V_{out}) = (V_{DD}, \mathtt{GND})$ [7]. For this reason a full range switching waveform is impossible to achieve. More information on this topic will be provided in Section 3.7.4.

Due to these problems it is not possible to derive the Eigenfunctions of an Inverter in this fashion. Actually, full range switching waveforms in `InvHy` are only possible when the model is crucially reduced, in detail by completely removing either the n- or pMOS. This results in two separate descriptions for the Inverter, one for a rising output transition and one for a falling one, whereat both have to be expected to be very inaccurate. Although we were actually able to fully specify the model and simulate pulses, we did not pursue this approach any further due to its very bad characteristics (e.g., the trace below/above V_M for rising/falling transition has to be approximated).

Eigenfunction with Limited Delay

In the preceding analysis, the assumption $V_{out}(t) = V_{DD} - V_{in}(t)$ caused severe problems. Thus we conclude, that a certain delay between in- and output, as it is also observed in physical systems, is a necessary prerequisite. Let

$$\frac{\mathrm{d}}{\mathrm{d}t} V_{out}(t) = F[V_{out}(t), V_{in}(t)] \tag{3.8}$$

[7] In Chapter 5 we will identify V_M as the metastable voltage.

be the ODE describing the behavior of a buffer. To obtain the ODE for the Eigenfunction we have to set $V_{in}(t) = V_{out}(t + \Delta)$ with $\Delta > 0$ being the time shift between in- and output. Replacement in (3.8) leads to

$$\frac{\mathrm{d}}{\mathrm{d}t} V_{out}(t) = F[V_{out}(t), V_{out}(t + \Delta)]. \tag{3.9}$$

Since the future value at time $t + \Delta$ is required to calculate the derivative at present time t, we end up with a retarded ODE, which is, unfortunately, at the moment only solvable for linear equations. Nevertheless, we are cautiously optimistic that it is possible to derive solutions also for non-linear ODEs, have, however, not yet succeeded in solving this hard problem. Other approaches, like rearranging the equation to achieve an explicit form for $V_{out}(t)$, failed as well.

One possibility to avoid retarded ODEs is to calculate the Eigenfunction in the reverse direction, i.e., by defining the final values and going back in time. In this case (3.8) transforms to

$$\frac{\mathrm{d}}{\mathrm{d}t} V_{out}(t) = -F[V_{out}(t), V_{out}(t - \Delta)]$$

which is solvable. The result has to be interpreted again time-reversed to achieve the correct Eigenfunction. For example, for the rising Eigenfunction a falling transition is derived which must not be confused with the falling Eigenfunction.

The main problem with this approach is the accurate definition of the Eigenfunction in the start interval $[t_0 - \Delta, t_0]$ (actually the tailing in the natural time direction) as it, and the trace resulting from it, have to fulfill several properties and conditions:

- The derivative at time t depends on the value of V_{out} at time t and $t - \Delta$. This is already a very stringent property, which renders a lot of imaginable traces impossible, as all three values have to match at each point in time. Please note that Δ is constant along the whole trace, which implies that the newly defined value at time t is used to determine the slope at time $t + \Delta$, which again has to fit.
- The extension of the start interval for $t \to +\infty$ has to approach the correct value, i.e., either V_{DD} or `GND`. This is actually a very challenging task since the signal shape can only be controlled by the values chosen in the initial interval.

Consequently, whereas it is in theory possible to determine the Eigenfunction using this time-reversed approach, for practical applications, a suitable initial interval is way too hard to find.

3.7.3 Approximation Functions

Since the exact calculation of $f_\uparrow$ and $f_\downarrow$ turned out to be very challenging, we are focusing in the sequel on approximations using specific families of analytic functions. The main advantage is, that it is not necessary to store the whole waveform but instead single parameters are sufficient to calculate the required values on demand.

Observing typical HSPICE simulations for a FRSW (cf. Figure 3.6) reveals a continuous function, whose derivative gradually in-/decreases until an intermediate value is reached. From there onward the behavior is inverse, i.e., the derivative de-/increases to slowly approached the final value. In the course of our research we investigated some promising candidates which, more or less, match this overall description.

Exponential Function

At a first sight an exponential does not fit our demands well, since it experiences the highest derivative already at the start. Nevertheless, functions of the form

$$f_{\downarrow}(t) = V_{DD} - f_{\uparrow}(t) = V_{DD} \cdot e^{-k \cdot t}$$

are very well suited to approximate later parts of the waveform, which determine how the final value is approached.

Sigmoid Function

A sigmoid, whose switching behavior can be described by

$$f_{\uparrow}(t) = V_{DD} - f_{\downarrow}(t) = V_{DD} \cdot (e^{-k \cdot t + d} + 1)^{-1} ,$$

is primarily interesting due to its smooth shape, which is very close to actual FRSWs from start to end. This was already noticed in the past, as Plahte, Mestl, and Omholt [112] showed how a complete logic can be built on top of sigmoids.

Unfortunately, even the sigmoid is not ideal: (1) Start- and endvalue are only approached but never reached, which makes it hard to properly define a start point of the transition. This problem can be circumvented by using the $V_{DD}/2$ crossing time $(t = d/k)$. (2) The function is symmetric, which implies that it takes the same amount of time to bridge the gap between an arbitrary voltage V_x and $V_{DD}/2$ in both directions.

To depict the severe consequences of the latter consider this (exaggerated) example: Assume a rising transition that occurred one hundred years ago. Since the waveform has an exponential shape it has not reached V_{DD} yet. Switching to $f_{\downarrow}$ right now would thus take another one hundred years to hit $V_{DD}/2$, which is simply not reasonable.

Hill Function

Due to the shortcomings of the sigmoid we searched for a waveform that is still close to an actual FRSW but in addition also asymmetric. As one possible candidate we identified the Hill Function [153], which can be described by

$$f_{\uparrow}(t) = V_{DD} - f_{\downarrow}(t) = V_{DD} \cdot \frac{t^n}{k^n + t^n}$$

It has several advantages:

1. $f_{\uparrow}(0) = \texttt{GND}$ and $f_{\downarrow}(0) = V_{DD}$

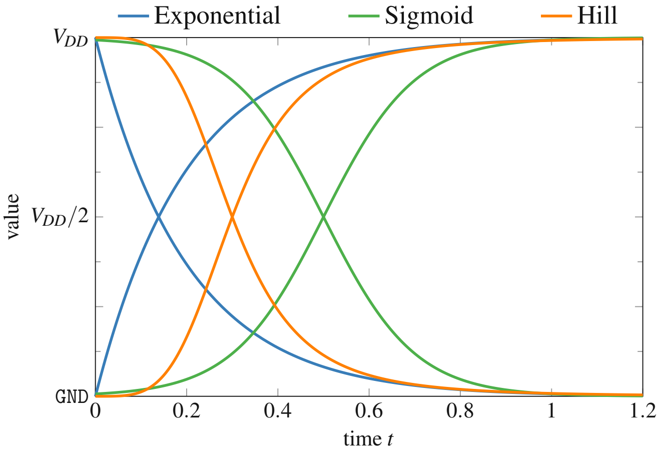


Figure 3.27: Comparison of our approximation functions. The beginning of the Exponential function is very steep and not according to reality. The other two continuously change their derivative whereat the asymmetry of the Hill function can be seen clearly. The final value is approached by all in the same fashion. Please note that the Sigmoid has a value > 0 *at time zero.*

2. The time to reach $V_{DD}/2$ is adjustable by the parameter k as $f_{\uparrow}(k) = f_{\downarrow}(k) = V_{DD}/2$. The exponent n solely determines the curvature, whereat a high value results in a very steep curve.

3. The function is asymmetric, i.e, the initial value is quickly left whereat the final value is slowly approached. This is actually very close to physical reality.

Figure 3.27 shows examples for all the approximation functions introduced so far.

Generating Arbitrary Waveforms in HSPICE

In order to incorporate arbitrary waveforms, like our approximation functions, in HSPICE simulations, one can use a *voltage controlled voltage source* (Exxx), which allows to determine any function using the *vol* argument. For example, a sigmoid is realized between Terminals 1 and 0 by

```
E1 1 0 vol='VDD/( 1+exp( a*(-TIME+c) ) )'
```

3.7.4 Pulse Modeling

While having an accurate description of the FRSWs is already a big success, it is not sufficient to depict the complete behavior of real circuits. Analog waveforms may reverse their direction before the opposite value (GND resp. V_{DD}) was reached, forming a pulse. For accurate results we thus need the possibility to model such pulses. In the sequel we will analyze how this can be achieved.

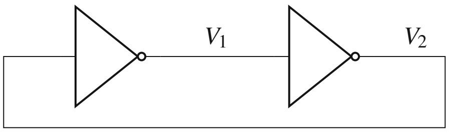


Figure 3.28: Layout of an Inverter loop used to simulate pulses using `InvUni`*.*

General Considerations

To generate pulses from FRSWs we initially investigated the possibility to manipulate the second derivative, e.g., by multiplying it with -1. The achieved results are, not surprisingly, unreasonable. Just imagine a rising FRSW at the output, whose first derivative has the shape of a positive pulse and the second shows the first period of a sine wave. Flipping the latter in the second half would result in $V''_{out} > 0$ and consequently an increasing V'_{out} resulting in an ever increasing V_{out}.

Simulations using the Uniform Model

Deriving pulses automatically from our simplified transistor models would make it possible to either (i) record proper pulse shapes or (ii) calculate them on demand. We thus investigated this possibility by properly initializing an Inverter loop (see Figure 3.28) and running a simulation based on the Uniform Model.

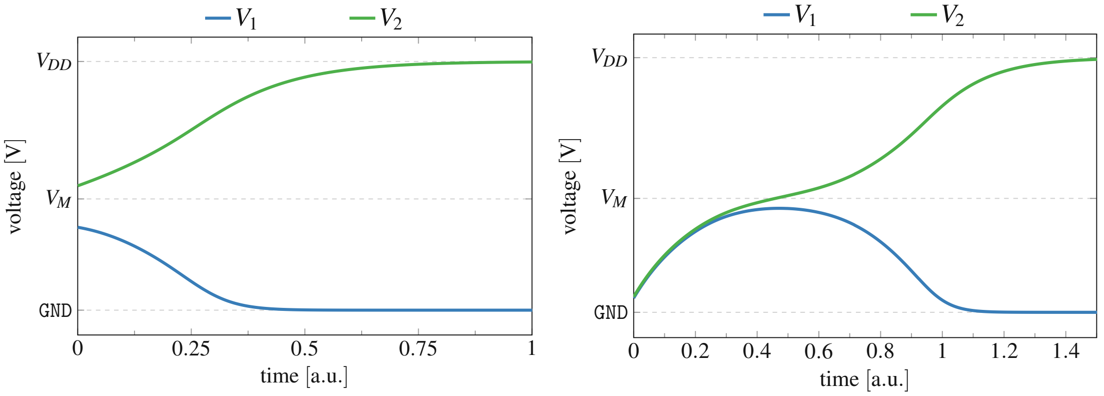
By recalling the static transfer function of an Inverter (cf. Figure 3.5) it should become clear, that for $V_1 = V_2 = V_M$, with $V_{out}(V_M) = V_M$, the voltage values will not change over time[8]. For $V_1 < V_M < V_2$ the stable state $(V_1, V_2) = (\mathtt{GND}, V_{DD})$ is approached, while $V_1 > V_M > V_2$ resolves to $(V_1, V_2) = (\mathtt{GND}, V_{DD})$. In these cases V_1 and V_2 support each other, such that quick resolution is guaranteed. Nevertheless, pulses are not achievable as V_M is never crossed (see Figure 3.29a for simulation results).

If both V_1 and V_2 are on the same side of V_M the behavior changes significantly (cf. Figure 3.29b). In this case both move initially in the same direction and thus compete against each other. Eventually one signal conquers. It keeps on moving in the initial direction, while the other signal is pushed in the opposite one. Please note that the node "winning" the competition is the one that started closer to V_M. If both start at exactly the same value, the point $(V_1, V_2) = (V_M, V_M)$ is approached. Figure 3.30 shows this resolution behavior graphically. Although pulse shapes are finally achieved they still do not cross V_M (see Figure 3.29), rendering it an incomplete description.

Adding Full-Range Switching Waveforms

Since all our efforts so far failed to model pulses accurately, we present, at last, a method that is heavily based on fitting without providing any insights on the physical processes. As this clearly contradicts the main goal of this thesis, we only roughly sketch the general idea, also because the actual implementation is still a topic of ongoing research.

[8] This is actually the metastable state that we will discuss in Chapter 5.



(a) $V_1 < V_M < V_2$

(b) $V_1 < V_2 < V_M$

Figure 3.29: MACS simulations of the Inverter loop. Depending on the initial conditions either direct resolution (left) or pulse creation (right) can be observed.

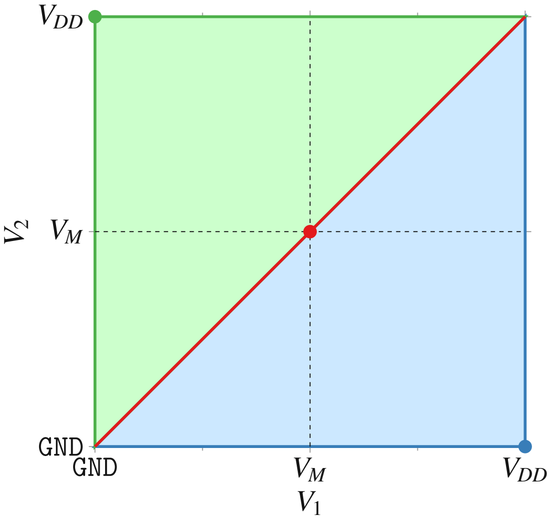


Figure 3.30: Convergence Plane for starting values V_1 *and* V_2 *of the Inverter loop. For all points in the blue area the stable point* $(V_1, V_2) = (V_{DD}, \mathit{GND})$ *is approached, while for the ones in the green area* $(V_1, V_2) = (\mathit{GND}, V_{DD})$ *is achieved. A special case are the points on the red line* $V_1 = V_2$*, which end up in* $(V_1, V_2) = (V_M, V_M)$*.*

Considering the unique FRSWs, large pulses can be quite accurately described by following the respective waveform and, at some point, switching to the other one. Applying this approach to shorter pulses as well, i.e., by reducing the temporal distance between the FRSWs, reveals, that a pulse can be interpreted as a continuous transition from $f_\uparrow$ to $f_\downarrow$ and vice versa.

Consequently, the waveform at the beginning and the end of the pulse approaches the FRSWs, which raised the question whether the region in between can also be approximated

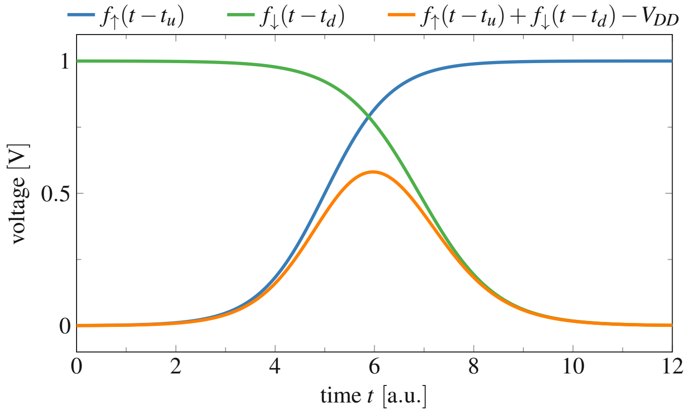


Figure 3.31: Creating pulses by properly adding shifted versions of the FRSWs. In this figure sigmoids were used. For $t_d - t_u > 0$ an up pulse is created, otherwise a down one. For asymmetric FRSWs, values below GND*/above V_{DD} due to incorrect time shifts might appear, which have to be handled with care.*

by (a suitable combination of) $f_\uparrow$ and $f_\downarrow$. The solution we derived is a simple addition

$$V_{out}(t) = f_\uparrow(t-t_u) + f_\downarrow(t-t_d)\ [-V_{DD}]\ ,$$

where t_u and t_d represent reasonable time shifts. The bigger $|t_u - t_d|$ the bigger the pulse. Note that the subtraction of V_{DD} is only required for pulses starting and ending at GND. These can be achieved by $t_d > t_u$, while for ones starting and ending at V_{DD}, $t_d < t_u$ is required[9]. An example trace is shown in Figure 3.31.

We verified this approach via HSPICE simulations of an Inverter, whereat the FRSWs were extracted from analog simulations (note that in this case $f_\uparrow \neq V_{DD} - f_\downarrow$). The pulses were then fitted by moving $f_\uparrow$ and $f_\downarrow$ against each other, adding them up and calculating the difference to the HSPICE simulation. The optimal fitting, i.e., the configuration with the minimal error, was finally stored. This procedure was repeated for each single pulse.

It turns out that the achievable approximations fit qualitatively very well to HSPICE (see Figure 3.32), whereat the largest errors are observed for small pulses. This can be explained by the low signal slopes, which can not be properly modeled here. Consequently the maximum/minimum value of the pulse is over-/underestimated and thus might result in digital transitions although the analog trajectory actually stays below/above the threshold value. Note that a similar effect can also be observed for large pulses, however, much less pronounced.

The achieved results indeed suggest, that it is possible to model the analog waveform solely by knowing the FRSWs and their respective time shifts. To predict the propagation of analog traces throughout the circuit it is, however, additionally required to predict the parameters of the output waveform based on those on the input. Figure 3.33 shows the relationship between them for the investigated Inverter.

[9] Note that this is strictly only valid for symmetric waveforms, i.e, $f_\uparrow = V_{DD} - f_\downarrow$.

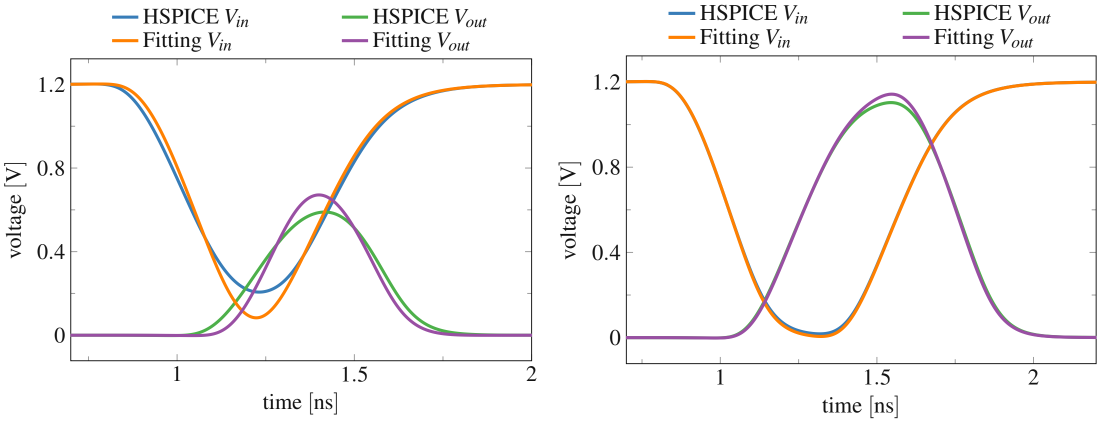


Figure 3.32: Fitting of pulses achieved by adding FRSW to HSPICE simulations. For larger pulses the fitting is much better compared to smaller ones.

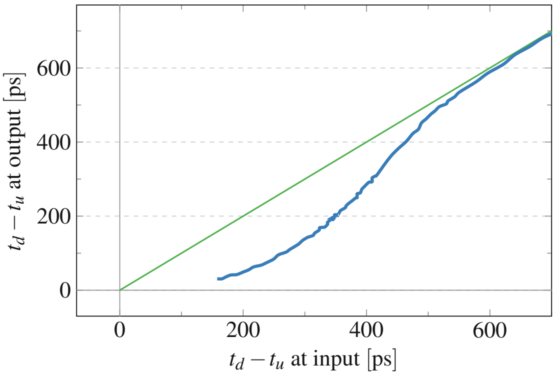


Figure 3.33: Relationship between $t_d - t_u$ at output and input. The green line shows the first median and indicates the steep decrease of the output parameters. Most changes are caused by a shift of the second transition while the first one only slightly deviates.

Although a mapping is clearly possible, we were not yet able to work out the details. Currently, there is a master thesis in progress with the goal of developing a fully automatic characterization tool. It determines, based on HSPICE simulations, proper fittings and a mathematical function to translate from input to output parameters. The main difficulty that is encountered here is the very large search space of possible fitting functions and their corresponding parameter values as well as the fair evaluation of the fitting error. Based on this tool, we are very optimistic that the development of a simplified analog simulation suite is possible. By propagating very few waveform parameters, it is, ideally, possible to approximate the analog behavior throughout the circuit. Due to the low complexity, the results are expected to be reasonable accurate, while only demanding a fraction of the run time compared to HSPICE. For this reason, we consider this a very promising path towards executing analog waveform analysis for larger circuits.

CHAPTER 4

Delay Modeling

The results of our various attempts to reduce the complexity of analog models in the previous chapter indicate, that there is not much hope to scale them up to large circuits. Discrete-state continuous-time models, based on delay estimations, have the potential for a viable alternative. In fact, digital delay estimation is a crucial task in modern circuit development. Representing analog signals by zero time transitions happening when some threshold value V_{th} is crossed[1] by the analog waveform, enables on one hand the analysis of large circuits in a short amount of time but on the other hand results in a significant loss of information. Thus accurate delay models are instrumental for any attempt to faithfully cover a circuit's behavior in the digital domain.

In this chapter we will thus focus on the question how digital delay prediction methods can be enhanced using physical considerations. At first, we provide a short introduction of state-of-the-art delay estimation methods, whereat the IDM will be explained in detail. We then thoroughly analyze whether/how the IDM matches the behavior of real circuits, which reveals several interesting and important facts. The gathered information enables us to enhance the approach by (i) calculating the utilized delay functions, (ii) automatizing the simulation of a circuit in our InvTool using the IDM, (iii) evaluating more complex logic gates and comparing the predictions to analog results, (iv) adding non-determinism, which is especially interesting for formal verification, and (v) relaxing some very stringent constraints, which enlarges applicability and simplifies the overall model characterization effort.

We want to emphasize, that the Involution Delay Model is not supposed to be a replacement for existing delay estimation methods, but rather an enhanced alternative. Although it provides, in general, a behavioral description that is very close to physical reality, and thus enables the identification of a wide range of malicious behaviors, it is also computationally expensive. Overall, simplistic approaches might be sufficient for early rough estimations, whereat, the IDM might be more suitable to evaluate all the

[1] Note that this corresponds to $V_{LO} = V_{HI} = V_{th}$ from Chapter 1.

possible behaviors of single critical components. In any case, digital approaches are only able to indicate a possible upset. If it actually manifests for the given application has to be verified with precise analog simulations or measurements.

4.1 Introduction

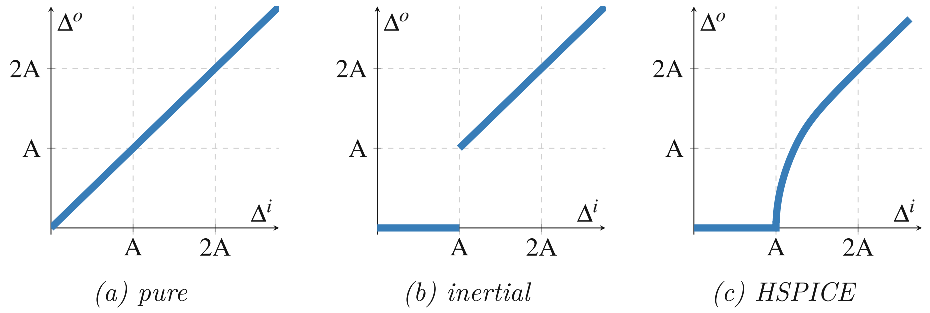
The term *delay* describes, in the digital domain, the time difference δ between an input transition and the resulting output transition. The task of a delay estimation model is to predict δ, which obviously depends on the signal propagation and possible variations throughout a circuit. To reduce the simulation time, reasonable delay values are, in general, not calculated on the fly but picked during a preceding characterization phase.

Delay estimation has made huge progress since very early macro modeling approaches, for example by Brocco, McCormick, and Allen [133] in the year 1988, or simulation tools, e.g., by Bryant et al. [135] in 1987. More and more effects such as the overshooting at the beginning of a transition [46, 56, 72], input-to-output coupling and transistor gain [121], input slope effects [99, 129], short channel effects [101], crosstalk at the output [88] and proper modeling of the parasitic RC load network [131] have been considered. While some approaches purely rely on analog simulations and fittings [25, 33], others resort to analytic calculations of simplified transistor models (cf. Section 3.2) such as the α-power law [111, 117, 130] or charge based models [87, 119]. To reduce complexity, which results from considering every single operation region [43] of a transistor (cf. Section 2.5.2), crude simplifications, like using an average current during switching [93], are applied.

The approaches also differ in the way the input slope is handled: Either slow and fast slopes are considered separately [67] or are extrapolated based on the step input behavior [44]. Some approach even sacrifice a certain degree of accuracy to end up with simpler models such as done by Wang and Zwolinski [66]. Lately also neural networks [21] have been trained to generate predictions. Note that primarily the Inverter has been investigated in such studies, since complexity quickly rises for more advanced gates due to additional phenomena such as multi input switching (cf. Section 3.4),

A lot of effort is nowadays invested into determining the constant static delay value that can be used in digital timing analysis. The most prominent examples are current source models (CSM) like the Effective CSM (ECSM) by Cadence [33] or the Composite CSM (CCSM, also called CCS) by Synopsys [25]. Based on extensive analog simulations, the input and its corresponding output voltage (ECSM) / current (CCSM) are extracted for different input slopes, and load capacitance and stored in massive tables. Before starting the digital simulation of a given circuit, the surrounding of each gate is analyzed, input slope and load capacitances are extracted, and a suitable delay value is derived. *Static timing analysis* (STA) [59] considers this delay to determine important characteristics, like the maximum clock frequency.

To catch more evolved effects, e.g., signal degradation or interference, that lead to very short pulses, *timing simulations*, which simulate the propagation of an input trace through the circuit, are indispensable. To evaluate the most common methods, we assume a single pulse (two transitions in opposite directions) of width Δ^i at a gate's input. The



(a) pure *(b) inertial* *(c) HSPICE*

Figure 4.1: Output (Δ^o) over input (Δ^i) pulse-width of a single in- and output gate for the pure and inertial delay model compared to HSPICE simulations of a real circuit. The continuous degradation can not be modeled using those simple approaches.

possible output pulse-width is denoted by Δ^o. Note that we call the pulse an *up*-pulse of width $\Delta_\uparrow$ if the input was initially LO followed by a rising and falling transition and a *down*-pulse of width $\Delta_\downarrow$ in the reverse case.

In the *pure delay* model [150], each input transition is simply delayed by a constant amount of time, leading to $\Delta^o = \Delta^i$ (see Figure 4.1a). Please note that the delay for rising and falling transitions might differ, which results in a constant alternation $\pm D$ of the pulse-width in one direction and $\mp D$ in the other, for example $\Delta^o_\uparrow = \Delta^i + D$ respectively $\Delta^o_\downarrow = \Delta^i - D$. The standard *inertial delay* model [150] is very similar, with the only difference that input pulses with a width smaller than some threshold A are dropped, i.e., not propagated to the output at all. In Figure 4.1b one can clearly see the respective discontinuity at $\Delta^i = A$.

Although pulse suppression effects can also be observed in HSPICE simulations (see Figure 4.1c), Δ^o shows a gradual increase in this case, which cannot be modeled by pure and inertial delays. This is a direct consequence of using a static delay value, i.e., one that stays constant throughout the whole timing analysis. Despite these shortcomings, state-of-the-art industry-grade timing analysis tools still heavily utilize pure and inertial delays. The main reasons are their simplicity and thus, speed, and the fact that "normal" digital circuits are not supposed to operate in the regime of pulse-width degradation. Nevertheless, due to internal gate connections, short pulses and glitches can never be completely ruled out, even for very low input frequencies.

4.2 Single History Delay Models

To model propagation in circuits more realistically, non-constant delay values are mandatory. Below some threshold pulse-width, input pulses should get removed (canceled). After exceeding this threshold the output pulse-width Δ^o has to gradually increase with growing Δ^i. Consequently a delay *function* is required, which provides a delay depending on some parameters related to the input pulse-width. Naturally, many possibilities to pick a suitable parameter exist. The single history approach, which will be used in this

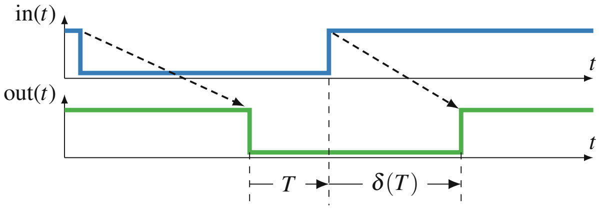


Figure 4.2: Delay estimation in a single history model. The delay δ is determined based on the parameter T, which is defined as the time difference between the current input transition and the previous output one.

thesis, derives the delay δ for the current input transition based on its temporal distance to the previous output transition, i.e., the *previous-output-to-input delay* T, as is shown in Figure 4.2. The fact that the input is only referred to the last output transition gave raise to the term *single history*. Clearly this approach could be extended to consider the last n output transitions, which will be the topic of Section 6.4.

With decreasing Δ^i the values for T and also $\delta(T)$ can become negative. While $T < 0$ can be easily retraced for very short input pulses, $\delta(T) < 0$ seems at a first glance counter-intuitive. Reality is causal, meaning that an event can only cause a reaction later in time, so how can this be reasonable? Well, from Figure 4.1, we know that $\Delta^i = A$ leads to $T + \delta(T) = 0$, i.e., a zero-time pulse at the output. For $\Delta^i < A$ analog simulations show trajectories, which do not cross V_{th} any more and thus are not visible in the digital domain, however, still can have large impact on the delay of the succeeding input transition. Thus it is mandatory to somehow represent these *sub-threshold* pulses also in the digital delay model.

This is realized in single history models by decreasing the delay, which leads to $T + \delta(T) < 0$, i.e., *cancelation*. The further the transition is pushed into the past the smaller the output pulse gets (this will become clearer in Section 4.3). Note that this has to be done with extreme care, since the models we are considering in the sequel calculate T always in reference to the latest output transition, even if it got canceled.

4.2.1 Degradation Delay Model

A concrete implementation of a single history model is the DDM, which has been introduced by Juan-Chico et al. [115] and was later extended several times, e.g., in [102]. A comprehensive overview of the model and all of its features is given by Bellido, Juan, and Valencia [82].

To determine the shape of the delay function[2], the authors used extensive HSPICE simulations with input ramps. The values for T resp. δ are extracted as shown in Figure 4.3. For varying input pulse-widths, different values are obtained and hence the delay function $\delta(T)$ can be determined numerically. Using ramps seems, at a first glance,

[2] In the original publications the delay function is called $t_p(T)$. For the sake of uniformity, however, we will use $\delta(T)$ throughout this thesis.

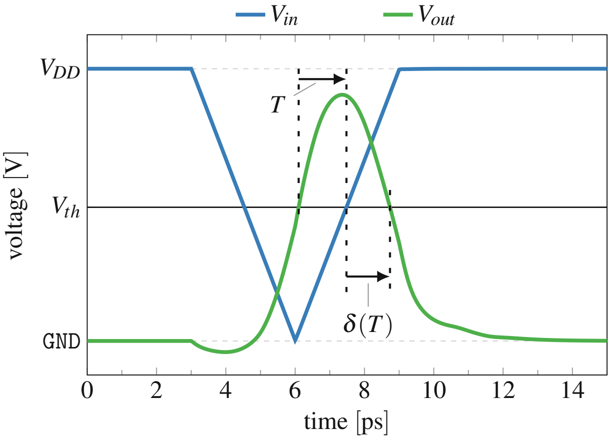


Figure 4.3: Characterization procedure of the DDM, showing the input slopes (V_{in}) and output trajectory (V_{out}) of an Inverter gathered from HSPICE simulations using technology (T15). T and δ are extracted as shown to determine $\delta(T)$ numerically.

not very realistic, however, as the simulation algorithm later propagates not only the switching time but also the slope, reasonable results are achieved.

Careful analysis of the gathered numerical values allowed Bellido-Díaz et al. [100] to fit their delay function to a decaying exponential, i.e.,

$$\delta(T) = t_{p0}\left(1 - e^{-\frac{T-T_0}{\tau}}\right), \tag{4.1}$$

where t_{p0} denotes the maximum delay, T_0 the crossing point of the x-axis and τ the rate of change. Moreover, the authors also provided qualitative physical explanations as well as characterization methods for the parameters T_0 and τ [100]. Note carefully, however, that extensive HSPICE simulations are still mandatory to obtain accurate parameters for the delay function.

Figure 4.4 shows an example delay function, where different operation regions, as defined by Bellido, Juan, and Valencia [82], are shown: In the *Normal propagation* region, the delay is nearly constant and very close to t_{p0} such that $\Delta^o \approx \Delta^i$. In the *Degradation effect* region, a significant reduction of the delay can be observed, leading to continuously decreasing Δ^o. Finally in the *Pulse filtering* region, an input pulse is suppressed completely, resulting in no output transition at all. Note that we do not agree with Bellido, Juan, and Valencia regarding the onset of pulse filtering, which was defined by them at $\delta(T) = 0$. In this case the output pulse-width, according to Figure 4.2, results in $\Delta^o = T + \delta(T) = T > 0$. Thus there is no cancelation. In our opinion, pulse filtering starts when the second median is crossed, i.e., at $T = -\delta(T)$.

Although DDM looks promising and powerful, Függer, Nowak, and Schmid [27] were able to show that all *bounded* delay models, including DDM, are not faithful, meaning that some behavior observed in real circuits can not be reproduced in the model and vice versa. Bounded in this regard refers to the fact that $\delta(T) = -\infty$ can not be achieved.

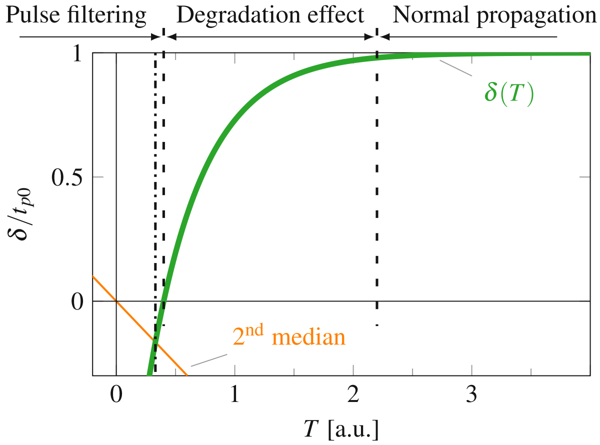


Figure 4.4: Exponential approximation for the DDM delay function $\delta(T)$. Different regions (dashed lines) are distinguished by the authors. We disagree with the onset of pulse filtering at $\delta = 0$, which should be $T = -\delta(T)$ marked by the dashed-dotted line. Heavily inspired by [82].

Providing for unbounded negative delays again seems not very reasonable, however, will turn out to be a crucial property.

The main issue with DDM is the non-proper cancelation of pulses. In more detail, an ε-pulse (glitch) at the input, i.e., whose width approaches zero, should have no impact whatsoever on the output. In the digital model this is equal to restoring the last output transition before the glitch, as is shown in Figure 4.5. Since the previous output transition can be arbitrarily far back in the past, an unbounded negative delay is required.

One might argue that the inaccuracies are negligible since the exponential, which approximates the delay function in DDM, drops very quickly. Although this is true for "normal" circuits, there may be others where improper glitch propagation may severely affect the correctness or power consumption of a circuit. In addition, incorrect predictions are very hard, and sometimes even impossible, to detect in the digital domain, as will be shown in Section 4.6. Thus a more reliable model like the IDM, which will be introduced in the succeeding sections, certainly makes sense.

4.3 The Involution Delay Model

After showing the unfaithfulness of existing approaches, Függer et al. proposed in [34] the unbounded single-history Involution Delay Model (IDM) and proved its faithfulness regarding the *Short-Pulse Filtration* problem:

Definition 1 (Short-Pulse Filtration)**.** A circuit with a single input and a single output port solves Short-Pulse Filtration (SPF), if it fulfills the following conditions:

1. The circuit has exactly one input and one output port. *(Well-formedness)*

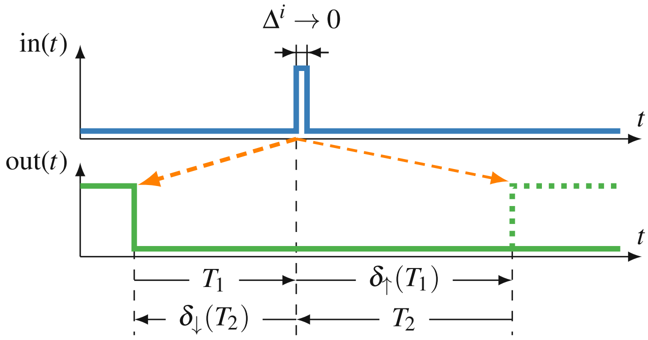


Figure 4.5: Proper cancelation of an input glitch ($\Delta^i \to 0$) for a delay model. Since the glitch is supposed to have no impact on the output, the scheduled output transition (dotted line) has to be canceled, such that the previous output transition (solid line) is restored.

2. A zero input signal produces a zero output signal. *(No generation)*

3. There exists an input pulse such that the output signal is not the zero signal. *(Nontriviality)*

4. There exists an $\varepsilon > 0$ such that for every input pulse the output signal never contains a pulse of length less than ε. *(No short pulses)*

It can be shown that this problem can be solved in a real circuit only in unbounded time. While IDM successfully predicts this behavior, any other model fails.

The distinguishing property of IDM can be derived directly from Figure 4.5, where we already used different delay functions for rising ($\delta_\uparrow$) and falling ($\delta_\downarrow$) input transitions. To restore the output transition in the case of an input glitch, $T_1 = -\delta_\downarrow(T_2)$ and $T_2 = -\delta_\uparrow(T_1)$ have to be satisfied. Combining both leads to

$$T_1 = -\delta_\downarrow(-\delta_\uparrow(T_1)). \tag{4.2}$$

Since this has to be true for all values T_1 the single delay functions have to form a mathematical *involution*, hence the name. The major difference between DDM and IDM is the way how values at and below the 2nd median, i.e., $\delta(T) = -T$, are handled. Using the definition $\delta_\uparrow(-\delta_{\min}) = \delta_{\min}$ as the onset of cancelation for a rising transition in combination with (4.2) leads to

$$\delta_\downarrow(-\delta_\uparrow(-\delta_{\min})) = \delta_\downarrow(-\delta_{\min}) = \delta_{\min}$$

and consequently to the necessary condition

$$\delta_\uparrow(-\delta_{\min}) = \delta_{\min} = \delta_\downarrow(-\delta_{\min}) \; . \tag{4.3}$$

Consequently, $\delta_\uparrow$ and $\delta_\downarrow$ have to meet at the 2nd median. Recall at this point that $\delta(T) < T$ is used to model sub-threshold output pulses and thus cannot be extracted

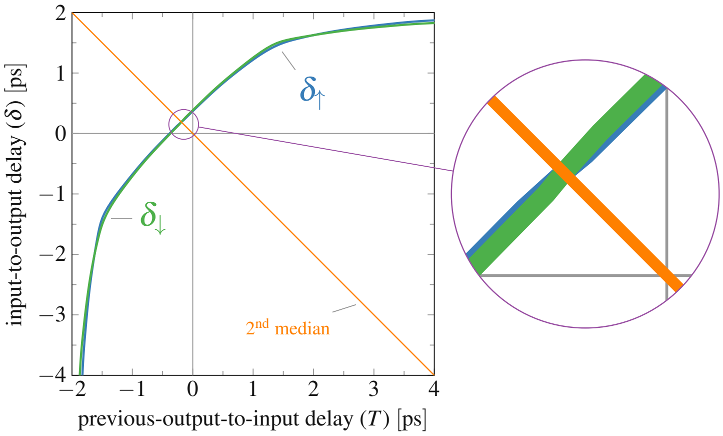


Figure 4.6: IDM delay function of an Inverter in technology (T15), numerically derived by extensive analog simulations in HSPICE.

from analog simulations. Instead the delay function during cancellation is extrapolated from the involution property, i.e., by mirroring the other delay function around the 2nd median such that $\delta_\downarrow(T)|_{T<-\delta_{\min}} = -\delta_\uparrow^{-1}(-T)$. In this case $\delta_\uparrow^{-1}(.)$ represents the inverse delay function, such that $\delta_\uparrow^{-1}(\delta_\uparrow(T)) = T$.

A comparison with (4.2) finally reveals, that an involution delay function (see Figure 4.6) is characterized by two strictly increasing delay functions $\delta_\uparrow : (-\delta_\infty^\downarrow, \infty) \to (-\infty, \delta_\infty^\uparrow)$ and $\delta_\downarrow : (-\delta_\infty^\uparrow, \infty) \to (-\infty, \delta_\infty^\downarrow)$ such that both $\delta_\infty^\uparrow = \lim_{T\to\infty} \delta_\uparrow(T)$ and $\delta_\infty^\downarrow = \lim_{T\to\infty} \delta_\downarrow(T)$ are finite and

$$-\delta_\uparrow\big(-\delta_\downarrow(T)\big) = T \qquad \text{and} \qquad -\delta_\downarrow\big(-\delta_\uparrow(T)\big) = T \tag{4.4}$$

The extraction from analog simulations is comparable to DDM. To mask the fact that IDM does not consider input slopes, properly shaped signals are used in the characterization process. In a chain of equal gates, DDM focuses on the first unit with linear input. IDM, on the other hand, investigates a gate further down the chain, where natural analog waveforms are reached. Therefore the $\delta(T)$ contains all crucial parameters such as output load and input driving strength. Consequently every single gate in a circuit has a unique delay function, which makes it crucial to find fast and easy ways for characterization. This will be further investigated in Section 4.5 and Section 6.2.

Comparing the extracted delay functions from DDM and IDM reveals deviations mainly close to the 2nd median. The reason is that in the case of settled waveforms shorter input pulses also have a decreased slope which leads to a less steep output. Consequently the first output transition is pushed further into the future (see Figure 4.7), which accelerates the decrease of T and thus increases $\delta_{\min}$ (cf. Figure 4.4 and Figure 4.6). While this seems, intuitively, to also provide an explanation for $\delta(0) > 0$ in the case of IDM and $\delta(0) < 0$ for DDM, this property primarily depends on the respective choice of the digitization thresholds, which will be analyzed in detail in Section 4.4.2.

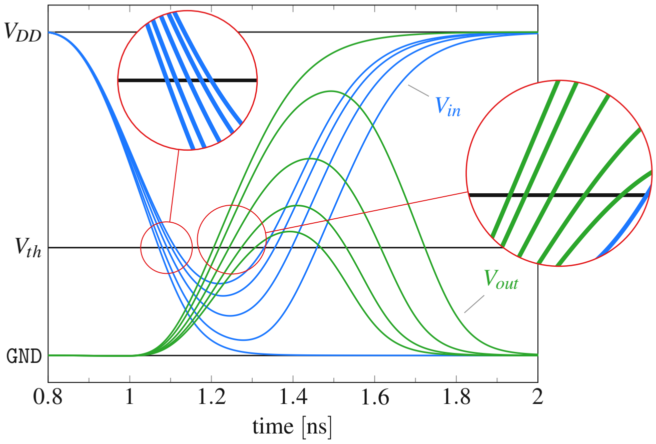


Figure 4.7: Changes of the analog in- and output waveforms for varying pulse-widths of an Inverter. For shorter input pulses, less pronounced (“shallower”) output pulses are observed leading to deviations in the relative timings (T and δ).

Unlike for DDM, we did not find an analytic fitting function for $\delta_\uparrow$ resp. $\delta_\downarrow$ yet. Note that even the exponential approximation for the DDM shows some deviations near the 2^{nd} median, for example in the results published in [115]. In this thesis we will thus rely more on calculations than on fittings.

In the IDM a circuit is modeled by Boolean, zero-time gates that are connected by single input-single output *involution channels*. These have the task to properly delay incoming signals. In the next section we will shortly review these channels. For further information the interested reader is referred to the original publication [17].

4.3.1 Analog Channel Model

When introducing the IDM in [34], Függer et al. have shown that its self-inverse delay functions arise naturally in a (generalized) standard analog model. It consists of a pure delay component, a slew-rate limiter with generalized switching waveforms, and an ideal comparator, as shown in Figure 4.8. First, the incoming, binary-valued input u_i is delayed by a pure delay $\delta_{\min} > 0$, which is necessary to assure causal channels, i.e., $\delta_{\uparrow/\downarrow}(0) > 0$. For every transition on u_d, the generalized slew rate limiter immediately switches to the corresponding waveform ($f_\downarrow$ for a falling and $f_\uparrow$ for a rising transition) such that the value at u_r, representing the analog output voltage, does not jump. Finally, the comparator generates the output u_o by discretizing the value of this waveform w.r.t. the threshold voltage V_{th}. Note that $f_\uparrow$ and $f_\downarrow$ need not have to be, and in general are not, equal to the FRSWs, i.e., rail-to-rail transitions observable in analog simulations.

To calculate the delay function $\delta_\downarrow(T)$, as detailed in [34], the value of u_r at the arrival of the falling transition on u_d has to be determined as well as the time it takes u_r to return to V_{th}. For this purpose, we compute the delay of a perfectly idle channel ($T = \infty$)

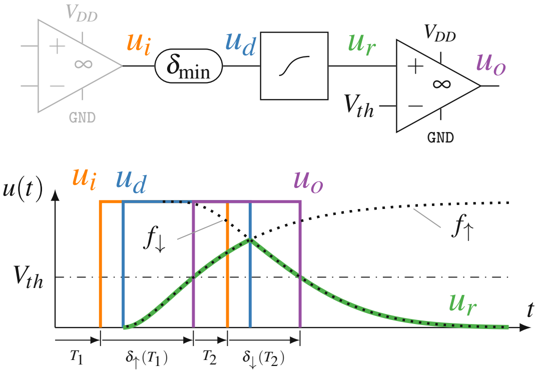


Figure 4.8: Simple analog channel model (upper part) with a sample execution (bottom part). The switching from $f_\uparrow$ to $f_\downarrow$ is done instantly when a transition on u_d arrives, leading to continuity in u_r but a jump in its derivatives. Heavily inspired by [34].

from a transition on u_i to reaching V_{th} on u_r as

$$\delta_\infty^\uparrow = \delta_{\min} + f_\uparrow^{-1}(V_{th}) \text{ and } \delta_\infty^\downarrow = \delta_{\min} + f_\downarrow^{-1}(V_{th}). \tag{4.5}$$

The main difference for $T < \infty$ is that u_r holds an intermediate value $V_s = f_{\uparrow/\downarrow}(t_s)$ at the switching time, which reduces the time to return to V_{th} by exactly t_s (see Figure 4.9). Consequently the general delay functions can be defined as

$$\delta_\uparrow(T) = \delta_\infty^\uparrow - t_s(T) \qquad \text{and} \qquad \delta_\downarrow(T) = \delta_\infty^\downarrow - t_s(T). \tag{4.6}$$

For the case shown in Figure 4.9 we get $V_s = f_\uparrow(t_{th} + T + \delta_{\min}) = f_\uparrow(f_\uparrow^{-1}(V_{th}) + \delta_{\min} + T)$, which can be transformed using (4.5) to

$$V_s = f_\uparrow(\delta_\infty^\uparrow + T).$$

Mapping this value back to the time domain, i.e., $t_s = f_\downarrow^{-1}(V_s)$, and plugging it into (4.6), and executing the same for the reverse direction finally yields the general delay functions

$$\begin{aligned} \delta_\downarrow(T) &= \delta_\infty^\downarrow - f_\downarrow^{-1}\left(f_\uparrow(T + \delta_\infty^\uparrow)\right) \quad \text{and} \\ \delta_\uparrow(T) &= \delta_\infty^\uparrow - f_\uparrow^{-1}\left(f_\downarrow(T + \delta_\infty^\downarrow)\right). \end{aligned} \tag{4.7}$$

Clearly the switching waveforms have a huge impact on the delay functions. Let us investigate this for the trajectories used for analog fitting in Section 3.7.3. For an *Exp-channel*, i.e., $f_\downarrow(t) = V_{DD} - f_\uparrow(t) = V_{DD} \cdot e^{-t/\tau}$, we derive

$$f_\uparrow^{-1}(V) = -\tau \cdot \ln\left(1 - \frac{V}{V_{DD}}\right) \text{ and } f_\downarrow^{-1}(V) = -\tau \cdot \ln\left(\frac{V}{V_{DD}}\right)$$

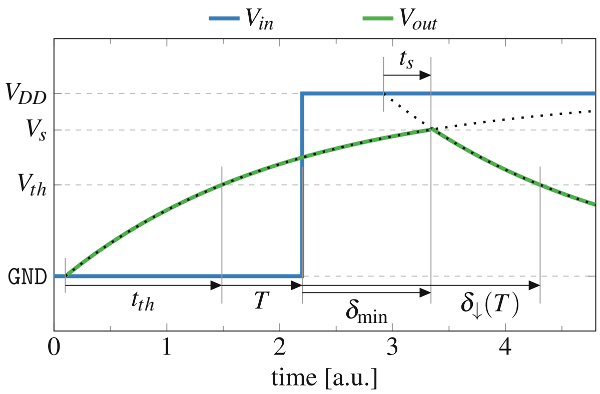


Figure 4.9: Graphical derivation of the IDM delay function. For finite values of T the downward switching waveform $f_\downarrow$ does start at an intermediate voltage $V_s = f_\downarrow(t_s)$ causing the time to reach V_{th} (t_{th}) to be reduced by t_s.

Plugging these into (4.5) and (4.7), and using $\overline{V_{th}} = V_{th}/V_{DD}$ leads to

$$\begin{aligned}
\delta_\uparrow(T) &= \delta_\infty^\uparrow + \tau \cdot \ln\left(1 - e^{-(T+\delta_\infty^\downarrow)/\tau}\right)\\
&= \delta_{\min} - \tau \cdot \ln\left(1 - \overline{V_{th}}\right) + \tau \cdot \ln\left(1 - e^{-(T+\delta_{\min} - \tau\ln(\overline{V_{th}}))/\tau}\right)
\end{aligned}$$

$$\begin{aligned}
\delta_\downarrow(T) &= \delta_\infty^\downarrow + \tau \cdot \ln\left(1 - e^{-(T+\delta_\infty^\uparrow)/\tau}\right)\\
&= \delta_{\min} - \tau \cdot \ln\left(\overline{V_{th}}\right) + \tau \cdot \ln\left(1 - e^{-(T+\delta_{\min} - \tau\ln(1-\overline{V_{th}}))/\tau}\right) \ .
\end{aligned}$$

For *Hill-channels* the waveforms satisfy $f_\uparrow(t) = V_{DD} - f_\downarrow(t) = V_{DD} \cdot \frac{t^n}{k^n+t^n}$. Recall that parameter k determines the time when $V_{DD}/2$ is reached and thus primarily depends on $\delta_\infty^\uparrow$ resp. $\delta_\infty^\downarrow$, $\delta_{\min}$ and V_{th}. For $V_{th} = V_{DD}/2$, the parameter n (the Hill coefficient) can be chosen almost freely and solely adjusts the waveform steepness. In all other cases it also has a certain impact on the threshold crossing time. By using

$$f_\uparrow^{-1}(V) = k_\uparrow \sqrt[n_\uparrow]{\frac{V/V_{DD}}{1 - V/V_{DD}}} \quad \text{and} \quad f_\downarrow^{-1}(V) = k_\downarrow \sqrt[n_\downarrow]{\frac{1 - V/V_{DD}}{V/V_{DD}}}$$

and inserting these once more into (4.7) and (4.5) we obtain

$$\begin{aligned}\delta_\uparrow(T) &= \delta_\infty^\uparrow - k_\uparrow \left(\frac{k_\downarrow}{T+\delta_\infty^\downarrow}\right)^{\frac{n_\downarrow}{n_\uparrow}}\\ &= \delta_{\min} + k_\uparrow \sqrt[n_\uparrow]{\frac{\overline{V_{th}}}{1-\overline{V_{th}}}} - k_\uparrow \left(\frac{k_\downarrow}{T+\delta_{\min}+k_\downarrow \sqrt[n_\downarrow]{(1-\overline{V_{th}})/\overline{V_{th}}}}\right)^{\frac{n_\downarrow}{n_\uparrow}}\end{aligned}$$

$$\begin{aligned}\delta_\downarrow(T) &= \delta_\infty^\downarrow - k_\downarrow \left(\frac{k_\uparrow}{T+\delta_\infty^\uparrow}\right)^{\frac{n_\uparrow}{n_\downarrow}}\\ &= \delta_{\min} + k_\downarrow \sqrt[n_\downarrow]{\frac{1-\overline{V_{th}}}{\overline{V_{th}}}} - k_\downarrow \left(\frac{k_\uparrow}{T+\delta_{\min}+k_\uparrow \sqrt[n_\uparrow]{\overline{V_{th}}/(1-\overline{V_{th}})}}\right)^{\frac{n_\uparrow}{n_\downarrow}}.\end{aligned}$$

Finally note that the symmetric sigmoids result in the delay function $\delta(T) = T + \delta_{\min}$, which is unreasonable. Therefore, these waveforms are not suitable choices for $f_\uparrow$ and $f_\downarrow$.

4.4 Analyzing the Involution Delay Function

The main ingredient of the IDM are clearly the delay functions. So far we have only considered the extraction from HSPICE simulations, which are executed in the following fashion: Let $t_0^I < t_1^I$ be the points in time the input pulse crosses V_{th} respectively $t_0^O < t_1^O$ for the output. Then we can determine the desired parameters as

$$T = t_1^I - t_0^O \quad \text{and} \quad \delta(T) = t_1^O - t_1^I.$$

By varying the input pulse-width, the threshold crossing times and thus T and $\delta(T)$ change, such that sufficient data for a successful characterization can be gathered. Especially for low values of T, i.e., near the crossing of the 2nd median, small changes have high impact, which requires an intelligent simulation algorithm that reduces the step size there.

This kind of characterization is quite cumbersome as lots of simulations are required. Although it is possible to fully automate this procedure it still consumes a lot of time, especially because it has, at least in theory, to be done for each gate of a circuit individually. For that reason we are searching for ways to find $\delta(T)$ in a more direct fashion, e.g., by calculation, which demands, however, more detailed information on the delay function. In this section we will therefore analyze and explore the impact of its parameters.

4.4.1 Prediction Inaccuracy

In Section 4.3 the characterization process for IDM was already shortly described. Recall that for both, IDM and DDM, the delay prediction of the first transition in a pulse is constant (t_{p0} respectively δ_∞). In contrast, HSPICE simulations show significantly

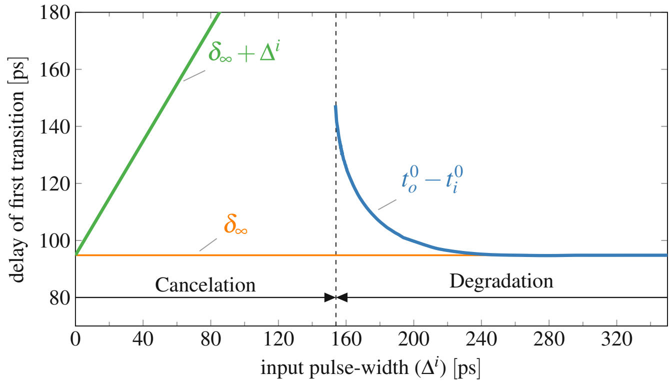


Figure 4.10: Variations in the delay of the first transition. The bound $\delta_\infty + \Delta^i$ must not be exceeded, as this would result, in the case of an input glitch, in a value $T < -\delta_\infty$.

different results (see Figure 4.10). The fact that smaller input pulses lead to less steep output waveform causes the delay of the first transition to actually increase with declining input pulse-width. As shown in Figure 4.7 this effect can already be observed for rather large pulses that almost reach all the way to V_{DD} resp. GND. Thus the IDM schedules the output transition too early in theses cases. The chances to encounter such a misbehavior thereby scales with the amount of degrading pulses that are analyzed and is thus, in general, non-negligible. Furthermore we want to stress that for balanced in- and output signals such delay variations can not be prevented: Only slowing down the output significantly would reduce the magnitude of the delay changes and the regions in which they appear. This can be easily be retraced by the fact that the gate in this case is in transition only for a very short period compared to the overall output switching time.

While this seems at a first glance just like a mere inaccuracy, it results in major drawbacks: The misplaced output transition influences the value of T for the succeeding input transition, which, in turn, results in a deviating delay value. We want to stress the significance of this result: Even the digital predictions for pulses, which have been used for characterization, are off by a certain amount during simulation, not to mention all remaining ones. In detail the first transition is predicted too early while the total output pulse width is overestimated.

For more realistic results it would be necessary to adapt δ_∞ depending on the input pulse-width. Since the latter is only available after the second transition has arrived this includes to alter the time of already scheduled transitions, which neither DDM nor IDM supports. This is actually a feature of higher order channels, which will be sketched in Section 6.4. An important question, which is also still unanswered, is how to estimate the delay of the first transition when entering cancelation (cf. Figure 4.10). Dropping to δ_∞ discontinuously does not seem physically reasonable. In any case it has to be assured that $\delta_\infty + \Delta^i$ is not crossed as this would result in $T < -\delta_\infty$ and thus violate the definition of the delay function.

To this end the only thing that would be realistically possible is to model Δ^o accurately. This could be realized by adapting the characterization procedure in the sense, that

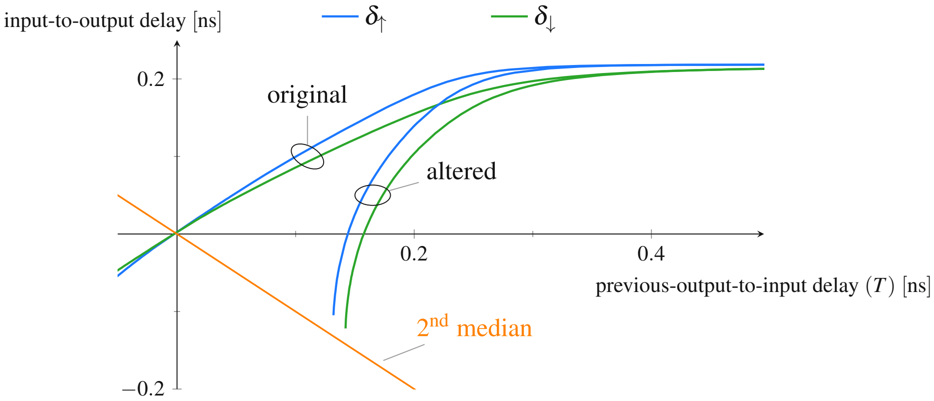


Figure 4.11: Correction of IDM delay functions to account for the non-constant delay of the first transition. In more detail, applying the altered delays results in more accurate output pulse-widths which are however still shifted in time. Please note the similarity of the altered $\delta_\uparrow$ and $\delta_\downarrow$ to DDM.

$t_i^0 + \delta_\infty$ is used as the reference for computing T instead of the actual first output V_{th} crossing. Figure 4.11 shows a comparison between the delay function extracted using the actual threshold crossings (original) and $T = t_i^1 - (t_i^0 + \delta_\infty) = \Delta^i - \delta_\infty$ (altered). Note that the latter results in an acausal channel, i.e., $\delta(0) < 0$, which can not be handled by the IDM[3]. Therefore this does not yield a practical solution.

4.4.2 Threshold Voltages

In delay modeling, the underlying circuit is necessarily heavily abstracted. This becomes evident when statements such as "when the threshold is crossed the output starts to switch" [84] are considered. As we know already from the analyses in Chapter 2 and Chapter 3, a transistor is a continuous device. So what does this ominous gate threshold voltage represent? How can it be determined and which values are reasonable? In this section, we will thus empirically explore the relation of gate delays and discretization threshold voltages by means of simulation results in technology (T15).

To obtain more general results, we will distinguish input (V_{th}^{in}) and output (V_{th}^{out}) threshold voltages in the sequel. DDM utilizes $V_{th}^{in} = V_{th}^{out} = V_{DD}/2$, which leads to acausal channels, i.e., $\delta(0) < 0$. This implies that simultaneous transitions on in- and output cause the latter to not reach V_{th}^{out} at all. Although this seems counter-intuitive, it can be well explained at the transistor level.

Be aware that several threshold voltages have to be distinguished in our context: (1) V_{th}^{in} and V_{th}^{out} denote the values used to digitize the analog in- and output trajectories, while (2) $V_{th,n}$ and $V_{th,p}$ refer to the thresholds of n- and pMOS transistors (cf. Section 2.5). In the sequel we will only investigate (1). These thresholds have, in contrast to (2), no

[3] An extension relaxing this condition will be presented in Section 4.8.

direct physical justification and can thus be chosen arbitrarily. Smaller values result in earlier transitions for rising waveforms and deferred transitions for falling trajectories.

Definition 2. The input and output discretization voltages V_{th}^{in} and V_{th}^{out} are called *matching* for a gate, if the induced delay functions $\delta_{\uparrow}(T)$, $\delta_{\downarrow}(T)$ fulfill the condition $\delta_{\uparrow}(-\delta_{\min}) = \delta_{\min} = \delta_{\downarrow}(-\delta_{\min})$. To stress that a pair of input and output discretization threshold voltages is matching, they will be denoted as V_{th}^{in*} and V_{th}^{out*}.

We will now characterize properties of matching discretization threshold voltages. They depend on many factors, including transistor threshold voltages [110] and the symmetry of the pMOS vs. nMOS stack. Since varying these physical parameters is commonly used in advanced circuit design to trade delay for power consumption [42, 58] and reliability [61], as well as for implementing special gates (e.g., logic-level conversion [113]), the range of suitable discretization threshold voltages could differ significantly among gates.

The following observation shows that there is an unlimited number of matching discretization threshold pairs for IDM:

Observation 3. For every choice of V_{th}^{in}, there is exactly one matching V_{th}^{out}. Fixing either of them uniquely determines the other and, in addition, also the pure delay $\delta_{\min}$.

Justification. Let us fix V_{th}^{out} and investigate how V_{th}^{in} and $\delta_{\min}$ can be determined. For this purpose, we consider an analog voltage pulse at V_{out} that barely touches V_{th}^{out}, i.e., results in a zero-time glitch in the digital domain. There is a *unique* positive and a *unique* negative analog output pulse with this shape, which is both confirmed by simulation results and analytic results on the underlying systems of differential equations (see Section 3.3). Now shift the positive and negative pulses in time such that their output voltages touch V_{th}^{out}, one from below and the other from above, at time t_o (see Figure 4.12). Due to the condition $\delta_{\downarrow}(-\delta_{\min}) = \delta_{\min} = \delta_{\uparrow}(-\delta_{\min})$, this implies that the falling transition of the positive pulse and the rising transition of the negative pulse at the input must both cross V_{th}^{in} at time $t_i = t_o - \delta_{\min}$. Thus, fixing V_{th}^{out*} uniquely determines both the matching V_{th}^{in*} and $\delta_{\min} = t_o - t_i$. □

Actually determining the matching V_{th}^{out*} for a given V_{th}^{in*} and vice versa is a challenging task. For a start, let us investigate the static case, with f_s being the static transfer function of a gate. In this setup, an output derivative $V_{out}' = 0$ is achieved for all values fulfilling the condition $V_{out} = f_s(V_{in})$ since f_s represents the stable states of the gate. To obtain high accuracy when discretizing the analog signal, one typically chooses the output threshold such that the respective output waveform for a full-range input pulse has a steep slope at this point. While $V_{th}^{out*} = V_{DD}/2$ is in general a good choice, the corresponding V_{th}^{in*} will differ significantly between balanced and high-threshold inverters, for example.

Besides these static considerations, for a dynamic input coupling capacitances cause a current at the output, which must be compensated via the gate-source voltages of the transistors as well. Obviously, the required overshoot w.r.t. V_{th}^{in}, and hence the time

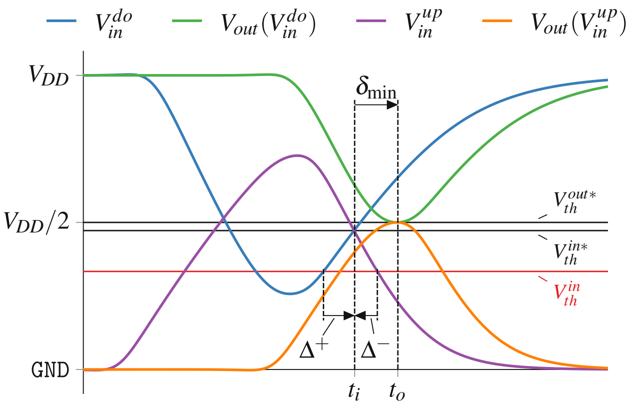


Figure 4.12: The unique relationship among V_{th}^{in*}, $\delta_{\min}$ and V_{th}^{out*} for a Buffer based on simulations in technology (T15).

until this value is reached, depends on many parameters like the size of the coupling capacitances and the slope of the input signal. A detailed analysis of $\delta_{\min}$ based on physical considerations will be presented in Section 4.4.3.

Observation 3 has a severe consequence for the simulation of circuits in any model, like IDM, where (4.3) has to be satisfied:

Observation 4. Fixing either V_{th}^{in} or V_{th}^{out} for a single gate G fixes the threshold voltages of all gates in the circuit simulated in a model where Observation 3 holds.

Since the detailed relation of V_{th}^{in*} and V_{th}^{out*} according to Observation 3 depends on the individual gate, this means that the discretization threshold voltages across a circuit may vary in *a priori* arbitrary ways, depending on the interconnect topology and the gate properties. In any case, it may take a large effort to properly characterize every gate such that the dependencies among discretization thresholds are fulfilled.

When starting at the back, forks (that is, joins) are problematic, since the input characterization thresholds of two distinct gates, which are driven by the same output, most certainly do not coincide. Reversing the direction of characterization, i.e., starting at the front and propagating towards the end, would solve this problem but adds a similar difficulty at the inputs of multi-input gates. Needless to say, feedback loops most probably make any such attempt impossible.

By contrast, an ideally composable delay model uses a uniform discretization threshold such as $V_{th}^{out} = V_{th}^{in} = V_{DD}/2$. To investigate if IDM allows such a uniform choice, we proceed with Observation 5:

Observation 5. Characterizing a gate with non-matching discretization thresholds V_{th}^{in} and V_{th}^{out*}, in the case where matching V_{th}^{in*} and V_{th}^{out*} lead to an IDM channel with pure delay $\delta_{\min}$, results in delay functions $\delta_\uparrow(T)$, $\delta_\downarrow(T)$, which satisfy $\delta_\uparrow(-\delta_{\min}^\uparrow) = \delta_{\min}^\uparrow$ and $\delta_\downarrow(-\delta_{\min}^\downarrow) = \delta_{\min}^\downarrow$ for $\delta_{\min}^\uparrow = \delta_{\min} + \Delta^+ \neq \delta_{\min}^\downarrow = \delta_{\min} + \Delta^-$. Δ^+ and Δ^- have opposite sign, with $\Delta^+ > 0$ for $V_{th}^{in} < V_{th}^{in*}$.

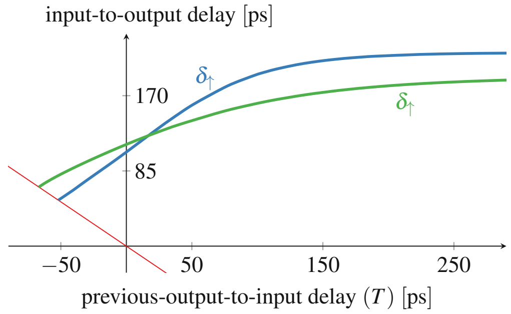


Figure 4.13: Characterizing a gate with $V_{th}^{in} = V_{th}^{out} = V_{DD}/2$. Clearly visible are the differing pure delay components $\delta_{\min}^{\uparrow} \neq \delta_{\min}^{\downarrow}$.

Justification. The observation follows from refining the argument used for confirming Observation 3, where it was shown how matching V_{th}^{in*} and V_{th}^{out*} are achieved. For the non-matching case, we increase resp. decrease V_{th}^{in}, starting from V_{th}^{in*}, while keeping everything else, i.e., electronic characteristics, waveforms and V_{th}^{out} the same. As illustrated in Figure 4.12 for $V_{th}^{in} < V_{th}^{in*}$, it still takes $\delta_{\min}$ from hitting V_{th}^{in*} (at time $t_o - \delta_{\min}$) to seeing a zero time glitch (at time t_o) at the output. W.r.t. V_{th}^{in}, the falling transition has already crossed V_{th}^{in*} when it hits on V_{th}^{in}, whereas the rising transition still has some way to go: Denoting the switching waveforms of the preceding gate (driving the input) by $f_\uparrow$ and $f_\downarrow$, the pure delay for the rising resp. falling transition evaluates to $\delta_{\min}^{\uparrow} = \delta_{\min} + \Delta^+$ and $\delta_{\min}^{\downarrow} = \delta_{\min} + \Delta^-$ with

$$\Delta^+ = f_\uparrow^{-1}(V_{th}^{in*}) - f_\uparrow^{-1}(V_{th}^{in}) \qquad \text{and} \qquad \Delta^- = f_\downarrow^{-1}(V_{th}^{in*}) - f_\downarrow^{-1}(V_{th}^{in}). \tag{4.8}$$

Consequently, $\delta_\uparrow(-\delta_{\min}^{\uparrow}) = \delta_{\min}^{\uparrow}$ and $\delta_\downarrow(-\delta_{\min}^{\downarrow}) = \delta_{\min}^{\downarrow}$ indeed holds. Finally, since $f_\uparrow$ must obviously rise and $f_\downarrow$ must fall, it follows that if $\Delta^+ > 0$ (the case in Figure 4.12) then $\Delta^- < 0$. □

Figure 4.13 shows the derived delay function for non-matching discretization thresholds. Clearly visible are the different pure delays $\delta_{\min}^{\uparrow} \neq \delta_{\min}^{\downarrow}$. Please note that in our justification of Observation 5, we focused on $\delta_{\min}$ and how it changes with varying discretization threshold voltages. If the rising and falling switching waveforms were always the same, as is assumed in the analog channel model for IDM (cf. Figure 4.8), this would result in delay functions that are fixed in shape and are simply shifted along the 2nd median. The actual delay functions of gates, obtained by analog simulations, for example, exhibit additional deviations (for $T \neq \delta_{\min}^{\uparrow/\downarrow}$), however, since the shape of the input switching waveforms also vary. Consequently, the difference between V_{th}^{in*} and V_{th}^{in} will not always be passed in constant time.

Finally, the dependency of the IDM on the particular choice of the discretization threshold voltages also reveals another problem:

Observation 6. Different choices of V_{th}^{out} can significantly change the digital model prediction of the IDM.

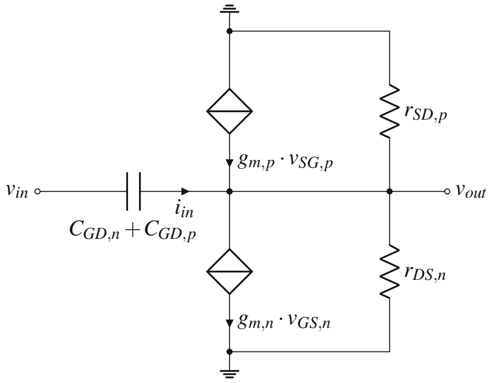


Figure 4.14: Simplified small signal representation of an Inverter using a current source (controlled by V_{Gy}*) and a resistor to model each transistor.* v_{in} *denotes the small displacement around the operating point* V_{in}*.*

Justification. Since sub-threshold pulses are automatically removed by the comparator in Figure 4.8, i.e., are completely invisible at the digital output of a gate that is fed to the successor gate, this can lead to the complete suppression of high-frequency oscillations at intermediate voltage levels: Assume an oscillatory behavior of a gate output with minimal voltage V_0 and maximal V_1. These oscillations would only be reflected in the digital discretization if $V_{th}^{out} \in (V_0, V_1)$. □

4.4.3 Pure Delay

In the previous section we have shown (i) how the pure delay component can be extracted from analog simulation results and (ii) that it depends on the choice of the threshold voltages. The question we left unanswered, however, is, if the pure delay is unavoidable and, if yes, what the physical causes are. This is not only important for the development of a proper model but also to predict deviations caused by parameter variations.

For this purpose we once again fix V_{th}^{out} and search, using the dynamic small signal representation of the Inverter (based on the ones of the transistor presented in Section 2.5.4) shown in Figure 4.14, for values of V_{in} that result in $V_{out}' = 0$. This implies that for these input values an output glitch can be observed, which makes it possible to calculate $\delta_{\min}$ for arbitrary choices of V_{th}^{in}. For the small signal analysis we start our considerations in (V_{in}^s, V_{th}^{out}) on the static transfer characteristic, i.e., $V_{th}^{out} = f_s(V_{in}^s)$, such that

$$\frac{\mathrm{d}}{\mathrm{d}t} V_{out} \bigg|_{V_{in}=V_{in}^s, V_{out}=V_{th}^{out}} = 0.$$

For very slow inputs, V_{in}^s is already the desired value. For bigger input derivatives a significant current i_{in} is induced over the input-output coupling capacitances, whereat

$$i_{in} = \frac{\mathrm{d}\, v_{in}}{\mathrm{d}t} \cdot (C_{GD,n} + C_{GD,p}) = \text{const} \;.$$

Since we are investigating very small deviations we approximated in the last step the input by a ramp, leading to a constant first derivative.

Conducting i_{in} via the resistors towards GND would induce a voltage Δv across the resistors and thus an increase of v_{out}. Since we fixed V_{th}^{out}, which demands $v_{out} = 0$, the current sources have to be adapted to compensate the additional charge carriers instead. In detail we are searching for a v_{in} such that

$$i_{in} = g_{m,n} \cdot v_{in} - g_{m,p} \cdot (-v_{in})$$

is satisfied. Simple arithmetic quickly leads to

$$v_{in} = \frac{i_{in}}{g_{m,n} + g_{m,p}} = \frac{\mathrm{d}\, v_{in}}{\mathrm{d}t} \frac{C_{GD,n} + C_{GD,p}}{g_{m,n} + g_{m,p}} \; .$$

Note that the fraction of capacitance and conductance is constant in close proximity to the operation point, i.e., $v_{in} \propto v_{in}'$. This implies that the dynamic stable points $(V_{in}^s \pm v_{in}, V_{th}^{out})$ are reached from (V_{in}^s, V_{th}^{out}) in a time that is independent of the input i_{in}'s slope and direction. Consequently every pair (V_{in}^s, V_{out}^s) on the static transfer function is a valid choice for V_{th}^{in*} and V_{th}^{out*}. The pure delay, i.e., the time difference between reaching the threshold at the input and the threshold at the output is finally the time it takes to build up v_{in}. Using a linear approximation, as usual in small signal analysis, we obtain

$$\delta_{\min} = v_{in} \cdot \left(\frac{\mathrm{d}\, v_{in}}{\mathrm{d}t} \right)^{-1} = \frac{C_{GD,n} + C_{GD,p}}{g_{m,n} + g_{m,p}} .$$

Note that this is only valid in close proximity around the static transfer characteristic. Thus for real circuits, minor deviations have to be expected.

4.4.4 Switching Waveforms

For given switching waveforms $f_\uparrow$ and $f_\downarrow$ the corresponding delay functions $\delta_\uparrow$ and $\delta_\downarrow$ can be uniquely calculated using (4.7). The reverse case, however, is fundamentally different: For a given set of delay functions infinitely many suitable switching waveforms, which only have to satisfy some simple condition, are imaginable. In this section we will thus show (i) how to achieve suitable $f_\uparrow$ and $f_\downarrow$ for given delay functions and (ii) further exploit the conditions that have to be satisfied.

From Delay Functions to Switching Waveforms

Recall that for a general expression of $\delta(T)$ in Section 4.3.1 we followed one waveform until V_s was reached and then switched to the other one (cf. Figure 4.9). The combination of both switching waveforms thus determines the time difference between reaching V_s and crossing $V_{th}^{in*} = V_{th}^{out*} = V_{th}^{out}$ (cf. Section 4.3.1), and consequently the delay. Note that the actual shape is thereby of no concern, which is good and bad news at the same time.

Clearly always both $f_\uparrow$ and $f_\downarrow$ are involved in switching, whereat $f_\uparrow(t)|_{t>t_{th}^\uparrow}$ and $f_\downarrow(t)|_{t<t_{th}^\downarrow}$ are used to derive $\delta_\downarrow(.)$ (V_s in $[V_{th}^{out}, V_{DD}]$), while $f_\uparrow(t)|_{t<t_{th}^\uparrow}$ and $f_\downarrow(t)|_{t>t_{th}^\downarrow}$ lead to $\delta_\uparrow(.)$ (V_s in $[\mathtt{GND}, V_{th}^{out}]$). Hereby $t_{th}^\uparrow$ and $t_{th}^\downarrow$ denote the point in time when the respective waveform reaches V_{th}^{out}. Picking some continuous shape for (parts of) a switching waveform thus also affects (the corresponding parts of) the other waveform. Consequently, only either $f_\uparrow$ or $f_\downarrow$ can be predefined in a specific voltage range $[V_1, V_2]$ with $\mathtt{GND} \leq V_1 < V_2 \leq V_{DD}$.

For the analysis presented in the sequel, we define both switching waveforms for $t > t_{th}$, i.e., $f_\downarrow$ in the range $[V_{th}^{out}, \mathtt{GND}]$ and $f_\uparrow$ in $[V_{th}^{out}, V_{DD}]$. Since the gates investigated in Chapter 2 showed an exponential behavior at the end of an output transition, we use

$$f_\uparrow(t)|_{t>t_{th}^\uparrow} = 1 - \exp\left(-\frac{t}{\tau}\right) \quad \text{and} \quad f_\downarrow(t)|_{t>t_{th}^\downarrow} = \exp\left(-\frac{t}{\tau}\right)$$

for some fixed τ. The task at hand is to determine the shape of $f_\uparrow(t)|_{t<t_{th}^\uparrow}$ and $f_\downarrow(t)|_{t<t_{th}^\downarrow}$. For a fixed T_1 let $\delta_1 = \delta_\downarrow(T_1)$. In the analog domain this corresponds to following $f_\uparrow$ for time $T_1 + \delta_{\min} + t_{th}^\uparrow$ before switching to $f_\downarrow$, which reaches V_{th}^{out} exactly after $\delta_1 - \delta_{\min}$. Since we defined $f_\uparrow(t)|_{t>t_{th}^\uparrow}$, V_s is known such that for $f_\downarrow$ the startvalue (V_s) and endvalue (V_{th}^{out}) as well as the time ($\delta_1 - \delta_{\min}$) to bridge the gap are known. Starting at $T_1 = -\delta_{\min}$ and increasing T_1 continuously allows one to generate $f_\downarrow(t)|_{t<t_{th}^\downarrow}$ that matches $f_\uparrow(t)|_{t>t_{th}^\uparrow}$ step by step. Note that $f_\uparrow(t)|_{t<t_{th}^\uparrow}$ can be calculated analogously.

Provided that we have only numerical data of the delay functions, given in the form of two discrete arrays T and δ, we calculate the switching waveforms from start to reaching V_{th}^{out} as discrete arrays v (value) and t (time). To determine $f_\downarrow$ the following algorithm has to be executed ($f_\uparrow$ can be handled analogously):

1. Define $f_\uparrow(t)$ for $t > t_{th}^\uparrow$.
2. $\forall i \in [1, \text{length}(T)]$ determine $V_s[i]$ as $f_\uparrow(T[i] + \delta_{\min} + t_{th}^\uparrow)$.
3. Set $t[F] = t_{th}^\downarrow$ and $v[F] = V_{th}^{out}$ with $F = \text{length}(T)$ to assure a continuous transition to the analytic definition for $t > t_{th}^\downarrow$.
4. By iterating $i \in [1, \text{length}(T)]$ time and value are calculated as

   $$\begin{aligned} t[F-i] &= t[F-i+1] - (\delta[i+1] - \delta[i]) \\ v[F-i] &= v[F-i+1] + (V_s[i+1] - V_s[i]) \ . \end{aligned}$$

   Note that the index is decreased as we are trying to determine the shape before the threshold is crossed, starting at the latter. The next value is chosen such that the difference in V_s is compensated during a time interval equal to the delay difference.

In our simulation we varied the parameter $\tau \in \{20, 60, 120\}$ ps (see Figure 4.15). For the slowest trace we see a steep drop at the beginning, which indicates that the final

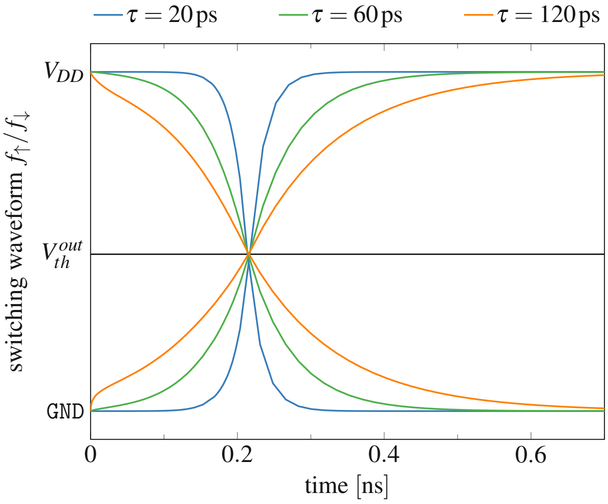


Figure 4.15: Different switching waveforms matching the same delay functions. The pieces after crossing V_{th} are initially fixed for varying time constant. Less steep ends lead to steeper beginnings.

value is approached too slow. For the fastest one, the start value is kept constant at the beginning to guarantee the demanded delay. Only for the intermediate value a “natural” shape is achieved.

Necessary Condition

Equation (4.7) introduces a very tight coupling between switching waveforms and delay functions. In fact, it induces a stringent relationship among the first derivatives, which we will explore in the sequel. For this purpose we enlarge the waveforms massively around V_{th}^{out}, which allows us to utilize a linear approximation, as shown in Figure 4.16. Two values of T are shown to investigate the changes based on the deviation ΔT: For $\Delta T > 0$ the rising trajectory is followed for a longer time such that the change to $f_\downarrow$ occurs at a higher voltage $V_s^1 > V_s^0$. The voltage difference ΔV_s has to be compensated by $f_\downarrow$, which increases the delay by $\Delta\delta$. For the linear approximation we get

$$\Delta V_s = \Delta T \cdot f'_\uparrow \quad \text{and} \quad -\Delta V_s = \Delta\delta \cdot f'_\downarrow \; .$$

Equating both expression leads to

$$\frac{\Delta\delta}{\Delta T} = -\frac{f'_\uparrow}{f'_\downarrow} \; .$$

This is very much in accordance to the results we got in the previous section, where also only the combination of $f_\uparrow$ and $f_\downarrow$ was of importance. For $\Delta T \to 0$ the differential term turns into the derivative, in detail

$$\frac{\mathrm{d}\delta_\downarrow}{\mathrm{d}T} = -\frac{f'_\uparrow(V_s)}{f'_\downarrow(V_s)} \quad , \quad \frac{\mathrm{d}\delta_\uparrow}{\mathrm{d}T} = -\frac{f'_\downarrow(V_s)}{f'_\uparrow(V_s)} \tag{4.9}$$

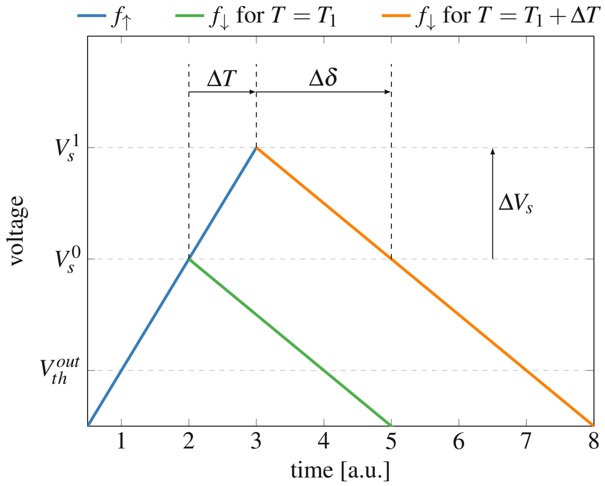


Figure 4.16: A linearized pulse barely exceeding V_{th} showing the difference in the delay based on the change of parameter T.

Clearly infinitely many combinations of $f_\uparrow$ and $f_\downarrow$ fulfill these conditions, whereat the exact value of V_s depends on the chosen waveforms. Note that the same results are derived by directly calculating the derivative of (4.7), which is, however, less intuitive.

We exemplarily evaluate (4.9) in the sequel for an Exp-channel and a Hill-channel. Recall that in Section 4.3.1 we already calculated $\delta(T)$, and thus easily get the derivative. Since we know $V_s(T)$ it is possible to calculate $f'_\uparrow(T)$ respectively $f'_\downarrow(T)$ whose ratio should then deliver the same results.

For the Exp-channel we shortly reiterate the waveforms

$$f_\uparrow = 1 - e^{-\frac{t}{\tau}} \qquad\qquad f'_\uparrow = \frac{1}{\tau} \cdot e^{-\frac{t}{\tau}}$$

$$f_\downarrow = e^{-\frac{t}{\tau}} \qquad\qquad f'_\downarrow = -\frac{1}{\tau} \cdot e^{-\frac{t}{\tau}}$$

and the resulting delay functions (for more details refer to Section 4.3.1)

$$\delta_\uparrow(T) = \delta_{\min} - \tau \ln\left(1 - \overline{V_{th}}\right) + \tau \cdot \ln\left(1 - e^{-(T + \delta_{\min} - \tau \ln(\overline{V_{th}}))/\tau}\right)$$
$$\delta_\downarrow(T) = \delta_{\min} - \tau \ln\left(\overline{V_{th}}\right) + \tau \cdot \ln\left(1 - e^{-(T + \delta_{\min} - \tau \ln(1 - \overline{V_{th}}))/\tau}\right).$$

In the sequel we will calculate $\delta'_\downarrow(T)$ whereat $\delta'_\uparrow(T)$ can be achieved analogously. Using $\delta_\infty^\uparrow = \delta_{\min} - \tau \cdot \ln(1 - \overline{V_{th}})$ the derivative evaluates to

$$\begin{aligned} \delta'_\downarrow(T) &= \tau \cdot \left(1 - e^{-(T + \delta_\infty^\uparrow)/\tau}\right)^{-1} \cdot \left(-e^{-(T + \delta_\infty^\uparrow)/\tau}\right) \cdot \left(-\frac{1}{\tau}\right) \\ &= \frac{1}{e^{(T + \delta_\infty^\uparrow)/\tau} - 1} \end{aligned} \tag{4.10}$$

According to our previous assumption this result must be equal to the ratio of the derivatives of the rising and falling switching waveform. Without loss of generality we assume $T = t_0 - \delta_{\min}$. At the switching point the upward switching waveform has a value of $f_\uparrow(t_0 + t_{th}^\uparrow) = f_\uparrow(t_0 + \delta_\infty^\uparrow - \delta_{\min}) = f_\uparrow(T + \delta_\infty^\uparrow)$. This value is reached by $f_\downarrow$ at time $t_s = -\tau \cdot \ln(f_\uparrow(T + \delta_\infty^\uparrow))$, thus the respective derivatives evaluate to

$$
\begin{aligned}
f'_\uparrow(T + \delta_\infty^\uparrow) &= \frac{1}{\tau} \cdot e^{-\frac{T+\delta_\infty^\uparrow}{\tau}} \\
f'_\downarrow(t_s) &= -\frac{1}{\tau} \cdot e^{-\frac{-\tau \cdot \ln(f_\uparrow(T+\delta_\infty^\uparrow))}{\tau}} \\
&= -\frac{1}{\tau} \cdot f_\uparrow(T + \delta_\infty^\uparrow) \\
&= -\frac{1}{\tau} \cdot \left(1 - e^{-\frac{T+\delta_\infty^\uparrow}{\tau}}\right)
\end{aligned}
$$

The ratio of both finally leads to

$$
-\frac{f'_\uparrow\left(T + \delta_\infty^\uparrow\right)}{f'_\downarrow(t_s)} = \frac{1}{e^{(T+\delta_\infty^\uparrow)/\tau} - 1}
$$

which perfectly matches (4.10).

Using $n_\uparrow = n_\downarrow = n$ for the Hill-channel simplifies the calculation and leads to

$$
\begin{aligned}
f_\uparrow &= V_{DD} \cdot \frac{t^n}{k_\uparrow^n + t^n} & \qquad f'_\uparrow &= -\frac{V_{DD} \cdot k_\uparrow^n \cdot n \cdot t^{n-1}}{(k_\uparrow^n + t^n)^2} \\
f_\downarrow &= V_{DD} \cdot \frac{k_\uparrow^n}{k_\uparrow^n + t^n} & \qquad f'_\downarrow &= \frac{V_{DD} \cdot k_\downarrow^n \cdot n \cdot t^{n-1}}{(k_\downarrow^n + t^n)^2}
\end{aligned}
$$

The derivative results to

$$
\delta'_\downarrow(T) = -\frac{f'_\uparrow(T + \delta_\infty^\uparrow)}{f'_\downarrow(t_s)}
$$

with $t_s = \frac{k_\downarrow k_\uparrow}{T+\delta_\infty^\uparrow}$, which was extracted from the definition of $\delta_\downarrow(T)$ in Section 4.3.1. After a short numeric calculation both methods deliver

$$
\delta'_\downarrow(T) = \frac{1}{k_\uparrow \cdot k_\downarrow}\left(T + \delta_\infty^\uparrow\right)^{-2}
$$

This shows that our approach is indeed valid, i.e., that the derivative of the delay function merely depends on the derivative of the switching waveforms and vice versa.

4.5 Calculating the Involution Delay Function

A closed form description of the delay function $\delta(T)$ offers several advantages over tabulated numerical values, such as (i) less storage requirements, (ii) higher accuracy since interpolating is not required any more, (iii) analytic circuit/delay composition and maybe even (iv) additional insights into physical/electrical processes governing circuit delays. Especially the last is important when arguing about the applicability to future technologies.

In this section, we will therefore investigate if and how an analytic description of the delay functions can be obtained. To the best of our knowledge, the authors of the DDM did not provide an explanation why an exponential fitting is appropriate in their case and whether it can be expected to apply also for future technologies. Based on physical considerations we (i) explain why and where an exponential approximation is accurate and (ii) develop appropriate abstractions that eventually lead to closed form analytic expressions[4]. Finally, (iii) we discuss how these expressions can be applied to the IDM.

For our analysis we used the simplistic Basic Model since it (i) facilitates analytic calculations and (ii) is actually capable of providing reasonably accurate predictions for the quantities we are aiming at. Let us recall at this point that our major goal is to explain the general shape of the delay functions. Whereas more accurate equations would lead to more fine-grained results, we conjecture that the differences are minor. In fact, considering that digital timing simulations are inherently inaccurate, we deem the deviations negligible.

To verify our modeling assumptions, we resort to HSPICE simulations as a golden reference; ten-stage Inverter chains are synthesized and parasitics extracted with Innovus using technologies (T15) and (T65). These different technologies allow us to verify whether our results have a chance to be technology independent, which indeed turns out to be the case. The 65 nm Inverter chain is further modified to include large 72 fF load capacitances instead of the relatively small parasitics. This allows us to pronounce effects that are otherwise too small and too fast to be observed, and to demonstrate that our approach works also in the presence of high fan-out. All our calibrations use the input and output signals of (i) the first Inverter in the chain, if trapezoidal input signals are required, and (ii) the seventh Inverter for shaped input signals.

4.5.1 General Remarks

For a start, we take a closer look at the linear input shape modeling originally used for DDM in [100], which simplifies calculations and analysis considerably. For IDM, we will later extend our results to the more general case of realistically shaped inputs. To get comparable results for DDM channels, we used the same settings as described in [82]: linear ramps as input signals, and V_{in} and V_{out} digitized at $V_{th}^{in} = V_{th}^{out} = V_{th} = V_{DD}/2$. The linear input slope at the first Inverter stage is chosen to have about the same rise/fall time as the shaped output signal (cf. Figure 4.3). Note that we investigate an optimal Inverter (single n- and pMOS with a load capacitance and no parasitics).

Figure 4.17 shows the HSPICE results of an up-pulse at the output. Recall that each transistor of the Inverter can operate in one of three operation regions, whereat we showed in Section 3.3.2 that only seven of the possible nine states are reachable

[4] Wherever this is not possible we will give at least an intuition for possible closed form expressions (IDM) or justifications for fittings (DDM, IDM).

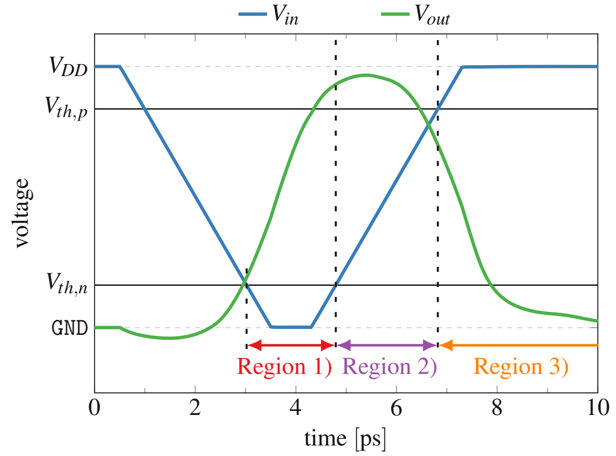


Figure 4.17: Overview of the Inverter operation regions in technology (T15) during switching. $V_{th,n}$ respectively $V_{th,p}$ represent the threshold voltages for n- and pMOS.

(the important subset for the evaluation of the up-pulse is shown in Figure 4.18). We distinguish three regions in the switching process:

Region 1) We start our considerations in state ① of Figure 4.18, i.e., when V_{in} drops below $V_{th,n}$ (the threshold of the nMOS) and thus opens the nMOS (non-conducting) while the pMOS is still in (SAT). As V_{out} increases, eventually the pMOS enters (OHM) ② , which reduces the current and thus the speed by which V_{out} increases. Only after V_{in} has exceeded the threshold $V_{th,n}$ of the nMOS in its rising transition, the latter starts to conduct again, causing a transition to ③. Note that quick input changes make it possible to transition from ① directly to ④.

Region 2) In the time period between V_{in} crossing $V_{th,n}$ and $V_{th,p}$ (the threshold of the pMOS), both transistors are conducting (③ and ④), thus both have to be considered. This is also the period where the trace of the output starts to deviate from the full range rising switching waveform and the maximum of the pulse is reached.

Region 3) Finally, the input reaches a value where the pMOS is opened and just the nMOS is conducting. At first, the latter is in (SAT) ⑤, i.e., the current stays nearly constant. Later, it enters (OHM) ⑥ to slowly approach the stable value.

In the sequel, we will derive an analytical solution for $\delta(T)$ for all $T > 0$. We start with a small output pulse, resulting in a small value of T, which just barely exceeds the threshold voltage V_{th} and thus operates in Region 2), i.e., ③ and ④, only. Later we increase the pulse-width to reach bigger values of T.

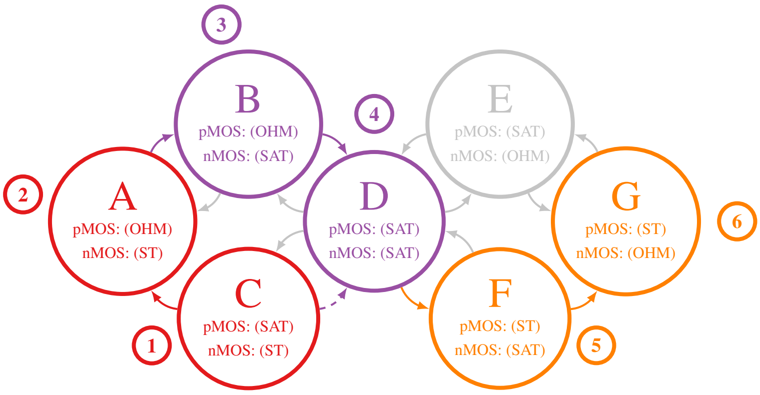


Figure 4.18: Transition graph of the transistor operation regions in an Inverter. The first line in a node shows the pMOS, the second one the nMOS. The colors correspond to Region 1) [red], 2) [purple] and 3) [orange].

4.5.2 Region 2), state (SAT) – (SAT)

Around the maximum of a small output pulse (barely exceeding the threshold voltage V_{th} used for digitization, corresponding to very low T), both transistors are in (SAT), thus their currents, according to the formalism we use, only depend on V_{in}. Furthermore, since we are trying to reason about DDM and are investigating an up-pulse, we pick a linear input with slope $k > 0$. Choosing a linear input has the advantage that coupling capacitances do not have to be considered as they always observe the same slope (hence draw a constant current). The input hits the threshold at $t = 0$, which we also assume to be the time when the output pulse reaches its maximum. This is a reasonable assumption, as it can be controlled by the choice of V_{th}. Note that this is only possible for DDM as an involution could not be derived in this fashion (cf. Section 4.4.2).

According to the transistor-level implementation of the Inverter (see Section 3.3), the derivative of the output is proportional to the difference between the current flowing through n- and pMOS, i.e.,

$$\frac{\mathrm{d}V_{out}}{\mathrm{d}t} = C_L \cdot I_{out} = C_L \cdot (I_{D,p} - I_{D,n}).$$

Without loss of generality, we can choose $C_L = 1$, since we are only interested in the general shape of the result. In the Basic Model the current through a transistor in its saturation region is approximated by a quadratic function, i.e., $I_{D,n} = S_n \cdot (V_{in} - V_{th,n})^2$ and $I_{D,p} = S_p \cdot (V_{DD} - V_{th,p} - V_{in})^2$ with $V_{in} = k \cdot t + V_{th}$. After subtraction and integration we end up with a polynomial of order three. Due to the fact that we demanded the output peak to be at $t = 0$, the linear term has to vanish, which results in the following general form:

$$V_{out} = C_3 \cdot t^3 + C_2 \cdot t^2 - \frac{3}{k^2}(C_0 - I) \tag{4.11}$$

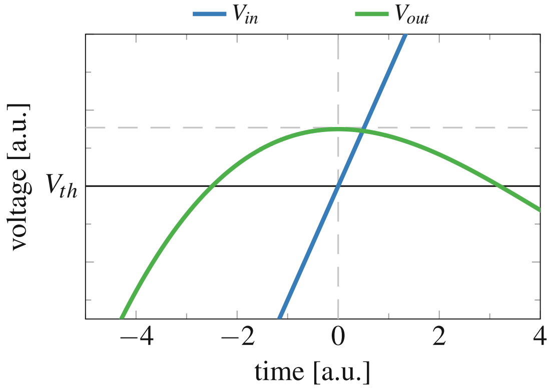


Figure 4.19: Cubic approximation of V_{out}. An arbitrary slope was chosen for V_{in} as k is hard to derive from the cubic function used for drawing the output curve.

with some integration constant C_0 and

$$C_3 = S_p - S_n, \qquad C_2 = -\frac{3}{k}(A \cdot S_p + B \cdot S_n)$$

$$A = V_{DD} - V_{th,p} - V_{th}, \qquad B = V_{th} - V_{th,n}$$

$$I = \frac{A^3 \cdot S_p}{3 \cdot k} + \frac{B^3 \cdot S_n}{3 \cdot k}.$$

Please note two important properties: (1) The peak value at $t = 0$ is primarily determined by the integration constant C_0 and (2) for an up-pulse a negative quadratic coefficient C_2 is achieved, while for down-pulses it is positive. Figure 4.19 shows an example trace which clearly reveals the cubic nature of V_{out}.

To calculate $\delta(T)$, we could vary C_0, i.e., the peak value, and observe the appropriate V_{th} crossing times. This tedious process can, however, be simplified significantly by analytically determining at which points in time the function given in (4.11) has the same value. Out of the three solutions, we are only interested in the ones closest to 0 on the negative ($-T$) and positive (δ) side. To get an explicit form, i.e., an expression for $\delta(T)$, we then have to derive δ as a function of T. Note that the specific values are actually of no concern for this analysis; just knowing the shape is sufficient.

In the easiest case $S_n = S_p$, which represents the situation that both transistors are driving with equal strength, the cubic part is zero and we end up with a quadratic function. As these functions are symmetric around zero, we get

$$\delta(T) = T,$$

i.e., the delay function is a ramp with slope 1. Since it is, however, almost impossible that both transistors are absolutely identical, we are more interested in the cases where $S_n \neq S_p$. Recall that we are looking for an explicit formula, so we need to find an expression that determines δ based on the knowledge of $T > 0$. As already mentioned,

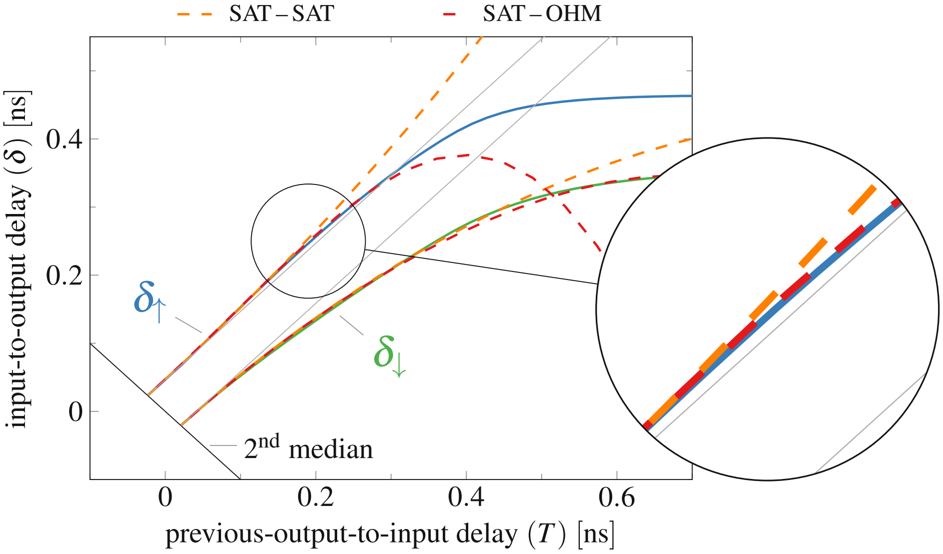


Figure 4.20: DDM delay function of the first Inverter in the 65 nm *Inverter chain (solid lines) vs. predictions based on our simplifications (dashed lines). Clearly visible is the super-linear growth of $\delta_\uparrow$ for $T < 0.2$ ns.*

we need a positive value δ with $V_{out}(\delta) = V_{out}(-T)$ for this purpose, i.e., by using (4.11), we need to solve

$$-C_3 \cdot T^3 + C_2 \cdot T^2 = C_3 \cdot \delta^3 + C_2 \cdot \delta^2.$$

Besides the obvious solution $\delta = -T$, which is irrelevant, we get two other ones, namely,

$$\delta(T) = \frac{-C_2 + C_3 \cdot T \pm \sqrt{C_2^2 + 2C_3C_2T - 3C_3^2T^2}}{2 \cdot C_3}.$$

One of those solutions is the desired result, provided that the constants C_2, C_3 and T do not cause the argument of the square root to become negative: Depending on the sign of C_2, the negative branch ($C_2 < 0$) or the positive branch ($C_2 \geq 0$) must be used. Comparing this estimation to delay functions simulated in HSPICE (see Figure 4.20) we observe good agreement for small values of T.[5] Note carefully that both delay functions initially have a slope of 1 (cf. the gray lines). Whereas the derivative of $\delta_\downarrow$ continuously decreases from there onward, the one of $\delta_\uparrow$ rises initially. This is in stark contrast to DDM, which demands sub-linear growth at all times. The 15 nm technology shown in Figure 4.21 appears better balanced, as no super-linear growth can be observed, implying a small cubic part.

[5] Note that the start position on the 2nd median was picked from the simulation results, as it depends on the choice of V_{th} and other parameters and cannot be determined analytically yet.

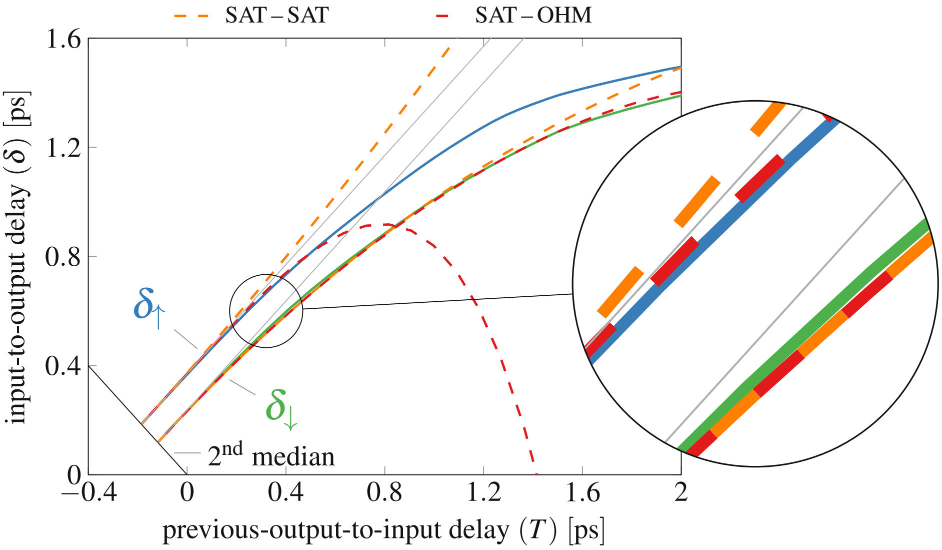


Figure 4.21: DDM delay function of the first Inverter in the 15 nm *Inverter chain (solid lines) vs. predictions based on our simplifications (dashed lines).*

4.5.3 Region 2), state (SAT) – (OHM)

The fitting developed for the (SAT) – (SAT) state of Region 2 in the previous section is only accurate up to a certain point. While the estimation keeps increasing, the simulated delay starts to decline. This is a consequence of the fact that, for slightly larger output pulses (that exceed V_{th} a little more), the Inverter is also in state ③ , where the pMOS delivers significantly less current than in ④. Therefore, we need to investigate this case separately. Using the same approach as before would cause I_{out} and hence V'_{out} to depend on V_{out}. Albeit the resulting ODE is solvable, its solution is far too complicated for being used here. Consequently, we will rely on appropriate abstraction instead.

As the pMOS operates in (OHM) it delivers less current than before. This implies that the peak of the output pulse shifts to a lower value of V_{in} since the nMOS has to close less to reach the current equilibrium $I_{D,n} = I_{D,p}$, and thus the peak value ($I_{out} = V'_{out} = 0$). With respect to our cubic fitting of V_{out}, this means that the peak is now at some time $t < 0$ instead of $t = 0$. We can approximate this behavior by artificially shifting the whole pulse. As a consequence, the time T between the first output V_{th} crossing to the input V_{th} crossing increases, while δ decreases by the same amount, see Figure 4.22. Note that this decreases the derivative of the resulting $\delta(T)$, and also guarantees a continuous transition between the low T situation of Section 4.5.2 and the higher T situation analyzed later.

However, we still need to answer the question how much the peak shall be shifted: Since I_D changes in (OHM) only linearly with V_{in} but quadratically with V_{out}, we carry out a time shift that depends quadratically on the peak value V_p. The resulting changes to determining a closed-form expression for $\delta(T)$ seems straightforward: just reduce T by $k \cdot V_p^2$ and increase δ by the same amount. In a real simulation several steps (shown

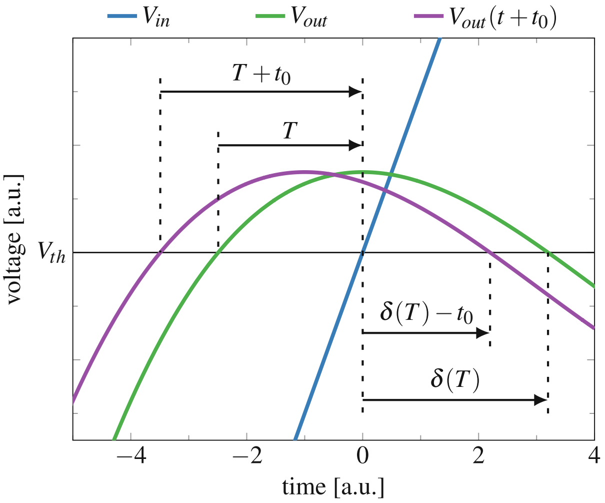


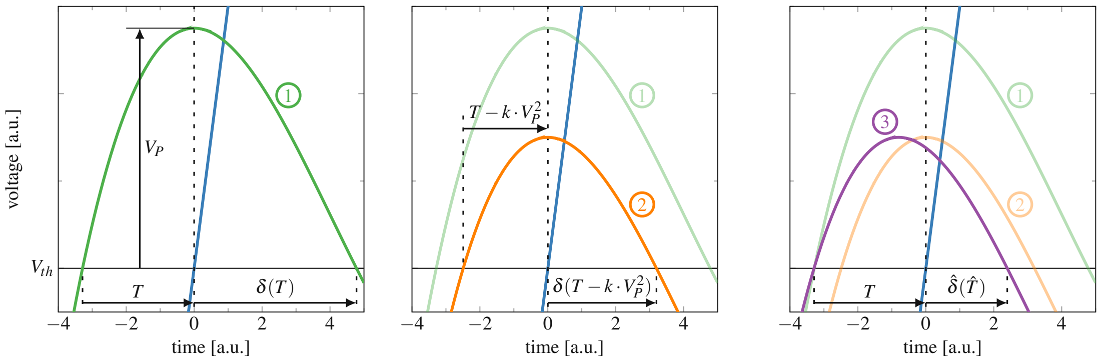
Figure 4.22: Shifting the cubic approximation of V_{out} by t_0 causes an increase in T and a decrease in δ. The slope of the input signal V_{in} is approximated.

in Figure 4.23) have to be executed to achieve the desired behavior. In step 1 the peak value is computed according to (4.11) as $V_p = V_{out}(0) - V_{out}(-T) = -C_3 \cdot T^3 + C_2 \cdot T^2$. In the next step we shrink the output pulse by defining $\hat{T} = T - k \cdot V_P^2$ and calculating the corresponding output waveform and delay. This is required to end up with a pulse that can be shifted in step 3 in time such that the previous-output-to-input delay T is achieved once again. The final delay can thus be determined by $\hat{\delta}(T) = \delta(\hat{T}) - k \cdot V_P^2$. Indeed, the predictions obtained with this approximation fits actual delay simulations, see Figure 4.20 and Figure 4.21. Qualitatively, the results look similar for both technologies, whereat a strong curvature in the approximation for $\delta_\uparrow$ can be observed. This forces us to investigate the region for big T separately.

4.5.4 Regions 1) & 3)

If the output pulse, and hence T, grows further, V_{out} is well above V_{th} when the rising input exceeds $V_{th,n}$ of the (open) nMOS, i.e., the Inverter is in Region 1) here. When the rising input eventually also exceeds $V_{th,p}$, the Inverter is in Region 3). This actually allows us to make radical reductions and thus simplifications. First of all, we assume that the part of the trajectory that lies in Region 2) is fixed, meaning that its shape and thus the contribution to T (T_2) and δ (δ_2) is constant (cf. Figure 4.17). This is reasonable, as we assume a linear input signal, which will therefore be the same for all pulses.[6] This also implies that the voltage gained in Region 1) has to be completely compensated in Region 3), which simplifies our calculations even further.

[6] Actually, the input slope has a big impact on the output through coupling capacitances. By keeping it constant, however, we effectively eliminate this influence completely.



(a) step 1

(b) step 2

(c) step 3

Figure 4.23: Single simulation steps in shifting the cubic approximation.

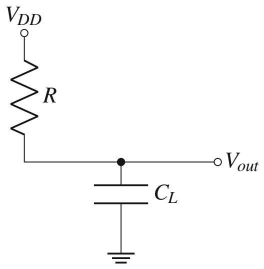


Figure 4.24: Inverter in Region 1).

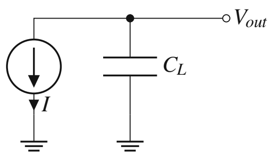


Figure 4.25: Inverter in Region 3).

After the V_{th} crossing of the rising output transition, the pMOS operates in (OHM). We can hence represent it as a simple resistor, leaving the overall Inverter in Region 1) as shown in Figure 4.24. Consequently, the capacitance C_L will be charged according to an exponential function, with a time constant $\tau = R \cdot C_L$. During most of the falling output transition, the nMOS is in (SAT), which causes the current in Region 3) to only change moderately with V_{out} (see Section 3.2.1). We repeat our assumptions of constant current in (SAT) here and replace the transistor by a constant current source, as shown in Figure 4.25. Figure 4.26 depicts these simplifications as fittings to a simulated HSPICE trace. In Region 1) & 3) (outside dashed lines) very good agreement can be observed.

Deriving an explicit formula for $\delta(T)$ is easy now. All the voltage ΔV gained by the exponential, which is followed for the time $T - T_2$, has to be compensated by the linear discharging current, which is in effect for the time $\delta - \delta_2$. We thus get

$$\Delta V = (V_{DD} - V_{th}) \cdot \left(1 - e^{-(T-T_2)/\tau_\uparrow}\right).$$

The time $\delta_\downarrow(T)$ it takes the output downward ramp with slope $k_\downarrow$ to compensate this voltage ΔV evaluates to

$$\delta_\downarrow(T) = \frac{\Delta V}{-k_\downarrow} + \delta_2 = \frac{V_{DD} - V_{th}}{-k_\downarrow} \cdot \left(1 - e^{-(T-T_2)/\tau_\uparrow}\right) + \delta_2$$

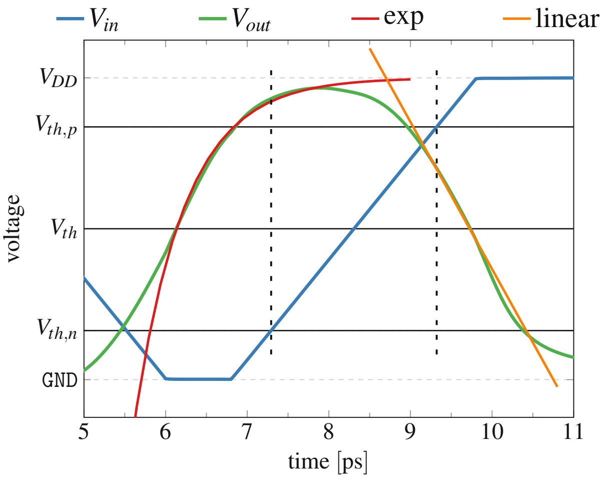


Figure 4.26: Simplification of V_{out} for Region 1) & 3) for 15 nm technology. The exponential increase is followed by a linear drop.

Similarly, the delay function for the rising output transition reads

$$\delta_{\uparrow}(T) = \frac{V_{th}}{k_{\uparrow}} \cdot \left(1 - e^{-(T-T_2)/\tau_{\downarrow}}\right) + \delta_2$$

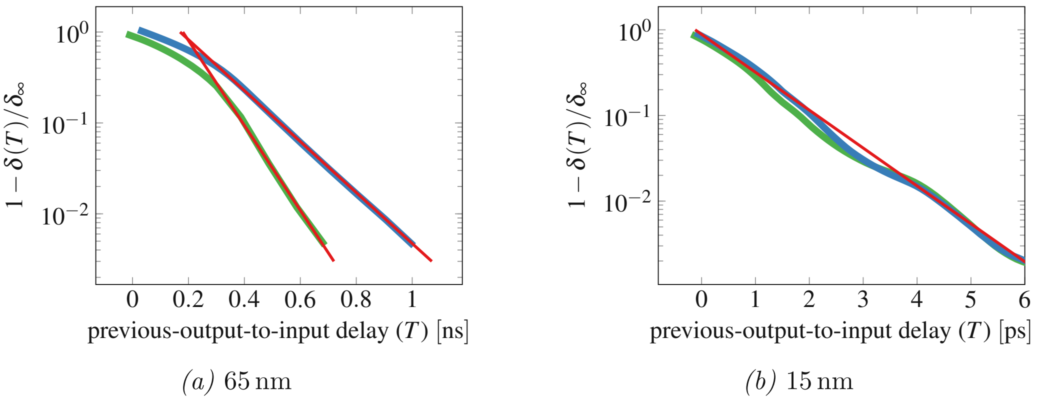
From these expressions, we can already deduct important parameters of the delay functions. In particular, their limiting values $\delta_{\uparrow}(\infty)$ and $\delta_{\downarrow}(\infty)$ solely depend on the choice of the output threshold voltage V_{th} and the current I_p resp. I_n (represented by $k_{\uparrow}$ resp. $k_{\downarrow}$) delivered by the active pMOS resp. nMOS transistor (plus some constant). Note carefully that $k_{\uparrow}$ (and analogously $k_{\downarrow}$) depends on the load capacitance via

$$\frac{\mathrm{d}V_{out}^{\uparrow}}{\mathrm{d}t} = k_{\uparrow} = C_L \cdot I_p.$$

We do not expect that accurately estimating these limiting values, which effectively correspond to the static delays and are hence usually well-characterized anyway, becomes urgent in the near future. They are interesting, though, for estimating the consequences of changing transistors and/or output load.

It can be seen clearly that the overall shape of the delay function for large T is determined by the RC constant of the transistor active in the first part, i.e., $\tau = R \cdot C_L$ from Figure 4.24. To determine R, one has to investigate the slope of I_D shown in Section 3.2.1 for $V_{GS} = V_{DD}$ and low values of V_{DS}. As there are different fittings possible, finding an appropriate value might be a challenging task.

Figure 4.27a shows the resulting delay functions in logarithmic scale, e.g., $\log(1 - \delta_{\uparrow}(T)/\delta_{\uparrow}(\infty))$. For large values of T, we get a linear dependency, i.e., an exponential behavior. In this region, the DDM delay function given by Bellido, Juan, and Valencia [82] is indeed correct. Unlike the cubic fitting established in the previous subsections, it can, however, not explain the significant curvature for T towards 0. Simulations on



(a) 65 nm

(b) 15 nm

Figure 4.27: DDM delay function of the first Inverter for different technologies in logarithmic space with linear fitting.

the 15 nm technology show a quite different picture (see Figure 4.27b), as no curvature is observable there. Instead the complete delay function may be fitted using a single exponential more or less accurately.

4.5.5 Extension to IDM

As pointed out earlier, the main difference when switching to IDM are the shaped input signals used for characterization, instead of the linear ones used by DDM. We simulated this by picking the seventh Inverter in our ten inverter chain. This "minor" change increases not only the overall complexity, as the changing input derivative induces varying currents via coupling capacitances, but also has an impact on the shape of the delay function. In general, an increased bending of the delay function can be observed. Nevertheless, our assumptions still seem valid as the projected trace is very close to the simulated one (see Figure 4.28). Solely in the transition region, where we have to switch between the different approximations, the accuracy slightly decreases.

4.5.6 Summary

Overall, the description of the delay function can be divided into up to three regions, where each requires a different model. While it is sufficient for low values of T to ignore the output voltage, we quickly run into troubles with this approach, as the delay for one direction would continuously increase. In our simulations, we see a significant reduction of the derivative shortly after the start, which we model, due to computational complexity, by shifting the output waveform in time depending on the maximum deviation to the threshold voltage. This way, the actual delay function can be approximated closely. For large values of T we are able to employ coarse abstractions, which resulted in the exponential function that was derived by the authors of DDM. Unfortunately, the transition to IDM turned out to be more challenging than expected. Albeit we observe

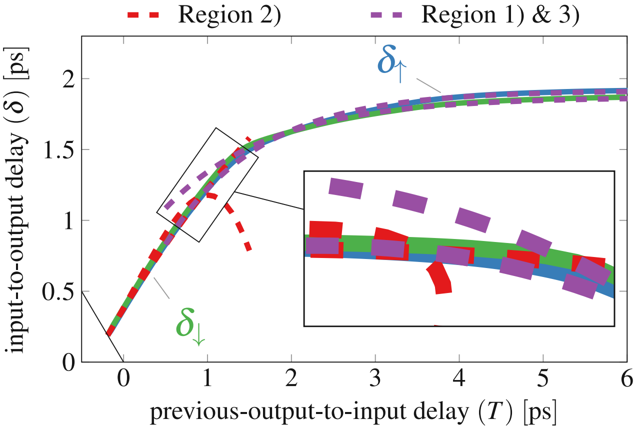


Figure 4.28: Fitting of the involution delay function for the seventh Inverter in the 15 nm *chain.*

quite good agreement also here, the inaccuracies while transitioning from low to high values of T is bigger.

Despite using the very simple models described in Section 3.2.1, which neglect a lot of phenomena present in modern technology, the fitting to accurate HSPICE simulations is generally very good. A comparison between the predictions of our models and the real delays observed in different technologies (cf. Figure 4.27 and Figure 4.28) allow us to conjecture, that we have identified a set of equations that is sufficiently parametrizable to properly model the delay functions both for DDM and IDM. Our research also revealed that the exponential fitting of DDM can be justified based on physical consideration, albeit only for large values of T; for smaller values it does not describe the real behavior well.

4.6 Simulating the Involution Delay Model

Although it is important that a delay model provides reliable results, it also has to be easily applicable to be of practical relevance. As this was not the case for the IDM, we developed the *Involution Tool* (InvTool)[7], which was first presented by Öhlinger [19]. In a nutshell, it is a complete framework for the systematic and automatic evaluation of several power and timing metrics, evaluated on different delay prediction methods and models. Switching between default and IDM simulation is thereby simply achieved by loading a different library, such that existing infrastructure, like test scripts and input vectors, can be reused without modification. Thanks to its ability to process the output of multiple simulation tools it even allows a comparison among various delay models.

In this section, we will shortly introduce the basic idea behind the InvTool. For a complete description the interested reader is referred to the original publication [7]. We

[7] The InvTool can be found on GitHub: https://github.com/oehlinscher/InvolutionTool

then utilize the tool to compare the predictions for more elaborate circuits, namely, an `OR Loop` with variable feedback, an `SR Latch` and an `Adder`, to analog and other digital delay simulations. Our results show a high correlation between IDM and analog simulations leading to highly reliable results, and reveal how commonly used approaches fail to describe a wide range of possible behaviors. Although its sometimes significant overhead in simulation time compared to inertial delay (up to 250 %), we consider the IDM a viable upgrade that allows to reliably identify potentially harmful locations resp. input trajectories in critical circuits.

4.6.1 Incorporating IDM in ModelSim

One of the main goals for the development of the InvTool was our desire, to perform circuit simulations using the IDM without the need to install and utilize additional software. For that reason, we used VHDL Vital as prototype for the development of the InvTool. Consequently, our solution not only has the same structure as VHDL Vital, but also responds to the same variables and is written in VHDL. Simulations in the InvTool are completely controlled by ModelSim, which makes it possible to use all its features without restrictions: Based on the next input transition time at the channel input, the algorithm determines T and the resulting $\delta(T)$, and adds the transition to the channel's output. This is done separately for each channel, as their parameters can differ.

For simulations using the InvTool, one hence needs exactly the same input files as for any standard post-layout simulation: the circuit, a testbench, and the timing characteristics stored in *.sdf* files. The latter contain the static delay of each gate (δ_∞) in the circuit and the interconnects in between. While VHDL Vital uses essentially pure/inertial delays with a priori given fixed delay values, the IDM calculates the parameters for the delay functions $\delta_\uparrow$ and $\delta_\downarrow$ as introduced in Section 4.3.1, using a user-defined value for the pure delay parameter $\delta_{\min}$. Currently Exp-channels, the sum of Exp-channels and Hill-channels are supported, whereat the user has to set for the latter the values of $n_\uparrow$ and $n_\downarrow$. Note that neither $\delta_{\min}$ nor $n_\uparrow$ and $n_\downarrow$ are easy to guess, so their impact has been extensively evaluated experimentally in [7].

A crucial task for using the InvTool in practice is to extend the set of available basic gates, which of course has to be done only once for a new gate. It essentially consists of modeling the Boolean functionality in VHDL and connecting in- and outputs via suitable IDM channels. Please note that in contrast to single input-single output channels, which are quite easy to handle, things get more complicated for multi-input gates, as there are different possible locations for placing the IDM channels.

4.6.2 Experimental setup

Starting from a Verilog netlist circuit description[8]. we utilize Genus and Innovus to place & route the design using technology (T15). Based on the final layout we are able to automatically extract the parasitics (.spef format) and static delay values (.sdf format),

[8] The simulation data is available on GitHub: https://github.com/jmaier0/idm_evaluation

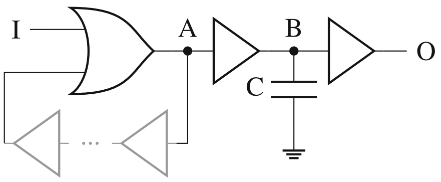


Figure 4.29: `OR Loop` *gate level implementation.*

two important ingredients for the succeeding analog and digital simulations. In the analog case, we back-annotate the extracted parasitics to a transistor level model, which is then processed using Spectre. Transient simulations deliver analog traces, which are dumped and finally plotted. Note that the analog results primarily serve as golden reference for the digital predictions, enabling a quick and easy evaluation regarding the correctness and behavioral coverage.

The digital simulations are run with ModelSim, which reads the .sdf file to parameterize the circuit netlist generated by Innovus. Two digital simulation approaches were executed: The default one provided by the tool (INE), essentially an inertial delay method in Verilog shipped with the technology library, and the Involution Delay Model. For the latter we utilized only the Exp-channel model. Since only $\delta_\infty^{\uparrow/\downarrow}$ are provided by the .sdf files, we set, for the sake of simplicity, the pure delay to a constant value of $\delta_{\text{min}} = 1\,\text{ps}$. The results of the simulations were, at last, dumped into a .vcd file and then plotted.

We want to emphasize at this point, that these simulations confirmed in an impressive fashion the simplicity of integrating the IDM into an existing design flow. Starting from the finished test setup for INE one only has to compile and link the gate descriptions and channel model for the IDM. However, due to the fact that the library delay model INE and the IDM are implemented in different hardware description languages (Verilog vs. VHDL) we were not able to use the same testbench in both cases. The reason is, that some commands, such as forcing signals, do not properly work across language boundaries. This made it necessary to implement the testbench twice, once in Verilog and once in VHDL, with, of course, the exact same behavior.

4.6.3 Circuits

In the sequel we describe the circuits used in our simulations in greater detail. Note that additional Buffers, which we add at the in- and output to emulate the settings far away from the chip boundaries, are not shown.

`OR Loop`

The circuit shown in Figure 4.29 has been used in the past for proving the faithfulness of IDM regarding the SPF problem [17]. It solely utilizes simple, single input-single output, Buffers to create a combinational loop, whereat up-pulses are inserted utilizing a single OR-gate. Based on the pulse-width Δ^I on signal I different behaviors are possible: For small Δ^I the gates in the loop lead to degradation, causing the pulse to vanish

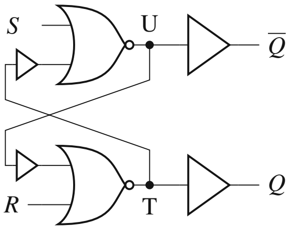


Figure 4.30: `SR Latch` *gate level implementation.*

eventually. A similar behavior can be observed for rather large Δ^I, however, in this case the remaining `LO` time in the loop decreases, leading to a constant `HI` at the output.

In between these cases there is a nonempty range of Δ^I causing the pulse train inside the loop to recreate itself infinitely. Depending on the length of the feedback path either a train of distinguished pulses is observable or just a constant, intermediate voltage value. While the former corresponds to a simple ring oscillator, the latter depicts *metastability* [53], an undesired state in digital circuits that causes combinational loops to settle at an intermediate voltage value for a possibly unlimited amount of time[9]. Since such intermediate values may be interpreted differently by succeeding gates, it is crucial to model metastable upsets in a suitable fashion in the digital domain. In our descriptions of oscillations and metastability we are going to use Δ_n^{HI} and Δ_n^{LO} to denote the high respectively low time of the n^{th} oscillation (= high + low pulse) at node A.

To stimulate different behaviors, we run simulations with a varying number of Buffers in the feedback path. The primary effect is an increased loop delay, which, as we already mentioned, has a big impact on the behavior. We are aware that longer delays could also be achieved by adding large capacitances. In our setup this would, however, lead to significantly different results since a capacitance serves as a low pass filter and thus suppresses short, i.e., high frequency, pulses very effectively. Using multiple Buffers in succession, on the contrary, increases the delay but keeps the signal shape intact, such that oscillating signals can be generated. Nonetheless we artificially add a large capacitance at node B, as this allows us to study the internal behavior in greater detail and to reveal possible shortcomings of the delay models more clearly.

SR Latch

So far, the IDM has almost exclusively been applied to single input-single output gates. Therefore, it is of major interest whether more elaborate circuits, still only utilizing Boolean gates and single-input single-output delay channels, can be properly described as well. For this purpose we investigate the `SR Latch`, a well-known circuit with the possibility for metastability, as shown in Figure 4.30. Note that we added a single Buffer on the coupling paths between the `NOR`-gates, to pronounce the observable effects and thus ease their detection.

[9] A detailed consideration of metastability follows in Chapter 5

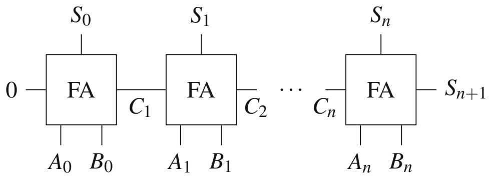


Figure 4.31: `Adder` gate level implementation.

The Set Reset Latch operates as follows: If the set (S) input turns `HI`; Q switches to `HI`, for a `HI` on the reset (R) input, Q changes to `LO`. $\overline{Q}$ represents the inverse of Q and thus transitions exactly in the opposite direction. Both inputs set to `HI` leads to an intermediate voltage value at nodes U & T and thus has to be prevented. Note the similarities between `SR Latch` and `OR Loop`: If one input is `LO`, the `SR Latch` behaves, w.r.t. the other one, just like the `OR Loop`. Very short pulses are blocked, very long ones immediately set the loop, while ones in between may lead to metastability. Significantly different behavior is possible, however, if both inputs are allowed to change. While one steers the loop into a metastable state the other one can either support or impair its resolution, a behavior that we will stimulate in our simulations.

Adder

To investigate the scaling of the IDM and its predictions on loop-free circuits we also simulated a simple ripple carry adder as shown in Figure 4.31, whereat we used $n = 4$. Each full adder block FA is defined at the gate level and implements the equations

$$\begin{aligned} S_i &= C_i \oplus A_i \oplus B_i \\ C_{i+1} &= (C_i \wedge (A_i \oplus B_i)) \vee (A_i \wedge B_i) \ . \end{aligned}$$

Regarding input stimuli, those leading to a maximum number of transitions are the most interesting for us, as they allow the investigation of the whole circuit in a single simulation run. Considering the fact that signals traverse from left to right we choose $B_0B_1B_2B_3 = 1111$, $A_0A_1A_2A_3 = 0000$ and introduce an up-pulse on signal A_0. If the pulse is wide enough this leads to a pulse on all internal carry signals C_i and all output signals S_i. For a down-pulse on signal A_0 we used a very similar setup, with the only difference that we set $A_0A_1A_2A_3 = 1000$ initially.

In the following sections, we will present and compare the analog resp. digital simulation results for all our the circuits. We start by studying oscillatory behavior and its digital counterpart for the `OR Loop` with long feedback delay. Subsequently we will remove the Buffers from the feedback path and investigate the effects on the (significantly changing) analog and (only slightly differing) digital simulation results. Afterwards we use the `SR Latch` to demonstrate the superior modeling power of IDM, which, in contrast to inertial delay, predicts metastable behavior quite well. Simulations of the `Adder` confirm the superior modeling power of the IDM but also reveal inaccuracies. Finally we compare the overhead and thus the price for the more reliable results.

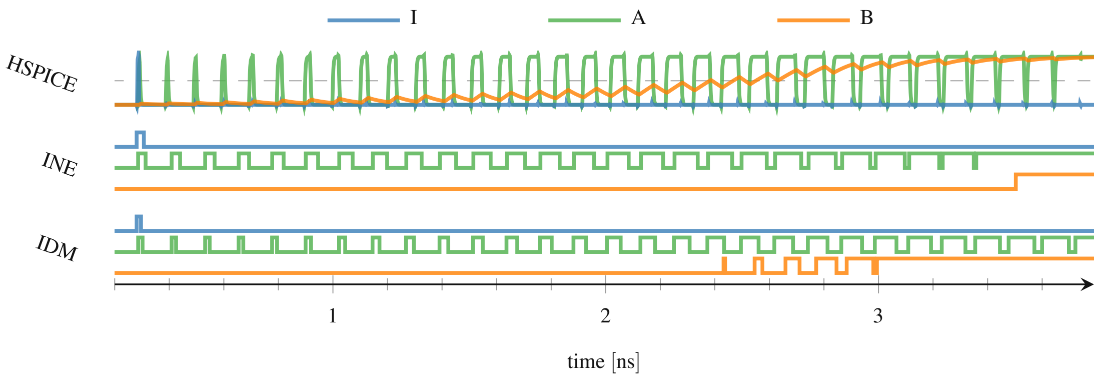


Figure 4.32: Analog and digital simulation results for the `OR Loop` *with long feedback.*

4.6.4 `OR Loop` with Long Feedback

For our first experiments we insert thirty Buffers into the feedback path of the `OR Loop` in Figure 4.29. In this setup it is, for $\delta_\infty^\uparrow = \delta_\infty^\downarrow$, possible to generate several periodic signals in the loop, since the signal rise/fall time, i.e., the time it takes to switch from `GND` to V_{DD} or reverse, is significantly smaller than the overall delay of the loop. However, the static delay values extracted after place & route did not match: Rising transitions are delayed less than falling ones, leaving only one Δ^I that perfectly compensates the increase in Δ_n^{HI} by pulse degradation effects and thus creates infinite oscillation.

HSPICE

Figure 4.32 (top) shows the analog simulation results for an initially very short pulse that grows and eventually settles the loop at V_{DD}. Clearly visible is also the impact of the high capacitive load: Since the transitions at node A are very fast compared to node B it actually seems as if charging and discharging curves are switched immediately when an input transition occurs. Recall that this perfectly matches the analog domain model of the IDM (cf. Section 4.3.1). Consequently the threshold (dashed line) is crossed multiple times, whereat the time difference between rising and falling crossing strictly increases.

Noteworthy is the high sensitivity of the feedback loop in this state and thus the very low probability to reach it. We had to vary Δ^I in steps of 1 as (10^{-18} s) in order to eventually generate an oscillation trace inside the loop that lasted up to 4 ns.

INE

At a first glance the inertial delay results shown in Figure 4.32 (middle) look very similar to the analog results. The short pulse in the beginning increases until, finally, the loop is constant `HI` and thus also node B gets `HI`. However, on closer examination severe shortcomings become apparent. First of all the shown pulse is the shortest Δ_0^{HI} that can be inserted into the loop. Smaller ones are removed by a high-delay Buffer upstream, since inertial delay blocks all pulse-widths smaller than the delay of the corresponding

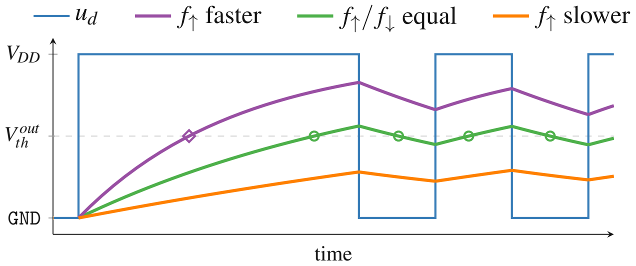


Figure 4.33: Analog waveform u_r of the IDM for various combinations of $f_\uparrow$ and $f_\downarrow$.

gate. This indicates a general problem: A gate with long delay close to the input of the circuit removes a large share of all possible input pulses. This may include relevant ones, as it is the case in this example, thereby making it impossible to detect any infinite oscillations or ones that eventually return to LO.

Regarding the output transition another important property can be observed. The signal at node B only switches to HI after the loop has fully settled, i.e., the oscillations have ceased. This can again be explained by the succeeding gate, whose delay is bigger than the feedback delay. Overall it thus serves as a metastability filter, which does not correspond well to the analog simulations, where the threshold is already crossed way before the loop is fully locked. We can conclude that INE is not well suited to properly describe the exact behavior of the circuit in such circumstances. In particular, it is impossible to achieve pulses at node B for the inertial delay model: only a single transition is observed or none at all.

IDM

Compared to INE the Involution Delay Model achieves a much more fine grained description of the analog behavior. First of all, any value of Δ_0^{HI} can be studied, also ones that quickly decay. Figure 4.32 (bottom) shows a simulation with increasing Δ_n^{HI} for ascending n: Internally $f_\uparrow$ is utilized more and more, increasing also the mean value of u_r steadily (cf. Figure 4.8), eventually crossing V_{th}^{out} and resulting in the digital oscillations on node B(cf. Figure 4.8). This is very much in accordance with our HSPICE simulations.

By properly tuning Δ^I it is even possible to achieve the infinite pulse train, i.e., the one that actually recreates itself. Note that this trace did not produce even a single transition on B, which reveals an unfavorable property of the IDM: The exact voltage range, where the internal, analog waveform stabilizes in an oscillatory manner, depends on $f_\uparrow$ & $f_\downarrow$ and might not include the threshold voltage. Based on the chosen V_{th}^{out}, the IDM possibly fails to reveal the internal unstable behavior at all, which was already mentioned in Observation 6. Figure 4.33 illustrates this situation by showing the analog trace u_r of the IDM for a given digital input and varying $f_\uparrow$ and $f_\downarrow$. Note that only for equally fast switching waveforms a pulse train is observed at the output in this case.

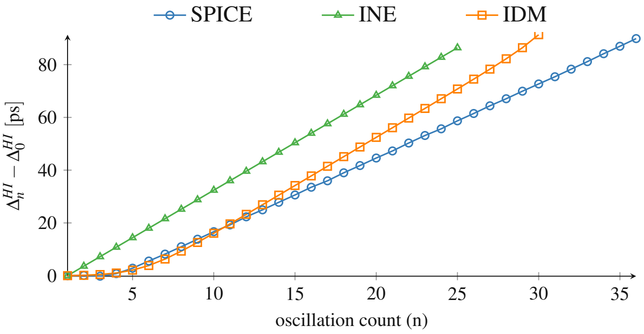


Figure 4.34: Increase of the pulse train high time compared to its initial value.

Comparison

A major difference between IDM and INE can also be identified by investigating the evolution of Δ_n^{HI}, as shown in Figure 4.34. The rate of growth is determined by the difference between the delay variation of falling and rising transition. For INE this variation is constant, leading to a constant rate and thus a linear shape. HSPICE and IDM show, however, a quite different behavior. For small n, Δ_n^{HI} increases only marginally, as we are running the circuit initially near the metastable point, i.e., where pulses recreate themselves. For bigger pulses the rate quickly increases.

Very interesting is the nonlinear increase of IDM. Intuitively, Δ_n^{HI} is expected to settle at a constant rate since for large values of T, the IDM and inertial delay are equal. While this is true, one has to consider that the increase in Δ_n^{HI} causes a drop of Δ_n^{LO}, which then experiences pulse-width degradation and thus further enhances the increase rate of Δ_n^{HI}. Note that the increasing differences to HSPICE are a result of inaccurate delay values extracted from the design tools, which we describe in detail in [7].

4.6.5 `OR` Loop with Direct Feedback

Reducing the Buffer count in the feedback path, and thus its delay, causes rising and falling transitions to be moved closer together, while leaving the rise and fall time untouched. Eventually they get so close that GND/V_{DD} are not reached any more. The effects of these changes on the infinite oscillatory behavior are as follows: As long as there is at least one gate in the loop still performing full range switching, which is possible due to differing parasitics, oscillations with a reduced amplitude, i.e., within the range $[V_L, V_H]$ with $V_L >$ GND and $V_H < V_{DD}$, are possible. In Section 4.6.6, we will investigate such a setup. Note that, due to the lower amplitude, the time between succeeding threshold crossings declines. Reducing the delay further eventually leads to a damped oscillation, which approaches a constant value, the metastable voltage. The time it takes to reach the final value decreases by reducing the number of gates in the path. When there is only a plain wire left, no more oscillations occur and the constant value is approached immediately. We chose exactly this setup for our simulations to evaluate how digital predictions represent such a behavior.

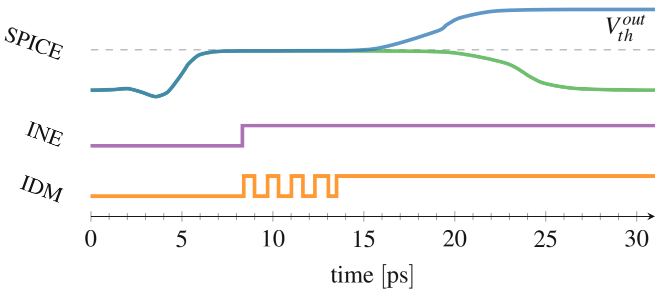


Figure 4.35: Analog and digital results of the `OR Loop` *with direct feedback (node A).*

HSPICE

Analog simulations confirm the intuitive explanation for the metastable case presented above. Figure 4.35 shows two traces on node A, which stay at a constant value near V_{th}^{out} for some time. Eventually they resolve to `LO` in one case and to `HI` in the other one. The fact that the corresponding Δ^I differ by merely 1 as (10^{-18} s) and, nonetheless, it is only possible to stay in the metastable state for a few picoseconds (10^{-12} s), indicates the very high sensitivity of the circuit.

INE

As described in Section 4.6.4 the shaping gates at the input already filter all small pulses. In fact, only pulses longer than the delay of the storage loop are able to pass, causing an immediate switch to `HI`. Thus, for INE, the simulation either delivers a single rising transition on all wires or none at all. While this coincides with HSPICE at a first glance very well, the metastable state, and thus the increase in delay, are not revealed, suggesting falsely a settled and well defined behavior.

IDM

Although the analog simulations did not show any V_{th}^{out} crossing during metastability, IDM again delivers an oscillatory behavior, which seems to be awfully wrong. Recalling, however, the analog representation of IDM, i.e., the switching between $f_\uparrow$ and $f_\downarrow$, it becomes apparent, that the closest u_r can get to a constant intermediate value is to oscillate with high frequency around it. Therefore, in IDM a pulse train happens to indicate metastability.

Overall a pulse train can thus either describe real oscillations, as shown in the previous example, or a metastable state. How can these scenarios be distinguished? Based alone on the digital predictions this is unfortunately impossible. The only major difference between different oscillatory traces is the sequence Δ_n^{HI} respectively Δ_n^{LO}, which does not yield much information on their own. Only in combination with the switching waveforms $f_\uparrow$ & $f_\downarrow$ or the static delays $\delta_\infty^{\uparrow/\downarrow}$ it is possible to estimate the voltage gain of u_r during the high and low period. However, as a rule of thumb, one can say that if Δ_n^{HI} (Δ_n^{LO})

is approximately or lower than $\delta_\infty^\uparrow$ ($\delta_\infty^\downarrow$), damping and hence metastability has to be expected. In our setup we extracted $\delta_\infty^\uparrow = 4.6\,\text{ps}$ and $\delta_\infty^\downarrow = 5.8\,\text{ps}$ for the OR-gate, which is clearly more than Δ_n^{HI} respectively Δ_n^{LO} in Figure 4.35. While this looks, at first sight, as a disadvantage compared to INE, be reminded that also for the latter comparisons with the delay values are necessary to determine, if a pulse is close to suppression. Since knowing the peak values in the analog domain is so important, we expanded the InvTool, which now enables the designer to investigate the underlying analog waveform u_r for desired areas and time spans. This is, however, only suited for rough estimations and is not intended to replace analog simulations at all.

Overall it has to be stated that an oscillating simulation trace does not automatically indicate undesired behavior. Only when the period gets too small, which depends on the actual circuit at hand, ill shaped pulses or even metastability have to be expected.

4.6.6 SR Latch

After studying the general behavior of digital simulation approaches on the rather artificial OR Loop, we turn to the SR Latch in Figure 4.30. Interestingly, INE again fails to cover very important parts of the real behavior and thus delivers overly optimistic results, while IDM stays close to the analog trace. The latter even enables us to explore unfavorable input conditions, which we will use to artificially prolong metastability.

Set or Reset Input Pulse

For constant LO on either S or R, the SR Latch degrades to the OR Loop for the other input. Simulations thus lead to very similar results, which are shown in Figure 4.36 for a single up-pulse on S. Once again, the shortest pulse for INE that is able to pass the input Buffers immediately sets the loop, leading to a single transition. This strongly contradicts the analog simulation results, which show an oscillation in a range between GND and V_{DD}. As discussed earlier, such a behavior is possible if one of the gates in the path, in this case the Buffers, still issue full range waveforms at their output.

In contrast, the IDM describes the behavior during, and also the resolution out of, metastability so faithfully, that the digital predictions enabled us to search for “malicious” input conditions that prolong the metastable state. As shown in Figure 4.36, a very long HI phase Δ_5^{HI} on node T appears right before Q switches to constant HI. To prevent the oscillation from resolving, it would be necessary to reduce Δ_5^{HI}, which simultaneously increases Δ_5^{LO}. This can be achieved by setting the reset input R to HI, driving the NOR gate and thus T to LO. Issuing a pulse on R at an appropriate point in time should be able to restore the continuous trace, i.e, push the resolving memory loop back into metastability. Essential for success is the time of the rising transition, as it determines the width of Δ_5^{HI}. On the other hand, the falling transition can be issued at any point in time during the HI period of the other NOR-gate input, since in this case the reset input is masked anyway. Reiher *et al.* [20] described a similar effect when “kicking” synchronizers, i.e., abruptly changing an internal voltage value, which also led to a potential extension of the metastable state.

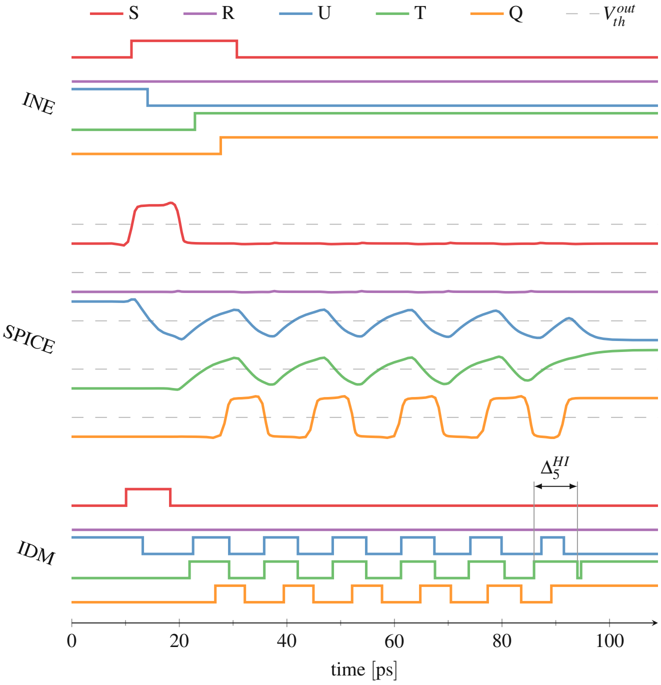


Figure 4.36: Analog and digital simulation showing metastability in the `SR Latch`*.*

Set and Reset Input Pulse

To verify the above considerations, we simply added a pulse on input R to the previous simulations. Results for INE, shown in Figure 4.37, reveal just one additional output transition delayed, as if the circuit was completely stable: Based on these predictions, one would assume the circuit is completely settled, i.e., operating under normal conditions. The instabilities the circuit is actually experiencing are invisible to INE.

HSPICE simulations shown in Figure 4.37 reveal the correctness of the IDM predictions. Not only is metastability extended but also a resolution to `HI` is forced. Please note that finding a proper spot for the reset pulse in HSPICE is a rather challenging task, as pulses already have an impact via coupling capacitances before they show up on the wire. This becomes obvious when observing, that the pulse on R that extends metastability appears way after the circuit had fully resolved in Figure 4.36. Separate simulations even reveal, that the signal on R is too short to have any impact on a fully settled memory loop. Only in combination with this particular circuit state a change in value becomes possible. Consequently, it is very important to investigate trains of very short pulses in combination with metastable states closely.

The theoretical predictions also fit very well to an actual execution of the IDM. Cutting Δ_5^{HI} indeed sets the loop back into metastability, resulting in a very realistic representation of the underlying analog behavior. Scheduling the reset pulse is, compared to HSPICE simulations, much easier, since only the `NOR` gate delay has to be considered.

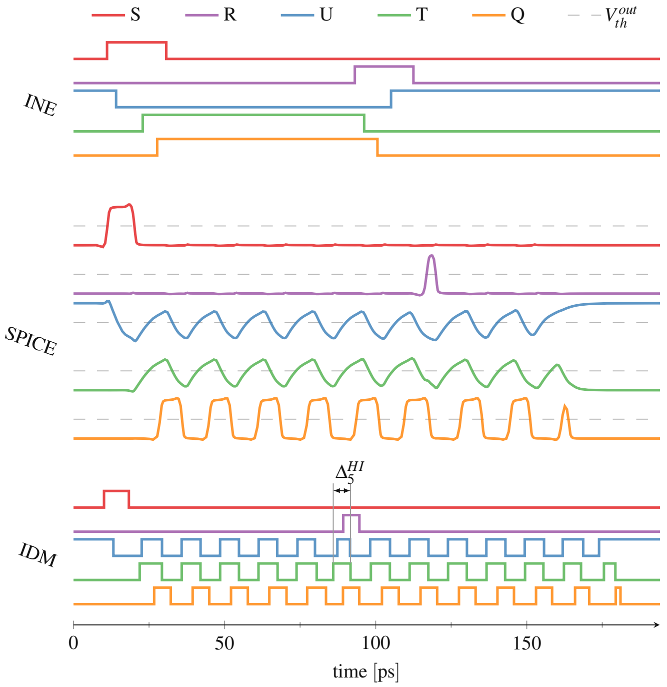


Figure 4.37: Analog and digital simulation of keeping the `SR Latch` *in metastability.*

4.6.7 **`Adder`**

At last we turn to our four Bit wide `Adder` shown in Figure 4.31. Analog HSPICE simulation results (see Figure 4.38) clearly show the propagation of the input pulse through the `Adder` and the corresponding degradation. Whether a pulse is observed on output S_i depends on (i) the initial input pulse-width on signal A_0 and (ii) the path length from S_i to A_0. The longer the path the bigger the input signal has to be to still have an impact. Interestingly the carry signals C_{i+1} seem to be generated faster than the sum value S_i. This can be seen very clearly by comparing S_3 and S_4 (the latter is actually the carry signal of the last full adder): While S_3 still barely crosses the threshold, S_4 already reaches all the way to GND/V_{DD}.

Overall, these results show the threat caused by glitches: Due to differing path lengths through the circuit, the input signal generates a varying number of output pulses with decreasing pulse-widths. This makes it more probable to steer a succeeding memory element into metastability: As on signals S_i many deviating transition times are generated, the chance to violate the setup and hold time of a succeeding Flip-Flop is elevated. Furthermore we want to emphasize that a metastable input value has the chance in this circuit to spread to five output signals and thus multiplies the effect of a single upset. This shows, once more, the importance of faithfully predicting glitches and metastability in the first place.

For INE, a very inconsistent buildup of transitions can be observed: Increasing the pulse-width of an input that only induces a pulse on S_1, by 1 fs, for example, causes a

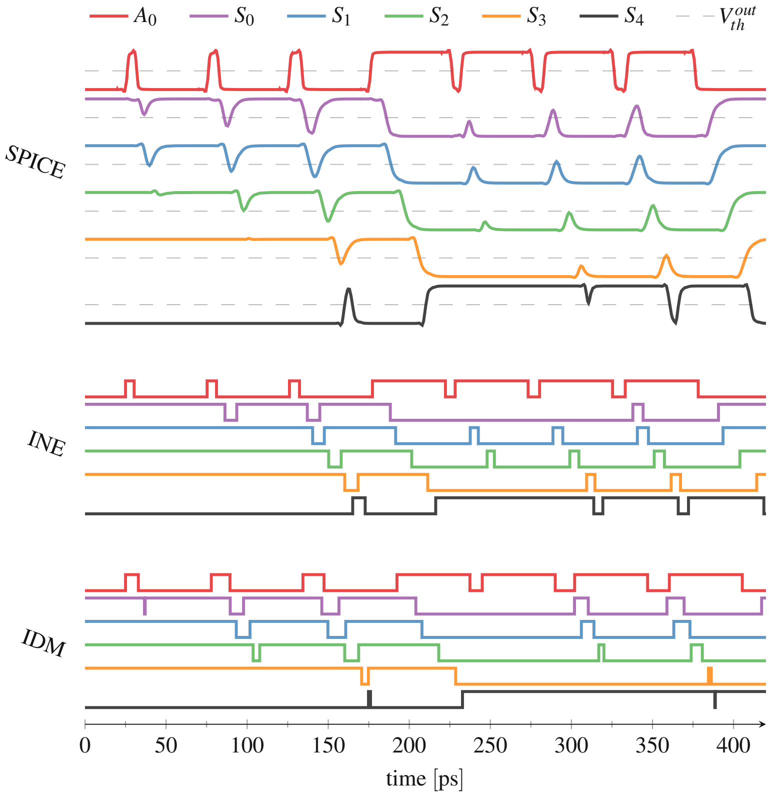


Figure 4.38: Analog and digital simulation of the `Adder` *with a glitch on its input.*

propagation all the way up to S_4. This is a direct consequence of the fact, that INE suppresses pulse-widths below a certain threshold, as was shown in Figure 4.1. For down-pulses on A_0, INE even delivers nonphysical results, as pulses on signal S_0 only appear after every other signal had been triggered. We retraced this to an unfortunate series of delays causing the signal closest to the input switch last, which is the actual opposite of what is seen in analog simulations. Finally note the constant shifts in pulse-widths, i.e., once a pulse appears on a signal it differs from the input pulse solely by a constant additive value. Since the respective values are very similar for each output comparable shapes are achieved.

A smooth increase of pulse-widths is naturally much better modeled by the IDM. In our simulations we even observe a strict causality among S_0 to S_4, i.e, S_i show a transition only after all $S_j, j < i$ have switched. Compared to INE this is a big improvement. Compared to HSPICE, however, some inaccuracies are still observable. For example, the quick increase on S_4 compared to S_3 is not well depicted. Possible causes are inaccurate delay values extracted from the design or the non optimal modeling of multi-input gates. Nonetheless, due to its accurate pulse-width degradation coverage, the IDM is able to provide overall very realistic results.

Be aware that in more complex circuits, e.g., the multiplier investigated in [96],

	INE		IDM		
#	$\overline{x}$ [s]	σ [s]	$\overline{y}$ [s]	σ [s]	overhead [%]
1	4.80	0.92	8.65	0.90	80.23
2	5.95	2.03	12.00	0.41	101.58
4	6.78	0.90	18.80	0.86	177.16
10	11.74	0.24	37.75	1.15	221.43
20	20.02	0.42	69.24	2.09	245.93
40	37.30	1.15	132.53	1.31	255.27
100	91.13	2.19	419.47	105.57	360.33
200	216.17	59.28	1492.03	317.88	590.20
400	1098.69	242.03	3674.48	584.66	234.44

Table 4.1: Mean simulation time and variance σ for both simulation methods and varying instances of the `Adder`*.*

glitches might also be triggered internally, leading to possible further signal degradations. This shows, once more, the importance of using the IDM also for obviously “harmless” input trajectories, since from an outside viewpoint internal race conditions can never be ruled out. The more reliable and trustworthy results, however, also come at a price in the form of a computational overhead, which will be thoroughly investigated in the sequel.

4.6.8 Overhead

Naturally, calculating the delay functions of the IDM, which includes exponential and logarithmic operations, is computationally more expensive than applying constant values paired with some minor removal checks for INE. To evaluate the overhead we thus ran extensive simulations and measured the execution time on our machine (Intel Xeon X5650, 1600 MHz, 32 GB RAM, CentOS 6.10). As test circuits we chose to use the `Adder` and the `Clock Tree` of an open source MIPS processor [13] that comprises of 227 inverters which drive 123 Flip-Flops. To also generate results for larger circuits we simply instantiated each unit multiple times, which had the expected impact on the overall simulation time.

For comparable results we have to ensure that INE and IDM process the same amount of transitions. Since their behavior mainly differs for high input frequencies, we use rather long pulses to assure no internal cancellations. Overall, 2×10^5 input transitions are applied per simulation run. The results are shown in Table 4.1 for the `Adder` and in Table 4.2 for the `Clock Tree`, whereat the first column denotes how often the circuit is instantiated. Most and foremost, we stress that, due to the high variance σ of the achieved execution times, we ran each simulation 30 times and calculated the average $\overline{x}$ respectively $\overline{y}$. Furthermore, the presented values only provide a lower bound, since real input signals may lead to very short internal pulses, which increases the workload of IDM compared to INE.

In essence the results show that the improved coverage of the IDM definitely comes at a cost. For the `Adder` the overhead increases with increasing circuit size. For 40

	INE		IDM		
#	$\overline{x}$ [s]	σ [s]	$\overline{y}$ [s]	σ [s]	overhead [%]
1	26.07	2.18	41.46	1.12	59.06
2	41.17	0.46	69.58	1.56	69.01
4	71.32	1.27	122.09	1.25	71.17
10	188.27	49.26	368.30	127.09	95.62
20	1016.23	265.44	1294.92	451.77	27.42
40	2430.30	406.60	3554.59	576.95	46.26

Table 4.2: Mean simulation time and variance σ for both simulation methods and varying instances of the `Clock Tree`.

instances it is almost 260 %. Please note that we consider the values for 100 and 200 not representative, since the disproportional increase in simulation time indicates a bottleneck of the computational platform that is not experienced by both methods in the same fashion. For the `Clock Tree` the overhead is lower and more constant, ranging from 27 to almost 100 %. We explain this result by the fact that only simple inverters are utilized, again showing that there is still a lot to be done in the IDM regarding multi-input gates.

4.6.9 Summary

In summary, our simulation results show that INE fails to model wide ranges of the analog behavior, especially high frequency oscillations and metastable intermediate voltages. The causes are single gates with larger delays, which have to be expected in almost every real world circuit. Relying exclusively on these predictions thus leads to a false sense of correctness. In these cases the IDM can significantly enhance the results, as it is able to stick much closer to the analog circuit behavior. This enables a more reliable identification of a wider range of malicious behavior in the digital domain and thus a better guidance of succeeding analog simulations. The latter are still mandatory to either confirm or dismiss the problems discovered in the digital domain.

Overall, state-of-the-art simulation suites tend to miss potentially malicious circuit behaviors like infinite oscillations or metastability and thus fail to deliver faithful predictions. Although an evaluation of the overhead showed a significant increase in simulation time, we think that the IDM poses a viable alternative to identifying malicious behavior, especially if confined to the most critical parts, and thus a significant enhancement of digital simulations.

One point deliberately neglected in this analysis is modeling accuracy. It is still computationally hard to characterize each single delay channel, relying heavily on analog simulations or crude approximations. Approaches that yield reasonable results based on available, or easily achievable, data are instrumental for making the IDM a truly competitive alternative to existing delay models.

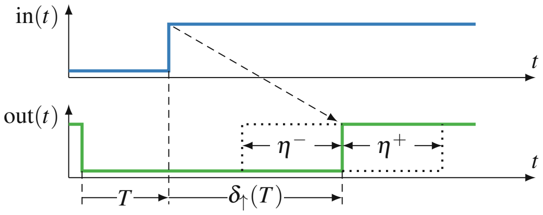


Figure 4.39: The $\boldsymbol{\eta}$-involution channel: Non-deterministic choice of the tentative output transition after applying $\delta_\uparrow(T)$.

4.7 Adding Non-Determinism

In the previous section, we have shown that simulating random traces using the IDM delivers good but not perfect approximations, primarily due to the applied simplifications. Since only pulses are used for characterization, and even those are not recreated perfectly as shown in Section 4.4.1, traces consisting of three or more transitions inevitably experience larger deviations. Furthermore, there are minor variations in the behavior of different instances of the same gate in a single chip or several chips. Therefore it would be convenient to cover such variations by allowing a small amount of non-determinism, which we will introduce in this section.

4.7.1 Introducing Adversarial Choice

In this section we generalize the circuit model from Section 4.3 to allow a non-deterministic perturbation of the output transition times after the application of the delay functions $\delta_\uparrow$ and $\delta_\downarrow$. The resulting output shifts of an $\boldsymbol{\eta}$-involution channel need *not* be the same for all applications of the delay functions; they can vary arbitrarily from one transition to the next. However, each perturbation needs to be within some pre-determined interval $\boldsymbol{\eta} = [-\eta^-, \eta^+]$. These non-deterministic choices can be used to model various effects in digital circuits that cannot be captured by single-history delay functions, ranging from arbitrary types of noise [24] to unknown variations of process parameters and operating conditions. Figure 4.39 shows the possible changes of the output transition time caused by the non-deterministic choices.

Formally, we change the notion of the *channel function* to accept an additional parameter: A channel has a channel function, which maps each pair (s, H) to an output signal, where s is the channel's input signal and H is a parameter taken from some suitable set of admissible parameters, i.e., any sequence of choices $\eta_n \in \boldsymbol{\eta}$ for $\boldsymbol{\eta}$-involution channels. The output transition generation is thus altered to $\delta_\uparrow(T) + \eta_n$ for a rising and $\delta_\downarrow(T) + \eta_n$ for a falling input transition.

Figure 4.40 depicts three example signal traces based on the same input. In the first case no shifts have been used while for the other two varying values were employed. One can observe that the adversary has the freedom to “de-cancel” pulses that would have canceled according to the delay function (second pulse in out$_2$), extend pulses (first pulse in out$_1$), and shift pulses (first pulse in out$_2$).

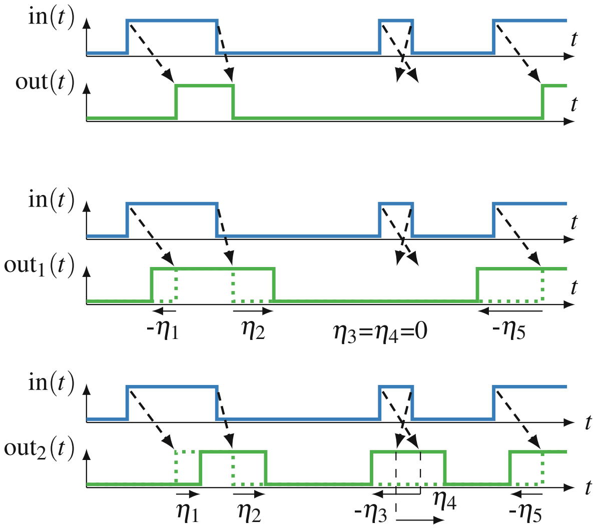


Figure 4.40: Multiple possible output behaviors for the same input trace, caused by different adversarial choices ($\eta_1, \eta_2, \dots$). The output transitions that would have been caused for $\eta_n = 0$ are dotted. Note that different adversarial choices usually change the history and, hence, T and thus $\delta(T)$.

4.7.2 Faithfulness of Involution Channels with Adversarial Choice

In this section, we will prove that $\boldsymbol{\eta}$-involution channels are faithful with respect to *Short-Pulse Filtration (SPF)* (cf. Definition 1). We start with the trivial direction: we prove that no circuit with $\boldsymbol{\eta}$-involution channels can solve the bounded-time variant of SPF (where the output must stabilize to constant 0 or 1 within bounded time). Note that this matches the well-known impossibility [146] of building such a circuit in reality. Indeed, the result immediately follows from the fact that the adversary is free to always choose $\eta_n = 0$, i.e., make the $\boldsymbol{\eta}$-involution channels behave like involution channels. In [17], it has been shown that no circuit with involution channels can solve bounded-time SPF, which completes the proof.

What hence remains to be shown is the existence of a circuit that solves SPF (with unbounded stabilization time) with $\boldsymbol{\eta}$-involution channels. We can prove that the circuit shown in Figure 4.41, which consists of a fed-back OR-gate forming the storage loop and a subsequent Buffer with a suitably chosen (high) threshold voltage (modeled as an Exp-channel), does the job. As a consequence, a circuit model based on $\boldsymbol{\eta}$-involution channels enjoys the same faithfulness as the involution channels of [34], even though its set of allowed behaviors is considerably larger.

Informally, we consider an input up-pulse of width Δ_0 at time 0 and reason about the behavior of the feed-back loop, i.e., the output of the OR gate. There are 3 cases: If Δ_0 is small, then the pulse is filtered by the channel in the feed-back loop. If it is large, the pulse is captured by the storage loop, leading to a stable output 1. For a certain range of Δ_0, the storage loop oscillates, possibly forever. In any case, however, it turns out that a

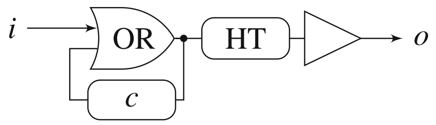


Figure 4.41: A circuit solving unbounded SPF, consisting of an OR-gate, with initial value 0*, fed back by channel c, and a high-threshold Buffer HT.*

properly chosen Exp-channel can translate this behavior to a legitimate SPF output.

Lemma 7. *If the input pulse's width Δ_0 satisfies $\Delta_0 \geq \delta_\infty^\uparrow + \eta^+$, then the output of the OR in Figure 4.41 has a unique rising transition at time 0, and no falling transition.*

Proof. Clearly, the output of the OR, hence the $\boldsymbol{\eta}$-involution channel's input, will have a rising transition at time 0. The corresponding rising transition occurs at the channel output at the latest at $\eta^+ + \delta_\infty^\uparrow \leq \Delta_0$. This guarantees the storage loop to lock, causing the output of the OR output to stick to 1. □

Lemma 8. *If the input pulse's width Δ_0 satisfies $\Delta_0 \leq \delta_\infty^\uparrow - \delta_{\min} - \eta^+ - \eta^-$, then the OR output in Figure 4.41 contains only the input pulse.*

Proof. The input signal contains only two transitions: one at time $t_1 = 0$ and one at time $t_2 = \Delta_0$. The earliest time when the output transition corresponding to the rising input transition can occur is $t_1' = \delta_\infty^\uparrow - \eta^-$. For the falling input transition, we thus get $T = \Delta_0 - \delta_\infty^\uparrow + \eta^-$, and observe that the corresponding falling output transition cannot occur later than $t_2' = \Delta_0 + \eta^+ + \delta_\downarrow(T)$. The two output transitions cancel iff $t_2' \leq t_1'$, which is equivalent to $X = \Delta_0 + \eta^+ + \delta_\downarrow(T) - \delta_\infty^\uparrow + \eta^- \leq 0$. Replacing Δ_0 with the upper bound from the lemma reveals $T \leq -\delta_{\min} - \eta^+$ and $X \leq -\delta_{\min} + \delta_\downarrow(-\delta_{\min} - \eta^+) \leq -\delta_{\min} + \delta_\downarrow(-\delta_{\min}) = 0$ by monotonicity of $\delta_\downarrow$ and (4.3), which concludes the proof. □

For an input pulse-width that satisfies $\delta_\infty^\uparrow - \delta_{\min} - \eta^+ - \eta^- < \Delta_0 < \delta_\infty^\uparrow + \eta^+$, the OR output signal may contain a series of pulses of widths $\Delta_0, \Delta_1, \Delta_2, \ldots$. In sharp contrast to standard involution channels [34], it is *not* the case that there is a unique value $\Delta_0 = \tilde{\Delta}_0$ that leads to an infinite series of (identical) pulses $\Delta_1 = \Delta_2 = \ldots$ Rather, due to the adversarial choices, there is a range of values for Δ_0 that may lead to a whole range of infinite pulse trains, with varying pulse-widths, which are difficult to bound.

An informal, high-level explanation of the approach that was eventually found to be successful is the following: we identified a self-repeating infinite "worst-case pulse train", which ensures that any adversarial choice that deviates from it at some point causes the subsequent pulses to die out, i.e., to resolve to a stable 1. In more detail, let Δ_0 be such that an infinite self-repeating pulse train $\Delta = \Delta_1 = \Delta_2 = \ldots$ exists, subject to the constraint that the adversary deterministically takes all rising transitions maximally (η^+) late and all falling transitions maximally (η^-) early. Note that this adversarial choice actually minimizes Δ_n for any given Δ_{n-1}. Therefore, given a pulse $\Delta_{n-1} = \Delta$, any other adversarial choice (as well as any larger $\Delta_{n-1} > \Delta$) leads to a subsequent pulse with $\Delta_n > \Delta$. As a consequence, Δ is an *upper* bound for the width of *every* pulse Δ_n, $n \geq 1$, occurring in an arbitrary *infinite* pulse train: if some $\Delta_{n-1} > \Delta$ ever happens, then $\Delta_{n+\ell} > \Delta$ for every $\ell \geq 0$ as well; in fact, Lemma 11 will reveal that the pulse train will only be finite in these cases.

Similarly, since the adversarial choice that minimizes the up-time Δ_n simultaneously maximizes the down-time Δ'_n of a pulse, we also get a a lower bound $\Delta'_n \geq P - \Delta$ for all pulses in an arbitrary infinite pulse train, where P is the period of our infinite self-repeating pulse train.

For these arguments to work, we need to restrict the adversarial choice for the feed-back channel in Figure 4.41:

$$\eta^+ + \eta^- < \delta_\downarrow(-\eta^+) - \delta_{\min} \tag{C}$$

Formally, we have the following Lemma 9:

Lemma 9. *Consider the circuit in Figure 4.41 subject to constraint* (C)*. Assume that the input pulse-width Δ_0 is such that it results in an infinite pulse train $\Delta_0, \Delta_1, \ldots$ occurring at the output of the OR. Then, for every $n \geq 1$, the up-time Δ_n satisfies $\Delta_n \leq \Delta$, the down-time Δ'_n (preceding the pulse with up-time Δ_n) satisfies $\Delta'_n \geq P - \Delta$, and $P_n = \Delta_n + \Delta'_{n+1} \geq P$. Herein, $\Delta = \delta_\downarrow(\eta^+ - \tau)$ with $\Delta < \delta_{\min}$ is the up-time of an infinite self-repeating pulse train with period $P = \tau$ and duty cycle $\gamma = \Delta/P$, with $\tau > 0$ denoting the smallest positive fixed point of the equation $\delta_\downarrow(\eta^+ - \tau) + \delta_\uparrow(-\eta^- - \tau) = \tau$, which is guaranteed to exist and satisfies $\eta^+ + \delta_{\min} < \tau < \min(-\eta^- + \delta^\downarrow_\infty, \eta^+ + \delta^\uparrow_\infty)$.*

Proof. In the circuit of Figure 4.41, the n^{th} input pulse of the $\boldsymbol{\eta}$-involution channel c is just its $(n-1)^{\text{th}}$ output pulse. Therefore, for all $n > 1$, the output pulse-width Δ_n under the worst-case adversarial choice of η^+-late rising and η^--early falling transitions evaluates to

$$\begin{aligned} \Delta_n = f(\Delta_{n-1}) \quad &= \quad \delta_\downarrow(\Delta_{n-1} - \eta^+ - \delta_\uparrow(-\Delta_{n-1})) \\ &\quad\quad + \Delta_{n-1} - \eta^- - \eta^+ - \delta_\uparrow(-\Delta_{n-1}) \ . \end{aligned} \tag{4.12}$$

The sought fixed point Δ of (4.12) resulting in a infinite pulse train is obtained by solving $\Delta = f(\Delta)$, which yields

$$\delta_\downarrow(\Delta - \eta^+ - \delta_\uparrow(-\Delta)) = \eta^- + \eta^+ + \delta_\uparrow(-\Delta) \ . \tag{4.13}$$

Applying the involution property to (4.13) results in $\Delta - \eta^+ - \delta_\uparrow(-\Delta) = -\delta_\uparrow(-\eta^- - \eta^+ - \delta_\uparrow(-\Delta))$ and further in

$$\Delta + \delta_\uparrow(\,-\eta^- - \eta^+ - \delta_\uparrow(-\Delta)) = \eta^+ + \delta_\uparrow(-\Delta) \ . \tag{4.14}$$

Defining $\tau = \eta^+ + \delta_\uparrow(-\Delta)$, rewriting it to $-\delta_\uparrow(-\Delta) = \eta^+ - \tau$ and applying the involution property, we observe

$$\Delta = \delta_\downarrow(\eta^+ - \tau) \ . \tag{4.15}$$

Using (4.15) and (4.4) in (4.14) yields the fixed point equation stated in our lemma:

$$\delta_\downarrow(\eta^+ - \tau) + \delta_\uparrow(-\eta^- - \tau) = \tau \ . \tag{4.16}$$

Now assume that the smallest fixed point $\tau > 0$ of (4.16), and hence Δ of (4.12), exists. Then, in any infinite pulse train, any pulse $\Delta_{n-1} > \Delta$, $n > 1$, and/or any non-worst-case adversarial choice (also in the case $\Delta_{n-1} = \Delta$) leads to a subsequent pulse with $\Delta_n > \Delta$. As a consequence, Δ is indeed an upper bound for the width of *every* such pulse.

We will proceed in our proof with establishing constraints on η^-, η^+ that guarantee the existence of a solution $\tau > 0$ of (4.16). For this purpose, we introduce the function

$$h(\tau) = \delta_\downarrow(\eta^+ - \tau) + \delta_\uparrow(-\eta^- - \tau) - \tau \ . \tag{4.17}$$

and show that there are values $\tau_0 < \tau_1$ where $h(\tau_0) > 0$ but $h(\tau_1) < 0$. Since $h(.)$ is continuous, this ensures the existence of $\tau_0 < \tau < \tau_1$ with $h(\tau) = 0$.

If we plug in $\tau_0 = \eta^+ + \delta_{\min}$ in (4.17), we find, by recalling (4.3), that $h(\eta^+ + \delta_{\min}) = \delta_\uparrow(-\eta^+ - \eta^- - \delta_{\min}) - \eta^+$. In order to guarantee that $h(\eta^+ + \delta_{\min}) > 0$ we need $\delta_\uparrow(-\eta^+ - \eta^- - \delta_{\min}) > \eta^+$. Rewriting this using the involution property requires $-\delta_\uparrow(-\eta^+ - \eta^- - \delta_{\min}) < -\delta_\uparrow(-\delta_\downarrow(-\eta^+))$ and hence $\eta^+ + \eta^- < \delta_\downarrow(-\eta^+) - \delta_{\min}$ as stated in constraint (C). Note that this implies $\eta^+ < \delta_{\min}$, since $\eta^+ + \eta^- \geq 0$.

For $h(\tau) < 0$, we simply obtain $-\infty$ from $\delta_\downarrow(\eta^+ - \tau)$ or $\delta_\uparrow(-\eta^- - \tau)$ by plugging in $\tau_1 = \min(-\eta^- + \delta_\infty^\downarrow, \eta^+ + \delta_\infty^\uparrow)$ in (4.17), noting that the involution property guarantees $-\infty = \delta_\uparrow(-\delta_\infty^\downarrow) = \delta_\downarrow(-\delta_\infty^\uparrow)$. Since all other terms of $h(.)$ are finite, the result is definitely < 0.

We still need to assure that the boundary interval for τ is not empty, i.e., that $\tau_0 = \eta^+ + \delta_{\min} < \tau_1 = \min(-\eta^- + \delta_\infty^\downarrow, \eta^+ + \delta_\infty^\uparrow)$. This is trivially the case if $\tau_1 = \eta^+ + \delta_\infty^\uparrow$. If $\tau_1 = \delta_\infty^\downarrow - \eta^-$, we need $\eta^+ + \eta^- < \delta_\infty^\downarrow - \delta_{\min}$, which is implied by constraint (C). Putting everything together, we can indeed guarantee a solution τ of $h(\tau) = 0$, which satisfies

$$0 < \eta^+ + \delta_{\min} < \tau < \min(-\eta^- + \delta_\infty^\downarrow, \eta^+ + \delta_\infty^\uparrow) \tag{4.18}$$

as stated in our lemma.

We can now determine the upper bound for Δ: Recalling the definition $\tau = \eta^+ + \delta_\uparrow(-\Delta)$, the lower bound on τ implies $\delta_{\min} < \tau - \eta^+ = \delta_\uparrow(-\Delta)$. Using the involution property, we can translate this to $-\delta_{\min} = -\delta_\downarrow(-\delta_{\min}) < -\Delta$. Applying (4.3), we end up with

$$\Delta < \delta_{\min} \tag{4.19}$$

as asserted in this lemma.

Regarding the periods of our pulses, we recall that our adversary takes all rising transitions maximally late and all falling transitions maximally early to minimize the high-times of the generated pulse train. The period $P_n = \Delta_n + \Delta'_{n+1}$ of the high-pulse Δ_n, measured from the rising transition of Δ_n to the rising transition of Δ_{n+1}, is $P_n = \delta_\uparrow(-\Delta_n) + \eta_n^+$, which is not difficult to see from the considerations leading to

(4.12). Hence, P_n only depends on the up-time Δ_n and the adversarial choice $\eta_n^+ \leq \eta^+$. It follows that the adversarial choices used for generating our minimal up-time pulse train simultaneously *maximize* both the period ($P = \delta_\uparrow(-\Delta) + \eta^+$) and the down-time ($P - \Delta$). As the adversary cannot further shrink the up-times of the pulses, it cannot further extend the down-times, without running into cancellations.

Formally, by the same argument as used for Δ, we find that no infinite pulse train can contain a pulse with a downtime strictly smaller than $P - \Delta$, where $P = P'$ is the period of our infinite Δ pulse train: analogously to P_n above, we find that the down-period $P'_n = \Delta'_n + \Delta_n$, measured between the falling transitions of Δ'_n and Δ'_{n+1}, evaluates to $P'_n = \delta_\downarrow(-\Delta'_n) - \eta_n^-$, which decreases with both Δ'_n and $\eta_n^- \leq \eta^-$. If $\Delta'_n < P - \Delta$ ever occurred, this would lead to $P'_n > P' = \delta_\downarrow(-P + \Delta) - \eta^-$. Since obviously $P' = P$, this implies $\Delta_n = P'_n - \Delta'_n > \Delta$, which contradicts the previously established upper bound $\Delta_n \leq \Delta$, however.

It hence only remains to evaluate $P = \delta_\uparrow(-\Delta) + \eta^+ = \tau$, which completes the proof. □

Lemma 10. *Consider the circuit in Figure 4.41 subject to constraint* (C). *The duty cycle γ_n of any pulse Δ_n, $n \geq 1$, in an infinite pulse train at the output of the OR-gate satisfies $\gamma_n \leq \gamma < 1$.*

Proof. According to Lemma 9, we have $\gamma_n = \frac{\Delta_n}{P_n} \leq \frac{\Delta}{P} = \gamma = \frac{\Delta}{\delta_\uparrow(-\Delta)+\eta^+} < \frac{\delta_{\min}}{\delta_{\min}+\eta^+} \leq 1$ for every $n \geq 1$ as asserted. □

We remark that $\eta^+ > 0$ allows strengthening constraint (C), which allows sharpening some inequalities in Lemma 9, namely, $\eta^+ + \eta^- \leq \delta_\downarrow(-\eta^+) - \delta_{\min}$, $\Delta \leq \delta_{\min}$, and $\eta^+ + \delta_{\min} \leq \tau$, without violating $\gamma < 1$ established in Lemma 10.

The following lemma implies that if $\Delta_1 > \Delta$ for Δ according to Lemma 9, then the sequence of generated output pulses Δ_n, $n \geq 1$, will be strongly monotonically increasing. Consequently, we will only get a bounded number of pulses at the output of the OR gate, with a stabilization time in the order of $\log_a(1/(\Delta_1 - \Delta))$ with $a = 1 + \delta'_\uparrow(0) > 1$.

Lemma 11. *For $f(.)$ given in* (4.12) *with fixed point Δ, we have $f(\Delta_1) - \Delta \geq (1 + \delta'_\uparrow(0)) \cdot (\Delta_1 - \Delta)$ if $\Delta_1 > \Delta$.*

Proof. Differentiation of (4.12) provides

$$\begin{aligned} f'(\Delta_1) &= (1 + \delta'_\uparrow(-\Delta_1))\Big(1 + \delta'_\downarrow(\Delta_1 - \eta^+ - \delta_\uparrow(-\Delta_1))\Big) \\ &\geq 1 + \delta'_\uparrow(0) \end{aligned} \tag{4.20}$$

because $\delta'_\uparrow(-\Delta_1) \geq \delta'_\uparrow(0)$ as $\Delta_1 > \Delta > 0$ and $\delta'(T) > 0$ is decreasing for all T as $\delta(.)$ is concave and increasing [17]. The mean value theorem of calculus now implies the lemma. □

The following lemma allows to extend the validity of the statement of Lemma 11 from the first output pulse Δ_1 to the initial input pulse Δ_0.

Lemma 12. *There is a unique* $\tilde{\Delta}_0$ *such that every input pulse-width* $\Delta_0 \geq \tilde{\Delta}_0$ *guarantees* $\Delta_1 \geq \Delta$ *as given in Lemma 9. Moreover,* $\Delta_1 - \Delta \geq (1 + \delta'_\uparrow(0)) \cdot (\Delta_0 - \tilde{\Delta}_0)$ *for* $\Delta_0 > \tilde{\Delta}_0$*, provided* $\Delta_0 < \delta^\uparrow_\infty + \eta^+$*.*

Proof. For the first pulse under the same worst-case adversarial choice as in Lemma 9, the analogous considerations as in the proof of Lemma 8 reveal

$$\Delta_1 = \delta_\downarrow(\Delta_0 - \eta^+ - \delta^\uparrow_\infty) + \Delta_0 - \eta^- - \eta^+ - \delta^\uparrow_\infty \ .$$

Defining the auxiliary function $g(\Delta_0) = \delta_\downarrow(\Delta_0 - \eta^+ - \delta^\uparrow_\infty) + \Delta_0 - \eta^- - \eta^+ - \delta^\uparrow_\infty$, it is apparent that $\Delta_1 = g(\Delta_0)$. Now, as $\lim_{\Delta_0 \to \eta^+ + \delta^\uparrow_\infty - \delta_{\min}} g(\Delta_0) \leq 0$ due to (4.3) and $\lim_{\Delta_0 \to \eta^- + \eta^+ + \delta^\uparrow_\infty} g(\Delta_0) = \delta_\downarrow(\eta^-)$, which is certainly (much) larger than Δ, cf. Lemma 9, there is indeed a unique $\tilde{\Delta}_0$ with $g(\tilde{\Delta}_0) = \Delta$ with the desired properties. The Lipschitz property is obtained exactly as in the proof of Lemma 11, by differentiating $g(\Delta_0)$ and using $\Delta_0 < \delta^\uparrow_\infty + \eta^+$. □

We summarize the consequences of the previous lemmas in the following theorem, which extends [41, Thm. 12] to the $\boldsymbol{\eta}$-involution model:

Theorem 13. *Consider the circuit in Figure 4.41 subject to constraint* (C)*. The fed-back OR gate with a strictly causal* $\boldsymbol{\eta}$*-involution channel has the following output when the input pulse has width* Δ_0*:*

- *If* $\Delta_0 \geq \delta^\uparrow_\infty + \eta^+$*, then the output has a single rising transition at time* 0*.*
- *If* $\Delta_0 \leq \delta^\uparrow_\infty - \delta_{\min} - \eta^+ - \eta^-$*, then the output only contains the input pulse.*
- *If* $\delta^\uparrow_\infty - \delta_{\min} - \eta^+ - \eta^- < \Delta_0 < \delta^\uparrow_\infty + \eta^+$*, then the output may resolve to constant* 0 *or* 1*, or may be an (infinite) pulse train, with* $\Delta_n \leq \Delta$ *and duty cycle* $\gamma_n \leq \gamma = \frac{\Delta}{\delta_\uparrow(-\Delta)+\eta^+} < 1$ *for* $n \geq 1$*.*

  If $\Delta_0 > \tilde{\Delta}_0$*, the output resolves to* 1 *within a stabilization time in the order of* $\log_a(1/(\Delta_0 - \tilde{\Delta}_0))$ *with* $a = 1 + \delta'_\uparrow(0) > 1$*.*

Proof. The statements of our theorem follow immediately from Lemmas 7, 8 and 9. Lemma 11 in conjunction with Lemma 12 reveals that the number of generated pulses is in the order of $\log_a(1/(\Delta_0 - \tilde{\Delta}))$ with $a = 1 + \delta'(0)$. □

For dimensioning the high-threshold Buffer, we can re-use Lemmas 13 and 14 from [41]:

Lemma 14 ([41, Lem. 13])**.** *Let* C *be an Exp-channel with threshold* V_{th} *and initial value* 0*, and let* $0 \leq \Gamma < V_{th}$*. Then there exists some* $\Theta > 0$ *such that every finite or infinite pulse train with pulse-widths* $\Theta_n \leq \Theta$*,* $n \geq 0$*, and duty cycles* $\Gamma_n \leq \Gamma$*,* $n \geq 1$*, is mapped to the zero signal by* C*.*

Lemma 15 ([41, Lem. 14])**.** *Let* $\Theta > 0$ *and* $0 \leq \Gamma < 1$*. Then, there exists an Exp-channel* C *such that every finite or infinite pulse train with pulse-widths* $\Theta_n \leq \Theta$*,* $n \geq 0$*, and duty cycles* $\Gamma_n \leq \Gamma$*,* $n \geq 1$*, is mapped to the zero signal by* C*.*

By choosing $\Gamma = \gamma(1+\varepsilon) < 1$ for some $\varepsilon > 0$ sufficiently small and Θ so large that the feed-back loop in Figure 4.41 has already locked to constant 1 at time $T+\Theta$, where T is the time when some pulse Δ_n, $n \geq 1$, of the feed-back loop with duty cycle $\gamma(1+\varepsilon)$ has started, we get the following: If SPF input pulse-widths Δ_0 and adversarial choices are such that no Δ_n reaches duty cycle $\gamma(1+\varepsilon)$, the output of the Exp-channel is constant zero; otherwise, there is a single up-transition (occurring only after $T+\Theta$) at the output. Therefore:

Theorem 16. *There is a circuit that solves unbounded SPF.*

Proof. If $\Delta_0 < \delta_\infty^\uparrow - \delta_{\min} - \eta^+ - \eta^-$, Theorem 13 ensures that the input of the high-threshold Buffer is constant 0, and so is the output. If $\Delta_0 > \delta_\infty^\uparrow + \eta^+$, then the input of the high-threshold Buffer experiences a single up-transition (at time 0), and so does the output (eventually).

For Δ_0 in between, we distinguish two cases: (i) Suppose Δ_0 and the adversarial choices are such that no Δ_n ever reaches duty cycle $\gamma(1+\varepsilon)$. Then, the minimality of the period P of the worst-case pulse train guaranteed by Lemma 9 implies that the input of the high-threshold Buffer sees pulses with duration at most Θ and duty cycle at most Γ. Hence, Lemma 15 guarantees a zero-output in this case.

For the other case (ii), which is guaranteed to happen when $\Delta_0 > \tilde{\Delta}_0$ (but may also occur for smaller values of Δ_0 in the case of certain adversarial choices), there is some time T where a 1-pulse Θ_n starts at the input of the Exp-channel that will (along with its subsequent 0) have a duty cycle $\Gamma_n \geq \Gamma > \gamma$. Moreover, by time $T+\Theta$, the last input transition (to 1) has already occurred. Lemma 15 not only guarantees that all pulses occurring before T cancel, but also the ones that occur before time $T+\Theta$: after all, even a single, long pulse $\Theta_n = \Theta$ would still be canceled. Therefore, since the input of the Exp-channel is already stable at 1 at time $T+\Theta$, only this final rising transition will eventually appear at the output. □

4.7.3 Simulations

In this section, we complement the proof of faithfulness provided in the previous section with simulation experiments and measurement results, which confirm that our $\boldsymbol{\eta}$-involution model indeed captures reality better than the original involution model [38]. Whereas more experiments, with different technologies and more complex circuits (including multi-input gates), would be needed to actually claim improved model coverage, our results are nevertheless encouraging.

We employ the same experimental setup as in [38], which uses UMC-90 nm and technology (T65) CMOS 7-stage Inverter chains as the primary targets. For technology (T65), we resorted to HSPICE simulations of a standard cell library implementation, for UMC-90, we relied on a custom ASIC [52]. The latter provides a 7-stage Inverter chain built from 700 nm x 80 nm (W x L) pMOS and 360 nm x 80 nm nMOS transistors, with threshold voltages 0.29 V and 0.26 V, respectively, and a nominal supply voltage of $V_{DD} = 1$ V. As all Inverter outputs are connected to on-chip low-intrusive high-speed analog sense

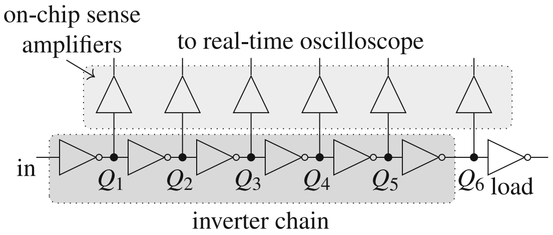


Figure 4.42: Schematics of the ASIC used for validation measurements. It combines an Inverter chain with analog high-speed sense amplifiers.

amplifiers (gain 0.15, -3 dB cutoff frequency 8.5 GHz, input load equivalent to 3 Inverter inputs), see Figure 4.42, which can directly drive the 50 Ω input of a high-speed real-time oscilloscope, the ASIC facilitates the faithful analog recording of all signal waveforms. Independent power supplies and grounds for Inverters and amplifiers also facilitate measurements with different digital supply voltages V_{DD}. For convenience, we provide the delay functions determined in [38] in Figure 4.43 ($\delta_\downarrow$ for UMC-90, measurements).

In order to validate the $\boldsymbol{\eta}$-involution model, we use the following general approach: Given simulated/measured output waveforms of a single Inverter excited by input pulses of different width, we compare (i) the digital output obtained from the simulated/measured waveforms with (ii) the predictions for some given delay function. The differences of the transition times of predicted and real digital output is a measure of modeling inaccuracy of the original involution model. If these differences can be compensated by suitable output shifts within $[\eta^-, \eta^+]$, however, we can claim that the $\boldsymbol{\eta}$-involution model matches the real behavior of the circuit for the given waveforms. Since faithfulness puts the severe constraint $\eta^+ + \eta^- < \delta_\downarrow(-\eta^+) - \delta_{\min}$ on η^+, η^-, recall Lemma 9, it is not clear under which conditions this claim indeed holds. In our evaluation, η^+ was first set to a suitable value ($\eta^+ > 0$) and afterwards η^- was calculated according to $\eta^- = \delta_\downarrow(-\eta^+) - \delta_{\min} - \eta^+$. Clearly, this results in different $\boldsymbol{\eta}$ bounds in each of the following figures.

The particular questions addressed in our experiments are, if the allowed range for η^+ and η^- is sufficient for the $\boldsymbol{\eta}$-involution model to capture the following:

(a) The circuit behavior under varying operation conditions: After all, circuit delays change with varying supply voltage and temperature, so the question remains to what extent the resulting fluctuations are covered by the $\boldsymbol{\eta}$-involution model.

(b) The circuit behavior under process variations: In general, circuit delays vary among manufactured chips and even across a single chip, so the question arises whether the $\boldsymbol{\eta}$-involution model based on a "typical" delay function covers typical variations.

(c) The real behavior of our Inverter chain with a (suitably parametrized) standard involution function, in particular, for Exp-channels. This would simplify model calibration, as it is typically easier to determine the Exp-channel model parameters for a given circuit [100], rather than its entire delay function.

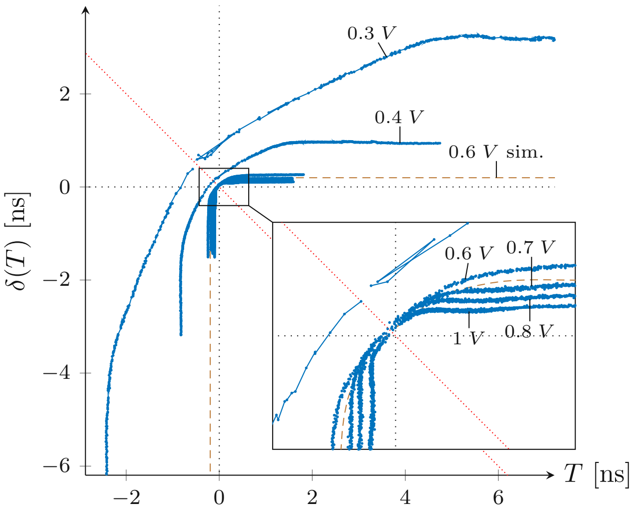
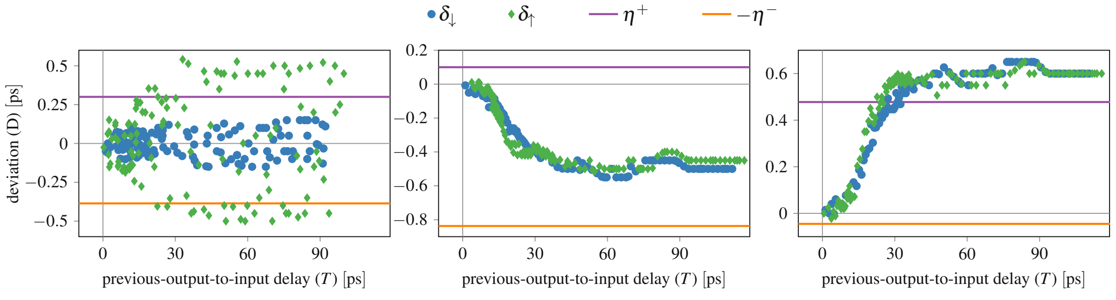


Figure 4.43: Measured $\delta_\downarrow$ for UMC-90 Inverter chain for $V_{DD} \in \{0.3, 0.4, 0.6, 0.7, 0.8, 1\}$ V and simulated (dashed brown) $\delta_\downarrow$ for $V_{DD} = 0.6$ V, taken from [38, Fig. 7].

To investigate question (a), i.e., the robustness against voltage variations, we added a sine wave to the voltage supply source (nominally $1.2\,\text{V} = V_{DD}$) with a period similar to the full range switching time of the Inverter and a magnitude of $0.012\,\text{V}$ (1 % of V_{DD}). We apply pulses with different widths to the input of the Inverter and record the output, whereat the phase of the sine wave is set for each pulse randomly between 0 and 360 degrees. In Figure 4.44, the deviation D between the prediction and the actual crossing over the previous-output-to-input delay T is shown. Despite the stringent bounds on $\boldsymbol{\eta}$, it is possible to fully cover the resulting delay variations for low T. For higher values, however, the $\boldsymbol{\eta}$-involution model does no longer apply. Please note that the huge difference between $\delta_\downarrow$ and $\delta_\uparrow$ can be easily explained by the fact that $\delta_\uparrow$ results in a falling transition at the output of the Inverter. In this case, the transistor connecting the output to the power supply gets closed more and more, reducing also the impact of the voltage variations. (When varying the ground level, the reverse can be observed.)

To answer question (b), we chose to vary the transistor width, which increases/decreases the maximum current and allows us to model variations of resistance and capacitance as well. The simulations themselves were carried out in the same fashion as described in the last paragraph, except that $V_{DD} = 1.2\,\text{V}$ was constant. Figure 4.44 (b) shows the results for 10 % wider transistors, where the $\boldsymbol{\eta}$-bound is even bigger than required. In contrast, the deviations for 10 % narrower ones [Figure 4.44(c)] exceed the $\boldsymbol{\eta}$-bound with increasing values of T. Unlike V_{DD} variations, varying transistor sizes, as expected, either increases or decreases the delay. This can be seen very clearly in the figures, as one trace is well below and one well above $D = 0$.

For question (c), we tried to fit an Exp-channel to the measurement data published in [38] and evaluated the deviations D between the resulting model predictions and the real digital output. Whereas the deviations over the whole range of T exceed the feasible $\boldsymbol{\eta}$-bounds, one can observe in Figure 4.45 that even the very simple Exp-channel only



(a) V_{DD} variation of 1 %

(b) transistor width +10 %

(c) transistor width -10 %

Figure 4.44: Deviation between predicted and actual V_{th} crossings for different variations. The η-bounds reveal good coverage for the $\boldsymbol{\eta}$-involution model.

results in minor mispredictions near $T = 0$.

We hence conclude that the $\boldsymbol{\eta}$-involution model indeed improves the modeling accuracy of the original involution model, despite the fact that the allowed non-determinism, i.e., $\boldsymbol{\eta}$, is quite restricted. Moreover, our simulation experiments indicate that the absolute deviations $|D|$ between model predictions and real traces is increasing with increasing previous-output-to-input delay T, making it possible to fully compensate D via $\boldsymbol{\eta}$ near $T = 0$. This is crucial, as our $\boldsymbol{\eta}$-bounds result from proving faithfulness, which involves the range $T \in [-\delta_{\min}, 0]$ only. For larger T, D grows bigger, but in this region, it might be feasible to also increase the allowed non-determinism as these values are almost irrelevant w.r.t. faithfulness.

4.7.4 Summary

We proved the surprising fact that adding non-determinism to the delays of involution channels, the only delay model known so far that is faithful for the SPF problem, does not invalidate faithfulness. Since this enables a wide ranges of changes to the signal, i.e., de-cancellation of removed pulses or pulse-width adaptions, we had to introduce an upper and lower bound for the possible shift values. As confirmed by some simulation experiments and even measurements, noise, varying operating conditions and process parameter variations hence do not a priori rule out faithful continuous-time, binary valued models.

The $\boldsymbol{\eta}$-involution model provides also a possible solution for the issues for varying threshold voltages we experienced in Section 4.4.2. Using it makes it possible to apply a constant time shift on each transition which essentially represents a translation between different threshold voltages. Although this approach sounds promising and even preliminary simulations on random HSPICE traces showed good results we decided to address the case of non-matching thresholds differently, as elaborated in the following section.

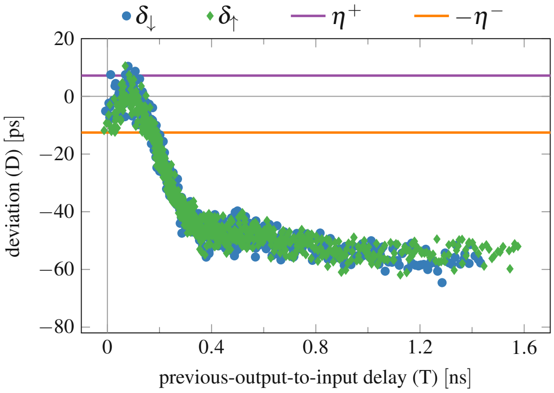


Figure 4.45: Fitting an Exp-channel involution to measured data. For small values of T the $\boldsymbol{\eta}$-bounds are sufficient to cover the deviations.

4.8 The Composable Involution Delay Model

In Section 4.4.2 we discussed several issues connected to the necessity of matching in- and output thresholds. From a physical point of view this makes little sense, as a unique analog waveform is of course consistent with any choice of threshold voltages. The step to arbitrary V_{th}^{in} and V_{th}^{out} in our digital abstraction is the goal of the Composable Involution Delay Model (CIDM), which will be defined in this section. It enables the composition of successive gates, simplifies their characterization, and exposes canceled transitions at the gate interconnect. While CIDM is not strictly equivalent to IDM, we are able to show that every CIDM circuit has an equivalent IDM description. This allows a transfer of properties known to be true for IDM to CIDM; in particular, faithful propagation of glitches. Simulations finally reveal a significantly improved accuracy compared to IDM.

4.8.1 Model Definition

According to Observation 5, using non-matching thresholds introduces a pure delay shift. The major building blocks of our CIDM are hence *PI channels*, which consist of a pure delay shifter with different shifts Δ^+ and Δ^- for rising and falling transitions[10] followed by an IDM channel. In order to also alleviate the problem of invisible oscillations identified in Observation 6, we re-shuffle the internal architecture of the original involution channels shown in Figure 4.8 to expose trains of canceled transitions on the interconnecting wires.

Theorem 17 (PI channel properties)**.** *Consider a channel PI formed by the concatenation of a pure delay shifter (Δ^+, Δ^-) with $\Delta^+ \in \mathbb{R}$ for rising and $\Delta^- \in \mathbb{R}$ for falling transitions followed by an involution channel c, given via $\delta_\uparrow(.)$ and $\delta_\downarrow(.)$ with minimum delay $\delta_{\min}$. Then PI is* not *an involution channel, but rather characterized by delay functions*

$$\overline{\delta}_\uparrow(\overline{T}) = \Delta^+ + \delta_\uparrow(\overline{T} + \Delta^+) \qquad \overline{\delta}_\downarrow(\overline{T}) = \Delta^- + \delta_\downarrow(\overline{T} + \Delta^-). \tag{4.21}$$

[10] A pure delay shifter with $\Delta^+ \neq \Delta^-$ causes a constant extension/compression of up/down input pulses by $\pm(\Delta^+ - \Delta^-)$.

These functions satisfy

$$\overline{\delta}_\uparrow(-\overline{\delta}_\downarrow(\overline{T}) - (\Delta^+ - \Delta^-)) = -\overline{T} + (\Delta^+ - \Delta^-) \tag{4.22}$$
$$\overline{\delta}_\downarrow(-\overline{\delta}_\uparrow(\overline{T}) + (\Delta^+ - \Delta^-)) = -\overline{T} - (\Delta^+ - \Delta^-) \tag{4.23}$$
$$\overline{\delta}_\uparrow(-\overline{\delta}^\uparrow_{\mathrm{min}}) = \overline{\delta}^\uparrow_{\mathrm{min}} \tag{4.24}$$
$$\overline{\delta}_\downarrow(-\overline{\delta}^\downarrow_{\mathrm{min}}) = \overline{\delta}^\downarrow_{\mathrm{min}} \tag{4.25}$$

for $\overline{\delta}^\uparrow_{\mathrm{min}} = \delta_{\mathrm{min}} + \Delta^+$ *and* $\overline{\delta}^\downarrow_{\mathrm{min}} = \delta_{\mathrm{min}} + \Delta^-$.

Proof. Consider an input signal consisting of a single up-pulse. Let t_i' resp. t_i be the time of the rising resp. falling input transition, t_p' resp. t_p the time of the rising resp. falling transition at the output of the pure delay shifter, and t_o' resp. t_o the time of the rising resp. falling transition after the involution channel. With $T = t_p - t_o'$, we get $\delta_\downarrow(T) = t_o - t_p$ as well as $t_p' = t_i' + \Delta^+$ and $t_p = t_i + \Delta^-$.

For the delay function $\overline{\delta}_\downarrow(\overline{T})$ of the PI channel, if we set $\overline{T} = t_i - t_o' = t_i - t_p + t_p - t_o' = -\Delta^- + T$, we find

$$\begin{aligned}\overline{\delta}_\downarrow(\overline{T}) &= t_o - t_i = t_o - t_p + t_p - t_i = \delta_\downarrow(T) + \Delta^- \\ &= \Delta^- + \delta_\downarrow(\overline{T} + \Delta^-)\end{aligned} \tag{4.26}$$

as asserted. By setting $\overline{T} = -\delta_{\mathrm{min}} - \Delta^-$ and using $\delta_\downarrow(-\delta_{\mathrm{min}}) = \delta_{\mathrm{min}}$ the equality $\overline{\delta}_\downarrow(-\delta_{\mathrm{min}} - \Delta^-) = \Delta^- + \delta_{\mathrm{min}}$ is achieved, which confirms (4.25).

By analogous reasoning for a down-pulse at the input, which results in the same equations as above with Δ^- exchanged with Δ^+ and $\delta_\downarrow(T)$ with $\delta_\uparrow(T)$, we also get

$$\begin{aligned}\overline{\delta}_\uparrow(\overline{T}) &= t_o - t_i = t_o - t_p + t_p - t_i = \Delta^+ + \delta_\uparrow(T) \\ &= \Delta^+ + \delta_\uparrow(\overline{T} + \Delta^+)\end{aligned} \tag{4.27}$$

as asserted. Setting $\overline{T} = -\delta_{\mathrm{min}} - \Delta^+$ and using $\delta_\uparrow(-\delta_{\mathrm{min}}) = \delta_{\mathrm{min}}$ confirms (4.24) as well. Using a simple parameter substitution equations transforms (4.26) and (4.27) to

$$\delta_\downarrow(T) = \overline{\delta}_\downarrow(T - \Delta^-) - \Delta^- \tag{4.28}$$
$$\delta_\uparrow(T) = \overline{\delta}_\uparrow(T - \Delta^+) - \Delta^+. \tag{4.29}$$

Utilizing these in the involution property of $\delta_\uparrow$ and $\delta_\downarrow$ provides

$$\begin{aligned}T &= -\delta_\uparrow(-\delta_\downarrow(T)) \\ &= -\overline{\delta}_\uparrow(-\delta_\downarrow(T) - \Delta^+) + \Delta^+ \\ &= -\overline{\delta}_\uparrow(-(\overline{\delta}_\downarrow(T - \Delta^-) - \Delta^-) - \Delta^+) + \Delta^+ \\ &= -\overline{\delta}_\uparrow(-\overline{\delta}_\downarrow(T - \Delta^-) + \Delta^- - \Delta^+) + \Delta^+.\end{aligned}$$

If we substitute $\overline{T} = T - \Delta^-$ in the last line, we arrive at

$$\overline{T} - (\Delta^+ - \Delta^-) = -\overline{\delta}_\uparrow(-\overline{\delta}_\downarrow(\overline{T}) - (\Delta^+ - \Delta^-)), \tag{4.30}$$

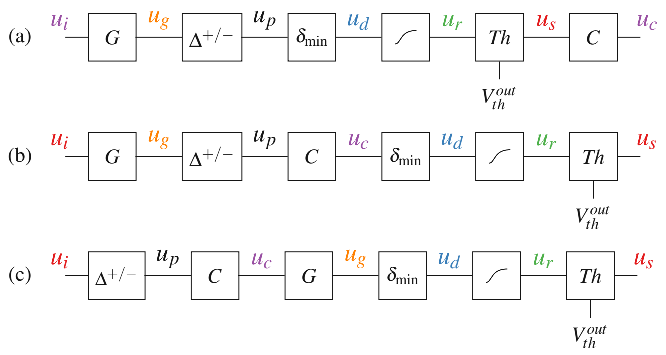


Figure 4.46: Candidate channel models for the CIDM.

which confirms (4.22).

Doing the same for the reversed involution property, provides

$$\begin{aligned} T &= -\delta_\downarrow(-\delta_\uparrow(T)) \\ &= -\overline{\delta}_\downarrow(-\delta_\uparrow(T) - \Delta^-) + \Delta^- \\ &= -\overline{\delta}_\downarrow(-(\overline{\delta}_\uparrow(T - \Delta^+) - \Delta^+) - \Delta^-) + \Delta^- \\ &= -\overline{\delta}_\downarrow(-\overline{\delta}_\uparrow(T - \Delta^+) + \Delta^+ - \Delta^-) + \Delta^-. \end{aligned}$$

If we substitute $\overline{T} = T - \Delta^+$ in the last line, we arrive at

$$\overline{T} + (\Delta^+ - \Delta^-) = -\overline{\delta}_\downarrow(-\overline{\delta}_\uparrow(\overline{T}) + (\Delta^+ - \Delta^-)), \tag{4.31}$$

which confirms (4.23).

□

Equation (4.21) implies that $\overline{\delta}_\uparrow(.)$ resp. $\overline{\delta}_\downarrow(.)$ are the result of shifting $\delta_\uparrow(.)$ resp. $\delta_\downarrow(.)$ along the 2nd median by Δ^+ resp. Δ^-. It is apparent from Figure 4.13, though, that the choice of Δ^+, Δ^- cannot be arbitrary, as it restricts the range of feasible values for T via the domain of $\delta_\uparrow(.)$ resp. $\delta_\downarrow(.)$ (see Definition 20 for further details).

This becomes even more apparent in the analog channel model. Figure 4.46 (a) shows an extended block diagram of an IDM channel, where we applied two changes: First, we added a (one-input, one-output) zero-time Boolean gate G. Second, we split the comparator at the end into a thresholder *Th* and a cancellation unit C. The *thresholder unit Th* outputs, for each transition on u_d, a corresponding V_{th}-crossing time of u_r, independently of whether it will actually be reached or not. For sub-threshold pulses, the transition might even be scheduled in the past. The *cancellation unit* C only propagates transitions that are in the correct temporal order. Obviously *Th* and C together are equivalent to a comparator.

At the beginning of the channel, the Boolean gate G (we assume a single-input gate for now) evaluates the input signal u_i in zero time and outputs u_g, which is subsequently delayed by the pure delay shifter $\Delta^{+/-}$. Here lies the cause of the problem: Since either

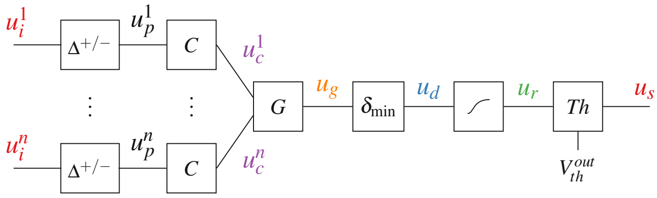


Figure 4.47: Channel model for CIDM.

$\Delta^+ < 0$ or $\Delta^- < 0$ it is possible that transitions on u_p are in reversed temporal order which, after being delayed by the constant pure delay $\delta_{\min}$, have to be processed in this fashion by the slope delimiter. The latter is, however, only defined on traces encoded via the alternating Boolean signal transitions' *Waveform Switching Times (WST)*, which occur in a strictly increasing temporal order and mark the points in time when the switching waveforms shall be changed. Moving the cancellation unit further to the front of the channel [see Figure 4.46 (b)] solves the problem, however, introduces another one at the gate G, which also expects transitions in the correct temporal order (note that this is not equal to WST since the pure delay is still missing).

One possible solution is to place the gate inside the channel, i.e., after the cancellation unit, as shown in Figure 4.46 (c). This solves our present problems but has the consequence, that transitions are interchanged among gates using the *Threshold Crossing Times (TCT)* encoding: The TCT encoding gives, in sequential order, the points in time when the analog switching waveform would have crossed V_{th}^{out} (it is not required that it actually does). Consequently, a signal given in TCT also exposes canceled transitions. Actually this is very convenient, since it allows us implicitly to detect oscillations independent of the chosen output threshold and thus solves the issue described in Observation 6.

Not all signals in Figure 4.46 can actually be mapped to TCT or WST; by suitably recombining the components in our CIDM channel, however, these encodings will be sufficient for our purposes. More specifically, TCT will be created by the thresholder Th, subsequently modified by the delay shifter, altered by the cancellation unit C, evaluated by the Boolean gate and finally transformed to WST by $\delta_{\min}$.

Now we are finally ready to formally define a CIDM channel (see Figure 4.47 for a general illustration). Note that, although a PI channel differs by its internal structure significantly from the CIDM channel, they are equivalent with respect to Theorem 17.

Definition 18. A CIDM channel comprises in succession of a pure delay shifter, a cancellation unit, a Boolean gate, a pure-delay unit, a shaping unit and a thresholding unit [see Figure 4.46(c)].

One may wonder whether CIDM channels could be partitioned also in a different fashion. The answer is yes, several other partitions are possible. For example, one could transmit signal u_g and move the slew-rate limiter and the thresholder to the succeeding channel. This would, however, mean that properties of single CIDM channels depend on the properties of both predecessor and successor gate, which complicates channel characterization and parametrization.

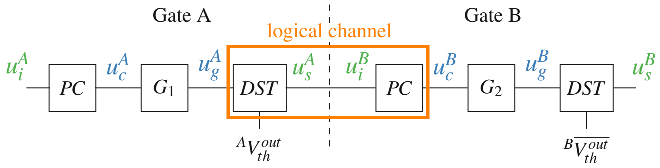


Figure 4.48: Channel model for proofs of the CIDM. Signals in blue have data type WST, those in green TCT.

The main practical advantage of a CIDM channel, which is a generalization of an IDM channel (just set $\Delta^- = \Delta^+ = 0$), is the additional degree of freedom for gate characterization in conjunction with the encapsulation of a single gate in a channel.

4.8.2 Glitch Propagation in the CIDM

Since CIDM channels do not satisfy the involution property, the question about faithful glitch propagation arises. After all, the proof of faithfulness of IDM [17] rests on the continuity of IDM channels, which has been shown only for involution delay functions. In this section, we will show that, for every modeling of a circuit with our CIDM channels, there is an equivalent modeling with IDM channels. Consequently, faithfulness of the IDM carries over to the CIDM.

For this purpose, we consider two successive CIDM channels and investigate the *logical channel*, i.e., the interconnection between two gates A and B as shown in Figure 4.48. For conciseness, we integrated the $\delta_{\min}$ pure delay, the slew-rate limiter and the threshold unit *Th* in a new block *DST*, and $\Delta^{+/-}$ followed by the cancellation unit *C* in the new block *PC*. Using this notation, the logical channel consists of the *DST* block of the predecessor gate G_1 and the *PC* block of the successor gate G_2. Overall this is just an IDM channel followed by an arbitrary pure delay shifter, which will be denoted in the sequel as *IP channel*. The following Theorem 19 proves the somewhat surprising fact that every IP channel satisfies the properties of an involution channel:

Theorem 19 (IP channel properties)**.** *Consider an IP channel formed by an involution channel given via $\delta_\uparrow(.)$, $\delta_\downarrow(.)$, followed by a pure delay shifter (Δ^+, Δ^-) with $\Delta^+, \Delta^- \in \mathbb{R}$. Then, it is an involution channel, characterized by some delay functions $\overline{\delta}_\uparrow(.)$, $\overline{\delta}_\downarrow(.)$.*

Proof. Consider an input signal consisting of a single up-pulse. Let t_i' resp. t_i be the time of the rising resp. falling input transition, t_c' resp. t_c the time of the rising resp. falling transition at the output of the involution channel, and t_o' resp. t_o the time of the rising resp. falling transition after the pure delay shifter. With $T = t_i - t_c'$, we get $\delta_\downarrow(T) = t_c - t_i$ as well as $t_o' = t_c' + \Delta^+$ and $t_o = t_c + \Delta^-$.

For the delay function $\overline{\delta}_\downarrow(\overline{T})$ of the IP channel, if we set $\overline{T} = t_i - t_o' = t_i - t_c' + t_c' - t_o' = T - \Delta^+$, we find

$$\begin{aligned} \overline{\delta}_\downarrow(\overline{T}) &= t_o - t_i = t_o - t_c + t_c - t_i = \Delta^- + \delta_\downarrow(T) \\ &= \Delta^- + \delta_\downarrow(\overline{T} + \Delta^+). \end{aligned} \tag{4.32}$$

By analogous reasoning for a a down-pulse at the input, which results in the same expressions with Δ^- exchanged with Δ^+ and $\delta_\uparrow(T)$ with $\delta_\downarrow(T)$, we also get

$$\begin{aligned}\overline{\delta}_\uparrow(\overline{T}) &= t_o - t_i = t_o - t_c + t_c - t_i = \Delta^+ + \delta_\uparrow(T)\\ &= \Delta^+ + \delta_\uparrow(\overline{T} + \Delta^-). \end{aligned} \tag{4.33}$$

Equations (4.32) and (4.33) are equivalent to

$$\delta_\downarrow(T) = \overline{\delta}_\downarrow(T - \Delta^+) - \Delta^- \tag{4.34}$$

$$\delta_\uparrow(T) = \overline{\delta}_\uparrow(T - \Delta^-) - \Delta^+ \tag{4.35}$$

which can be used in the involution property of $\delta_\uparrow$ and $\delta_\downarrow$ to achieve

$$\begin{aligned}T &= -\delta_\uparrow(-\delta_\downarrow(T))\\ &= -\overline{\delta}_\uparrow(-\delta_\downarrow(T) - \Delta^-) + \Delta^+\\ &= -\overline{\delta}_\uparrow(-(\overline{\delta}_\downarrow(T - \Delta^+) - \Delta^-) - \Delta^-) + \Delta^+\\ &= -\overline{\delta}_\uparrow(-\overline{\delta}_\downarrow(T - \Delta^+)) + \Delta^+ \end{aligned} \tag{4.36}$$

which confirms that the IP channel is indeed an involution channel. □

Note two very important properties: (1) $\overline{\delta}_{\min}$ of the IP channel is in general different from $\delta_{\min}$ of the constituent IDM channel. (2) The IP channel is strictly causal only if Δ^+ and Δ^- assure that the required condition $\overline{\delta}_\uparrow(0) > 0 \Leftrightarrow \overline{\delta}_\downarrow(0) > 0$ is satisfied, which is transformed, using (4.32) and (4.33), to

$$\overline{\delta}_\uparrow(0) = \Delta^+ + \delta_\uparrow(\Delta^-) > 0 \Leftrightarrow \overline{\delta}_\downarrow(0) = \Delta^- + \delta_\downarrow(\Delta^+) > 0. \tag{4.37}$$

At this point, the question arises whether it can be ensured that the logical channels in Figure 4.48 are always strictly causal. The answer is *yes*, provided that the interconnected gates are *compatible*, in the sense that the joined PC block of G_2 and the DST block of G_1 are compatible w.r.t. Observation 5. More specifically, the pure delays Δ^+ resp. Δ^- have to denote the time the rising resp. falling output transition of the DST block in G_1 needs to bridge the gap between V_{th}^{out*} of gate A and V_{th}^{in*} of gate B.

Defining Δ^+ and Δ^- in this fashion has one strong implication: Evaluating (4.32) and (4.33) with these values results in $\overline{\delta}_{\min} = \delta_{\min}$. To show the latter recall our analysis from Section 4.3.1 (especially the output trajectory in Figure 4.9) whereat V_{th}^{out*} of gate A corresponds to V_{th} and V_{th}^{in*} of gate B to V_s. For $V_{th}^{in*} > V_{th}^{out*}$, which corresponds to the case shown in the figure, it is possible to pick $T_1 \in [-\delta_{\min}, \infty]$ such that $\Delta^+ = T_1 + \delta_{\min}$ and consequently we get $\Delta^- = -(\delta_\downarrow(T_1) - \delta_{\min})$. Note that these values represent the relative distance of the point $(T_1, \delta_\downarrow(T_1))$ and $(-\delta_\downarrow(T_1), -T_1)$ on the delay function to $(-\delta_{\min}, \delta_{\min})$ as is shown in Figure 4.49. Plugging Δ^+ and Δ^- into (4.32) and (4.33) leads to mirrored movements that exactly compensate each others. More specifically, we get $\overline{\delta}_\downarrow(\overline{T}) = -(\delta_\downarrow(T_1) - \delta_{\min}) + \delta_\downarrow(\overline{T} + T_1 + \delta_{\min})$ and $\overline{\delta}_\uparrow(\overline{T}) = T_1 + \delta_{\min} + \delta_\uparrow(\overline{T} - (\delta_\downarrow(T_1) - \delta_{\min}))$, which lead, for $\overline{T} = -\delta_{\min}$ and by using the involution property $\delta_\uparrow(-\delta_\downarrow(T_1)) = -T_1$, to

$$\begin{aligned}&\overline{\delta}_\downarrow(-\delta_{\min}) = \delta_{\min} - \delta_\downarrow(T_1) + \delta_\downarrow(-\delta_{\min} + T_1 + \delta_{\min}) = \delta_{\min} \quad \text{and}\\ &\overline{\delta}_\uparrow(-\delta_{\min}) = T_1 + \delta_{\min} + \delta_\uparrow(-\delta_{\min} - \delta_\downarrow(T_1) + \delta_{\min}) = \delta_{\min}. \end{aligned} \tag{4.38}$$

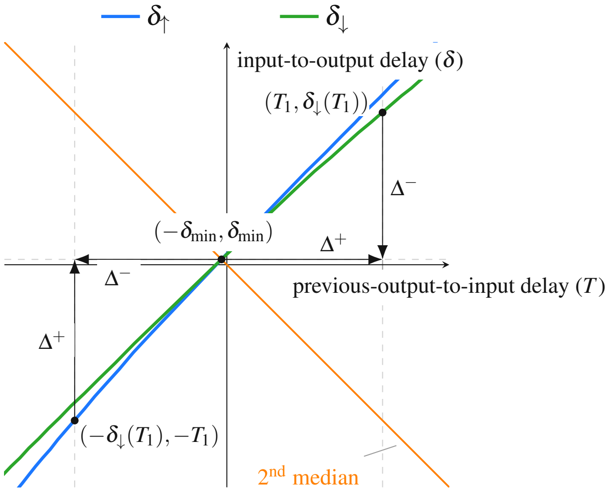

$\delta_\uparrow$
$\delta_\downarrow$
input-to-output delay (δ)
$(T_1, \delta_\downarrow(T_1))$
Δ^-
$(-\delta_{\min}, \delta_{\min})$
Δ^+
Δ^-
previous-output-to-input delay (T)
Δ^+
$(-\delta_\downarrow(T_1), -T_1)$
2nd median

Figure 4.49: Δ^+ and Δ^- shown on the delay functions of gate A. Due to the mirroring along the 2^{nd} median the values once refer to $\delta_\downarrow$ and once to $\delta_\uparrow$.

This shows that for proper choices of Δ^+ and Δ^-, the equality of the pure delays $\delta_{\min} = \overline{\delta}_{\min}$ is fulfilled. Note that after the shift the outer dots in Figure 4.49, i.e., at $\delta_\downarrow(T_1)$ and $\delta_\uparrow(-\delta_\downarrow(T_1))$ are located at $(-\delta_{\min}, \delta_{\min})$. The case $V_{th}^{in*} < V_{th}^{out*}$ and hence $\Delta^- > 0$, $\Delta^+ < 0$ can be handled analogously.

Definition 20 (Compatibility of CIDM channels)**.** Two interconnected CIDM channels are called *compatible*, if the logical channel between them is strictly causal.

Consequently, a logical channel connecting G_1 and G_2 is strictly causal if Δ^+ and Δ^- have been determined in accordance with Observation 5. If this is not the case, non-causal effects, like an output pulse crossing V_{th}^{out} without the corresponding input pulse crossing V_{th}^{in}, could appear.

Every chain of gates properly modeled in the CIDM can be represented by a chain of Boolean gates interconnected by causal IDM channels, with a "dangling" *PC* block at the very beginning and a *DST* block at the very end. Whereas the latter is just an IDM channel, this is not the case for the former. Fortunately, this does not endanger the applicability of the existing IDM results: As stated in property C2) for a circuit in [17, Sec. III], the original IDM assumes 0-delay channels for connecting an output port of a predecessor circuit C_1 with an input port of a successor circuit C_2. In the case of using CIDM for modeling C_1 and C_2, this amounts to combining the *DST* block of the gate that drives the output port of C_1 with the *PC* block of the input of the gate in C_2 that is attached to the input port. Note that the analogous reasoning also applies to any feedback loop in a circuit.

In addition, for an "outermost" input port of a circuit, we can demand that the connected gate must have a threshold voltage matching the external input signal, such that $\Delta^+ = \Delta^- = 0$ for the dangling *PC* component. Finally, in hierarchical simulations, where the output ports of some circuit are connected to the input ports of some other circuits, the situation explained in the previous paragraph reappears.

As a consequence, all the results and all the machinery developed for the original IDM [17] could, in principle, be applied also to circuits modeled with CIDM channels. Both impossibility and possibility results, and hence faithfulness, hold, and even the IDM digital timing simulation algorithm, as well as the InvTool, could be used without any change. Using the CIDM for circuit modeling is nevertheless superior, because its additional degree of freedom facilitates a more accurate characterization of the involved channels w.r.t. real circuits.

4.8.3 Experiments

In this section, we validate our theoretical results by means of simulation experiments. This requires two different setups: (i) To validate the CIDM, we incorporated a suitable simulation algorithm in the InvTool and compared the predictions for CIDM to other models. (ii) To establish the mandatory prerequisite for these experiments, namely, an accurate characterization of the delay functions of the gates, we employed a fairly elaborate analog simulation environment.

Comparable to our experiments in Section 4.6.2, we again relied on simulations using technology (T15). At first we developed a Verilog description of our circuits and used Genus & Innovus for optimization, placement and routing. We then extracted the parasitic networks between gates from the final layout, which resulted in accurate Spectre models. These results were used both for gate characterization and as a golden reference for our digital simulations.

Like in [17], our main target circuit is a custom Inverter chain. In order to highlight the improved modeling accuracy of CIDM, it consists of seven alternating high- and low-threshold Inverters. These are implemented by increasing the channel length of p- respectively nMOS transistors, which varies the transistor threshold voltages [110, Fig. 2], recall Section 2.5.3. For comparison, we also conducted experiments with a standard Inverter chain.

Regarding gate characterization for IDM, we used two different approaches. Recall from Observation 4 that fixing a single discretization threshold pins the value of all consistent $\delta_{\min}$, V_{th}^{in} and V_{th}^{out} throughout the circuit. In the variant of IDM called IDM*, we chose $V_{th}^{out*} = V_{DD}/2$ for the last Inverter in the chain, and determined the actual value of its matching V_{th}^{in*} by means of analog simulations. To obtain consistent discretization thresholds for the whole circuit, we repeated this characterization, starting from $V_{th}^{out*} = V_{th}^{in*}$ for the next Inverter up the chain. We thereby obtained values in the range $[0.301, 0.461]$ V, with $V_{th}^{in*} = 0.455\,\mathrm{V}$ for the first gate. Obviously, characterizing a circuit in this fashion is very time-consuming, as only a single gate in a path can be processed at a time.

Alternatively, we characterized every gate separately for $V_{th}^{out*} = V_{DD}/2$ and determined the matching V_{th}^{in*}, which we will refer to as IDM+. Note carefully that the discretization thresholds of connected gate out- and inputs differ for IDM+, such that an error is introduced at every interconnecting edge. Since the signals are very steep

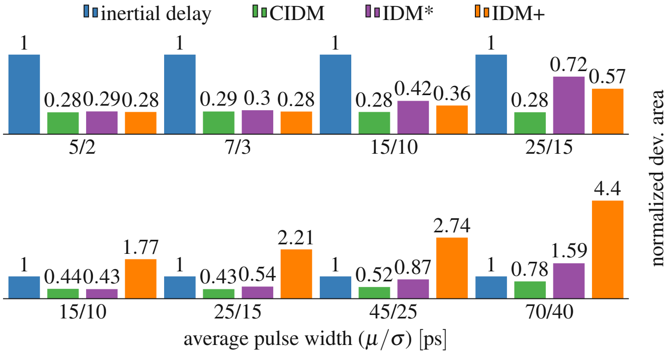


Figure 4.50: Accuracy, expressed as the normalized total deviation area of the digital predictions, relative to Spectre for the standard Inverter chain (top) and high/low threshold Inverter chain (bottom). Lower bars indicate better results.

near $V_{DD}/2$, we consider the deviation in general rather small. This circumstance is even more pronounced by the natural amplification of CMOS gates, causing the deviation of the input thresholds to be, in general, smaller than for V_{th}^{out}. Note that this was verified by our simulations of the standard Inverter chain.

However, although the misprediction is small, it is introduced for each transition at every gate. While this might be negligible for small circuits like our chain, the error quickly accumulates for larger devices leading to deviations even for very broad pulses. Thus, the IDM+ can be expected to deliver worse results than pure/inertial delay while being a computationally much more expensive approach. Indeed, for the gates used in our standard inverter chain, we recognized a clear bias towards $V_{th}^{in*} < V_{DD}/2$ for $V_{th}^{out*} = V_{DD}/2$. Finally, characterizing gates for CIDM was simply executed for $V_{th}^{out} = V_{th}^{in} = V_{DD}/2$.

The results for stimulating the standard Inverter chain, with 2500 normally distributed pulses of average duration μ and standard deviation σ, obtained by the InvTool for IDM*, IDM+, CIDM and the default inertial delay model, are shown in Figure 4.50 (top). The accuracy of the model predictions are presented relative to the digital predictions extracted from our golden Spectre simulations. For short pulses, IDM*, IDM+ and CIDM perform similarly. For broader pulses, we observe a reduced accuracy of IDM* and IDM+, which is primarily an artifact of the imperfect delay function approximation by the InvTool. We even observed settings, where CIDM does not even beat the inertial delay model, which can also be traced to this cause.

For our custom Inverter chain [Figure 4.50 (bottom)], CIDM outperforms, as expected, the other models considerably, whereas the IDM+ predictions are poor, even compared to inertial delays. This is a direct consequence of the non-matching threshold values and the accumulating error. IDM* achieves much better results, but still falls short compared to CIDM. For broader pulses, the latter performs comparable to inertial delay, since both use the same maximum delay $\delta_\infty^\uparrow$ and $\delta_\infty^\downarrow$. The degradation of IDM* is once again a result of the imperfect delay function approximations above.

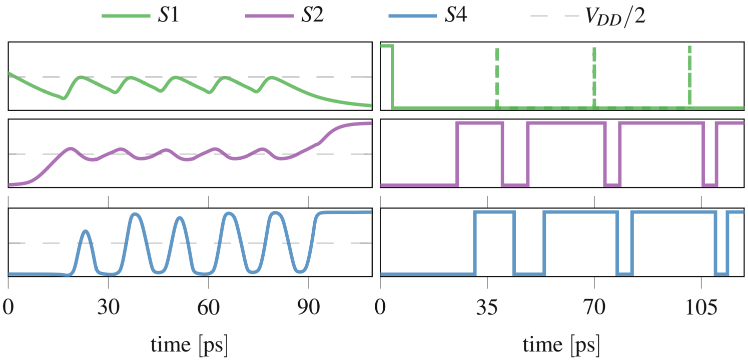


Figure 4.51: Analog and digital prediction of recovering sub-threshold waveform.

Finally, analog simulations in Figure 4.51 revealed that an oscillation slightly below $V_{DD}/2$ at the input of a low-threshold Inverter can still result in full range switches at the end of the chain. For IDM+ such traces get removed, whereat for IDM* this particular trajectory is actually visible. Nevertheless there are still infinitely many other possibilities that can not be detected by the latter. Please note that even if such traces do not propagate further it is important to know if the circuit has stabilized or not, e.g., for power estimations. The digital simulation results for the CIDM, shown on the right hand side of the figure, correctly predicts the regeneration of the pulses.

To summarize the results of our experiments, we highlight that the characterization procedure for IDM either requires high effort (IDM*) or may lead to modeling inaccuracies (IDM+). The CIDM clearly outperforms all other models w.r.t. modeling accuracy for our custom Inverter chain, and is also the only model that can faithfully predict the "de-cancellation" of sub-threshold pulses.

4.8.4 Summary

In this chapter we presented the Composable Involution Delay Model (CIDM), a generalization of the Involution Delay Model (IDM) that retains its faithful glitch-propagation properties. Its distinguishing properties are wider applicability, composability, easier characterization of the delay functions, and exposure of canceled pulse trains at interconnecting wires. The CIDM and our novel digital timing simulation algorithm have been developed on sound theoretical foundations, which allowed us to rigorously prove their properties. Analog and digital simulations for Inverter chains were used to confirm our theoretical predictions.

CHAPTER 5

Metastability Modeling

Zero time transitions in the digital domain solely indicate the point in time when the underlying analog waveform crosses the values V_{LO} resp. V_{HI}. The exact trajectory is not visible and thus a steep slope is implicitly assumed. Unfortunately, real signal do not necessarily behave in this fashion. For example, gradual transitions and even stalling at intermediate values for an arbitrary amount of time is well possible. While such *metastable* events are invisible in the digital domain they have an actual impact on the circuit, such as massively increased power consumption or inconsistent interpretation among succeeding gates. The latter is even more severe than a Byzantine fault in a communication network described by Lamport, Shostak, and Pease [142]. In this setup a faulty sender may transmit different information to its single receivers, but it is still assumed that the information is some legitimate digital value. By contrast, a metastable output is outside the digital model and, hence, impossible to contain/mask by classic digital fault-tolerance techniques, see [18]. Consequently separate metastability analyses are required that (i) identify problematic locations inside the circuit, (ii) determine possible erroneous behaviors and (iii) estimate their probabilities.

In this chapter we will investigate how physical considerations can be used to improve metastability analysis. After a short introduction we use the transistor models from Chapter 3 to study intermediate voltages in some simple circuits. Of interest for us is the S/T, which will be investigated in greater detail: After recalling previous research that revealed the potential of metastable behavior of some circuits, we introduce several approaches that allow to characterize an S/T both in a static and dynamic fashion. We then use these methods to evaluate modern implementations, i.e., we compare the achieved behavior to calculations on an optimal device and explore possible input trajectories to drive an S/T into metastability. After shortly reviewing the effects of cascading multiple units we finally present a novel approach towards finding a single representative number that quantifies the risk of metastability.

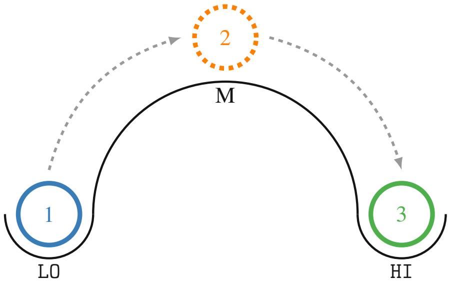


Figure 5.1: Graphical explanation of the metastable state. When moving a ball between the stable states `LO` *and* `HI` *the metastable position M has to be crossed. In M the ball may stay for an infinite amount of time.*

5.1 Metastability Analysis

Since the seminal work by Kinniment and Edwards [149], Chaney [145], Veendrick [144] and Marino [143, 146] it has been known that the source of long-time intermediate voltages inside a circuit are state-holding devices. The concrete values depend on various parameters such as the internal structure and transistor sizings. Depending on the discretization threshold voltages V_{LO} and V_{HI}, multiple representations in the digital domain, such as (i) a (late) transition, (ii) two transitions or (iii) no transition at all, are possible. Note carefully that also logic gates can provide an intermediate output voltage. In contrast to state-holding devices, however, they require an intermediate input voltage to do so. That is, they only propagate metastability but do not generate it.

In fact Marino [143] showed that metastability can be observed in any bistable unit. For this purpose he associated stable states to energy minima. This representation is in accordance to our analyses in Chapter 2, where we saw that electrons always try to achieve the lowest possible energy. Recall that the energy bands in a semiconductor had to be continuous, which infers that between two minima a maximum, which is called in this case the metastable state, has to be encountered.

This circumstance is shown in Figure 5.1. Depending on how much energy one applies to the ball at `LO` it either goes all the way to `HI` (full range switch), rises a little bit and drops back down to `LO` (sub-threshold pulse) or stays near the top for some time before dropping either towards `LO` or `HI`. This representation already shows three very important properties of metastability: 1) The barrier at M is perfectly flat, enabling the ball to stay at this point for an unbounded amount of time. Consequently metastability could be, in principle, maintained forever. 2) Every transition between `LO` and `HI` passes M, meaning that metastability cannot be avoided. It can only be made less probable. Note that in this delicate position tiny disturbances already have huge impact. 3) If the ball leans towards one side it will quickly picks up pace and thus *resolves* the metastable upset to either side within the *resolution time*.

Within a single clock domain, metastable upsets can be well controlled by careful design. However, at clock domain crossing boundaries the risk due to asynchronous inputs cannot be avoided altogether. As the countermeasure of choice *synchronizers* [53]

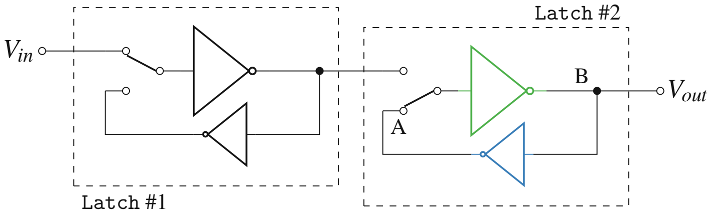


Figure 5.2: Internal structure of a Flip-Flop, consisting of two Latches in succession. Please note that the switches are controlled by the same signal such that one Latch stores a value while the other one propagates its input.

are utilized, which align the incoming transitions to the clock signal and, consequently, reduce the chances for metastability by several orders of magnitude. Since a synchronizer is essentially a chain of Flip-Flops, the latter has been the main target of metastability research in the past, which was performed either analytically (e.g. by Chaney [145]) or based on measurements and simulations (e.g. by Kacprzak and Albicki [136], Jones, Yang, and Greenstreet [62] or Beer et al. [54]).

In a Flip-Flop two Latches, which either are transparent (propagate the inverted input value) or opaque (cut off the input and store the value), are installed in succession (see Figure 5.2). The shown switches are controlled in an inverting fashion by the clock signal such that one Latch is opaque while the other is transparent and vice versa. Overall the Flip-Flop thus stores the input only at a transition on the clock signal, which is in general very short. Let us consider the case that Latch #1 is transparent and consequently Latch #2 opaque (the situation shown in the figure) for a LO clock value. Switching the clock to HI causes Latch #1 to cut off the input and store the current value, while Latch #2 simply forwards the value. In this setup the input thus gets sampled at a rising clock edge. Note that a falling edge triggered Flip-Flop is achieved by assigning the shown circuit setup to the HI clock phase.

The question to answer is: How can such a device become metastable? Problematic are situations where input and clock change in close proximity, such that V_{in} has an intermediate value as Latch #1 becomes opaque. Using the static transfer function f_s of an Inverter shown in Section 3.3 it is possible to determine voltage values at node A (V_A) and B (V_B) that perfectly recreate each other in a loop, as is shown in Figure 5.3. Besides the obvious stable configurations $(V_A, V_B) = (V_{DD}, \mathtt{GND})$ and $(V_A, V_B) = (\mathtt{GND}, V_{DD})$, a third option $(V_A, V_B) = (V_M^A, V_M^B)$, i.e., the metastable state, which satisfies $f_s^A(V_M^A) = V_M^B$ and $f_s^B(V_M^B) = V_M^A$, is visible.

Figure 5.3 can also be used to estimate the temporal behavior of the Latch. Considering the fact that $V_B = f_s^A(V_A) = f_s^A(f_s^B(V_B))$ and assuming a time delay between each "iteration", it is possible to observe a continuous increase/decrease of V_A respectively decrease/increase on V_B. The circuit thrives to move away from the metastable point whereat the speed of this process depends on the deviation to (V_M^A, V_M^B). Consequently the resolution time t_{res} increases with $V_A \to V_M^A$, whereat for $V_A = V_M^A$ ultimately $t_{res} = \infty$ (metastability cannot be resolved in limited time). Clearly a high amplification

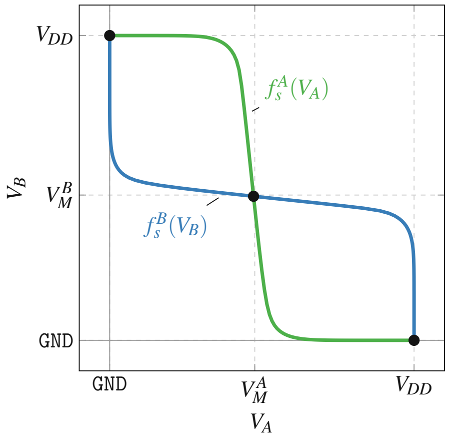


Figure 5.3: Static transfer characteristic f_s of the forward and backward Inverter in Latch #2 of Figure 5.2. The black dots mark the two stable configurations at the outskirts and the metastable one in the middle.

is beneficial for small t_{res} since the voltage gained by each "iteration" is increased.

Note carefully that $V_M^A = f_s^B(f_s^A(V_M^A))$ and thus metastability is always possible, although in general $f_s^A(V_M^A) \neq V_M^A$. To somehow quantify the risk of a unit getting metastable Veendrick [144] developed a metric called Mean Time Between Upsets (MTBU) which states how much time in average passes until a metastable upset is observed. Note that this is a statistical mean value, implying that the chance of observing an upset way before the estimated point in time is not zero! The MTBU can be calculated as

$$\text{MTBU} = \frac{T_{clk}}{\lambda_{dat} \cdot T_0} \cdot e^{\frac{t_{res}}{\tau}}$$

with T_{clk} the time between two consecutive rising clock edges, λ_{dat} the input data rate and t_{res} the available resolution time. The parameters T_0 and τ are technology-dependent and have to be determined for each implementation separately. One key assumption in these calculations are uniformly distributed data transitions times, which is in general not the case. Thus MTBU results require proper interpretation and analysis.

5.2 Analog Metastability Simulations

After having seen how memory elements can become metastable, we investigate in the sequel if the simplified models presented in Section 3.2 are able to predict such a behavior. Note that actually driving a circuit into metastability is very challenging as the circuit shows a highly nonlinear and sensitive behavior in this operation region. For our analyses we will investigate two specific circuits: a Latch and an `OR Loop`.

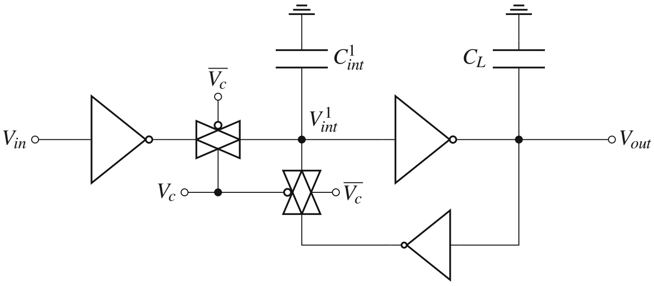


Figure 5.4: Uniform Model implementation of a Latch using Transmission gates.

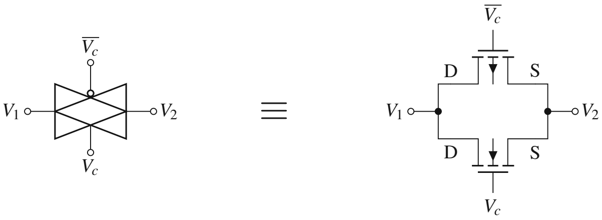


Figure 5.5: Internal structure of the Transmission gate. For $V_c = GND$ both transistors prevent charge transport while for $V_c = V_{DD}$ both fully conduct. The Transmission gate is thus the equivalent to a common switch, which propagates current in both directions.

5.2.1 Latch

Recall that we already investigated the basic component of the Latch, i.e., an Inverter loop, in Section 3.7.4. In the corresponding analysis we showed that the metastable value V_M is approached if the internal nodal voltages are initially equal, however, we neglected back then the resolution time completely. For more realistic results we extend the Inverter loop by an input Inverter and Transmission gates as shown in Figure 5.4. The latter are used to realize the clock-controlled switches in Figure 5.2 and consist of two transistors in parallel, as depicted in Figure 5.5. Note that the Transmission gate is required to enable charge transport equally in both directions. For asymmetric transistors it is thus beneficial that source and drain of n- and pMOS are on the same side such that at least one is fully conducting.

Unfortunately our MACS tool is not yet capable to properly model the transmission gate. The main reason is that the uniform model behaves strangely when V_{Dy} is smaller than zero, causing the model to fail. Although we are confident that the problem can be resolved, a suitable solution seems to be a non-trivial task. Therefore we used a quick, yet accurate, workaround: In a nutshell the Transmission gates serve the purpose of limiting the conductivity and thus the amount of current that can be delivered from the input and feedback Inverter to the capacitance C^1_{int}. We model this by applying a non-constant, sigmoidal shaped multiplicative factor $m(t) \in [0, 1]$ to the current through one inverter and the inverse, i.e., $1 - m(t)$, to the other one. More specifically, recalling

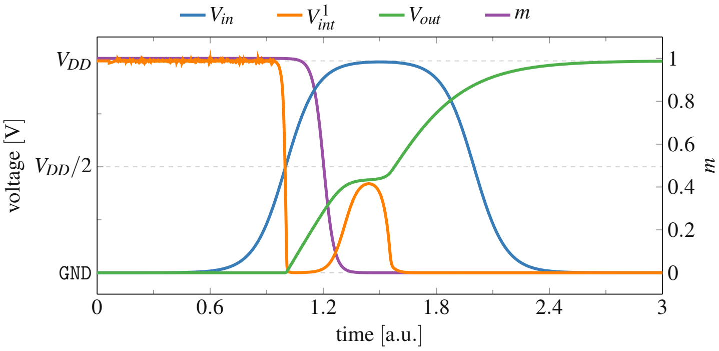


Figure 5.6: Simulation of the Latch in metastability using MACS. By varying the transition time of $m(t)$, which models the Transmission gates, longer metastable states (flat part of V_{out}) can be achieved.

Eq. (3.7) in Section 3.3.3 the following set of equations has to be evaluated:

$$\begin{aligned} C_L \cdot \frac{\mathrm{d}}{\mathrm{d}t} V_{out} &= I_{inv}(V^1_{int}, V_{out}) \\ C^1_{int} \cdot \frac{\mathrm{d}}{\mathrm{d}t} V^1_{int} &= m(t) \cdot I_{inv}(V_{in}, V^1_{int}) + [1 - m(t)] \cdot I_{inv}(V_{out}, V^1_{int})/A \end{aligned}$$

The capacitance C^1_{int} is, thereby, a lot smaller compared to C_L and only serves the purpose of being able to calculate the voltage V_1 properly. For our simulations we choose a ratio of $C_L/C^1_{int} = 200$ and a scaling factor $A = 2$.

Finally we ran simulations of this model with the goal to reach metastability: We applied a sigmoidal shaped pulse to the input and simultaneously switched the Latch to opaque (rising transition on $m(t)$), whereat the start point of the latter was varied in time. Using a systematic approach enabled us to derive the behavior shown in Figure 5.6. Note that very precise control was required here: changing the onset of m by 10^{-4} % of the full range output switching time already determined if the waveform resolved towards HI or LO. This confirms the high sensitivity of circuits in this regime.

5.2.2 OR Loop

The second circuit we are investigating is the OR Loop which was already evaluated using analog and digital simulations in Section 4.6. In contrast to the Latch, where the relative timing of the input transitions influenced the output behavior, this circuit is completely determined by the input pulse width. This circumstance simplifies the analysis and even enables verification. In consequence we evaluated the circuit shown in Figure 5.7 not in MACS but immediately utilized C2E2.

Figure 5.8 shows the achieved results. For a constant trapezoidal input (top) we slightly varied the initial value of V_{out}, leading to significantly different output traces (middle): While some quickly resolve to GND and V_{DD} others stay at an intermediate

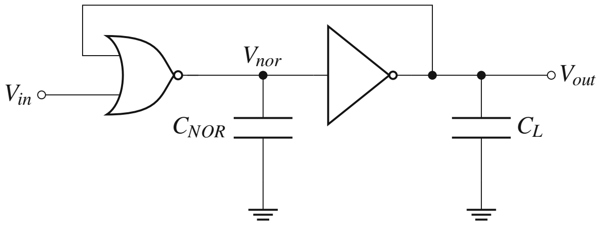


Figure 5.7: Uniform Model of an `OR Loop`*. To reuse available models the OR-gate was split up into a NOR-gate and an Inverter.*

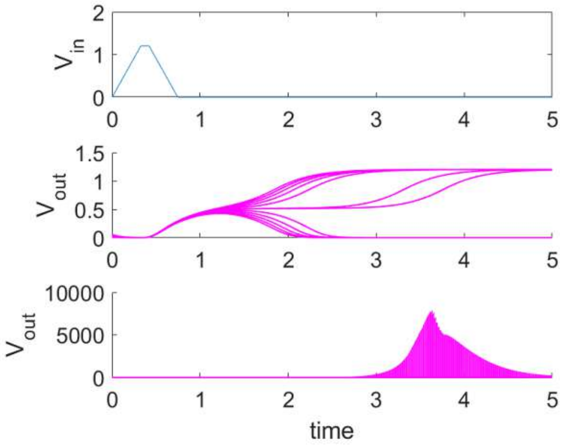


Figure 5.8: Metastability analysis of an `OR Loop` *in C2E2 using technology (T65). For a single input pulse (top) significantly different simulation traces (middle) have been achieved by changing the initial value of V_{out} minorly. The reachtube (bottom), corresponding to the trace sticking longest in metastability, reaches unreasonable high values, which indicates a very high sensitivity.*

value for a very long period, i.e., in a metastable state. This verifies that metastable behavior is covered by simplified transistors models.

Very interesting is the reachtube for the trace staying longest in metastability (shown at the bottom of the figure), which is used internally for verification purposes and depicts all possible values the output may achieve. While the blow up to several thousand Volts is physically unreasonable, it does indicate the very high sensitivity of the underlying system of ODEs in this region: Even the slightest disturbances of the initial state result in grossly deviating trajectories, in particular, in very different metastability resolution times. This is not only in accordance to common knowledge about metastability, but also, to the best of our knowledge, the first reachability analysis of circuits demonstrating metastable behavior.

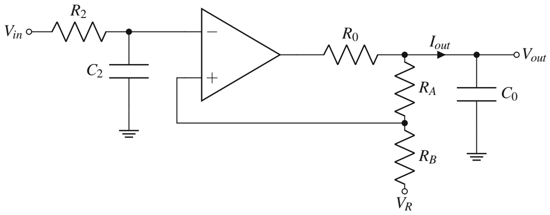


Figure 5.9: Dynamic model of the S/T studied by Marino [146].

5.3 The Metastable Schmitt Trigger

Since metastability cannot be prevented on the transistor level, it has to be handled at the gate level, e.g., by introducing additional components like the earlier presented synchronizer. A different approach is to filter intermediate voltage values by introducing gates that incorporate different thresholds for rising (V_H) and falling (V_L) transitions. For $V_H > V_L$ we end up with a Schmitt Trigger (S/T), which we already introduced in Section 3.5. Note that such a unit has been used, e.g. by Greenstreet [108] and Nyström and Martin [94], to confine a possible metastable upset to a bounded region in the circuit. Nevertheless, it is important that late transitions still cannot be prevented.

In early days the S/T was assumed to be free of metastability, which led to a discussion, e.g. by Wormald [147] or Chaney [145], whether it can be used to develop a synchronizer Flip-Flop that is immune to metastability. Early investigations by Marino [146] revealed that also the S/T itself is prone to metastability. This follows immediately from our analysis presented in Section 5.1, since inside the hysteresis two stable states for a unique input value are possible, which implies the existence of a metastable state.

Despite these early revelations little is still known about the behavior of modern S/T implementations. For this reason, we will present a thorough analysis of the Schmitt Trigger in the succeeding chapters. We start with a revision of the model provided by Marino and then turn to novel methods to characterize an implementation efficiently. Based on the gathered information we evaluate specific circuits statically and dynamically. Finally we investigate the possibility to derive an MTBU estimation also for the S/T.

5.3.1 Metastability Model for the Schmitt Trigger

Marino investigated a simplified and optimal dynamic model for the S/T circuit as shown in Figure 5.9. Based on this representation he derived a phase plane description (V_{out}' in the V_{in}-V_{out} plane) shown in Figure 5.10, with A denoting the gain of the differential amplifier and M being the output saturation voltage (possible output values in $[-M, M]$). The functions γ_1 and γ_3 represent the stable states while γ_2, which connects the former, the metastable ones. Note that all points on γ_1, γ_2 and γ_3 share the property $V_{out}' = 0$, since, at least theoretically, all of them could be preserved for an unlimited amount of time. These results have fundamental implications: In contrast to the Latch, which had a single metastable and two stable states, there are now infinitely many (meta-)stable values ranging from the lower (GND) continuously to the upper (V_{DD}) supply voltage.

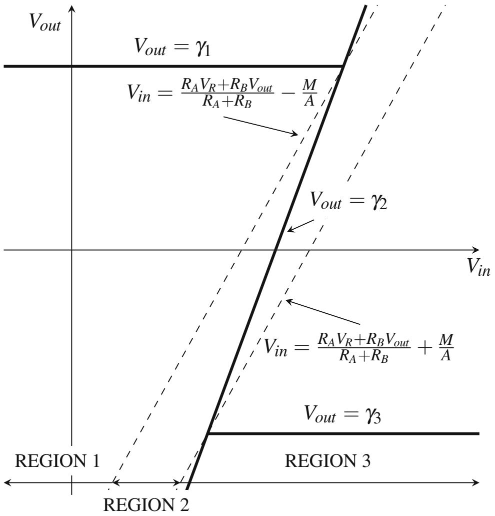

V_{out}
$V_{out} = \gamma_1$
$V_{in} = \frac{R_A V_R + R_B V_{out}}{R_A + R_B} - \frac{M}{A}$
$V_{out} = \gamma_2$
V_{in}
$V_{in} = \frac{R_A V_R + R_B V_{out}}{R_A + R_B} + \frac{M}{A}$
$V_{out} = \gamma_3$
REGION 1
REGION 3
REGION 2

Figure 5.10: Phase plane representation (V'_{out} in V_{in}-V_{out} plane) of the S/T as derived by Marino [146]. The bold black lines (γ_1, γ_2 and γ_3) denote $V'_{out} = 0$.

Additionally the input cannot be disconnected and has to be at an intermediate value to enable metastability, which we actually attributed to combinatorial gates earlier. The S/T, however, differs from the latter by enabling the resolution of metastability for constant V_{in} and can thus be seen as an intermediate step from the Latch to a purely combinatorial gate.

Consequently, varying the input has a big impact on the overall metastable behavior of an S/T. To express this by an analogy, envision the Latch as a long stick that is placed on a table. While the upright position represents the metastable state, lying flat indicates the stable ones. In the case of an S/T we are balancing the stick on our palm, comparable to an inverted pendulum. By proper movement one can actively enhance or even counteract metastability resolution and thus alter the probability for metastability, but also the resolution speed.

Marino was able to divide the output behavior in the phase plane in three different regions: the upper and lower saturation (Regions 1 and 3; approach the final value) and the “linear region” 2 (escape metastability) in between. For these he provided an analytic description of the output derivative, a very important figure of merit as we will see in the sequel, resulting in the following equations (for their detailed derivation please refer to the original publication):

$$\text{Region 1:} \qquad \frac{\mathrm{d}V_{out}}{\mathrm{d}t} = V'_{out} = -\frac{1}{\tau_1}(V_{out} - \gamma_1) \tag{5.1}$$

$$\text{Region 2:} \qquad \frac{\mathrm{d}V_{out}}{\mathrm{d}t} = V'_{out} = +\frac{1}{\tau_2}(V_{out} - \gamma_2) \tag{5.2}$$

$$\text{Region 3:} \qquad \frac{\mathrm{d}V_{out}}{\mathrm{d}t} = V'_{out} = -\frac{1}{\tau_3}(V_{out} - \gamma_3) \tag{5.3}$$

The region boundaries (dashed lines in Figure 5.10) mark the zero crossing point of V''_{out}, i.e., where the absolutely growing slope of the waveform V_{out} leaving the metastable state γ_2, starts to decline again. For each point $(V_{in}, V_{out}) = (V_1, V_2)$ to the left of γ_2 we achieve $V'_{out} > 0$, meaning that the stable state γ_1 is approached. This follows immediately from (5.1) to (5.3), where we derive for Region 1 and $V_{out} < \gamma_1$ an output derivative $V'_{out} > 0$. For the parts of Region 2 to the left of γ_2 we get $V_{out} > \gamma_2$ and thus also a positive output derivative. A similar analysis can be carried out for points to the right of γ_2, with the difference, that in these cases γ_3 is approached.

One might wonder why V'_{out} is interesting for our analysis. First of all it indicates how quickly metastability is left and thus directly impacts the resolution time. It, furthermore, provides a measure how hard it is to drive the S/T into metastability: Only when the input is able to change faster than the output, intermediate values can be hold for longer time periods, meaning that a larger V'_{out} makes metastability overall less likely. Therefore V'_{out} is an intuitive quantity for the susceptibility of an S/T towards metastability. In addition, as we will explore later, (meta-)stable states share the property of having $V'_{out} = 0$, which gives us a simple measure to determine truly metastable points. This will be extensively used during characterization in Section 5.4.

The above insights might suggest that limiting the dynamics of the input signal can prevent the S/T from getting metastable (see also [14]). Unfortunately this is not the case, as has already been shown by Marino. In (5.2) one can see that it is always possible to find a small enough corridor around γ_2 to allow an appropriately controlled V_{in} to reach a metastable point, no matter how restricted its dynamics (V'_{in}) may be. However, it takes an extremely precise control of V_{in} to remain in a sufficiently narrow corridor. So while limiting V'_{in} cannot safely rule out metastability of the S/T, it *does* aid in making metastability less probable.

In the sequel we will investigate how an S/T can be efficiently and accurately characterized. Besides identifying the (meta-)stable states, we present methods that are able to highlight properties in the whole V_{in}-V_{out} plane, such as time constants, resolution time or output derivative.

5.4 Characterizing the Schmitt Trigger

The phase diagram, as proposed by Marino, can be understood as a finger print of a Schmitt Trigger implementation, which helps the designer to understand and optimize the circuit. While the stable states on γ_1 and γ_3 are easily determined, the task to identify γ_2 is much harder. Obviously, efficient methods are required if various circuits shall be analyzed in a systematic fashion. In the sequel we are, thus, proposing multiple approaches to determine the phase diagram in a fast, simple and accurate fashion. First, we relate our results to the simple model of Marino, which enables an analytic evaluation, and later turn to state-of-the-art implementations.

Although calculations, e.g., based on the Uniform Model, are in theory possible, they experience major shortcomings: Primarily the achieved results are rather inaccurate, as can be seen in Figure 5.11 for a modern implementation. Moreover, the determination

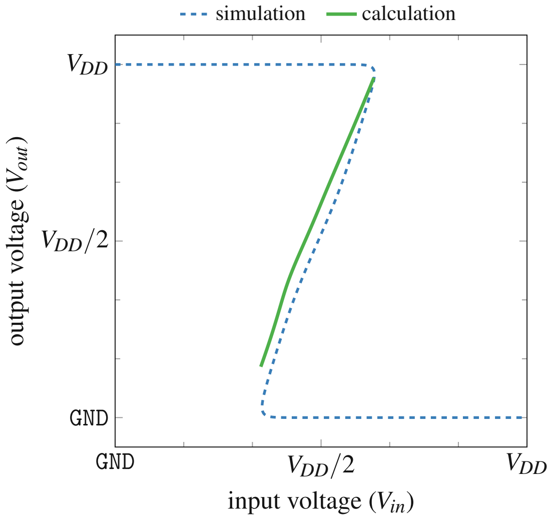


Figure 5.11: Calculation of γ_2 using the Uniform Model. The calculation not only has limited accuracy but also rather long run time ($\approx$ 3.3 s per metastable point).

of just a single metastable point takes several seconds, which is prohibitive, especially when considering the overall accuracy. For this reason we decided to base our analysis on HSPICE simulations using technology (T28). Note that comparable results have been achieved for technology (T65) which shows the general applicability of our approach.

5.4.1 General Characteristics

For the selection of a Schmitt Trigger circuit that is suitable for a given purpose, as well as for potential optimizations, its characteristics must be precisely understood, particularly with respect to metastable behavior. In this section, we present a number of approaches to characterize the overall properties of an S/T implementation, whereas in the subsequent Sections 5.4.2 and 5.4.3 we are going to investigate the metastable states γ_2 and the resolution behavior.

In order to explain and evaluate our approaches, they are first applied to the circuit `opamp` shown in Figure 5.12, which is a slightly modified version of Marino's idealized model implementation (cf. Figure 5.9). This allows a simple accuracy assessment of all methods, as an analytical ground truth can be obtained from theory. Since Marino's parametrization, leading to a close to ideal behavior with a narrow Region 2, is not comparable to real-world circuits, we chose $A = 50$ and $C_L = 10\,\text{pF}$ to obtain a more realistic behavior. Furthermore, the parameters $R_A = 10\,\text{M}\Omega$, $R_B = 4\,\text{M}\Omega$, and $R_0 = 5\,\Omega$ keep the voltage drop across R_0 at a low level. Albeit these changes, the model calculations still apply. For comparison we verified some of our results also for $A = 10\text{k}$, which required higher numeric accuracy (and hence computational effort) but overall still delivers valid numbers. Finally, note that we added an offset of $V_{DD}/2$ at the output of the OpAmp and chose $V_R = V_{DD}/2$ since our tool was designed for the voltage range $[0, V_{DD}]$.

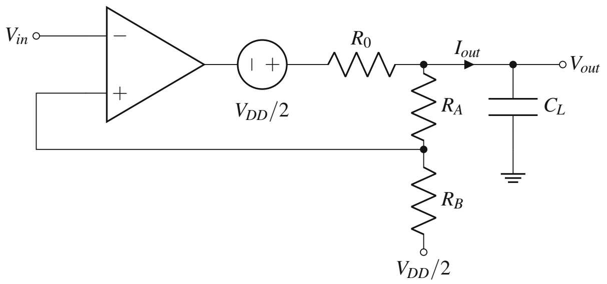


Figure 5.12: Circuit level implementation of `opamp`*.*

Hysteresis (`hyst`)

The stable states on γ_1 and γ_3 can be easily obtained by starting two DC analyses, one sweeping V_{in} from `GND` to V_{DD} and one in the opposite direction. From the results, the threshold voltages V_L and V_H can be identified right away: Exceeding the relevant threshold value on V_{in} leads to a major jump on the corresponding stable value V_{out} (cf. Section 3.5). For the circuit `opamp` we get $V_H = 570\,\text{mV}$ and $V_L = 330\,\text{mV}$, which is in agreement with analytic evaluations.

Exponential Voltage Trajectories

Marino already showed that V_{out}, and thus in consequence also V'_{out}, evolves exponentially over time in all regions, albeit with different time constants: In Regions 1 and 3 the values of τ_1 and τ_3, respectively, are constituted[1] by $R_0 C_L$, while in Region 2 the activity of the non-saturated operational amplifier reduces τ_2 by a factor of $\frac{1}{kA-1}$ with $k = \frac{R_B}{R_A+R_B+R_0}$.

This exponential trend can be verified by depicting V_{out} and V'_{out} in a semi-logarithmic plot over time – which, within one region, yields a perfectly straight line for the `opamp` case. The resolution trajectories of V_{out}, which start within Region 2 and are described in (5.2), are of specific interest in this work. The solution of the differential equation is an exponentially growing function of the shape

$$V_{out} = V_M \pm V_x \exp\left(\frac{t-\widehat{t}}{\tau}\right) \tag{5.4}$$

$$V'_{out} = \pm\frac{1}{\tau} V_x \exp\left(\frac{t-\widehat{t}}{\tau}\right) \quad , \tag{5.5}$$

where $\widehat{t}$ denotes the unknown time shift, $V_x > 0$ the unknown scaling factor of the exponential function and τ a general time constant. Note that although V_x and $\widehat{t}$ could be easily combined into a single parameter, since multiplying an exponential with a constant is equivalent to a time shift, i.e.,

$$V_x \exp\left(\frac{t-\widehat{t}}{\tau}\right) = \exp\left(\frac{\widetilde{t}}{\tau}\right)\exp\left(\frac{t-\widehat{t}}{\tau}\right) = \exp\left(\frac{t-\widehat{t}+\widetilde{t}}{\tau}\right)$$

[1] In detail, the effective R is the parallel resistance of R_0 and $R_A + R_B$.

holds, we kept them separate to better support an intuition. In consequence, the signal shape and especially the derivative remain unchanged in any arbitrary point V_1 on the function. In other words, the trajectory is independent of whether the voltage value V_1 represents the (static) starting point or is (dynamically) “passed by”. This is a specific property of a first-order system that becomes invalid for higher order ones.

For $\lim_{t \to -\infty}$, the time evolution expressed in (5.4) asymptotically approaches the metastable voltage V_M, i.e., the value of interest. More precisely, $V_M(V_{in})$ is a function of the input voltage and essentially corresponds to Marino’s γ_2. To remain consistent, we will use γ_2 to refer to the whole function, while V_M denotes the one metastable voltage for a specific V_{in}.

The resolution behavior in the form of exponentially growing trajectories has major implications. Firstly, it suggests a behavior comparable to the Flip-Flop, with the main difference that the metastable voltage V_M is not unique but varies with V_{in}. Secondly, it gives us the possibility to infer the metastable voltage V_M by recording just a short piece of the resolution trajectory and matching the parameters, as will be leveraged in Sections 5.4.2 and 5.4.2. Thirdly, it is possible to observe a common time constant τ within Region 2. While this is perfectly valid for `opamp`, it will become apparent in real-world circuits (see Section 5.5) that also τ varies with V_{in} – which makes the exponential description less ideal as well.

Voltage Derivative and Current

(Meta)stable states can be uniquely identified by checking for $V'_{out} = 0$; all points on γ_1, γ_2 and γ_3 share this property, cf. (5.1)–(5.3). This trivially follows from the fact that (meta)stability implies a constant V_{out}, i.e., no changes over time. In the circuit `opamp`, V_{out} denotes the voltage drop across the parasitic load capacitance C_L at the output. Using the well known relation between voltage and current at a capacitor, namely $I_C = C\, V'_C$, we can further conclude that the current flowing into C_L, i.e., I_{out} in Figure 5.12, also has to vanish in the (meta)stable state. Overall, we thus obtain a direct proportionality between I_{out} and V'_{out} in the form

$$I_{out} = C_L\, V'_{out} \ . \tag{5.6}$$

For real circuits (cf. Section 5.5) the relationship is more complicated, since these represent higher-order dynamic systems. In the sequel we will, thus, investigate the implication of an active feedback path, with its own dynamic behavior, by means of a simplified model (as shown in Figure 5.13). Note that the circuit is not actively driven from the outside in our analysis but instead we define an initial condition for u_1 and u_2 and observe the temporal evolution.

The signal u_2 is amplified with factor $-A_1$ resulting in signal u_1, which in turn is connected to u_2 via an amplifier with gain $-A_2$ [2]. With a choice of $A_1 > 1$, we can express active feedback. The parasitics at the amplifier outputs are modeled by a single

[2] Note that we used negative gain here, since we derived this model from an inverter loop.

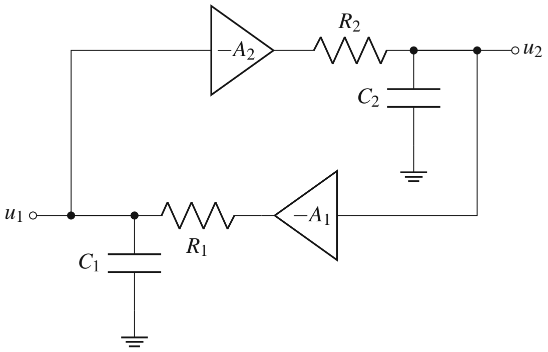


Figure 5.13: Simplified feedback loop model with active feedback path.

RC component, which means we obtain a first-order lowpass. The circuit analysis yields the equations

$$u_1 = -A_1 u_2 - R_1 C_1 \frac{\mathrm{d}u_1}{\mathrm{d}t}$$
$$u_2 = -A_2 u_1 - R_2 C_2 \frac{\mathrm{d}u_2}{\mathrm{d}t}$$

Transformation to the Laplace domain, followed by some reordering leads to

$$U_2(s) = \frac{\tau_2 u_2^0 (1+\tau_1 s) - A_2 \tau_1 u_1^0}{(1+\tau_2 s)(1+\tau_1 s) - A_1 A_2}$$

with the time constants $\tau_i = R_i C_i$ and the initial deviations u_i^0, $i = 1,2$, from the metastable point ($u_1 = u_2 = 0$). The inverse Laplace transformation, using the simplification $\tau_1 = \tau_2 = \tau$, then leads to

$$u_2(t) = \frac{u_2^0 - u_1^0\sqrt{k}}{2} \exp\left(t\frac{\sqrt{A_1A_2}-1}{\tau}\right) + \frac{u_2^0 + u_1^0\sqrt{k}}{2} \exp\left(-t\frac{\sqrt{A_1A_2}+1}{\tau}\right) \tag{5.7}$$

with $k = A_2/A_1$. Since the product A_1A_2 is in general rather large, the second term decays quickly and can be neglected. The remaining dominant term indicates that u_2 grows exponentially over time. Note that u_1 shows a similar behavior, but resolves in the opposite direction.

In a static analysis of the circuit, all transient processes have decayed. This is equivalent to neglecting the dynamics of the feedback path, or by demanding an instantaneous reaction on u_1. This behavior is achieved by assuming $R_1 = 0$ and $C_1 = 0$ which leads to the simplified set of equations

$$u_1 = -A_1 u_2 \tag{5.8a}$$
$$u_2 = -A_2 u_1 - R_2 C_2 \frac{\mathrm{d}u_2}{\mathrm{d}t} \ , \tag{5.8b}$$

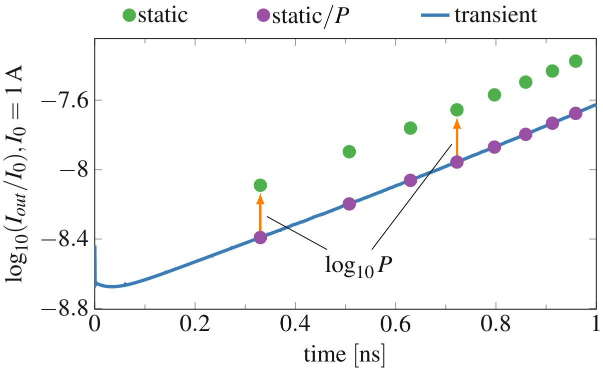


Figure 5.14: Comparison of the output current for transient and static simulations.

and finally the solution

$$u_2(t) = u_2^0 \exp\left(t\frac{A_1A_2 - 1}{\tau_2}\right) \ . \tag{5.9}$$

In this case we again obtain an exponential resolution trace, however, now with a much higher multiplicative factor in the exponent. For this circuit, the effective resolution time constant can be computed as the RC constant divided by the loop gain (A_1A_2) minus one. A similar relationship has been derived for opamp, where the resolution time constant is $\tau = \frac{RC}{kA}$. Recall that kA denotes the loop gain, with A being the gain of the forward path and k (< 1) that of the (passive) backward path. This similarity to the system (5.8) is no surprise, as setting R_1 and C_1 to zero has turned our second order system into a first order one.

For $\tau_2 = \tau$ the ratio P of the resolution time constants for the static and dynamic case from (5.7) and (5.9), respectively, evaluates to

$$P = \frac{(A_1A_2 - 1)/\tau}{(\sqrt{A_1A_2} - 1)/\tau} = \frac{(\sqrt{A_1A_2} - 1)(\sqrt{A_1A_2} + 1)}{\sqrt{A_1A_2} - 1} = \sqrt{A_1A_2} + 1 \ .$$

This shows that static analyses are indeed able to predict the transient resolution dynamics, if the feedback based coefficient P is carefully considered. Although determining P is a simple task in our small example, for real circuits this can be quite challenging, also because the value P depends on the operating point.

To provide evidence for the validity of the ratio P, we performed a comparison between static and transient simulations (see Figure 5.14). Therein, whenever the transient output trajectory reaches a grid point of map(see next section), the corresponding static current is indicated by a green dot. It can be seen clearly that (i) I_{out}, and thus V_{out}, is proportional to $\exp(t/\tau)$ and that (ii) the static analysis overestimates the current by a constant multiplicative factor P, which appears as an additive shift on the logarithmic axis in Figure 5.14. To obtain correct results for V'_{out}, this discrepancy has to be compensated by introducing the modified load capacitance

$$C_L^* = C_L\, P \ . \tag{5.10}$$

This correction has been used throughout Section 5.5.

Phase Diagram (`map`)

Without prior knowledge, a very pragmatic approach for obtaining the phase diagram is to cover the V_{in}-V_{out} plane with a regular grid and determine V'_{out} for each grid point. Albeit this initially appears quite unfocused and laborious, the resulting phase diagram, also denoted as `map`, not only allows to interpolate γ_2, but also provides a good intuition of the overall behavior, especially while resolving metastability.

Instead of extracting V'_{out} from transient simulations we opted to use I_{out} and the proportionality (5.6) between I_{out} and V'_{out}. To derive values for the current, an additional constant voltage source is attached in parallel to the output capacitance C_L in Figure 5.12 and its value V_{out} is chosen according to the investigated grid point. In the steady state, this forces the current to/from the capacitance to vanish since $V'_{out} = 0$. I_{out} is flowing through the voltage source, where it can easily be recorded. The respective values can then be used to indicate how fast C_L would be (dis)charged (resulting in a corresponding V'_{out}), once the voltage source is disconnected.

This approach is considerably faster and much simpler to execute, while providing the same level of accuracy for the metastable voltage[3]. The required simulations are run using built-in commands from HSPICE, as detailed in Listing 5.1: V_{in} is swept from `GND` (0) to V_{DD} (*supp*) in steps (*width*) corresponding to the grid spacing. For each value V_{in} of the input voltage, V_{out} is varied in the same fashion where the number of steps (*count*) can be different. The current through the voltage source V_{out} is determined in the second code line. For all the analysis presented in this thesis we used $width = 900$ ($\Delta V_{in} = 1\,\text{mV}$) and $count = 9000$ ($\Delta V_{out} = 100\,\mu\text{V}$).

```
.DC vIn 0 supp width SWEEP vOut LIN count 0 supp    $$ run DC analysis
.PROBE DC I(vOut)                                   $$ record current at output
```

Listing 5.1: Deriving I_{out} in the V_{in}-V_{out} plane in HSPICE

Most certainly we won't be fortunate enough to exactly hit $I_{out} = 0$ (or equivalently $V'_{out} = 0$) this way. Nevertheless, γ_2 can already be confined between two adjacent grid points with changing sign. In a first step, contour plots can be used to show (interpolated) lines for constant output current, especially $I_{out} = 0$ corresponding to the line of (meta)stable states. The resulting `map` provides the possibility to quickly identify the most important parameters of an S/T implementation (e.g. threshold voltages, (meta)stable values, gradients of V'_{out}) and thus to coarsely predict the overall behavior.

The obtained results for the circuit `opamp` are shown in Figure 5.15 and illustrate the very good agreement with the analytic considerations. In particular, horizontal interpolated contour lines (with linear spacing) in Regions 1 and 3, as implied by (5.1) and (5.3), are visible. Within a corridor of width $\frac{2M}{A} = \frac{2\cdot 450\,\text{mV}}{50} = 18\,\text{mV}$ around γ_2, whereat we used the definitions of A from Section 5.4.1 and $M = V_{DD}/2$, the contour lines run parallel to γ_2, in accordance with (5.2).

[3] Recall that $V'_{out} = 0$ directly translates to $I_{out} = 0$. Thus considering C_L, including the uncertainties and non-linearities possibly associated with it, is not required.

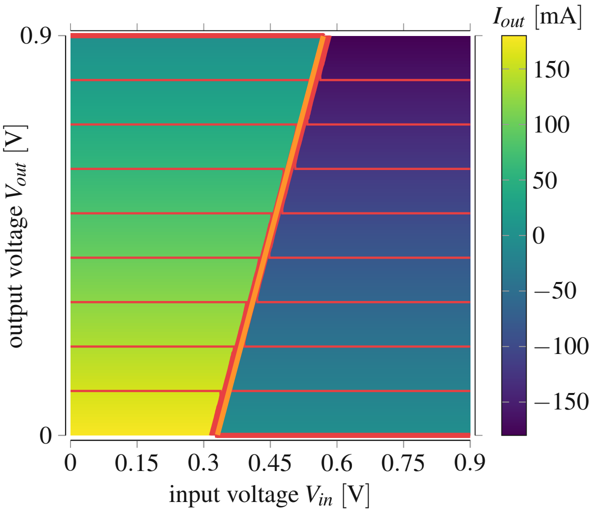


Figure 5.15: `map` *results for* `opamp` *with a grid size of* $\Delta V_{in} = 1\,\mathrm{mV}$ *and* $\Delta V_{out} = 100\,\mu\mathrm{V}$*. Clearly visible are the horizontal contour lines.*

5.4.2 Precisely Identifying the Metastable States

The approaches presented so far allow a rough overview of a given S/T implementation, whereat the metastable points constituting γ_2 are obtained by interpolation. In the following, we thus present methods that allow a more precise description and apply them to `opamp` for illustration and validation.

Transient Estimation (`expTran`)

In Section 5.4.1 we expressed the metastable voltage in (5.4). In fact by combining (5.4) and (5.5), V_M can be determined from a pair of corresponding values V_{out} and V'_{out} as

$$V_M = V_{out} - \tau\, V'_{out}\ . \tag{5.11}$$

Note that the unknown V_x and $\widehat{t}$ are eliminated. Consequently, τ is the only remaining parameter, which can be determined by applying the natural logarithm to (5.5), yielding

$$\ln(|V'_{out}|) = \ln\left[\frac{1}{\tau}\,V_x\,\exp\left(-\frac{\widehat{t}}{\tau}\right)\right] + \frac{t}{\tau} = K + \frac{t}{\tau}\ . \tag{5.12}$$

The linear correspondence between t and $\ln(|V'_{out}|)$ expressed in (5.12) allows to extract τ as the inverse of the slope between any two values of V'_{out} in a semi-logarithmic plot.

Overall we thus obtain the following strategy: From a transient simulation starting in an arbitrary point (V^*_{in}, V^*_{out}), we use the time and value differences among simulated values of V'_{out} to determine τ according to (5.12), while a single consistent pair of $(\widehat{V}_{out}, \widehat{V}'_{out})$ suffices to finally obtain V_M from (5.11). As outlined in Section 5.4.1, the time constant of the exponential function changes as soon as the operational amplifier

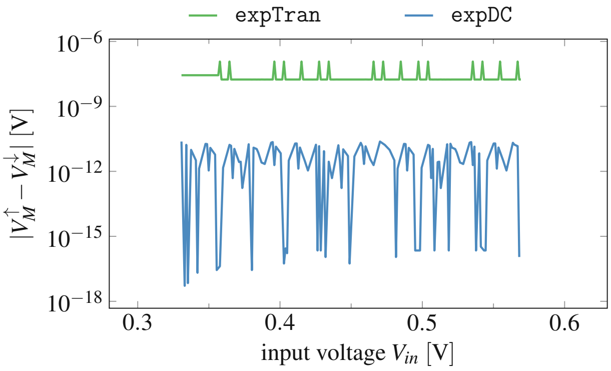


Figure 5.16: Absolute deviation between V_M predictions for resolution towards V_{DD} ($V_M^{\uparrow}$) and GND ($V_M^{\downarrow}$).

saturates, i.e., when the trajectory leaves Region 2 and enters Region 1 or 3. Since we are interested in the former, we must take care to only use values from within Region 2 for fitting the parameters.

Note that `expTran` can be run twice for each input voltage, since resolution traces to GND and V_{DD} are possible. Ideally both would render the same results. In reality, however, we get slightly different values $V_M^{\downarrow}$ and $V_M^{\uparrow}$. These can be retraced to numerical issues due to the limited accuracy of the simulations. Figure 5.16 shows the difference. To improve accuracy, we thus determine the intersection point of both linear functions resulting from (5.11), i.e., one for each direction.

We experienced that the resolution time constant τ can be comparably accurate (relative error $\approx 10^{-7}$) determined from HSPICE simulations, however, results for V'_{out} show a relative error of up to $1\,\%$. Fortunately, this mismatch can be largely mitigated by a careful choice of the consistent pair $(\widehat{V}_{out}, \widehat{V}'_{out})$: Equation (5.11) can be interpreted as correcting an inaccurate initial value $\widehat{V}_{out}$ by the term $\tau\,\widehat{V}'_{out}$. With a choice of $\widehat{V}_{out}$ close to V_M (in our case $|\widehat{V}_{out} - V_M| \approx 10^{-5}\,$V) that correction term becomes small. Considering the $1\,\%$ relative error of V'_{out}, a deviation less than $10^{-7}\,$V can be achieved, which is still quite big compared to other methods (see Figure 5.17). Increasing the simulator accuracy improves the results, but also leads to a prolonged computation time.

Static Estimation (expDC)

Equation (5.2) relates the output voltage V_{out} and its derivative V'_{out} to the metastable voltage γ_2. Since V'_{out} and I_{out} are directly related by (5.6), also the latter can be used for a good estimate of V_M. In fact, we can rewrite (5.2) as

$$V'_{out} = \frac{I_{out}}{C_L} = \frac{1}{\tau}\left(V_{out} - V_M\right) . \qquad (5.13)$$

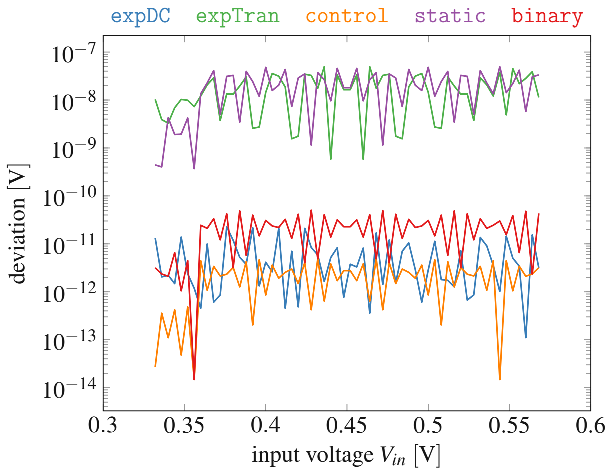


Figure 5.17: Absolute deviation between V_M predicted by various approaches compared to analytic calculations for `opamp`*.*

Note that this is a linear function of the form $I_{out} = kV_{out} + d$, where the desired metastable value V_M can be expressed as

$$V_M = -\frac{\tau\, d}{C_L} = -\frac{d}{k} \; . \tag{5.14}$$

The values of k and d can be easily obtained by extracting several values of $I_{out}(V_{out})$ (e.g. by running `map`) and fitting a linear function. In this fashion, the metastable voltage V_M is derived extremely fast. Although this process has to be repeated for numerous choices of V_{in} to obtain γ_2, its execution time is still one of the lowest.

Similar to `expTran`, the method `expDC` can be run with traces towards `GND` as well as towards V_{DD}. As can be seen in Figure 5.16, the difference between the values $V_M^{\uparrow}$ and $V_M^{\downarrow}$ is much lower than for `expTran`. This improved accuracy seems to originate from avoiding the use of V'_{out}. For better results we again determine the intersection of the linear functions resulting from (5.14) for both directions.

Binary Search (`binary`)

A more pragmatic approach called `binary` sweeps V_{in} from V_L to V_H (cf. Section 3.5), and for each value V_{in} a binary search is performed to find a value of V_{out} such that I_{out} becomes zero. This bisection principle is also applied in [74].

To our advantage HSPICE has a built-in mechanism called *Bisection* to run a binary search. The corresponding code is shown in Listing 5.2. The first line states that we want to bisect, and at most 40 steps shall be carried out. Note that this narrows down the initial interval by a factor of 2^{40}. Most of the time, the algorithm finishes earlier, as the demanded accuracy, which is specified by the parameter *RELIN*, is reached first. In our case we demand that the difference between two consecutive values must be smaller than

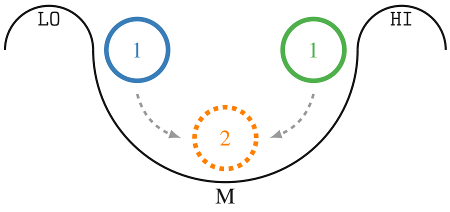


Figure 5.18: Flipping (meta-)stable states by current overcompensation.

0.01%. As can be verified in Figure 5.17, the residual error in V_M w.r.t. the analytical results (5.2) is indeed extremely small.

```
.MODEL optMod1 OPT METHOD=BISECTION RELIN=1e−4 ITROPT=40  $$ stop conditions
.PARAM outVal=optFunc1(vdd/2, vout_VL, vout_VH)            $$ search interval
.DC vIn inVal inVal 1 SWEEP OPTIMIZE=optFunc1              $$ run the Bisection
  + RESULTS=optMeasure MODEL=optMod1
```

Listing 5.2: Bisection in HSPICE

The second line sets the parameter *outVal*, which determines the initial value of the output voltage V_{out} ($V_{DD}/2$) and its sweep range. To ensure that the stable states are not contained in the latter, we set its boundaries to the last stable output values on γ_1 and γ_3. In the case of `opamp`, this corresponds to $\gamma_3(V_L)$ and $\gamma_1(V_H)$, respectively. The third line finally launches the *DC* analysis for the input voltage $V_{in} = inVal$ which means that this analysis has to be executed for each value of V_{in} separately.

Closed-loop Control (`control`)

The fact that all trajectories strive to leave the unstable states makes it so hard to achieve accurate values for V_M. In the sequel we will, thus, investigate a method that uses a simple proportional controller to overcompensate the current flowing into the load capacitance. This effectively converts the unstable equilibria into stable ones while the previously stable ones are now metastable (see Figure 5.18). In this fashion entering and observing metastability becomes a trivial task since (almost) independent of the initial condition the circuit on its own navigates towards $V_{out} = V_M$.

By overcompensating the current into the load capacitance, a situation that would cause V_{out} to increase leads actually to a reduction and vice versa. The setup for `opamp` is shown in Figure 5.19. Due to the fact that our goal is $I_{out} = 0$ and thus also $I_L = 0$ a proportional controller $I_L = K \cdot I_{out}$ is sufficient. This has also been verified by extensive control theory considerations.

One major challenge is to properly determine K, since picking excessive values cause oscillations and lead to unreasonable results. Clearly choosing $K = 1 + \varepsilon$, with $\varepsilon > 0$ arbitrarily small, is valid and will eventually lead to the correct results, but also increases the simulation time unnecessarily. Thus we were searching for ways to derive close to optimal results. Finally the poles and zeros of the transfer function turned out to be a good starting point. The respective information is gathered in HSPICE by the code

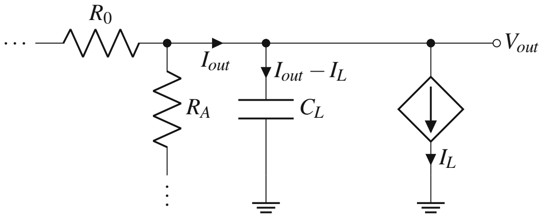


Figure 5.19: Circuit setup to characterize the linearized system dynamics and to overcompensate the current flow into the load capacitance.

shown in Listing 5.3. Over the optimal voltage source *Vmeas*, which is used to extract I_{out}, the capacitance C_L (line 2) is connected to the circuit. At the same node the *AC* current source *IL* (line 3) introduces the current, which is used as reference in line 5 to calculate gain and phase of I_{out}. Finally in line 6, the poles and zeros are evaluated and exported.

```
1  vMeas out outC DC 0 AC 0 0        $$ voltage source to measure current
2  cL outC 0 C_L                     $$ output capacitance
3  iL outC 0 DC 0 AC 1 0             $$ introduce reference current
4  .AC DEC 10 1 10000G               $$ run AC analysis from 1 Hz to 10 THz
5  .PROBE AC IDB(vMeas) IP(vMeas)    $$ record gain and phase
6  .PZ I(vMeas) iL                   $$ export pole-zero statistic
```

Listing 5.3: `control` *analysis in HSPICE*

To finally achieve V_M transient simulations (see Listing 5.4) are utilized. The controller is implemented as a current control current source (CCCS) (line 3) that multiplies the current through *Vmeas* (line 1), i.e., the one into the artificial capacitance *COUT* (line 2), by the conservative value $K = 2$. V_M if finally extracted as the value of V_{out} after 1 µs (line 4). Unlike for `opamp`, we will use $K(V_{in})$ for real world implementations, i.e., a separate amplification for each input value.

```
1  vMeas 5 out DC 0                           $$ voltage source to measure current
2  cOut out 0 10p                             $$ output capacitance
3  fP out 0 vMeas K                           $$ current source for overcompensation
4  .MEAS TRAN finalVal FIND V(out) AT=1us     $$ determine final value of Vout
5  .TRAN 1ns 1us                              $$ run transient analysis
```

Listing 5.4: Deriving V_M for `control` *in HSPICE*

The achieved results for `opamp` are shown in Figure 5.17 (`control`). We observe an absolute error of ≈ 1 pV compared to the analytic calculations, which is among the best. Note that increasing the simulator time steps allowed us to increase the simulation time while maintaining a constant computation time, which in turn leads to more accurate results.

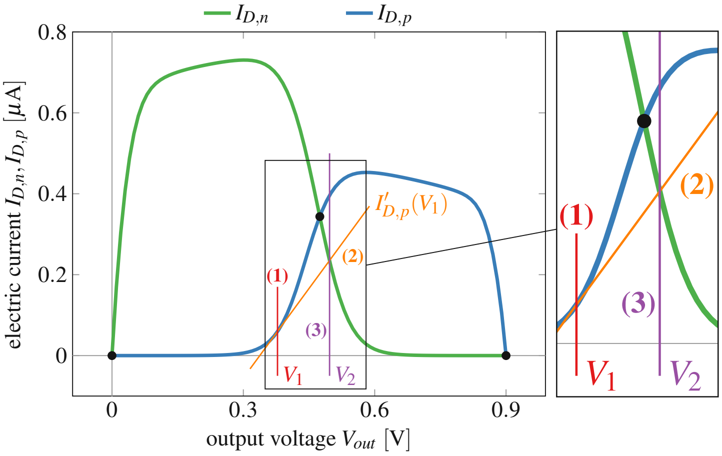


Figure 5.20: Application of the Newton-Raphson algorithm to find a statically stable V_{out} for fixed V_{in} (black dots).

DC Analysis (`static`)

In the course of our research we discovered that plain *DC* simulations on the unmodified implementation are already sufficient to determine the metastable values γ_2. We backtracked this feature to the operation principle of the *Newton-Raphson* algorithm that HSPICE utilizes to determine *DC* operating points in general [28]. To arrive at a stable operating point, the algorithm thrives to achieve $I_{out} = 0$ which is, as presented, a property of metastable states as well.

Figure 5.20 depicts a showcase execution of the algorithm, where we assume a constant V_{in} and try to find a suitable V_{out}. Before the search starts, HSPICE sweeps V_{out} and records the current through the n- ($I_{D,n}$) and p-stack ($I_{D,p}$) of the gate driving the output. As expected the traces cross three times, i.e., for three values of V_{out} the currents exactly compensate, resulting in $I_{out} = 0$. While the outermost intersections mark the stable states of the S/T, the inner one represents the metastable state. The algorithm is started by an initial guess $V_{out} = V_1$ that can be provided by the user. The subsequent steps are (i) to determine the derivative of $I_{D,p}(V_1)$, (ii) find the crossing point of the first-order approximation of $I_{D,p}(V_1)$ with $I_{D,n}$ at $V_{out} = V_2$ and finally (iii) restart the procedure with $V_{out} = V_2$, i.e., the value at the crossing point. The iteration stops when the voltage difference ΔV between two succeeding steps drops below a user-defined value.

Naturally, the initial guess determines which of the three crossings is approached. Thus, by starting close to V_M we can assure to determine the metastable state. Since a deviation of up to $V_{DD}/4$, i.e., several tens of millivolts, could be tolerated in our simulations, connecting γ_1 and γ_3 by a straight line already provides suitable values.

The respective HSPICE code is shown in Listing 5.5. After setting the initial output value in line 2, the *DC* analysis is started in line 3 in the range between $lowVal$ and $highVal$ with a step width of $stepWidth$. Since γ_2 is continuous, using the value found for the previous V_{in} as an initial guess (as it is done in HSPICE) is already very close, leading to a fast convergence and therefore low computing time. For the circuit `opamp` (see Figure 5.17) the achieved accuracy is roughly 10^{-8} V. This rather high value can be retraced to the limited number of decimal places in the output data format. If all other approaches are equally limited, their results are comparable.

```
.PROBE DC V(out)                          $$ record output voltage
.NODESET out=outVal                       $$ initialize output node
.DC vIn lowVal highVal stepWidth          $$ run DC analysis
```

Listing 5.5: Executing `static` in HSPICE

5.4.3 Dynamic Metastability Behavior

Complementary to determining the static metastability values, we also want to efficiently and accurately determine dynamic properties of an S/T implementation. Specifically, we are interested in the resolution time t_{res} and the corresponding time constant τ.

Resolution Time Constant

The resolution time constant τ characterizes the exponential growth of a waveform resolving metastability and is thus a very important parameter. Implicitly, we have already utilized it in some of the methods presented in Section 5.4.2. The questions we address in the sequel are (i) whether the derived results are suitable to predict τ and (ii) if there exist other, overall simpler, methods to achieve this goal.

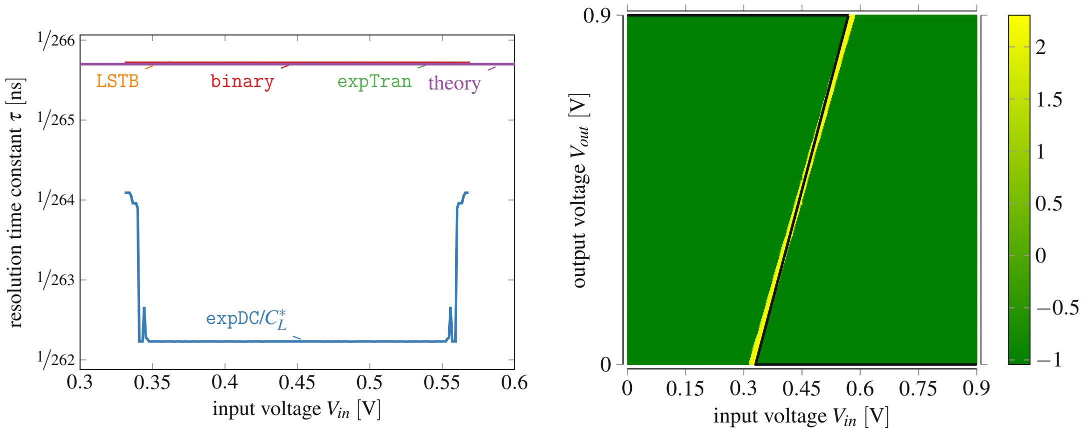
Method `expTran` The straightforward approach, i.e., starting a transient simulation near the predicted metastable value V_M and fitting the resulting analog waveform, was implemented with `expTran`. There, the resolution time constant τ was only calculated as a by-product since it was required in (5.11) to calculate the metastable voltage V_M.

For `opamp`, all points within one region share the same τ. It thus suffices to pick any segment of the resolution trajectory that does not cross region boundaries. Hence, any starting point close enough to γ_2 is suitable to compute the resolution time constant. In fact, the grid points obtained by `map` are already sufficient for this purpose. Figure 5.21a reveals the perfect matches of τ compared to the analytic computation.

Unfortunately, state-of-the-art circuits, which will be discussed in Section 5.5, show slightly non-exponential resolution waveforms and thus variations in the resolution time constant within one Region. For comparison, we thus determined τ also in deep metastability by starting transient simulations in the metastable points delivered by `binary`. For `opamp` the differences are negligible, as can be seen in Figure 5.21a (`binary` vs. `expTran`).

Method `expDC` τ is also computed as a by-product in `expDC`. In (5.13), the slope k of $I_{out}(V_{out})$ is

$$k = \frac{C_L}{\tau} \ . \tag{5.15}$$



(a) resolution time constant τ

(b) $\mathrm{sgn}(x)\log_{10}(|x/A|+1)$ *with* $x = \frac{V''_{out}(t)}{V'_{out}(t)}$

Figure 5.21: Results for opamp showing (a) the resolution time constant τ along γ_2 and (b) the output behavior in the whole V_{in}-V_{out} plane.

This relation allows a simple and accurate determination of τ, given that the output capacitance C_L is precisely known. Unfortunately, this is not the case for state-of-the-art circuits: In advanced CMOS technologies, C_L is constituted by gate capacitances, which change with bias and are subject to significant tolerances and noise. This makes it necessary to determine appropriate values of C_L through transient simulations, giving rise for minor numerical inaccuracies. Thus, for opamp our results for the resolution time constant were already slightly off the theoretical values (see Figure 5.21a; 1/262.2 vs. 1/265.7). Furthermore, recall that the direct correspondence between I_{out} and V'_{out} was derived for a capacitor in Section 5.4.1, so the relationship must be re-evaluated for systems with a dynamic order higher than 1.

Method PZ An alternative approach, referred to as PZ (Pole-Zero), determines τ directly in the frequency domain, without resorting to *DC* or transient simulations. It is rooted in the frequency domain and utilizes the transfer function $G(s)$ which collapses for opamp to

$$G(s) = \frac{1}{\tau s - 1}.$$

It can be seen clearly that the time constant is fully specified by the pole [91], whose value $\tau = 3.763\,\mathrm{ps}$ for the circuit opamp is in perfect agreement with theory. In HSPICE we used the command *.PZ* to automatically derive a list of all poles and zeros.

As we will show in Section 5.5, state-of-the-art circuits in general have multiple poles. Basically this makes their characterization using a (single) resolution time constant, as it is usually done in the context of metastability, questionable – since that implies a first-order dynamic behavior. Nevertheless, only considering the pole with the smallest positive real part, thus approximating the higher-order system with a first-order one, still delivers remarkably accurate results.

Application to Circuit `opamp`

To obtain a general understanding of the resolution behavior, we defined a regular grid in the plane, started a transient simulation with V_{out} close to the metastable value γ_2 and then determined for each grid point the respective slope of V'_{out} on a logarithmic scale, i.e., $x = \ln'(V'_{out}) = V''_{out}/V'_{out}$. Note that for exponential trajectories this results in $x = 1/\tau$. For realistic implementations we observe, however, variations of x, whereat very large positive and negative values in the V_{in}-V_{out} plane are encountered. In order to (i) scale all values by a factor A[4] and (ii) to remove non-interesting regions around the value 0 while preserving 0 itself we show the function $f(x) = \mathrm{sgn}(x)\log_{10}(|x/A| + 1)$. Property (ii) is of specific interest as the value zero marks the boundary between resolving out of metastability and towards the final value, i.e., between Region 2 and 1 respectively 3. The results for `opamp` shown in Figure 5.21b are in perfect agreement to the analytic calculations.

Resolution time

The reasons to investigate the resolution time t_{res} are manifold: When analyzing metastability, especially in synchronous designs, the time to reach uniquely identifiable states is crucial. In a flip-flop, t_{res} merely depends on how deep the circuit is in the metastable state, i.e., how close to the real metastable point the resolution starts. In contrast, the situation is much more intricate for the S/T, since further parameters become relevant (even if we assume constant V_{in}). Firstly, the resolution in an S/T can start in any point on (or close to) γ_2. Depending on the actual choice, the circuit needs to overcome a specific voltage difference to reach the closest digitization threshold value V_{dig}^{HI} or V_{dig}^{LO} beyond which a clear logic `HI` or `LO` level, respectively, is detected[5]. Secondly, as we will see later in Section 5.5, the resolution time constant τ varies over the phase plane and, thus, the dynamics of the resolution process change depending on where the resolution starts. Combining these aspects, it may happen that a given resolution trajectory that overcomes only a small voltage difference but with a large time constant exhibits a longer resolution time than another trajectory that crosses a larger voltage difference with a small τ. This potentially counter-intuitive behavior is further investigated by computing the resolution time t_{res}.

In detail, t_{res} expresses the time it takes the output voltage V_{out} to reach the digitization threshold. It consists of t_{res}^S, which denotes the time spent in Region 1 and 3 before reaching V_{dig}^{HI} or V_{dig}^{LO}, respectively, potentially extended by t_{res}^M, describing the time for moving away from the metastable state inside Region 2. Using (5.3), we can formulate the output trajectory towards `GND` starting from an arbitrary value V_s within

[4] For our analyses we chose $A = x_{max}/200$ with x_{max} being the maximum value in the plane.

[5] Please note that V_{dig}^{HI} or V_{dig}^{LO} are thresholds at the *output*, while V_L and V_H refer to the *input*.

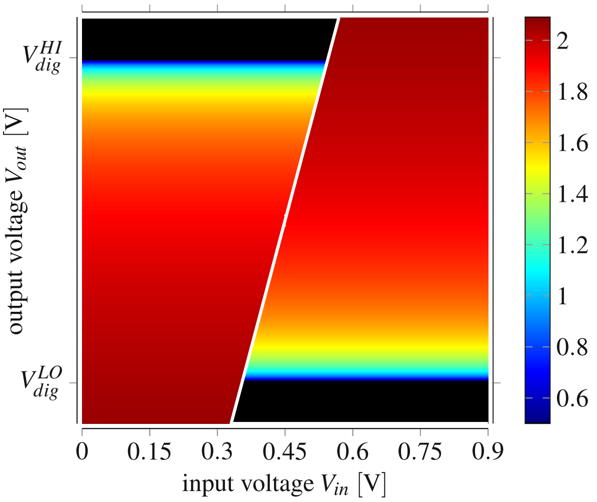


Figure 5.22: Resolution time $\log_{10}(t_{res}/t_0), t_0 = 1\,\text{ps}$ *for* `opamp` *in the* $V_{in} - V_{out}$ *plane.*

Region 3 as $V_{out} = V_s \exp(-t/\tau_3)$. Reordering leads to a resolution time t_{res}^S for reaching V_{dig}^{LO} of

$$t_{res}^S = -\tau_3 \ln\left(\frac{V_{dig}^{LO}}{V_s}\right) .$$

Analogously, for resolution towards V_{DD} (Region 1)

$$t_{res}^S = -\tau_1 \ln\left(\frac{V_{DD} - V_{dig}^{HI}}{V_{DD} - V_s}\right)$$

is obtained. For V_s within Region 2, t_{res}^S is constant and denotes the time span from reaching the boundary value between Region 2 and either 1 or 3, described by V_w, and the associated digitization threshold V_{dig}^{HI} or V_{dig}^{LO} respectively. The additional time it takes the circuit to move from V_s to V_w is denoted by t_{res}^M and can be derived by solving (5.2) as

$$t_{res}^M = \tau_2 \ln\left(\frac{V_w - V_M}{V_s - V_M}\right) .$$

Based on the simulation results from Section 5.4.3, we computed for each grid point the time until the corresponding digitization threshold $V_{dig}^{HI} = 0.9\,V_{DD}$ or $V_{dig}^{LO} = 0.1\,V_{DD}$ is reached. For the example of `opamp`, the combined resolution time $t_{res} = t_{res}^M + t_{res}^S$ over the whole V_{in}-V_{out} plane is depicted in Figure 5.22. Although the used grid definitely hits the rather narrow Region 2, no significant increase in t_{res} can be observed there. The simple explanation is that close to the border, the contribution of t_{res}^M to t_{res} is small. Starting halfway between γ_2 and the border to Region 1 or 3 results in $t_{res}^M = \tau_2 \ln(2)$ which evaluates in our case to $\approx 2.608\,\text{ps}$ – much less than t_{res}^S. To pronounce the discontinuity at V_M (since $t_{res}^M \to \infty$ for $V_s \to V_M$), we plotted the metastable values γ_2 in white.

5.5 Evaluating Schmitt Trigger Implementations

Provided that an appropriate HSPICE description of the circuit is available, the complete characterization process can be carried out without human interaction. For this reason we implemented all the approaches presented so far in the tool *MEAT* which is publicly available[6]. In this section, we present the results of simulations that we performed with this tool using technology (T28) with the following aims:

- We evaluate and compare the presented methods in a practical application. To this end, we apply all characterization methods for characterizing to three different implementations of S/Ts, as other circuits in literature are heavily based on these: a) the standard 6T implementation (`std`) b) an inverter loop (`loop`) [138] and c) an adjustable hysteresis type (`adjust`) [126]. For each circuit we determined the (meta-)stable states for 900 equally spaced values of V_{in}.
- We investigate how much the behaviors differ among them and also from theoretical results [146]. Our circuits are analyzed as pre-layout circuits, i.e., without parasitics, since we investigate integrated components here and thus (i) expect them to be very small (compared to the gate capacitances) and (ii) either way would get heavily layout-dependent results otherwise.

As we do not have a precise theoretical model available that would provide a ground truth for these implementations (like we had it for `opamp`), we need a different approach to verify the accuracy of the computed metastable values V_M^c. To this end, we start a transient simulation in each of them and then calculate the output deviation $M = |V_{out}(t_0) - V_{out}(0)|$ at a fixed time $t_0 > 0$. Due to the strictly monotonic nature of the resolving trajectories, a higher M corresponds directly to a larger initial inaccuracy ϵ, i.e., $|V_M^c - V_M|$. Considering the exponential nature of the resolving waveform as shown in Section 5.4.2 and using the resolution time constant τ (cf. Section 5.4.3) allows us to compute ϵ as

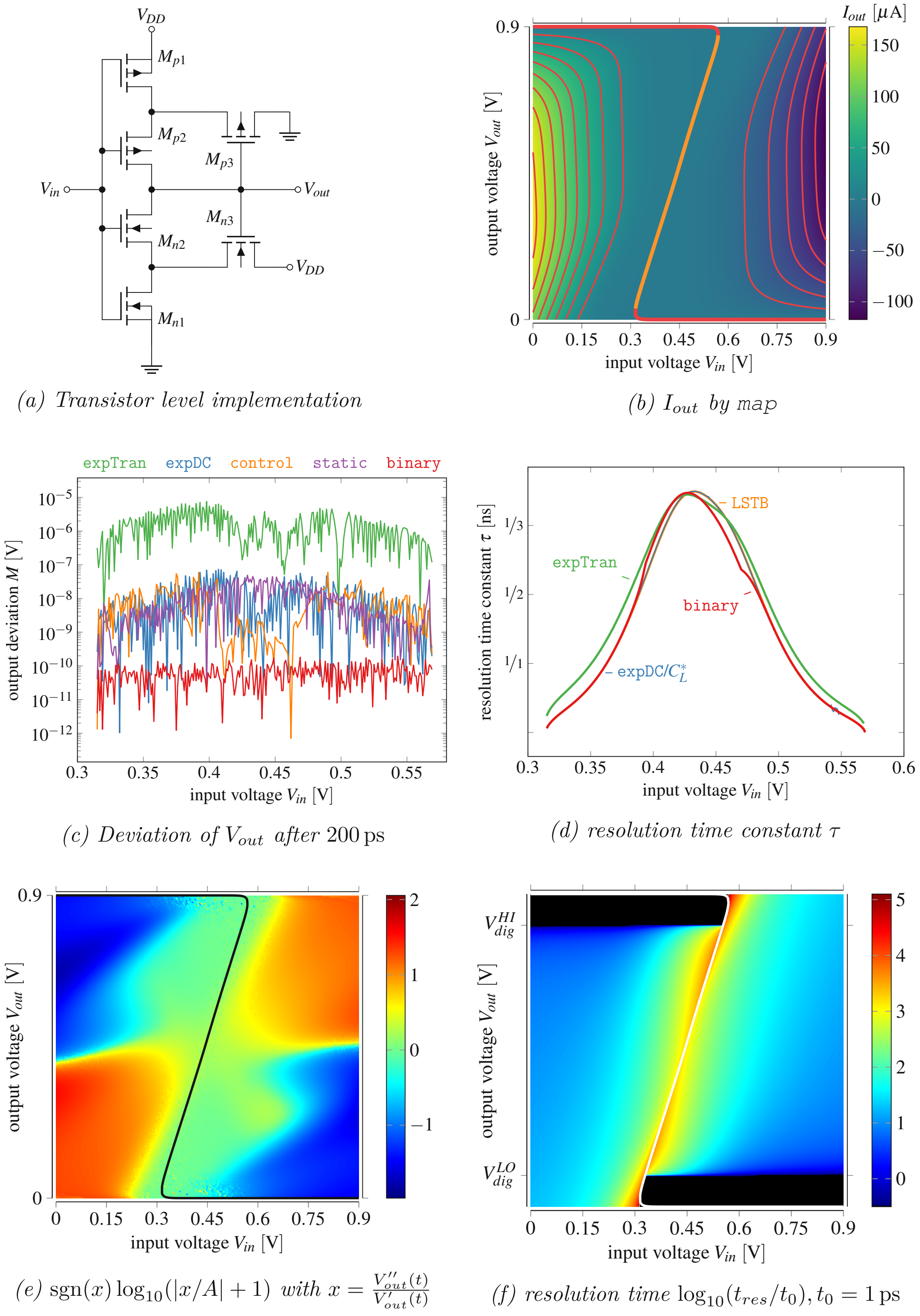
$$\epsilon = \frac{M}{\exp\left(\frac{t_0}{\tau}\right)} \; . \tag{5.16}$$

Due to the fact that the feedback paths in real circuits exhibit their own dynamics, turns them into a second-order dynamic system (in good approximation). This stands in contrast to `opamp` with its passive feedback path. As a consequence, the relation between I_{out} and V'_{out} becomes more complicated, as shown in Section 5.4.1. Consequently the results presented in the following were achieved using the modified load capacitance C_L^*.

5.5.1 Standard Implementation (`std`)

The transistor level circuit is shown in Figure 5.23a, and the obtained (meta-)stable line in Figure 5.23b (γ_1 and γ_3 in solid red, γ_2 in solid orange). In contrast to the analysis of

[6] https://github.com/jmaier0/meat



(a) Transistor level implementation

(b) I_{out} by `map`

(c) Deviation of V_{out} after 200 ps

(d) resolution time constant τ

(e) $\mathrm{sgn}(x)\log_{10}(|x/A|+1)$ *with* $x = \frac{V''_{out}(t)}{V'_{out}(t)}$

(f) resolution time $\log_{10}(t_{res}/t_0), t_0 = 1\,\mathrm{ps}$

Figure 5.23: Simulation results for `std`.

Marino, γ_1 and γ_3 are neither constants nor linear functions. Instead, the stable values start to deviate from `GND` or V_{DD} when the respective threshold voltage is approached.

The heat map of the output current (see Figure 5.23b), which utilizes linear spacing between the contour lines, reveals only moderate changes in I_{out} close to the metastable line, as expected from the exponential resolution trajectories predicted by theory. However, in contrast to the calculations of Marino [146], where V'_{out} only depends on the distance to the final, stable state (with horizontal contour lines, recall Figure 5.15), our results for the circuit `std` show mostly vertical contour lines, along with the maximum and minimum of I_{out} both located near $V_{DD}/2$. A resolution trajectory following a vertical (portion of the) contour line (which happens for constant V_{in}) exhibits a constant V'_{out} and hence a linear slope of V_{out} rather than an exponential curve.

Figure 5.23c shows an accuracy comparison for the metastable voltage V_M: `binary`, with a deviation of $M \approx \pm 0.5\,\text{nV}$ after 200 ps, performs clearly the best. `expDC`, `static` and `control` are comparable and moderately precise, while `expTran` performs worst.

Finally Figure 5.23d shows the resolution time constant τ determined using the methods `expTran`, `expDC`, `binary` and `PZ`. Their results match remarkably well. Note that this plot shows τ over V_{in} under the assumption that all initial condition pairs (V_{in}, V_{out}) lie on γ_2 and consequently τ represents resolution from (almost) perfect metastability. It can be seen clearly that, even within Region 2, the resolution time constant varies significantly over V_{in}, with the smallest values in the middle, around $V_{in} = 0.45V$. At the outskirts significantly worse values are observed meaning that these states are resolved slower.

A more general view is given in Figure 5.23e, which depicts V''_{out}/V'_{out} as a heat map in the whole V_{in}-V_{out} plane. The z-scaling is the same as for Figure 5.21b, whereat in contrast to `opamp` significant variations are visible. In particular, grid points far away from the respective stable state tend to have positive values (red regions), while closer ones have negative ones (blue regions). Considering the usual switching behavior of real-world circuits, this makes perfect sense: The output trajectory for an arbitrary constant $V_{in} < V_L$ (vertical cut in the figure) shows in the first part an exponentially growing behavior that turns into an exponential decaying one and asymptotically approaches V_{DD}. In both Regions 1 and 3, we observe relatively large absolute values while they decrease when approaching γ_2 (and, not surprisingly, at the transition from increasing to decaying exponential behavior). This is in contrast to Marino's results that predict a significantly larger τ for Regions 1 and 3 compared to Region 2 (cf. Figure 5.21b). Obviously, the `opamp` model does not sufficiently match `std` in this respect.

Finally, the map of the resolution time t_{res} is shown in Figure 5.23f. We observe, in comparison to `opamp` (cf. Figure 5.22), a further reaching and flatter dependence on the "horizontal" distance to the metastable value γ_2, but a similar clear dependence on the "vertical" distance from the corresponding stable state. This indicates a weaker impact of the input on the overall S/T behavior. Remarkable is the fact that metastable values near $V_{DD}/2$ (assuming the same distance to V_M) resolve faster than values on the outskirts of γ_2, although the distance to the stable state might be shorter.

5.5.2 Inverter Loop (`loop`)

The second circuit we investigate is a latch-like storage element (see Figure 5.24a for the transistor level implementation). It consists of a pair of cross-coupled inverters, of which the input stays permanently connected via an input inverter. The hysteresis of the input/output behavior is defined by the relation between the driving strength of the input inverter (transistors M_{p1} and M_{n1}) and that of the (weak) feedback inverter from the storage loop (M_{p2} and M_{n2}). For the latter we thus reduced the width to one tenth of their input counterparts.

The I_{out} map, see Figure 5.24b, significantly differs from the one of `std`. Much like for `opamp` (Figure 5.15), the contour lines are horizontal at the border. Near γ_2, the current changes much more rapidly than for `std`, which also leads to lower values of the resolution time constant τ (one order of magnitude, see Figure 5.24d), i.e., metastability is resolved much quicker. Interestingly, an increase in τ can be identified near $V_{DD}/2$. This may be due to the fact that in this region all transistors are saturated, meaning that voltage changes along the channel have little impact on the amount of conducted current. The methods `binary` and `PZ` once again deliver comparable results for τ. While `expTran` is only slightly off, `expDC` fails to deliver accurate values for this circuit. The main reasons are difficulties for the estimation of C_L^*.

For this circuit, the metastable voltage V_M is computed most accurately by the method `binary` (see Figure 5.24c), with `static` being very close and `expTran` performing significantly worse. Of special interest is `control` which occasionally outperforms `binary` in terms of accuracy.

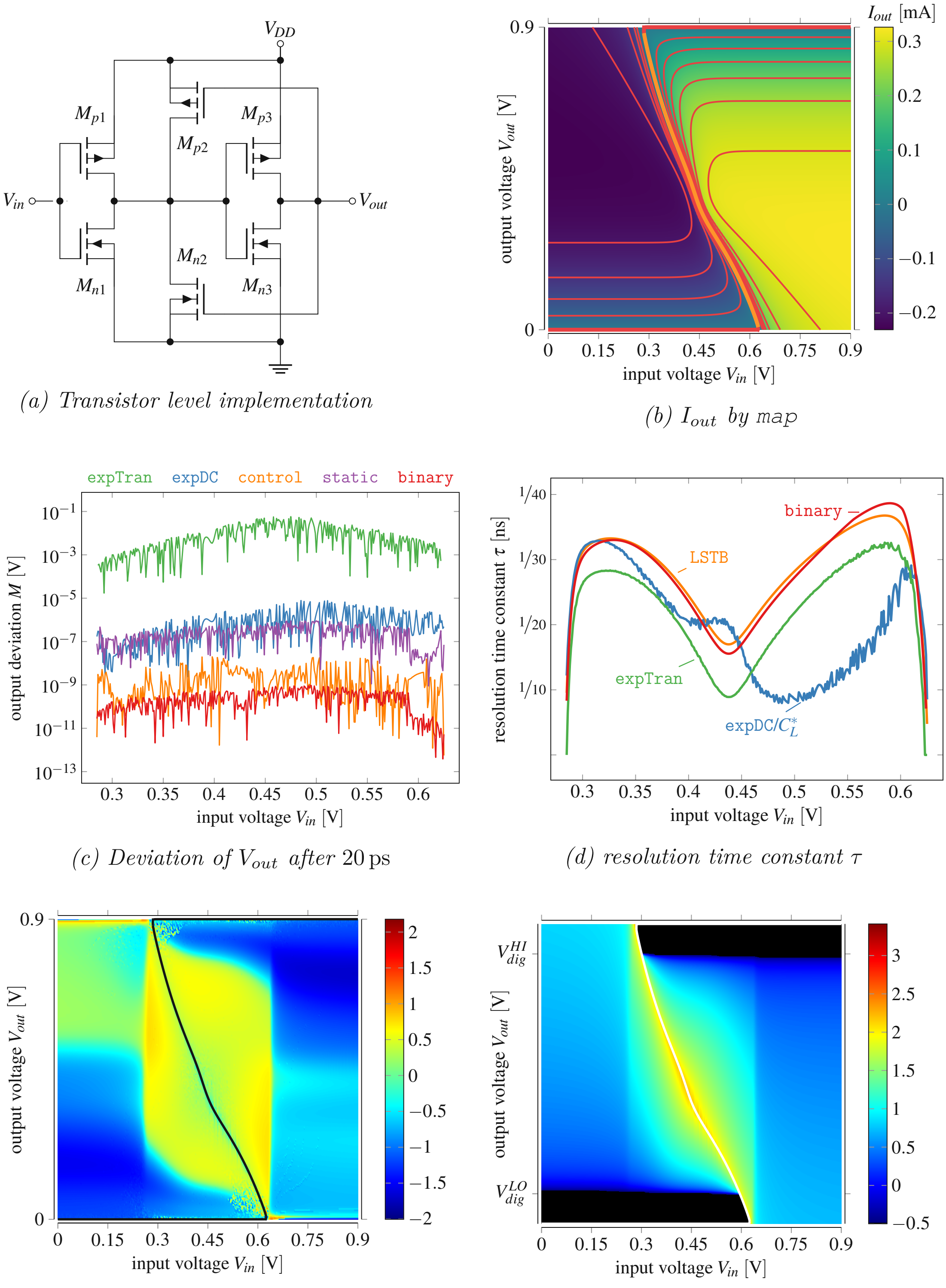
The global view shown in Figure 5.24e differs significantly from what we observed for `std`, cf. Figure 5.23e. Outside the metastable region ($V_{in} < V_L$ and $V_{in} > V_H$) almost exclusively negative values are visible indicating the near-exponential behavior of V_{out} (highest derivative at start). Inside the hysteresis, lower values that further decrease near $V_{in} = 0.45\,\mathrm{V}$, can be observed (cf. Figure 5.24d). Note that the region boundary in Figure 5.24e changes swiftly w.r.t. V_{in} near V_L and V_H. While inside the metastable region a slow and steady increase is visible, rapid changes are observable outside of it.

The resolution time plot in Figure 5.24f finally matches very well the already obtained results. The increased resolution time constant τ around $V_{in} = 0.45\,\mathrm{V}$ results in a significant increase in the combined resolution time t_{res}.

5.5.3 Adjustable Hysteresis (`adjust`)

In some applications it is important to adjust the hysteresis of the S/T during operation. One circuit that can be used for this purpose is called `adjust` and is shown in Figure 5.25a. The additional input V_B alters the position and width of the hysteresis. In our simulations we used $V_B = V_{DD}$ as in this case the hysteresis is the widest and thus has the largest amount of stable states.

The first remarkable aspect to be observed in Figure 5.25b is the vertical section of γ_3 with its relatively large peak value of V_{out}. It reaches up to about 0.3 V which is one third of the supply voltage and almost certainly in the forbidden region, i.e., above V_{dig}^{LO}. Recall



(a) Transistor level implementation

(b) I_{out} by `map`

(c) Deviation of V_{out} after 20 ps

(d) resolution time constant τ

(e) $\operatorname{sgn}(x)\log_{10}(|x/A|+1)$ *with* $x = \frac{V''_{out}(t)}{V'_{out}(t)}$

(f) resolution time $\log_{10}(t_{res}/t_0), t_0 = 1\,\mathrm{ps}$

Figure 5.24: Simulation results for `loop`*.*

from Section 5.4.1 that those states can be easily reached by a V_{in}-ramp stopping at a defined value, which implies low resilience against metastability for the circuit `adjust`. The `map` in Figure 5.25b shows similarities to that of `std` (cf. Figure 5.23b), especially in the right half. In the left half, the vertical contour lines are even more pronounced.

Due to the high similarity to `std` on the transistor level the achieved accuracy levels (shown in Figure 5.25c) and the (initial) resolution time constant τ (shown in Figure 5.25d) very much related. In absolute terms, the circuit `adjust` exhibits the largest peak value for the resolution time constant τ. This also becomes apparent in the global map shown in Figure 5.25e, where the large green area indicates slow changes. Nevertheless, the resolution time characteristics depicted in Figure 5.25f are comparable to that of `std`, cf. Figure 5.23f.

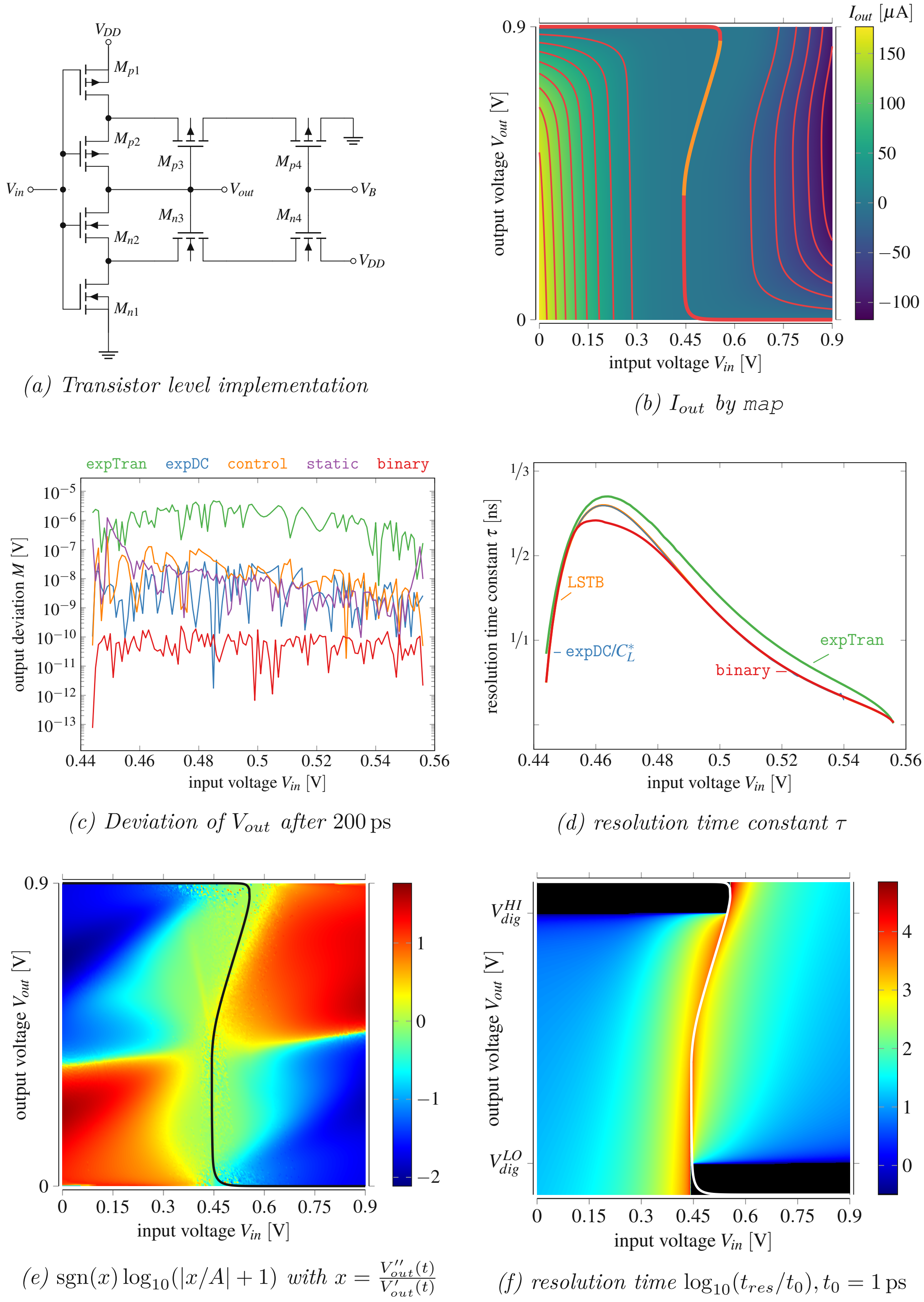
5.5.4 Comparison

In this section, the experiences gained throughout the characterization of three real-world S/T implementations are utilized for comparing and evaluating the methods that have been introduced in Sections 5.4.1, 5.4.2 and 5.4.3. Naturally, our main criteria for this evaluation are accuracy, resolution, ease of use, computing time and scalability.

The difficulty in doing an objective comparison is that trade-offs between these parameters can be made. For example, the accuracy can often be increased by investing more computing time. Similarly, the grid resolution of all presented methods can be made arbitrarily high. In practice, however, limitations apply such as the finite simulator precision (internal number format), the required computing time and the available output file formats. The latter raised significant issues for `expDC` as we only managed to export results with 7 positions after the decimal point from HSPICE, while for all other methods 10 positions were possible. Due to the above reasons, we restrict ourselves to qualitative analyses in this work. Nevertheless, to allow for a quantitative classification of the evaluated methods, Table 5.1 lists the computing times required to obtain the results that were presented in this paper on our machine (Intel Xeon X5650, 1600 MHz, 32 GB RAM, CentOS 6.10). Note that the hysteresis and therefore the number of metastable grid points between V_L and V_H differ among the S/T implementations.

`hyst` Using the method `hyst` the hysteresis curve is determined by two *DC* analyses starting at V_{in} = `GND` and $V_{in} = V_{DD}$, respectively. Due to the fact that γ_1 and γ_3 are almost constant, their exact characterization is simple and fast. The accuracy is directly dependent on the simulation tool and the circuit element model, and is therefore excellent, since no assumptions (e.g. on signal shapes) apply.

For obtaining V_L and V_H with high resolution, as well as exploring the non-ideal shape of γ_1 and γ_3 in the proximity of these threshold voltages, a small step size is desirable. Fortunately, the computation time scales only linearly with the number of samples. Consequently, state-of-the-art designs can be processed with excellent resolution within several seconds.



(a) Transistor level implementation

(b) I_{out} by `map`

(c) Deviation of V_{out} after 200 ps

(d) resolution time constant τ

(e) $\operatorname{sgn}(x)\log_{10}(|x/A|+1)$ *with* $x = \frac{V''_{out}(t)}{V'_{out}(t)}$

(f) resolution time $\log_{10}(t_{res}/t_0), t_0 = 1\,\text{ps}$

Figure 5.25: Simulation results for `adjust`.

	computing time [s]		
circuit	`std`	`loop`	`adjust`
metastable grid points	282	378	125
`hyst`	1.817	2.297	1.842
`binary`	238.834	370.135	115.272
`map`	661.302	676.525	797.702
`expTran`	841.718	1089.741	334.582
`expDC`	791.997	890.129	340.751
`control`	3581.520	2340.700	3258.726
`static`	2.572	2.818	2.627
τ(`binary`)	1194.648	552.158	537.934
`PZ`	730.889	1064.792	241.888
t_{res}/τ map	14 102.607	3939.382	11 228.703

Table 5.1: Computing times of S/T characterization methods.

`map` Similar statements as for the method `hyst` can be made for the phase diagram `map` with the significant difference, however, that the grid is now two-dimensional (V_{in} and V_{out}), and therefore an increase of the resolution has a quadratic impact on the computing time. In this paper we used a regular grid in the whole V_{in}-V_{out} plane, but much smarter choices are conceivable. The effort could, for example, be massively reduced if the grid is continuously refined with decreasing distance to γ_2.

Using I_{out} as an indirect measure for V'_{out}, reduces on the one hand the computing time significantly. On the other hand, however, the accuracy suffers due to uncertainties associated with C^*_L, as outlined in Section 5.5.1. Consequently the contour lines obtained using the method `map` should be considered as a qualitative result only – which is, nevertheless, often sufficient. If accurate quantitative results are required, the overhead of directly determining V'_{out} must be accepted.

`expTran` While other approaches purely rely on the data points derived from analog simulations, `expTran` incorporates analytic considerations as well. In fact, a few values extracted from the simulation are used to parameterize a known (exponential) function. The latter then allows to quickly derive τ and V_M with a maybe significantly improved resolution compared to the simulation tool. This property makes `expTran` appealing.

However, our results show that it achieves the worst accuracy for the metastable values γ_2, and quite some deviations for the resolution time constant τ. The reasons for these imperfections are (i) the relatively poor accuracy of HSPICE for determining V'_{out}, and (ii) the fact that in real-world circuits the resolution trajectories are not perfect exponential functions, even in Region 2. In detail, we have observed that V'_{out} changes more rapidly than an exponential function in the vicinity of the region boundaries. At the same time, care must be taken that the data points are extracted from within Region 2 (recall, it may be very small), as the trajectory definitely follows a different function outside. Therefore, the method `expTran` is more challenging to apply.

For each value of V_{in} within the hysteresis, two transient simulations are run. Since the initial values are taken from `map`, a finer grid for this `map` also improves the accuracy of `expTran`. However, the computation time scales quadratically for this method.

`expDC` This method is very similar to `expTran`, as also the grid points of `map` closest to γ_2 are utilized. In contrast, however, no separate simulations are required to predict the metastable voltage, which enables a rather quick execution of typically a few seconds. Unfortunately, it is necessary to determine the implicit load capacitance C_L^* in advance (cf. Section 5.4.1), whereat its value varies among operating points and thus requires transient simulations and proper averaging.

Similar to `expTran`, the method `expDC` also has the potential to speed up the simulation by leveraging the knowledge of the resolution trajectories being exponential, but then suffers in accuracy when this assumption is not perfectly met by the circuit. Still, for trajectories originating from γ_2, the results prove to be very accurate.

The resolution time constant τ is extracted from (5.14) and (5.10), which is fitted to the slope of $V'_{out}(V_{out})$. Although this is easily possible, the challenges regarding C_L^* described above and fitting to numerically noisy simulation data, sometimes lead to poor results. Consequently, for predicting τ, `expDC` is, on its own, only of limited use.

`binary` For each value of V_{in} within the hysteresis a binary search has to be executed. While the overall amount of simulations thus scales with the grid granularity of V_{in}, the amount of binary steps has hardly any impact on the computation time. We experienced a reduction by only 10 % when lowering the number of iterations from 40 to 20 whereas the accuracy was degraded by four orders of magnitude. The results achieved for V_M are among the most accurate ones, however, τ cannot be directly computed.

`control` The approach `control` essentially relies on a controller that stabilizes the dynamic system, i.e., the S/T, in a metastable state. Naturally, the parametrization of that controller significantly influences how fast and accurate V_M will be approached: With a slow controller, convergence will be unproblematic and robust, while a fast controller introduces overshoot, ringing and eventually even instabilities. That is why the choice of the controller gain K is critical. Although, theoretically, the metastable voltage V_M can be approached perfectly accurate, it would take an infinite amount of time to do so. For this reason, and also due to the limited accuracy of HSPICE, we settled for a simulation time in which the controlled circuit approximately reaches a steady state. In this fashion, very accurate results could be achieved. Fortunately it is possible to decrease the accuracy of the simulation to enlarge the simulation time horizon that can be processed with the same computational effort, while in turn improving the accuracy of the obtained metastable value V_M.

Overall, this approach, while being elegant in exposing the metastable value V_M for direct extraction, turns out to be rather time consuming. We primarily see its application in cases where little to no information about the metastable behavior or the circuit itself is available.

`static` The fact that the *DC* analysis in HSPICE reuses a preceding stable configuration as starting point for the succeeding iteration, makes this type of simulation very fast and accurate. In general, results can be obtained within a couple of seconds. The

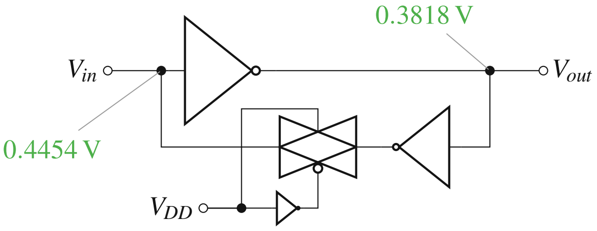


Figure 5.26: Latch circuit used for DC metastability analysis. The correct values are automatically achieved for setting both nodes to $V_{DD}/2$.

biggest problem we faced is the limited output data format, which reduced the achievable (exportable) accuracy significantly.

Essentially, the approach `static` heavily relies on the *Newton-Raphson* algorithm that HSPICE uses internally for static analyses. This means that any future changes in the internals of HSPICE may invalidate the method, although we are optimistic that this will not be the case. Furthermore the resolution time constant τ cannot be estimated.

`PZ` Determining the resolution time constant τ based on the smallest positive pole of the circuit works remarkably well, even in the presence of multiple poles. Clearly, one has to run an *AC* analysis in advance to extract the respective data. This has to be done for each value of V_{in} with the initial condition $V_{out} = V_M$, so the computational effort increases linearly for finer granularity.

Resolution Time Constant τ and Resolution Time t_{res} The heat maps for τ and t_{res} are the result of two transient simulations per value of V_{in}. Although the simulation time is kept short using a simple heuristic, starting in deep metastability with low driving strength leads to a significant computational effort and consequently long run time. In this case not only the computation time of HSPICE has to be considered; the extraction of τ and t_{res} from the simulation traces also creates non-negligible computational efforts.

General Observations While using both transient and *DC* analyses, overall we experienced that the former are much harder to handle. The reason is that more parameters have to be defined, most notably the time period of the simulation. In addition further complications, such as extracting a specific part of the simulation in `expTran` or finding an appropriate controller gain K for `control`, have to be overcome. In total, *DC* analyses achieve better results with less computation time and simpler methods. In this regard *AC* analyses are comparable to *DC*.

Finally we want to emphasize that the proposed methods are not restricted to S/Ts. We verified this by exemplarily running `static` and `control` on a latch formed by a loop of asymmetric inverters (width ratio 1/10) with a transmission gate (see Figure 5.26). In both cases the metastable configuration was quickly achieved, which indicates a potential general applicability of our analyses and thus a large application area.

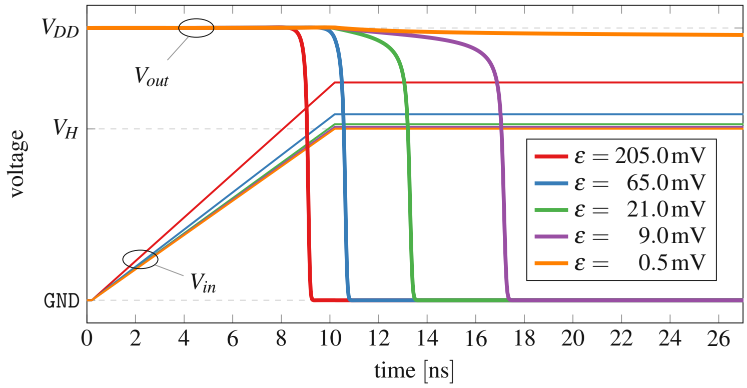


Figure 5.27: Analog HSPICE simulations of late transitions caused by a linear input traces that stop at varying values $V_H + \varepsilon$. The smaller ε the longer it takes until the stable output state $V_{out} = \texttt{GND}$ is achieved.

5.5.5 Driving into Metastability

For the `std` implementation we further investigated in technology (T65) how intermediate values can be forced by proper adjustment of the input. As we will show with simple ramps, that stop at a certain value, it is possible to generate late transitions, while more fine grained control of V_{in} even enables arbitrary waveforms at the output.

Monotonic Inputs

In this section we investigate the behavior of the S/T when provided with a monotonic input, which may stall at an arbitrary value V_S. This corresponds to a gate, e.g. a Flip-Flop, that starts a transition and then enters metastability. To simplify our analysis we focus on the case of an increasing input, starting at $(V_{in}, V_{out}) = (\texttt{GND}, V_{DD})$. The reverse case can be analyzed analogously.

As we have already discussed earlier, exceeding the threshold V_H causes a negative output derivative, which can only be reverted by reducing V_{in}. Since we consider a monotonic input, this is not possible, which implies that V_{out} has to change from V_{DD} to GND. Problematic are, however, cases where the input transition stalls near the threshold, i.e., when $\varepsilon = V_S - V_H$ is very small. In our previous analyses we have shown that the output derivative declines as the (meta-)stable lines are approached, since V'_{out} is zero there. For this reason we expect increasingly delayed output transitions for decreasing ε.

The corresponding simulation results for various values of ε are shown in Figure 5.27. As predicted, monotonic inputs lead to steep monotonic output transitions, but their delay varies significantly. Note that the output trajectories differ close to V_{DD} which shows impressively the continuously decreasing output derivative towards γ_2. While the relative deviations among 205 mV, 65 mV and 21 mV are rather low, switching from 9 mV to 0.5 mV massively increases the delay. This effect can be observed in Figure 5.28, where the delay based on ε for falling and rising input transitions is shown. For $\varepsilon \to 0$

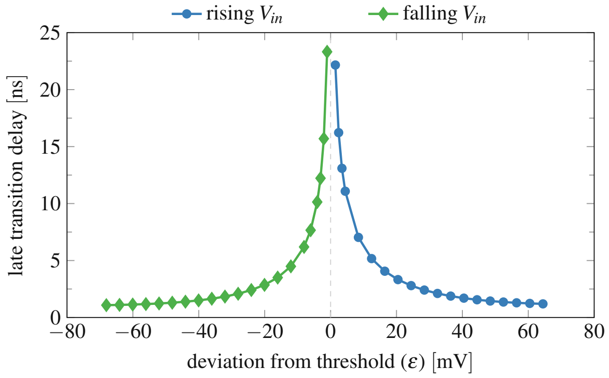


Figure 5.28: Late transition delay for rising and falling V_{in} stopping in distance ε above V_H / below V_L. The closer to zero the longer the delay with a pole at $\varepsilon = 0$.

the delay is actually unbounded; a behavior that is comparable to proximate clock and input transitions at a Latch [29], with the difference, that the S/T does not drive an intermediate value.

Driving Intermediate Output Voltages

To actually achieve arbitrary intermediate voltages at the S/T output, more sophisticated input trajectories are required. Assume that we start on γ_1: Once V_{in} exceeds V_H the output is pulled towards GND. In order to reach a metastable state on γ_2 we need to reduce V_{in} sufficiently before $V_{out} =$ GND is reached. Recall that our simulations revealed an increase of V'_{out} with increasing distance to γ_2, which, in turn, demands higher input dynamics. Despite this insight it is not possible to provide a lower bound for V'_{in} that is required to achieve metastability since there exists always a corridor surrounding γ_2 where metastability can be maintained (cf. Section 5.3), even for $\max(V'_{in}) \to 0$.

As γ_2 connects γ_1 and γ_3, every output value in between the latter can be approached. Even worse, one can switch among different metastable values. On the bright side we have to note that a very, very precise control of the input voltage is required, which also has to stay near $V_{DD}/2$ the whole time (input resolution implies S/T resolution). In real circuits such series of events can be assumed to be very unlikely.

Creating Arbitrary Output Trajectories

In principle, by appropriately navigating in the phase plane, one can achieve (almost) arbitrary output trajectories: For every value of V_{out} an appropriate V_{in} can be applied to obtain the desired gradient V'_{out} (by crossing γ_2 even the sign can be changed). However, within a limited range of V_{in} only a limited range of V'_{out} can be covered; in other words, the dynamics of V_{out} is naturally confined by the system dynamics. The second restriction is, as already described previously, the dynamics of V_{in}. To stay in metastability the differences induced by the changing V_{out} have to be compensated. If this is not possible

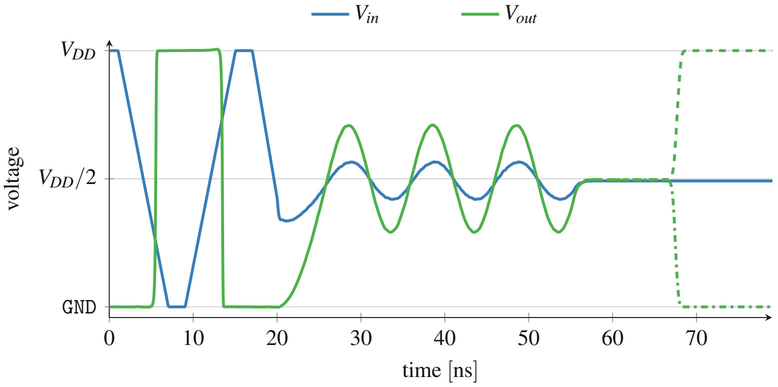


Figure 5.29: Creating an arbitrary waveform at the output of the S/T. After an initial phase a sine with a frequency of 100 MHz *is created. Note that, albeit looking that way,* V_{out} *does not follow* V_{in} *in this phase. Instead* V_{in} *is used to slow the output down. In the end the S/T is allowed to resolve starting from minorly differing values.*

there is no way of preventing the trajectory from approaching the saturation of V_{out} in a monotonic trace.

To show this we aim to create an arbitrary waveform at the output, more specifically a sine wave with frequency 100 MHz and a voltage swing of 0.5 V, in a simulation. Our results (see Figure 5.29) reveal that such a behavior can be achieved by means of non-monotonic inputs. In the first part regular operation is presented to demonstrate the dynamics of the S/T as well as its thresholds. At 20 ns the sine output is started. Albeit looking as if V_{out} follows strictly V_{in}, actually the reverse is the case: While the output tries to escape metastability, e.g. towards V_{DD}, we increase V_{in} such that we get closer to γ_2 and thus reduce the output derivative. As the peak of the sine wave is reached we exactly hit the metastable state. Finally, the simulated S/T is driven into deep metastability with the input being constant starting at 58 ns. From there onwards the results of two simulation runs, whose final value differ by ≈ 30 nV, are presented. While one (dashed line) resolves to V_{DD} the second one (dashed-dotted line) resolves to GND after about 10 ns. This again shows how accurate one has to hit the metastable value. In the phase plane, depicted in Figure 5.30, it can be seen that the generation of the slow (w.r.t. its regular switching speed) sine demanded to stick close to γ_2. The metastable resolution at the end is depicted by the vertical line segments at $V_{in} \approx V_{DD}/2$.

Overall our simulations showed that arbitrary output behaviors with little constraints can be realized. Nevertheless, we also experienced that it takes an *extremely* precise control of the voltage (in the range of nV) in order to (i) steer into metastability and (ii) to stay there. One major conclusion is that a monotonic trajectory will not lead to metastability inside the S/T, which implies that the latter can safely recover Flip-Flop metastability without glitching. This, however, only holds in a value-safe environment, since late transition cannot be prevented.

Although metastability cannot be ruled out completely, we conclude that the chances

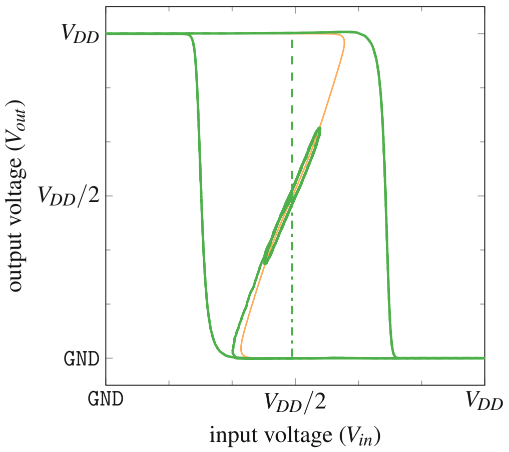


Figure 5.30: Phase space representation for the output waveform presented in Figure 5.29. During the sine wave behavior the trace sticks closely to γ_2 (orange line).

that an S/T enters a metastable state in a real application are very low. Albeit having said that, we will investigate in the succeeding section if and how the situation may be improved by cascading multiple S/Ts.

5.6 Cascading Schmitt Triggers

From the previous analysis we can conclude that a single S/T stage improves the signal quality as (i) almost all analog values in the forbidden region are mapped to either HI or LO and (ii) slowly creeping, yet monotonic, signals result in a clean steep transition. So, in principle, a subsequent S/T stage should obtain a similar improvement and thus reduce the susceptibility for entering metastability further compared to a single stage. Although a comparable approach on synchronizers, where an additional Flip-Flop increases the MTBU significantly, has already been shown to be successful [53], it is not yet clear whether this property also holds for the S/T. In order to come to a conclusive answer, the following questions will be addressed in the sequel:

(Q1) In which cases does the second S/T stage improve the behavior?

(Q2) Can the behavior get worse? Are there new types of (likely) behavior?

(Q3) Is the second stage equally likely to become metastable as the first one?

(Q4) Can metastability of the last stage be completely avoided, possibly by forming a longer cascade?

(Q5) How are the static properties of the cascade determined (is it still an S/T, and if so, which hysteresis does it have)?

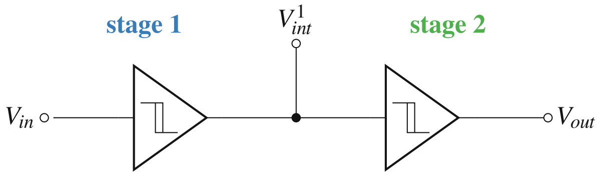


Figure 5.31: Two stage S/T cascade used in this thesis. We chose to use two inverting S/Ts, however, any combination of (non-)inverting ones are possible.

(Q6) How are the dynamic properties determined (regular delay, output slope, is there a performance penalty in using a cascade)?

(Q7) Are there any rules for optimal dimensioning of the cascade (combination of fast and slow stage, different hystereses,...)?

To answer these questions comprehensively we will, in the following, elaborate on the behavior of a (2-stage) S/T cascade. For our analyses we consider two equal inverting S/Ts in the cascade, whereat for more general statements, e.g. the combined phase plane representation, we also investigate arbitrary combinations.

5.6.1 Behavior of a Schmitt Trigger Cascade

For a start we consider a two stage S/T cascade as shown in Figure 5.31. In Section 5.5.5 we argued that an S/T can exhibit almost any output behavior. This means that the first stage qualitatively does not restrict the second one's input space, and, as a consequence, stage 2 has unrestricted output behavior as well. At this point we can already answer question (Q4) about complete avoidance of metastability through a S/T cascade: This is simply not possible.

In continuation of the physical analogy given in Section 5.3 we can view the second stage as an additional vertical stick balancing on top of the first one. This analogy nicely illustrates that it becomes much more unlikely to see metastability in the second stage (i.e. actually find a balance for the second stick) – thus giving an intuitive answer to (Q3) – , while still being physically possible.

To get closer to a quantitative answer, we will analyze in the sequel how the different output behaviors of the first stage, which we described in the previous section, are handled by the second one. Note that we use in this context V_{int}^1 to denote the voltage level on the internal wire connecting the S/Ts (cf. Figure 5.31).

Regular Behavior

Let us start with the regular behavior: Assume a starting point $V_{in} < V_{L,1}$ (the case of $V_{in} > V_{H,1}$ can be handled analogously), whereat the subindex 1 indicates that these are the threshold voltages of stage 1. Due to the inverting behavior of each of our S/T stages, we end up with V_{int}^1 being HI and V_{out} LO. As V_{in} increases, V_{int}^1 and V_{out} stay constant until V_{in} reaches $V_{H,1}$. Beyond that point V_{int}^1 will switch to LO. With a strictly monotonic V_{in}, this transition of V^1_{int} will be rapid, causing in consequence also the second stage to switch, namely when crossing the corresponding threshold $V_{L,2}$. Overall, we experience a clean rising transition at the output. Note that the threshold of the overall cascade in this case is $V_{L,1}$, while $V_{L,2}$ is irrelevant. Obviously, when using different implementations, the order in the cascade has a big impact. In general one prefers a wider hysteresis making it necessary to place a corresponding device at the beginning of the cascade. This is one option that can be considered when answering question (Q7), whereat we will present some more in the sequel.

Generally, in this mode of operation strong signal regeneration effects can be expected as the first S/T tends to switch very quickly when its threshold is reached, causing a similar behavior in the second one. Noteworthy is, however, the increased propagation delay, whose impact on metastability properties has already been investigated by Chaney [145] and Kleeman and Cantoni [137] for a synchronizer. They correctly state that the use of S/Ts is not beneficial for avoiding metastability there due to the additional delay, which even degrades the performance.

Late Transitions of Stage 1

According to our analysis in Section 5.3, a monotonic input stalling at a constant value near the threshold will cause a late but clean transition at the output of stage 1 (V^1_{int}). In that case the second stage perceives a clean input, which it simply conveys (adding its nominal propagation delay). Consequently late transitions are essentially not modified by the second stage.

For evaluation purposes ramps stopping at differing values are applied to the input. As shown in Figure 5.32, the first stage responds with late but clean transitions, while the second stage increases the slope. Note that again mainly the properties of the first stage are important, as the second stage does not yield further improvements. Thus, considering question (Q7), the ordering of the S/Ts has again an impact.

Pulse Propagation

A single S/T stage may or may not propagate a glitch or runt depending on the input pulse height and width. As the second stage shows the same behavior, glitches and runts are able to propagate through the whole cascade, but experience in the process significant degradation. This results from the fact that the thresholds have to be crossed before the output starts to move. On the other hand spurious pulses are transformed into stable transitions due to high amplification of both devices.

Simulation results shown in Figure 5.33 clearly reveal an increasing separation among the trajectories, which is a direct result of pulse-width degradation (cf. also Chapter 4). Comparing the longest pulse on V^1_{int} and V_{out}, i.e., the period of time they stick near GND/V_{DD}, might create the impression that the second S/T prolongs pulses. This is, however, just an optical illusion due to an improved transition slope. Checking the respective $V_{DD}/2$ crossing times confirms the constant width.

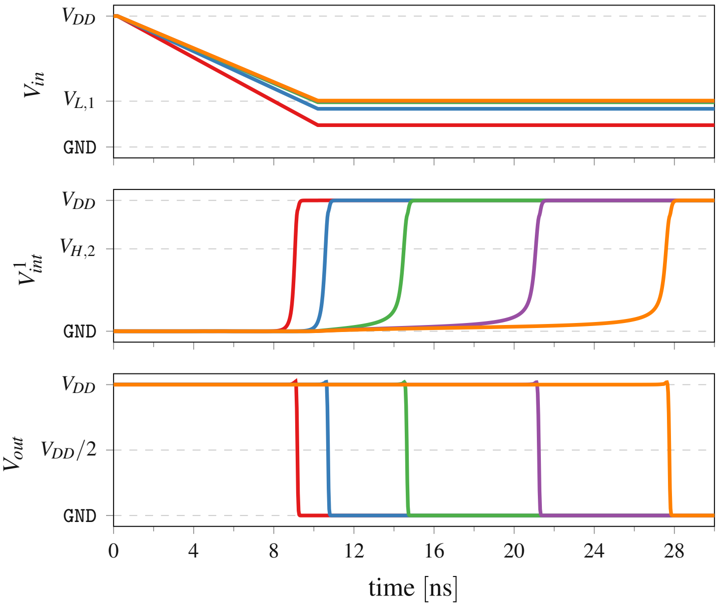


Figure 5.32: Analog simulations of V_{in}, V_{int}^1 and V_{out} for input slopes stalling at a constant value near $V_{L,1}$. Clearly stage 2 only increase the slope but leaves the transition time untouched.

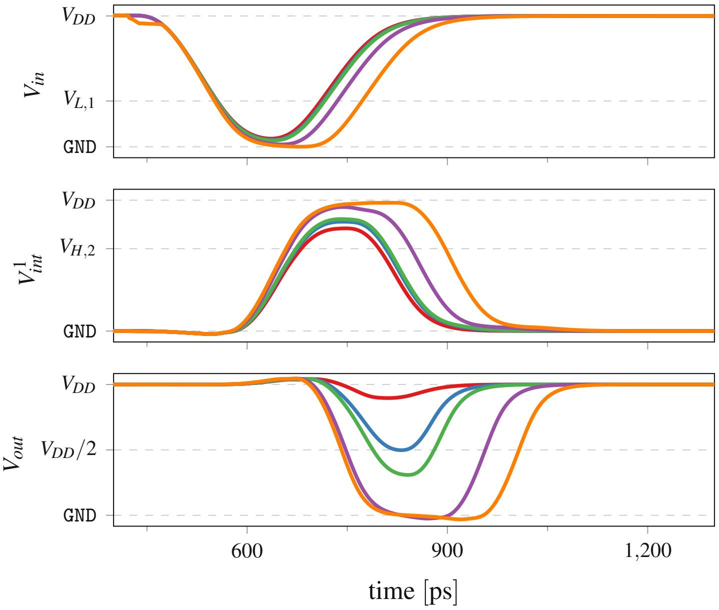


Figure 5.33: Analog simulations of the S/T cascade for input pulses. Traces initially very close show significant different output behavior due to pulse-width degradation effects, introduced by each single device.

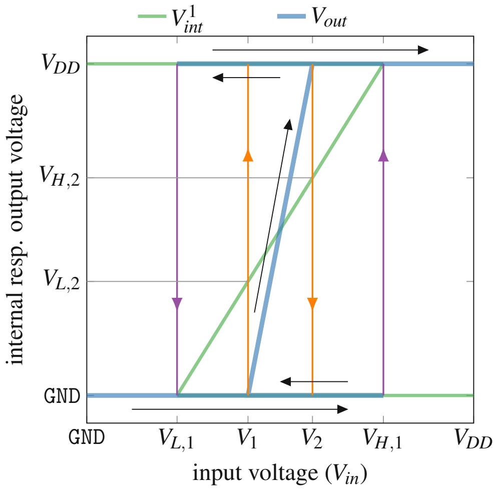


Figure 5.34: Theoretical analysis of the (meta-)stable states for the first S/T (green) and the cascade of two (blue). While the threshold points of the cascade are equal to the values of the first S/T (purple lines) with reversed direction, the metastable line γ_2 got much steeper since metastability is only possible when V_{int}^1 is driven between $V_{H,2}$ and $V_{L,2}$. The orange lines, which indicate a resolution of stage 2, must not be crossed to keep the cascade in metastability.

5.6.2 Phase Plane

In our analysis of a single S/T, we derived that metastable behavior is only possible if $V_L \leq V_{in} \leq V_H$ is fulfilled, i.e., inside the hysteresis. Applied to our cascade this means that we need $V_{L,2} \leq V_{int}^1 \leq V_{H,2}$ to drive the second stage into metastability. This leads to the general observation, that it is mandatory to drive the first $n-1$ S/Ts of a cascade into metastability to achieve a metastable value at the n^{th} Schmitt Trigger.

In the following we will show how to derive the (meta-)stable states for the cascade (blue line in Figure 5.34) and thus answer question (Q5). For simplicity reasons we start with the static case, meaning that the output value can be determined directly from the phase plane representation. For $V_{in} = \texttt{GND}$ the intermediate voltage V_{int}^1 results in V_{DD}, which leads, in turn, to $V_{out} = \texttt{GND}$. Overall we get $(V_{in}, V_{out}) = (\texttt{GND}, \texttt{GND})$, i.e., an non-inverting behavior. Increasing V_{in} does not have any effect (straight line at $V_{out} = \texttt{GND}$) until $V_{H,1}$ is surpassed. At this point it becomes possible to steer the first S/T into metastability by reducing V_{in} and consequently decreasing V_{int}^1. Continuing until $V_{in} = V_1$ causes the internal voltage to drop to $V_{L,2}$ which in turn makes it possible to drive the second S/T into metastability. Note that until this point the output did not change. Reverting the direction of V_{in} causes both, V_{int}^1 and V_{out}, to rise as well, due to the positive slope of γ_2. This setup is maintained until V_{int}^1 reaches $V_{H,2}$ causing the output to reach V_{DD}. Note that increasing V_{in} beyond this point would cause V_{out} to drop to GND as indicated by the downward orange arrow in the figure. To actually continue the blue line at $(V_{out} = V_{DD})$ it is necessary to resolve metastability in stage 1

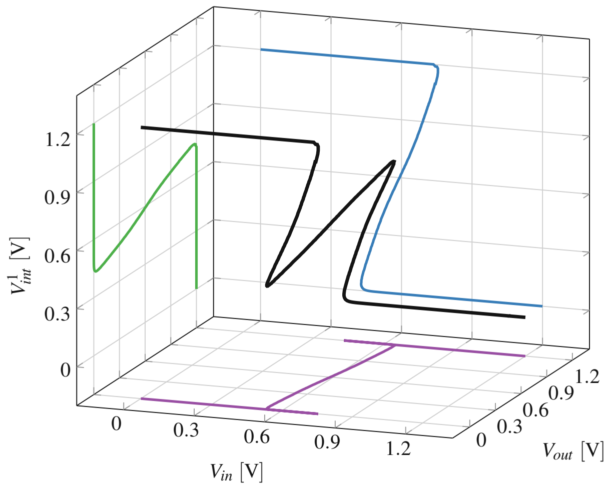


Figure 5.35: Phase plane representation of the cascade in three dimensions. The single faces show the characteristic for each single stage and the combined one.

first. This is done by decreasing V_{in} until $V_{L,1}$ is reached and then raise it to V_{DD}.

While the combined characteristic still inhibits properties of the single S/Ts it also shows significant differences. For example, the threshold voltages of stage 1 are preserved (purple lines in Figure 5.34) while the ones of stage 2 are encoded in the values V_1 and V_2. Since $V_2 - V_1 < V_{H,1} - V_{L,1}$ the slope of γ_2 increases, which emphasizes once more the importance of the order in the cascade. Overall, the shown characteristics suggests that the cascade has a substantially lower probability for metastability than a single S/T.

A three dimensional representation of the behavior is shown in Figure 5.35. It depicts the simulated (meta-)stable points in the $(V_{in}, V_{int}^1, V_{out})$-space. The blue line at the back represents V_{int}^1 over V_{in} and thus the (meta-)stable states of the first stage, while the projection to the left plane (V_{int}^1 over V_{out}) shows (a rotated image of) the second stage. The most interesting one is the projection of the curve to the ground plane, i.e., V_{out} over V_{in}, which indicates the (meta-)stable states of the cascade. This curve exactly matches our prediction from Figure 5.34.

Evaluation

To verify our predictions we run HSPICE simulations using technology (T65), where we not only drive the cascade into metastability but even force the output to arbitrary waveforms (in this example a sine wave). The achieved results are shown in Figure 5.36. At the beginning (first 15 ns) regular operation is shown which clearly reveals the hysteresis. Note that the first S/T dominates the behavior as a full range transition on V_{int}^1 also causes V_{out} to flip.

Afterwards we drive the first (≈ 15 ns) and finally the second S/T (≈ 28 ns) into metastability by carefully controlling V_{in}. It then is possible to create a sine wave at the output. In this phase the non-inverting behavior of the single stages and the amplification

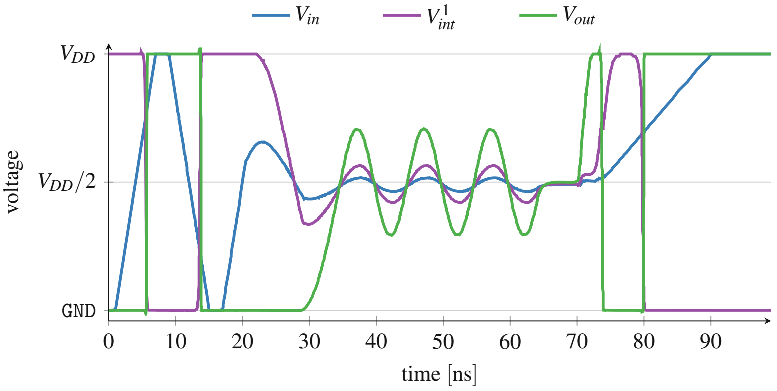


Figure 5.36: "Arbitrary" waveform derived by operating both S/Ts in metastability. After the initial normal operation a sine wave is enforced at the output. Note the amplification in this region, causing a rather small swing on V_{in} having large effects on V_{out}. In the end separate resolution of the single stages can be observed for a linear input.

of V_{in} to V_{int}^1 and then V_{out} can be observed very clearly. Once again note that V_{int}^1 and V_{out} do not follow V_{in} but are actually slowed down by the latter, effectively preventing resolution of metastability. Finally a constant input value is applied which causes the S/T in stage 2 to resolve to V_{DD} at first. Since stage 1 afterwards also resolves to V_{DD} the output switches back to GND. This is a very disadvantageous property of the cascade, which will be discussed in Section 5.6.3. Finally ($\approx$ 73 ns) the input is continuously increased which eventually forces an additional transition on V_{int}^1 and V_{out}.

Discussion

So far we only considered equal S/T implementations. In the sequel we thus investigate how the behavior changes if (1) two non-inverting S/Ts or (2) a mixture of various kinds is used. For the first case actually the same result is achieved. While this is rather obvious for the initial values, it seems counter-intuitive for γ_2. Nevertheless, in that case an increasing V_{in} would cause a decrease in V_{int}^1 which in turn again leads to an increase of V_{out}. Of course this statement is only valid for a cascade of even length. For an odd number the slope of γ_2 is negative, which differs from using solely inverting ones.

We already mentioned on various occasions that the order of varying Schmitt Trigger implementations is important. For example, the threshold voltages V_H and V_L of the cascaded system are determined by the first S/T alone, while V_1 and V_2 are influenced by both of them. Let V_{Nj} denote $(V_{H,j} + V_{L,j})/2$, i.e., the mean of the threshold voltages at stage j. If $V_{N1} = V_{N2} = V_{DD}/2$ the order in the cascade has no impact. In all other cases V_1 and V_2 might vary, however, with constant deviation, which evaluates to

$$V_2 - V_1 = \frac{1}{V_{DD}} (V_{H,1} - V_{L,1}) (V_{H,2} - V_{L,2}) .$$

The value $V_2 - V_1$ has ambivalent features: If it is small γ_2 is steep and thus hard to achieve, however, once in metastability it is easier to create oscillatory behavior with a lower voltage swing (cycling the orange lines in Figure 5.34), answering partly questions (Q2) and (Q5).

5.6.3 Resolving Metastability

We have shown previously, that, in order to achieve metastability at the last S/T in the cascade, all previous ones also have to be metastable. This means that for n stages n input transitions, in the sense that the trace had to revert its direction, were necessary. Such a cascade is thus able to store transitions. It has, however, even greater capabilities than that, being it intended or not. Assume that the S/Ts in the cascade resolve from end to start. In this case n transitions on the output are only visible if every single S/T resolves to the value contradicting the state of the succeeding one. In all cases, especially if the first stage resolves the earliest, only a single transition may be visible.

The cascade thus has multiple (un-)desired properties:

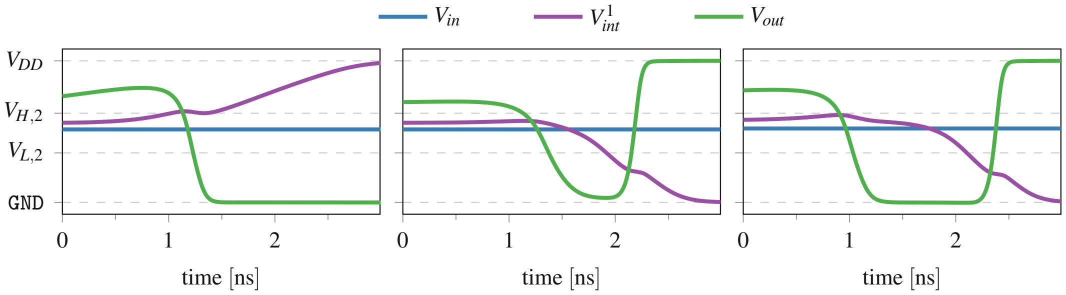
1. Transitions can be consumed, which is especially concerning for asynchronous circuits and systems.

2. Looking at the output of the cascade makes it impossible to determine if all S/Ts have resolved.

3. The resolution speed is amplified stage by stage due to a gain > 1. The further the stage that resolves is to the front, the quicker the resolution out of metastability.

An interesting behavior can be observed when, with both stages in metastability, V_{in} is increased to a value between V_2 and $V_{H,1}$, which causes the second stage to flip to LO (cf. Figure 5.34). A further (monotonic!) increase of V_{in} (beyond $V_{H,1}$) will then cause the first stage, i.e., V_{int}^1, to switch to LO, which, in turn, causes the second stage to flip back to HI. In this case we have observed a glitch at V_{out} that was caused by a monotonic transition of V_{in} (however, a non-monotonic V_{in} was initially required to bring both S/Ts into the metastable state in the first place).

Figure 5.37 shows further analog simulations of various possible resolution scenarios for a metastable cascade. In (a) stage 1 and 2 try to resolve to the same value (HI). At some point V_{int}^1 increased enough to force V_{out} to change direction and drop rapidly. The situation can be fundamentally different when the first S/T resolves at a later point in time, introducing a glitch at the output, which can be observed in (b). Recall that we already saw this behavior when creating the sine wave shown in Figure 5.36. While a proper investigation of the static plane can explain these effects it fails for the traces shown in (c). There stage 2 influences the behavior upstream, most probably due to coupling capacitances. In the figure one can see that initially the first Schmitt Trigger resolves towards V_{DD} until $V_{H,2}$ is reached, causing stage 2 to resolve to GND. This transition, however, leads to a dip on V_{int}^1 such that stage 1 reverts its direction and resolves towards GND itself, which, in turn, induces another output transition. In the stick



(a) resolution in same direction (b) stage 2 resolves first (c) stage 2 influences stage 1

Figure 5.37: Analog simulation results for resolving from metastability in the cascade. We observed an upstream coupling in (c) where the resolution of stage 2 causes stage 1 to switch from approaching V_{DD} to GND.

analogy introduced earlier, this corresponds to the case where the upper stick falls to one side and pushes the lower stick thereby to the side, which seems intuitively reasonable.

While these cases definitely represent new types of (often undesired) behavior not seen with a single stage – thus answering (Q2) positively – one should keep in mind that it takes an *extremely* precise control of V_{in} to navigate into this setup.

5.7 The Mean Time Between Upsets of Schmitt Triggers

In this chapter we already showed that the S/T can become metastable and even managed to characterize multiple implementations. Nevertheless, we did not succeed to answer the question how probable it actually is to achieve metastability during normal operation. In the sequel we are thus presenting first intermediate results towards deriving a characteristic number, just like the MTBU of a Flip-Flop, also for the Schmitt Trigger.

In Section 5.3 we argued that the S/T is situated in between the Flip-Flop and a purely combinatorial logic. In contrast to the former, the input cannot be decoupled and thus always has to be considered (recall the analogy of the stick once placed on a table and once on the palm). Consequently the S/Ts may experience at arbitrary times behavioral changes due to input variations, which makes the metastability analysis a lot harder. In consequence, simple timing information are not sufficient any more, meaning that the whole input trajectory has to be processed.

To simplify our analysis, we split the task into two questions and try to answer them separately: (1) How can metastability be entered and (2) how quickly is it resolved? Unfortunately we were not able to answer these in a quantitative fashion yet, i.e., to condense them into a single characteristic number. Nevertheless, we already achieved an accurate description of the single processes, which makes us optimistic that achieving a representative quantity is possible.

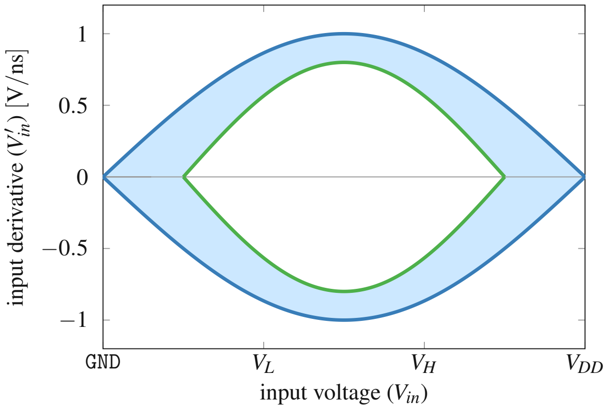


Figure 5.38: Brockett annulus specifying waveforms in the phase space, in this example for the input voltage V_{in}. For each value a region of allowed derivatives (blue area) is specified, which ensures some minimum dynamics but also limits it from above.

5.7.1 Entering Metastability

To gain a proper estimate of the MTBU it is crucial to determine how often the Schmitt Trigger actually experiences metastability. We already argued that only very specific input traces are capable to do that, however, there are still infinitely many possibilities. To better handle their vast amount we decided to use the *Brockett annulus* proposed by Brockett [132] (see Figure 5.38), which represents traces in the phase space. The advantage using this representation is that important boundary conditions, which have to be met at all costs, can be easily specified. Examples are the minimum output dynamics, i.e., to demand a certain slope (prevent stalls) at intermediate values, respectively the maximum input dynamics. The latter limits the rate the phase plane of the S/T can be traversed and thus increases the difficulty to reach γ_2.

Preventing Metastability

Often an S/T is introduced to convert the possible intermediate voltage V_M of a preceding, metastable binary storage element into a clean HI or LO, as done, for example, by Polzer and Steininger [48]. We have already shown that for certain waveforms, e.g. ramps, the S/T is able to filter metastability. Thus limiting the input behavior using a Brockett annulus can actually be used to prevent metastability altogether.

Recall from our simulations of ramps that monotonicity was only important near the threshold voltages. In a typical setting this can be easily assured since the (single!) intermediate output voltage V_M of the previous stage is in general near $V_{DD}/2$. With thresholds $V_L < V_M < V_H$ sufficiently separated from V_M it can be assured, that they are only crossed when metastability is already resolving, i.e., with a steep trajectory (for details see [137, 144]). However, care must be taken that it is indeed the S/T that decides upon the classification of V_M. As soon as any other stage (e.g. a decoupling buffer) is in between the metastability-producing element and the S/T, that element's (single!) input threshold will typically classify V_M in an undesired way. More specifically, glitches can be produced [48], with the S/T having no chance to mitigate these.

Chances of Metastability

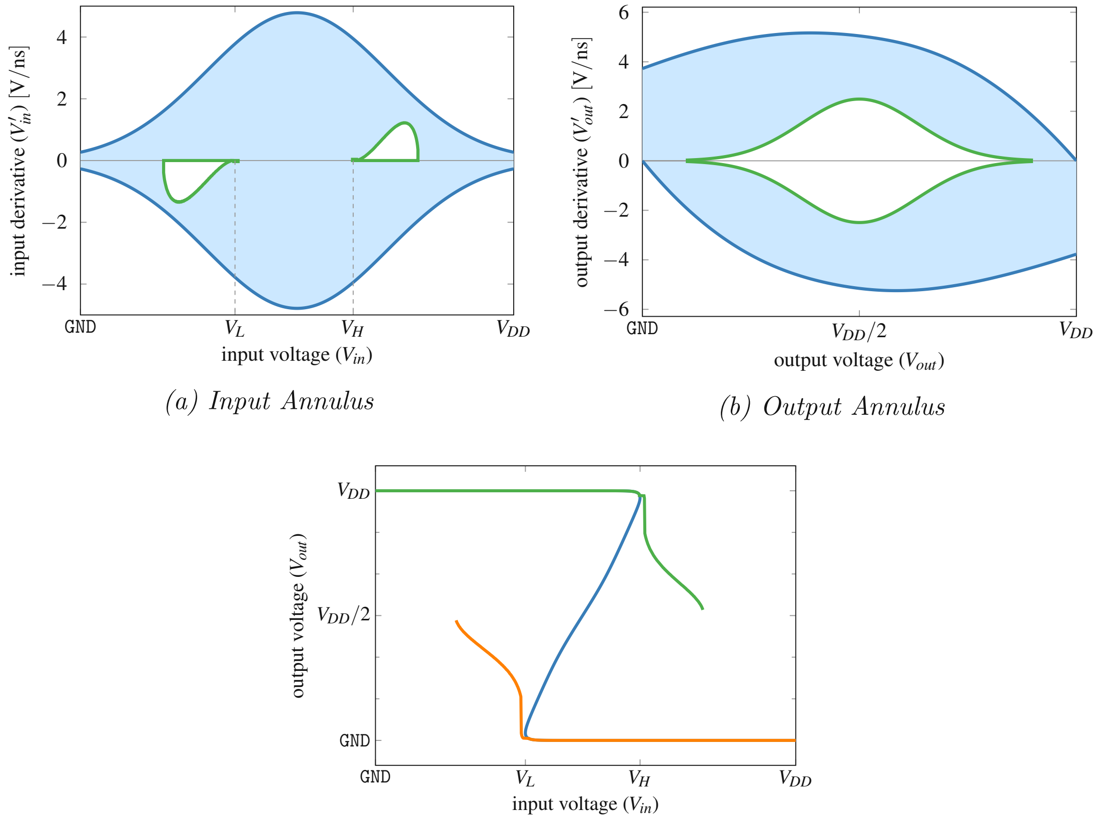
In Section 3.7 we argued that at a specific location in a circuit, only a small subset of all possible analog trajectories can be observed, which makes it possible to model the expected behavior by a rather limited Brockett annulus. This annulus can, in turn, be used to check if metastability is possible at all or not. Just consider the one shown in Figure 5.38. The input is not allowed to stall at intermediate values, i.e., only transitions among values well outside the hysteresis are possible.

Since γ_2 cannot be approached near the threshold, the only possibility left to reach metastability is to revert V_{in} before V_{out} has reached a reasonable value, leading to oscillations around γ_2. This behavior cannot be excluded based on the input annulus alone, since no information about the maximum output derivative is available. For that reason we started a proof-of-concept implementation, that, provided with (i) a phase plane representation of the implementation at hand, (ii) a Brockett annulus describing the maximum input derivative and (iii) a Brockett annulus that specifies the minimum output derivative, is able to determine the allowed range of input signals that satisfy the specified output constraints and, in addition, all achievable output traces.

We will explain the behavior of the algorithm on an example specified by the input annulus shown in Figure 5.39a and the output annulus in Figure 5.39b. Note that initially only the outer boundaries of the input- respectively the inner boundaries of the output annulus are available. While the outer limits of the output annulus are simply derived by determining the maximum resp. minimal V'_{out} for each output value in the phase plane, the restrictions for the inner limits of the input annulus are more evolved. Starting in $(V_{in}, V_{out}) = (\mathtt{GND}, V_{DD})$ (top left corner in Figure 5.39c) we are aiming to derive in the sequel the input restrictions around V_H. The reverse case, leading to the area around V_L, can be achieved analogously.

The behavior for $V_{in} < V_H$ is unproblematic since $V_{out} = V_{DD}$. Once the threshold is crossed we demand that the input derivative stays positive to prevent crossing γ_2. The lower bounds are determined by ensuring the minimum rate of change defined by the output annulus (recall that the output derivative rises in general with distance to γ_2). The worst case switching trajectories achieved in this fashion are shown in Figure 5.39c. Due to the low demands on V'_{out} at $V_{out} \approx V_{DD}$ a steep initial drop can be observed. With decreasing V_{out} the lower bound for V'_{out} increases and thus the minimum input derivative has to be increased. Actually the values we chose in our output annulus turned out to be rather large, requiring quite some distance of the worst case trajectory from γ_2. At last, i.e, after the peak in the boundary of V'_{out} has been passed, we allow an input derivative of zero and discontinue the worst case trajectory.

Although these are already very promising results, we are still lacking a description of how probable it is to derive metastability. For this purpose it would be mandatory to calculate (i) which share of possible trajectories violate the derived boundary conditions and (ii) how probable these are. For reasonable results, the user thus has to precisely



(a) Input Annulus

(b) Output Annulus

(c) Worst case switching traces

Figure 5.39: Results of our approach that determines the possible input behavior to prevent metastability in an S/T while fulfilling specific output requirements. Note that the tool is still under construction and not published yet.

specify the input annulus combined with a statistical expression for (malicious) input trajectories. Otherwise educated assumptions could help to achieve suitable values. Nevertheless, we expect the probability for metastability to be heavily overestimated in this fashion, since violating the boundary condition does not automatically lead to a metastable state.

5.7.2 Resolving Metastability

The other, very important, part when talking about the rate of metastable upsets, is the behavior after metastability has been achieved, i.e., how quickly non-perfect metastable points are resolved (remember that it takes an infinite amount of time to resolve perfect metastability). Naturally the faster this happens the less likely effects are observed in the remaining circuitry. In our analysis we already showed that an S/T leaves metastability following an exponential trace and determined the corresponding time constant τ and the resolution time t_{res} based on analog simulations. Unfortunately following this trajectory

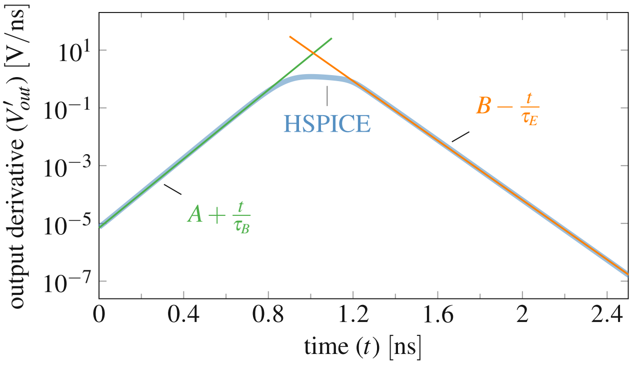


Figure 5.40: Resolving metastability of `std` *for $V_{in} = 0.64\,\mathrm{V}$ using technology (T65). The slopes can be very accurately modeled by an exponential approach.*

until the digitization threshold is reached is not reasonable, since real waveforms slow down as they approach their final value (recall the figures of $1/\tau$ in the $V_{in} - V_{out}$ plane presented earlier). We thus asked ourselves how a more physical modeling of this process can be achieved and what impact this has on the resolution time.

HSPICE simulations (see Figure 5.40) show, as expected, initially an exponential increase of V'_{out} followed by a plateau and an exponential drop. Note that the latter is in good agreement with our analyses in Section 4.5. We denote the rising time constant by τ_B and the falling one as τ_E. Note that some metastable states, especially in the middle of the metastable range, show an over-exponential growth of V'_{out} near the plateau, which renders a description using two exponentials less accurate. This adds to the already crude approximation near the plateau, which causes the resolution time to be underestimated.

5.7.3 Determining region boundaries

Although using two exponential waveforms already improves the situation, since in general only the trajectory out of metastability is considered, the absolute deviations in the resolution time are expected to be negligible. This becomes especially pronounced when comparing it to the possible infinite amount of time it takes to leave metastability. Therefore we will use the exponential fittings in the sequel for determining the value of V_{out} when V'_{out} stops increasing. Note that this is equivalent to the Region boundaries in the phase plane representation of Marino. His calculations lead to straight lines that encapsulate γ_2, while we observed for real world implementations severe variations. In the following we compare this prediction to data gathered from simulations, whereat we utilize two approaches:

1. We run analog HSPICE simulations and determine the value of V_{out} when $|V'_{out}|$ is maximal.

2. We use τ_B and τ_E to determine the voltage value where the single exponentials have to connect while assuring that the first derivative is continuous.

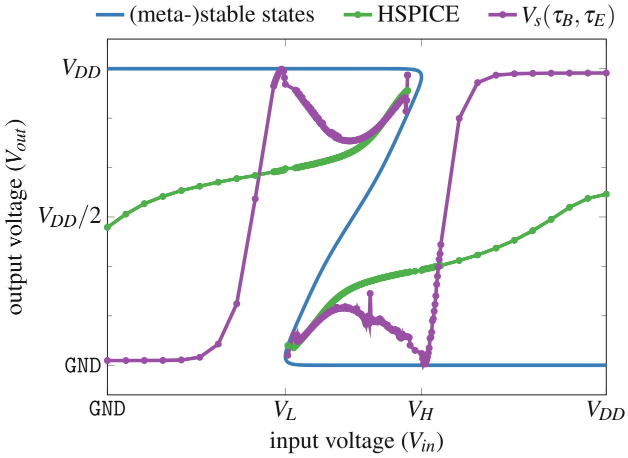


Figure 5.41: Resolution region crossing points ($V''_{out} = 0$) once derived directly from HSPICE simulations and once predicted based on the fitted τ_B and τ_E. Obviously the latter is not yet a viable alternative.

Let us quickly analyze the second approach in greater detail, whereat we will assume a resolution towards V_{DD} (the reverse case can be analyzed analogously). The task is to find $V_s \in [V_M, V_{DD}]$ where the waveform leaving metastability $V_B(t) = V_M + \exp(\frac{t}{\tau_B})$ and the one approaching V_{DD}, i.e., $V_E(t) = V_{DD} - \exp(-\frac{t}{\tau_E})$, have (a) the same value V_s and (b) the same derivative. This point has to exist as the slope of the former increases with time whereat the one of the latter decreases.

We express the properties stated above in mathematical terms by specifying $V_B(t_B) = V_E(t_E) = V_s$ which results in

$$e^{\frac{t_B}{\tau_B}} + e^{-\frac{t_E}{\tau_E}} = V_{DD} - V_M \tag{5.17}$$

$$\frac{1}{\tau_B} \cdot e^{\frac{t_B}{\tau_B}} = \frac{1}{\tau_E} \cdot e^{-\frac{t_E}{\tau_E}} \tag{5.18}$$

Replacing the second term on the left hand side in (5.17) by (5.18) leads to

$$e^{\frac{t_B}{\tau_B}} = (V_{DD} - V_M)\left(\frac{\tau_E}{\tau_B} + 1\right)^{-1} = V_s - V_M$$

where we used in the last step the definition of t_B. Rewriting this finally yields

$$V_s = V_M + \frac{\tau_B}{\tau_E + \tau_B} \cdot (V_{DD} - V_M)\,.$$

With increasing τ_B the value of V_s also increases, which sounds reasonable, as it takes more time to build up a derivative that is matched by $V_E(.)$.

Figure 5.41 shows results derived from HSPICE and our calculations. The achieved accuracy is close to γ_2 quite good, however, quickly derails. We suspect that the main issue is the insufficient approximation near the plateau. A more accurate description of the observed behavior demands, however, further investigations.

CHAPTER 6

Open Problems

Although many results have been presented in this thesis, there are still quite a number of topics available, where some of our efforts primarily revealed promising paths for future research. In this chapter we will introduce some of these shortly.

6.1 Analog Circuit Modeling

We have shown in Section 3.7.4, that (i) input and output trajectories can be fitted by adding-up arbitrary waveforms and that (ii) functions can be found that map input to output waveform parameters. A tool that automatizes this procedure is currently under development. Future research will be mainly devoted to evaluate this approach, i.e., compare performance and accuracy to HSPICE.

Unfortunately, lots of thrilling questions regarding circuit verification remained unanswered due to missing functionality of the C2E2 tool. More specifically, we would be interested in evaluations that determine the trajectories leading to an undesired behavior, such as:

- What are the possible initial input ranges that cause some (forbidden) output voltage range at time t?
- Which input slopes/shapes do not lead to a clean output transition before time t?

Consequently one additional research avenue is the improvement of C2E2. One major enhancement would be to define larger initial uncertainties without experiencing numerical inaccuracies in the future, which would enable more extensive verifications.

6.2 Predicting the Delay Function

The question how the delay function of the Involution Delay Model (IDM) can be determined and/or parameterized quickly and accurately has not been answered in a

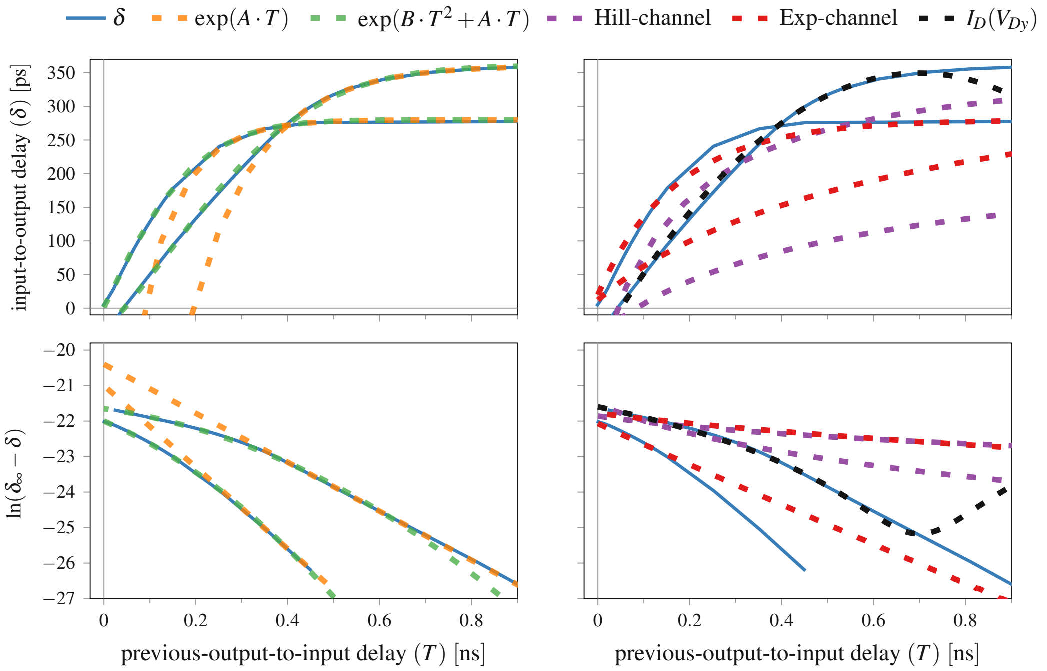


Figure 6.1: Multiple approaches to fit the DDM delay function of an optimal Inverter on a linear (upper) and logarithmic (lower) scale.

satisfying fashion yet. While we primarily focused on calculating the delay function in Chapter 4, it is also possible to approximate the latter by 1) fitting mathematical functions or 2) extrapolating existing numerical results. For both possibilities early results have been derived, which will be shortly presented in the sequel.

6.2.1 Fitting the Delay Function

A straightforward approach is to fit the numerical data to a mathematical function. In this section we thus investigate how well $\delta(T)$ can be approximated using (i) the delay functions of Exp- and Hill-channel introduced in Section 4.3.1 and (ii) arbitrary functions based on educated guessing.

For a systematic analysis we start with an optimal Inverter without real world parasitics, i.e., n- and pMOS with a big load capacitance of 72 fF. Our initial plan is to first investigate DDM, which can be fitted according to Bellido, Juan, and Valencia [82] to an exponential, and then transfer the insights to the IDM. Our simulations, which are depicted in Figure 6.1, are not able to confirm the exponential approximation for DDM in the first place, which is in agreement with the results presented in Section 4.5. While $\delta(T) \approx \delta_\infty$ fits very well, increasing deviations for $T \to 0$ can be observed. Using a polynomial of degree two in the exponent improves the results significantly.

Exp- and Hill-channels lead to even worse results. This is a consequence of the fact that all parameters are shared between $\delta_\uparrow$ and $\delta_\downarrow$. Approximating one leads to an

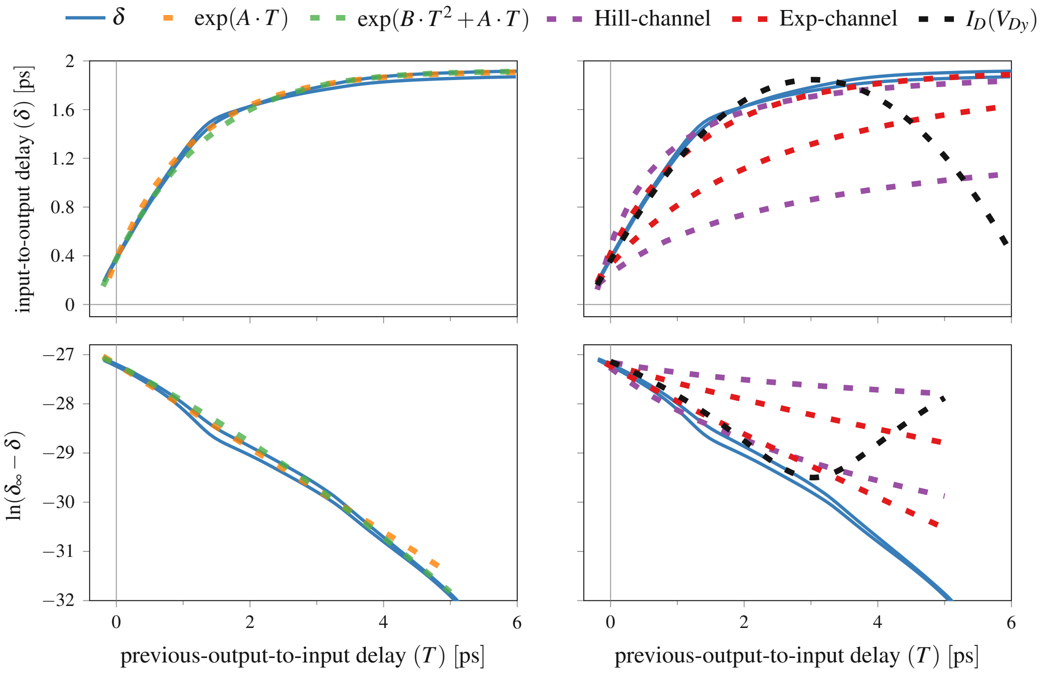


Figure 6.2: Multiple approaches to fit the IDM delay function of a place & routed Inverter using technology (T15) on a linear (upper) and logarithmic (lower) scale. The exponentials on the left side are only fit to one delay function since $\delta_\uparrow$ and $\delta_\downarrow$ differ solely marginally.

increasing deviation for the other and vice versa, as is shown in Figure 6.1. By qualitative observation we, furthermore, recognized a similarity of the delay function to the urrent $I_D(V_{GS}, V_{DS})$ (cf. Section 2.5). Thus we also investigate if the delay can be approximated by the expression which governs the behavior in (OHM) in the Basic Model. In detail $\delta = k \cdot (A - T/B) \cdot T$ was used, which led to good agreement.

Switching to an IDM characterization on more realistic circuit structures, which contain also extracted parasitics, altered the results dramatically (see Figure 6.2): The delay functions in this case behave more like exponentials, which simplifies the fitting. We suspect that this is (partly) a consequence of the large capacitance used earlier. Again the Exp- and Hill-channel can only be optimized for one delay function, rendering the approximation for the other quite inaccurate. The I_D fitting shows initially good agreement but quickly falls off for larger T.

Although these simulations reveal that exponentials, with a polynomial of degree one or two in the exponent, are quite accurate descriptions, we quickly realized that this characterization is not governed by the underlying physical behavior. Consequently we are not able to predict if other or future circuits are going to behave in the same manner. As there is currently no better alternative, Exp- and Hill-channels are still used in the InvTool to approximate the delay functions, despite their rather weak fittings.

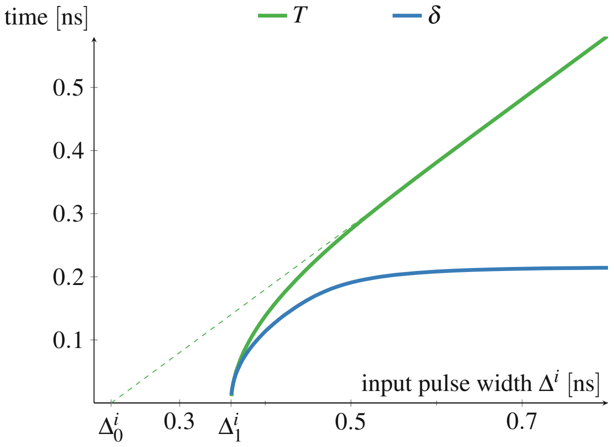


Figure 6.3: T and δ of an Inverter using technology (T65) as functions of the input pulse-width Δ^i. The combined delay function $\delta(T)$ looks similar to the blue curve whereat the lower part is stretched along the x-axis.

6.2.2 Formal Definition

Since simple fitting of $\delta(T)$ leads to quite untargeted and inaccurate results, we conjecture that it might be beneficial to analyze and describe single parameters in the characterization process and then combine their descriptions. One possibility (out of many) is to determine the output pulse-width (Δ^o) and the delay of the first output transition (δ_0) based on the input pulse-width Δ^i, i.e., to use the functions

$$f_1 : \Delta^i \to \Delta^o \qquad \text{and} \qquad f_2 : \Delta^i \to \delta_0$$

In this case, T and δ are given by

$$T = \Delta^i - f_2(\Delta^i) \ , \ \delta = f_2(\Delta^i) + f_1(\Delta^i) - \Delta^i = f_1(\Delta^i) - T.$$

Unfortunately the same problems remain: How can f_1 and f_2 be determined? Figure 6.3 shows simulation results for $T(\Delta^i)$ and $\delta(\Delta^i)$. While the former is changing linearly and the latter is constant for $\Delta^i \gg 0$, both experience severe bendings and a steep decrease during cancellation. Consequently, the combined function $\delta(T)$ is equal to $\delta(\Delta^i)$ for $\Delta^i \gg 0$ while for $\Delta^i \to \Delta^i_1$ it is stretched along the x-axis. This is in accordance to the results presented in Chapter 4.

For an analytic expression we tried to fit T and δ in the most simplistic fashion, i.e., using exponentials, leading to

$$T(\Delta^i) = k(\Delta^i - \Delta^i_0) \cdot \left(1 - e^{-(\Delta^i - \Delta^i_1)/\tau}\right)$$
$$\delta(\Delta^i) = \delta_\infty \left(1 - e^{-(\Delta^i - \Delta^i_1)/\tau}\right)$$

Unfortunately, it is not possible to combine these easily, only sophisticated computer programs are able to calculate $\delta(T)$. The corresponding solution is, however, far too complicated to be printed here and thus also unsuitable for actual computations.

Recall that in Section 4.4 we identified inaccuracies between characterization and actual simulation and proposed a modification that, at least, predicts the output pulse-width correctly. To achieve this we argued that T has to be calculated in reference to a fixed value $\delta_0 = \delta_\infty$, i.e., $T = \Delta^i - \delta_\infty$. Note that in this case there is an implicit solution, namely,

$$\delta(\Delta^i) = \delta_\infty \cdot \left(1 - e^{-(T+\delta_\infty-\Delta^i_1)/\tau}\right) ,$$

which is comparable to the results achieved for the Exp-channel.

6.2.3 Extrapolating the Delay Function

During our research we noticed, that all IDM delay functions look quite similar. After a steep increase for T near $-\delta_{\min}$ the delay starts to settle to the final value δ_∞. Therefore, the question arose, if it is possible to estimate the delay function of an arbitrary gate based on a single, well characterized, numerical set of $\delta_\uparrow$ and $\delta_\downarrow$. If done in a simple yet accurate fashion, this would yield a big improvement in characterization speed.

Unfortunately, simply scaling in the value domain combined with an appropriate time shift does not suffice. Our current approach, thus, utilizes three points that the new delay function has to cross: $\delta_{\min}$, δ_∞ and some value in between. For the latter we are currently utilizing $\delta(0)$ which might, however, turn out to be a bad choice, since $\delta_{\min}$ is in general very small. This implies that $\delta(0) - \delta_{\min} \approx 0$ and thus being heavily affected by numerical inaccuracies.

The task currently at hand is to properly determine $\delta_{\min}$, $\delta(0)$ and δ_∞. The easiest one to get is probably δ_∞, as it is also required in other delay estimation approaches (cf. Section 4.1), and can thus be assumed to be available. The others are, however, much harder to determine. For reasonable results it is mandatory to know the changes introduced by variations of some give parameter, such as input slope, output load or inner structure, which we will shortly investigate in the sequel.

Parameter Influence

Every single parameter has an impact on the resulting involution delay function: (i) Shifting V_{th}^{in*} causes a deviation of V_{th}^{out*} and $\delta_{\min}$ and thus most surely also of δ_∞ and $\delta(0)$. (ii) $\delta_{\min}$ is heavily influenced by the input slope and the coupling capacitances (cf. Section 4.4.3). (iii) δ_∞ naturally depends on the thresholds and the output slope. A crucial ingredient is also the internal structure of a gate, more specifically the relative driving strengths of the transistors. For technology (T65) multiple Inverter implementations with a varying number of transistors in parallel[1] are available, which all have slightly different delay functions. We even were able to verify a significant dependence of $V_{th}^{in*} = V_{th}^{out*} = V_{th}$ on the transistor width, as is shown in Figure 6.4. Unfortunately, a detailed analysis of all parameters and effects is still lacking.

[1] This leads to higher currents and an increased switching speed.

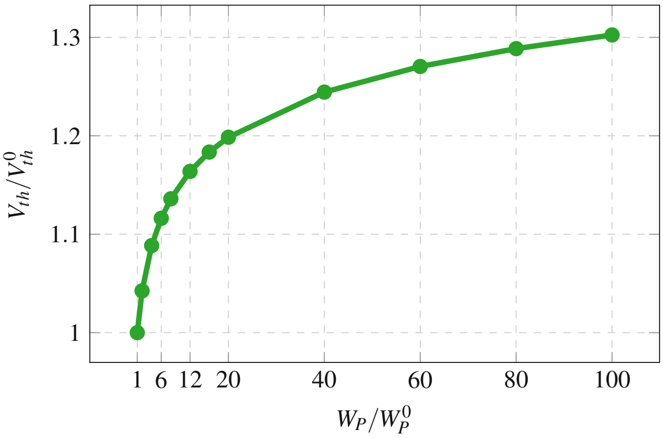


Figure 6.4: Change of the matching threshold voltage $V_{th}^{in*} = V_{th}^{out*} = V_{th}$ by varying the width of the pMOS transistor. The change in width and threshold are related to the original values of width W_P^0 and threshold V_{th}^0.

Prediction using Logical Effort

Predicting the relative variations between single gates is one of the key targets of Logical Effort (LE) developed by Sutherland, Sproull, and Harris [109]. In a nutshell LE provides a simple and fast method to quickly develop circuits by proper fragmentation and transistor scaling. It has, for example, led to the insight that amplifiers work best if they are built in a multi-stage approach, where the amplification increases exponentially.

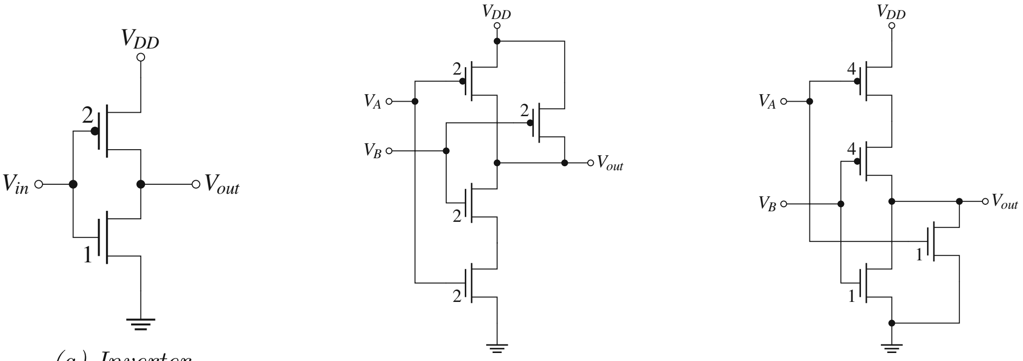
Over the years LE has been extended, e.g. by Lasbouygues et al. [77], who were able to capture couplings among wires, and Kabbani, Al-Khalili, and Al-Khalili [85], who considered in-series connected transistors, internodal capacitances and the effect of the input slope to derive more accurate results. Rahman, Tennakoon, and Sechen [49] added a distinction between rising and falling transition while Morgenshtein et al. [57] considered the RC interconnects to evaluate the optimal delay and minimize the design. LE has even been used in various delay estimations for example by Wang and Markovic [65] who implemented a slope correction, or by Consoli, Giustolisi, and Palumbo [50] who predicted δ_∞ for arbitrary gates.

For its estimations LE analyzes the load and the internal structure of a gate to predict its behavior compared to an Inverter. The overall delay is estimated as $d \cdot \tau$, where τ is the *delay unit* and represents the delay of an Inverter driving another Inverter of the same size. The factor d is calculated as

$$d = gh + p$$

where g represents the *logical effort*, h the *electrical effort* and p the *parasitic delay*.

The logical effort g represents a relation of input transistor sizes that are required to achieve the same current strength as the Inverter. Assuming that (i) nMOS transistors conduct twice the amount of pMOS and (ii) transistors in series only conduct half the amount of a single transistor, the sizings shown in Figure 6.5 are achieved. For the NOR



(a) Inverter

(b) NOR

(c) NAND

Figure 6.5: Transistor sizing of various gates to have the same conductivity as the Inverter. Taken from [109].

gate a logical effort of $g = \frac{4}{3}$ can be calculated while the NAND gate is slightly worse, with $g = \frac{5}{3}$.

The parasitic delay p results from the internal structure of the gate itself. For n-input NOR respectively NAND gates it is not calculated but, instead, a value of $n \cdot p_{inv}$ is assumed [109]. The electrical effort, finally, relates the load capacitance at the output to the input one as

$$h = \frac{C_{out}}{C_{in}}$$

To compare this theoretical predictions to our circuits we run simulations for multiple values of h and determine g and p for a NOR gate using technology (T65). While the delay indeed increased linearly, as predicted, g and p deviate (see Table 6.1). Please note that in our simulations the delay units slightly differed, i.e., $\tau_{\uparrow} = 4.35\,\text{ps}$ and $\tau_{\downarrow} = 4.95\,\text{ps}$, which is a result of the not perfectly matched n- and pMOS.

For future research, it remains to investigate whether the delay function parameters scale according to the predictions of LE. For this purpose extensive analog simulations are required to assure accuracy for all possible variations.

6.3 Multi-Input Delay Channels

The IDM represents circuits internally as zero time boolean gates connected by delay channels. Although it is possible to depict any desired circuit in this fashion, elaborate phenomena like the Charlie Effect (cf. Section 3.4) cannot be covered. The reason is simple: Since delay channels only have a single in- and output no interactions among multiple inputs can be modeled. To also depict these phenomena multi-input delay channels are necessary.

It is a well known fact that gates with multiple transistors in series / in parallel show a more elaborate behavior. Nevertheless, previous research mainly focused on collapsing complex gates to an equivalent Inverter [46, 92, 107, 122]. Although such approaches are

gate	input	g	p
Inverter	up	1	2.78
	down	1	2.28
NOR	A, up	1.39	2.04 p_{inv}
	A, down	2.38	2.57 p_{inv}
	B, up	3.25	1.15 p_{inv}
	B, down	3.60	1.71 p_{inv}

Table 6.1: Extracted LE parameters base on analog simulations using technology (T65). The values differ from the predictions stated in [109].

capable to estimate the static delay δ_∞, they are not well suited to investigate interference among inputs, simply because the gate is, again, reduced to a single in- and output. Thus, we have to follow a different route for future research. The results derived for the CIDM are, in this regard, very encouraging, as we managed to include the gate into the CIDM channel.

A big challenge with multi-input gates are the internal nodes, which can become floating, i.e., neither connected directly to `GND` nor V_{DD}. Thus their value at the beginning of a transition is unknown and may vary considerably, which has a big impact on the delay (as was also mentioned by Shoji [134]). This is especially important during characterization, where it becomes necessary to pin the initial voltage of these internal nodes to an appropriate value.

Preliminary results revealed already significant differences. In contrast to $\delta_\uparrow$ and $\delta_\downarrow$ of an Inverter, a two input gate is characterized by eight delay functions, which all represent one specific input state transition (single input switch only; see Figure 6.6). For a faithful model these have to be carefully interconnected such that proper cancellation is assured. In addition the gathered delay functions have (partially) an extraordinary shape, e.g., $\delta(T) < 0$ for $T \gg 0$. This makes the task even more challenging.

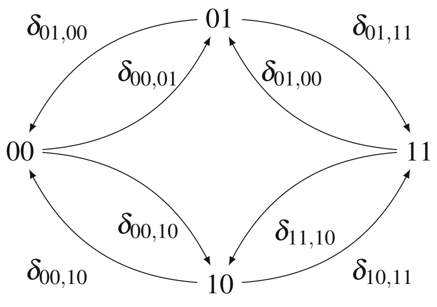


Figure 6.6: Interconnections of delay functions of Multi Input gates.

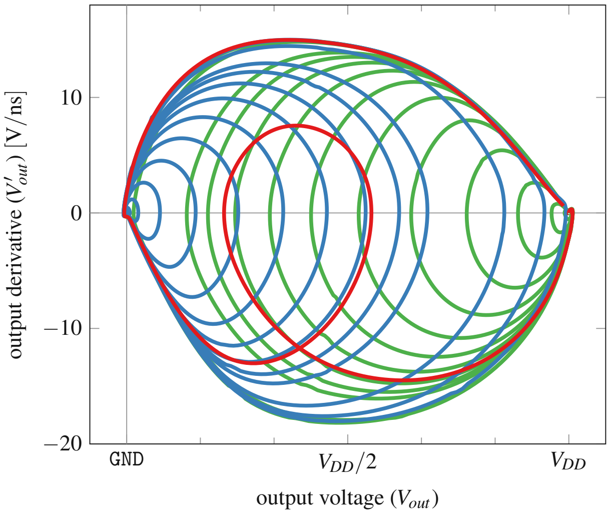


Figure 6.7: Phase space representation of pulses applied to a Buffer using technology (T65). Shown are up-pulse (blue), down-pulses (green) and a single trace consisting of four transitions (red), which was used to evaluate the approach. By picking reasonable switching points the latter can be approximated closely.

6.4 Second Order Extension

A possible extension of single history delay models is to increase the order of the approach. This can be achieved in various fashions, e.g., by considering the last two output transitions instead of just the last one. In the analog channel model of the IDM $f_\uparrow$ and $f_\uparrow$ are switched instantaneously, which is far from physical reality. By considering the last two output transitions it would be possible to continue V_{out} and V'_{out} smoothly, which would solve, besides others, the inaccuracy issues discussed in Section 4.4.1. In the sequel we shortly sketch how such a model could be realized.

6.4.1 Data Representation

To develop an approach that assures a continuation of V_{out} and V'_{out}, it is beneficial to represent analog trajectories in the phase space, since switching between rising and falling waveforms in this plane satisfies both demanded properties. HSPICE simulation results for single pulses applied to a Buffer are shown in Figure 6.7. Note that all possible traces lie within a region that is limited by the Full-Range Switching Waveforms (FRSWs) as those experience the highest respectively lowest slopes. Despite this rather constrained space, infinitely many pulses are possible: In fact every coordinate is hit by a unique up- and down-pulse, which implies, that switching from a rising to a falling transition and reverse is possible at every location in the phase space respectively point in time.

Obviously the future trajectory depends on when resp. where the present trajectory is left. This is a fundamental difference to the first order model, which just utilizes the outermost full-range trajectories and switches between these instantaneously. As already explained this preserves the output voltage but leads to a jump in the first derivative.

The main challenge for deriving suitable analog waveforms from the phase space is the determination of proper switchover points, i.e., when the present trajectory is left and the one in the opposite direction is followed. Utilizing V_{th}^{in} crossing times delivered unsatisfying results. To improve the estimations we even considered the slope of the input, however, with negligible success. Careful observation of Figure 6.7 finally revealed, that the output starts to deviate from the FRSW very early, which suggest to utilize a two threshold model, i.e., $V_{LO} \neq V_{HI}$. We are currently investigating this possibility.

To evaluate the achievable accuracy of an approach based on single pulses, we simulate a random trace consisting of multiple transitions and plot it in the phase space (see red curve in Figure 6.7). It turns out, that picking appropriate switching points, which we determined in this case by hand, the analog waveform can be approximated very closely. Larger deviations are only observed for traces that oscillate around intermediate values. An example is the small loop shown in Figure 6.7. For a good fit in the phase space, we actually have to follow the down-pulse further before switching to the respective up-pulse, although the desired analog value already deviates significantly. This, however, results in an overestimation of the slope and thus a shift in the time domain, which reveals one major shortcoming of this representation: time is used implicitly, which makes it hard to properly predict the effect of inaccuracies.

6.4.2 Analytic Calculation

While first order involution models are associated to an RC network (equivalent to an Exp-channel) the second order one corresponds to an RCRC network, which can also be evaluated analytically. In particular the general second order differential equation

$$f''(t) = A \cdot f'(t) + B \cdot f(t) + C \tag{6.1}$$

can be used to describe the behavior of the system. Solving for $f(t)$ leads to

$$f(t) = \alpha_1 \cdot e^{-\frac{t}{\tau_1}} + \alpha_2 \cdot e^{-\frac{t}{\tau_2}} + \alpha_3 \ . \tag{6.2}$$

Note that such a representation is versatile, as we have already shown in Section 3.7.4, where rising and falling switching waveforms where utilized to model output pulses.

For the second order model we are searching for a method to continue the analog output waveform from a given initial condition $(V_{out}, V'_{out}) = (v, k)$. Using $f(0) = v$ and $f'(0) = k$ in (6.2) leads to

$$f(t) = -\frac{\tau_1(v + \tau_2 k - \alpha_3)}{\tau_2 - \tau_1} \cdot e^{-\frac{t}{\tau_1}} + \frac{\tau_2(v + \tau_1 k - \alpha_3)}{\tau_2 - \tau_1} \cdot e^{-\frac{t}{\tau_2}} + \alpha_3.$$

Note that α_3 determines the final value of the waveform and thus the shape, as is shown in Figure 6.8.

The results we achieved so far are very promising, however, not yet fully satisfactory. One open challenge is to determine τ_1 respectively τ_2 such that the analog waveform

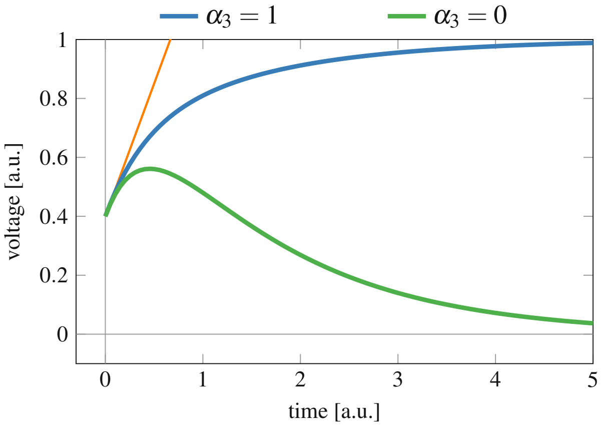


Figure 6.8: Continuation from an initial point (v, k) with $v \approx 0.4$ and $k > 0$ (orange line) depending on the final value α_3. In one case the trajectory keeps its current direction, while in the other a significant turning can be observed.

is closely approximated. Without loss of generality we assume $\tau_1 > \tau_2$ in the following. With decreasing τ_2 on the one hand steeper transitions are derived, but on the other hand also less voltage gain when reverting the direction. With further dropping τ_2 eventually a first order Exp-channel is achieved.

Having fixed the values τ_1 and τ_2, and thus the time until $V'_{out} = 0$ is achieved, it is still not clear how to pick an appropriate switching time. For future research we are therefore planning to check possible approximations or add additional parameters that allow a proper parametrization and reasonable estimations for all parameters.

6.4.3 Condensing

The analog output value v of a gate at time t_0 depends on the complete input trajectory for $t < t_0$. Since it is neither possible nor feasible to store and consider all of it during delay estimation it is mandatory to apply some form of condensing. In the first order model this was achieved by deriving, based on the previous output and the current input transition time, the succeeding V_{th}^{out} crossing, which represents the state of the channel.

For the second order model various approaches are possible. Based on HSPICE simulations we investigate m-to-n input condensing, i.e., to represent the current and future output behavior by the response to n input transition, based on the arrival times of the last m input transitions. All examined variation, i.e., 2-to-1, 3-to-1 and 3-to-2, delivered reasonable results. We did, however, not yet succeed in deriving suitable mapping functions.

6.5 Non-determinism

Future work will also be devoted to increase the level of sustainable non-determinism by relaxing the introduced bounds. Recall that the stringent conditions we achieved in our considerations are required to assure faithfulness at all costs. Nevertheless, those bounds are derived for $T \in [-\delta_{\min}, 0]$ so it might be possible to relax them in all other regions.

Finally we are aiming to include the $\boldsymbol{\eta}$-involution model in a state-of-the-art formal verification tool. Optimally, this allows to verify the proper operation for the projected circuit lifetime in one simulation run.

6.6 Metastability Modeling

In Section 5.7 we described first steps towards calculating a Mean Time Between Upsets for a Schmitt Trigger. The main challenge remaining is a concrete implementation in a state of the art verification framework, comparably to [69], where the behavior of an Arbiter was verified in almost the same fashion. Our primary goal is to transform an input annulus into a suitable output one, and vice versa, using verified methods that guarantee correctness. Hopefully, this even allows us to derive statements about the probabilities for respectively the absence of metastability.

A very interesting question we were not able to address yet is how a Schmitt Trigger (S/T) can be made more resilient against metastable upsets. To identify the crucial circuit components we are planning to perform small signal analysis (cf. Section 2.5.4). The corresponding model for the `std` implementation is shown in Figure 6.9. Identifying the bottlenecks would allow us to increase the electric current and, in consequence, to improve the resolution speed. Since we argued that higher V'_{out} also results in less traces to reach γ_2, this would decrease simultaneously the chances for metastability altogether.

In addition the small signal representation might be used to calculate characteristic values that we currently extract from extensive analog simulations (cf. Section 5.4). For $v_{in} = 0$, which significantly simplified the model, we are optimistic to derive results for I_{out} and consequently for the resolution time constant τ in deep metastability.

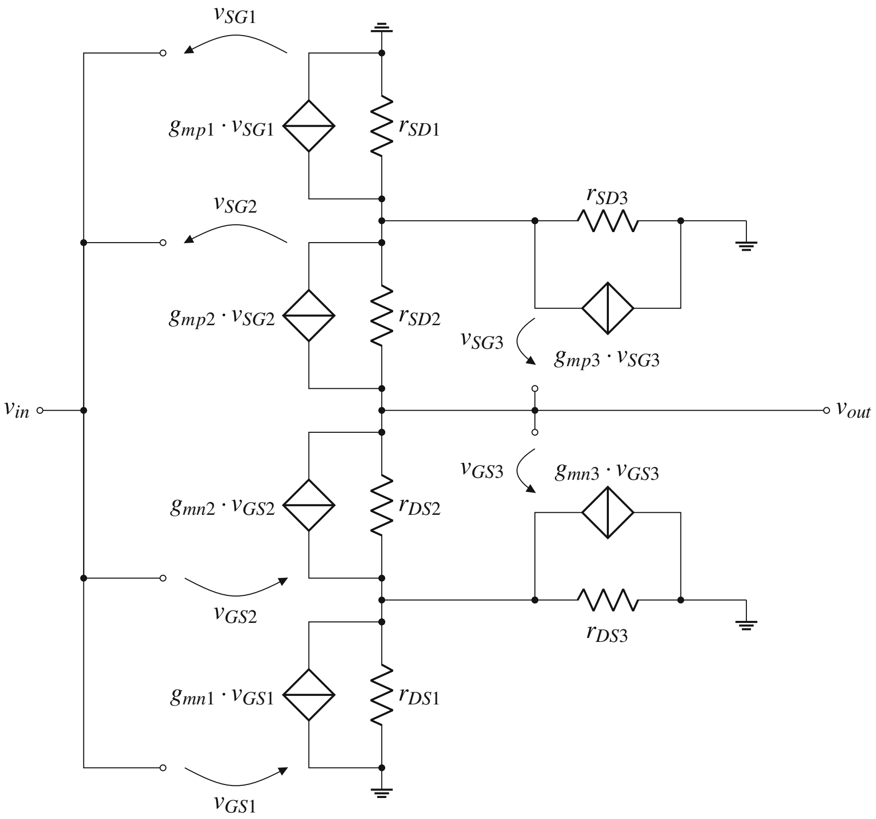


Figure 6.9: Small signal representation of the `std` *S/T implementation. Such a model may become useful for future research such as calculating the resolution constant or optimizing an implementation.*

CHAPTER 7

Conclusion

In this thesis, we showed that finding efficient and accurate abstractions for digital circuits, which is a mandatory task in modern circuit design, can be properly guided by investigating the physical processes governing the behavior in the analog domain. Since the latter is already very well understood, the main challenge nowadays is to identify the crucial model parameters that are required to faithfully capture the desired circuit properties. As our research revealed, no “silver bullet” that would fit all tasks can be provided. Instead, each case has to be evaluated separately.

In our analyses, we first focused on describing trajectories in the analog domain. Based on simplified transistor models, the behavior of standard gates was approximated and verified by simulations. Although the gathered results were quite accurate, the achieved evaluation time improvements were not significant enough to enable an analysis of large circuits in reasonable time. We, therefore, further abstracted the real trajectories by properly combining unique rising and falling full-range switching waveforms, whereat a simple addition turned out to be sufficient to model pulses. To speed up characterization, we investigated alternative approaches for finding suitable switching waveforms: Since calculations turned out to be infeasible, we resorted to fitting mathematical functions to HSPICE simulations. Although our results are very promising, an implementation and evaluation of a corresponding analog simulation tool (based on few distinct parameters that are propagated through a circuit), still needs to be done.

Regarding digital abstractions, we investigated the Involution Delay Model (IDM), whose purpose is to predict the retardations of a signal as it propagates through a circuit. A thorough analysis of the characterization process allowed us to reveal and/or explain specific (un)favorable properties of the existing model, e.g., the stringent conditions on the discretization threshold voltages, used to digitize analog waveforms, or the causes for pure delay. Based on these insight, we then proposed extensions and generalizations that improve the power and applicability of the IDM. More specifically, we succeeded to calculate the delay functions based on our simplified transistor models, which is a first step towards a quick and easy characterization. Furthermore, the development of the

InvTool, which runs delay simulations by utilizing existing tool infrastructure, alleviated the evaluation of an actual circuit. We used this tool to show the capabilities of the IDM in comparison to state-of-the-art delay estimation methods, whereat our simulations confirmed faithful and accurate predictions also for more elaborate gates. In addition, we were able to prove that non-deterministic delays and even acausal delay channels can be added without damaging the faithfulness property. While the former is, in our opinion, interesting for formal verification, the latter allows to relax the limitations on the threshold voltages, such that circuit analyses are considerably simplified.

Finally we turned our attention towards evaluation the metastable behavior of the Schmitt Trigger in detail. We provided various novel, simple and accurate characterization methods based on HSPICE simulations and collected them in a tool called MEAT. It allows a completely automatic evaluation and is thus a big step towards analyzing and comparing Schmitt Trigger circuits. The application to three modern implementations revealed surprising differences among each other and to theoretical predictions. We further showed, by simulation, how a Schmitt Trigger can be driven into metastability. Although essentially arbitrary trajectories are possible at the output, very precise analog input control is required for this purpose. Consequently, the chances of observing a metastable Schmitt Trigger in a real application are rather low. Following the example of synchronizers, we also investigated the effects of cascading two Schmitt Triggers, which decreased the chances for metastability but also revealed several unfavorable properties. Finally we introduced, for the first time, a promising path to deriving a characteristic number that quantifies the susceptibility of a Schmitt Trigger towards metastability. As it requires to also take the input waveform into account, this is a much more challenging task compared to, for example, the Latch.

Although many different aspects have been discussed in this thesis, there is still a lot to be done in the future. One of the most pressing issues is the development of a proper digital delay model extension for multi-input gates, which are currently covered only insufficiently. The main reason is input interference, which is impossible to describe using the single input-single output channels employed in the Involution Delay Model at the moment. Even a further increase in modeling accuracy, e.g., by basing the output predictions on multiple preceding transitions, is imaginable. For the Schmitt Trigger, we conjecture that some of the characteristic values could be derived analytically, which would render expensive analog simulations unnecessary.

List of Figures

List of Tables

Glossary

(OHM)

Operation region of the transistor where the output current depends linearly on the applied voltage.

(SAT)

Operation region of the transistor where the output current changes only mildly (may differ significantly for newer technologies).

(ST)

Operation region of the transistor where hardly any current is conducted, i.e., the sub-threshold regime.

Basic Model

Transistor model based on very basic equations designed for the earliest transistors.

C2E2

Compare Execute Check Engine. A simulation and verification suite using discrepancy functions. For details see https://publish.illinois.edu/c2e2-tool.

Elaborate Model

More elaborate transistor model considering several short channel effects.

Genus

Synthesis tool developed by Cadence. For this thesis version 19.11 has been used.

HSPICE

Analog simulation suite for electronic circuits and systems developed by Synopsys. For this thesis various versions have been used over time.

Innovus

Place & route tool developed by Cadence. For this thesis various versions have been used over time.

InvTool

A tool written to simulate a circuit in the analog and digital domain (using the IDM approach) and to evaluate the achieved results. Available open source at https://github.com/oehlinscher/InvolutionTool.

MACS

MAtlab Circuit Simulations: A tool written in Matlab that allows circuit simulations and export to verfication tools. Available open source under https://github.com/jmaier0/macs.

MATLAB

Mathematical framework program developed by MathWorks that we used for various calculations and simulations in version R2016b.

MEAT

MEtastability Analysis Tool: A tool written to characterize the metastable region of an S/T. Available open source at https://github.com/jmaier0/meat.

ModelSim

Industrial digital simulation suite for Hardware Description Languages by Mentor. For this thesis version 10.5c has been used.

Spectre

Analog simulation suite for electronic circuits and systems developed by Cadence. For this thesis version 15.1.0.627.isr12 has been used.

Uniform Model

Uniform transistor model that describes the transistor behavior with a single equation.

VHDL

Very High Speed Integrated Circuit Hardware Description Language: A programming language used to develop hardware.

Acronyms

nMOS
n-channel Metal Oxide Semiconductor Transistor

pMOS
p-channel Metal Oxide Semiconductor Transistor

CB
Conduction Band

CIDM
Composable Involution Delay Model

CMOS
Complementary Metal Oxide Semiconductor Technology

DDM
Degradation Delay Model

FET
Field Effect Transistor

FRSW
Full-Range Switching Waveform

IC
Integrated Circuit

IDM
Involution Delay Model

LE
Logical Effort

MIS
Multi Input Switching

MIS-FET
Metal Insulator Semiconductor-Field Effect Transistor

MOS-FET
Metal Oxide Semiconductor-Field Effect Transistor

MTBU
Mean Time Between Upsets

ODE
Ordinary Differential Equation

S/T
Schmitt Trigger

SCE
Short Channel Effect

SCR
Space Charge Region

SIS
Single Input Switching

TCT
Threshold Crossing Times

VB
Valence Band

WST
Waveform Switching Times

Bibliography

[1] Chuchu Fan, Yu Meng, Jürgen Maier, Ezio Bartocci, Sayan Mitra, and Ulrich Schmid. „Verifying nonlinear analog and mixed-signal circuits with inputs“. In: *IFAC-PapersOnLine* 51.16 (2018). 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018, pp. 241–246. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.08.041. URL: http://www.sciencedirect.com/science/article/pii/S2405896318311571.

[2] Jürgen Maier. *Modeling the CMOS Inverter using Hybrid Systems*. Tech. rep. TUW-259633. E182 - Institut für Technische Informatik; Technische Universität Wien, 2017. URL: http://publik.tuwien.ac.at/files/publik_259633.pdf.

[3] J. Maier, M. Függer, T. Nowak, and U. Schmid. „Transistor-Level Analysis of Dynamic Delay Models“. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019, pp. 76–85. DOI: 10.1109/ASYNC.2019.00019.

[4] Jürgen Maier, Daniel Öhlinger, Ulrich Schmid, Matthias Függer, and Thomas Nowak. „A Composable Glitch-Aware Delay Model“. In: *Proceedings of the 2021 on Great Lakes Symposium on VLSI*. GLSVLSI ’21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 147–154. ISBN: 9781450383936. DOI: 10.1145/3453688.3461519. URL: https://doi.org/10.1145/3453688.3461519.

[5] M. Függer, J. Maier, R. Najvirt, T. Nowak, and U. Schmid. „A faithful binary circuit model with adversarial noise“. In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2018, pp. 1327–1332. DOI: 10.23919/DATE.2018.8342219.

[6] Jürgen Maier. „Gain and Pain of a Reliable Delay Model“. In: *2021 24th Euromicro Conference on Digital System Design (DSD)*. 2021, pp. 246–250. DOI: 10.1109/DSD53832.2021.00046.

[7] Daniel Öhlinger, Jürgen Maier, Matthias Függer, and Ulrich Schmid. „The Involution Tool for Accurate Digital Timing and Power Analysis“. In: *Integration* 76 (2021), pp. 87–98. ISSN: 0167-9260. DOI: https://doi.org/10.1016/j.vlsi.2020.09.007. URL: http://www.sciencedirect.com/science/article/pii/S0167926020302777.

[8] D. Öhlinger, J. Maier, M. Függer, and U. Schmid. „The Involution Tool for Accurate Digital Timing and Power Analysis“. In: *2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. July 2019, pp. 1–8. DOI: 10.1109/PATMOS.2019.8862165.

[9] J. Maier and A. Steininger. „Efficient Metastability Characterization for Schmitt-Triggers“. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019, pp. 124–133. DOI: 10.1109/ASYNC.2019.00024.

[10] J. Maier, C. Hartl-Nesic, and A. Steininger. *Comprehensive Characterization of Schmitt-Triggers*. submitted to TCAS I June'21.

[11] A. Steininger, J. Maier, and R. Najvirt. „The Metastable Behavior of a Schmitt-Trigger“. In: *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2016, pp. 57–64. DOI: 10.1109/ASYNC.2016.19.

[12] A. Steininger, R. Najvirt, and J. Maier. „Does Cascading Schmitt-Trigger Stages Improve the Metastable Behavior?“ In: *2016 Euromicro Conference on Digital System Design (DSD)*. Aug. 2016, pp. 372–379. DOI: 10.1109/DSD.2016.56.

[13] Joao Canas Ferreira. *Physical synthesis with Encounter (Cadence)*. https://paginas.fe.up.pt/~jcf/ensino/disciplinas/mieec/pcvlsi/2015-16/tut_encounter/tut_encounter.html. 2015/16.

[14] http://electronics.stackexchange.com/questions/174079/do-schmitt-trigger-specs-give-requirements-to-avoid-metastability. 2015/11/22.

[15] *Semiconductors on NSM*. URL: http://www.ioffe.ru/SVA/NSM/Semicond/index.html (visited on 06/17/2021).

[16] Ulrich Tietze, Christoph [VerfasserIn] Schenk, and Eberhard [VerfasserIn] Gamm. *Halbleiter-Schaltungstechnik*. 15., überarbeitete und erweiterte Auflage. Berlin: Springer Vieweg. ISBN: 9783662483541.

[17] Matthias Függer, Robert Najvirt, Thomas Nowak, and Ulrich Schmid. „A Faithful Binary Circuit Model“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (Aug. 2019). DOI: 10.1109/TCAD.2019.2937748.

[18] Stephan Friedrichs, Matthias Függer, and Christoph Lenzen. „Metastability-Containing Circuits“. In: *IEEE Transactions on Computers* 67.8 (2018), pp. 1167–1183. DOI: 10.1109/TC.2018.2808185.

[19] Daniel Öhlinger. *Involution Tool*. Tech. rep. TUW-278633. E191 - Institut für Computer Engineering; Technische Universität Wien, 2018. URL: https://publik.tuwien.ac.at/files/publik_278633.pdf.

[20] J. Reiher, M. R. Greenstreet, and I. W. Jones. „Explaining Metastability in Real Synchronizers“. In: *2018 24th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2018, pp. 59–67. DOI: 10.1109/ASYNC.2018.00024.

[21] L. Valavala, K. Munot, and K. B. R. Teja. „Design of CMOS Inverter and Chain of Inverters Using Neural Networks“. In: *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*. Dec. 2018, pp. 269–274. DOI: 10.1109/iSES.2018.00065.

[22] J.C. Whitaker. *Microelectronics*. Electronics Handbook Series. CRC Press, 2018. ISBN: 9781420037593.

[23] Matthias Althoff and Dmitry Grebenyuk. „Implementation of Interval Arithmetic in CORA 2016“. In: *ARCH Workshop*. 2016, pp. 91–105.

[24] C. E. Calosso and E. Rubiola. „Phase Noise and Jitter in Digital Electronics“. In: *arXiv:1701.00094* (2016). URL: https://arxiv.org/pdf/1701.00094.

[25] *CCS Timing Library Characterization Guidelines*. Version 3.4. Synopsis Inc. Oct. 2016.

[26] Chuchu Fan, Bolun Qi, Sayan Mitra, Mahesh Viswanathan, and Parasara Sridhar Duggirala. „Automatic Reachability Analysis for Nonlinear Hybrid Models with C2E2“. In: *CAV*. Springer. 2016, pp. 531–538.

[27] Matthias Függer, Thomas Nowak, and Ulrich Schmid. „Unfaithful Glitch Propagation in Existing Binary Circuit Models“. In: *IEEE Transactions on Computers* 65.3 (Mar. 2016), pp. 964–978. ISSN: 0018-9340. DOI: 10.1109/TC.2015.2435791.

[28] *HSPICE® User Guide: Basic Simulation and Analysis*. Version L-2016.06. Synopsys®. June 2016.

[29] T. Polzer and A. Steininger. „A general approach for comparing metastable behavior of digital CMOS gates“. In: *2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*. 2016, pp. 1–6. DOI: 10.1109/DDECS.2016.7482456.

[30] B.G. Streetman and S. Banerjee. *Solid State Electronic Devices*. global seventh edition. Pearson Education Limited, 2016. ISBN: 9781292060552.

[31] Yannis Tsividis and Colin McAndrew. *The MOS transistor*. International Third ed. McGraw-Hill, 2016. ISBN: 978-0-19-809737-2.

[32] Yogesh Singh Chauhan, Darsen D. Lu, Sriramkumar Vanugopalan, Sourabh Khandelwal, Juan Pablo Duarte, Navid Paydavosi, Ali Niknejad, and Chen-Ming Hu. *FinFET Modeling for IC Simulation and Design: Using the BSIM-CMG Standard*. English. United States: Elsevier Inc., Feb. 2015. ISBN: 9780124200319. DOI: 10.1016/C2013-0-06812-0.

[33] *Effective Current Source Model (ECSM) Timing and Power Specification*. Version 2.1.2. Cadence Design Systems. Jan. 2015.

[34] Matthias Függer, Robert Najvirt, Thomas Nowak, and Ulrich Schmid. „Towards Binary Circuit Models That Faithfully Capture Physical Solvability“. In: *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 1455–1460.

[35] *HiSIM 2.9.0 User's Manual*. 2.9.0. http://home.hiroshima-u.ac.jp/usdl/HiSIM2/Open_Source/protect.cgi, accessed: 2016-05-02. Hiroshima University & STARC. Oct. 2015.

[36] Soonho Kong, Sicun Gao, Wei Chen, and Edmund Clarke. „dReach: δ-Reachability Analysis for Hybrid Systems“. In: *TACAS*. 2015, pp. 200–205.

[37] Mayler Martins, Jody Maick Matos, Renato P. Ribas, André Reis, Guilherme Schlinker, Lucio Rech, and Jens Michelsen. „Open Cell Library in 15Nm FreePDK Technology“. In: *Proceedings of the 2015 Symposium on International Symposium on Physical Design*. ISPD '15. Monterey, California, USA: ACM, 2015, pp. 171–178. ISBN: 978-1-4503-3399-3. DOI: 10.1145/2717764.2717783. URL: http://doi.acm.org/10.1145/2717764.2717783.

[38] Robert Najvirt, Ulrich Schmid, Michael Hofbauer, Matthias Függer, Thomas Nowak, and Kurt Schweiger. „Experimental Validation of a Faithful Binary Circuit Model“. In: *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*. GLSVLSI '15. Pittsburgh, Pennsylvania, USA: ACM, 2015, pp. 355–360. ISBN: 978-1-4503-3474-7. DOI: 10.1145/2742060.2742081. URL: http://doi.acm.org/10.1145/2742060.2742081.

[39] P. Chaourani, I. Messaris, N. Fasarakis, M. Ntogramatzi, S. Goudos, and S. Nikolaidis. „An analytical model for the CMOS inverter“. In: *2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. Sept. 2014, pp. 1–6. DOI: 10.1109/PATMOS.2014.6951894.

[40] P. Chaourani and S. Nikolaidis. „A unified CMOS inverter model for planar and FinFET nanoscale technologies“. In: *17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*. Apr. 2014, pp. 242–245. DOI: 10.1109/DDECS.2014.6868799.

[41] Matthias Függer, Robert Najvirt, Thomas Nowak, and Ulrich Schmid. „Faithful Glitch Propagation in Binary Circuit Models“. In: *arXiv:1406.2544* (2014). URL: http://arxiv.org/abs/1406.2544.

[42] Himanshu Gupta and Bahniman Ghosh. „Transistor size optimization in digital circuits using ant colony optimization for continuous domain“. In: *International Journal of Circuit Theory and Applications* 42.6 (2014), pp. 642–658. DOI: https://doi.org/10.1002/cta.1879. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cta.1879.

[43] F.S. Marranghello, A.I. Reis, and R.P. Ribas. „CMOS inverter analytical delay model considering all operating regions“. In: *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*. June 2014, pp. 1452–1455. DOI: 10.1109/ISCAS.2014.6865419.

[44] A. Rjoub and A. Ahmad. „Fast modeling technique for nano scale CMOS inverter and propagation delay estimation“. In: *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2014 24th International Workshop on.* Sept. 2014, pp. 1–4. DOI: 10.1109/PATMOS.2014.6951891.

[45] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. „Flow*: An analyzer for non-linear hybrid systems“. In: *CAV.* 2013, pp. 258–263.

[46] L. Ding, J. Wang, Z. Huang, A. Kurokawa, and Y. Inoue. „An analytical model of the overshooting effect for multiple-input gates in nanometer technologies“. In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013).* May 2013, pp. 1712–1715. DOI: 10.1109/ISCAS.2013.6572194.

[47] Johan Perols, Carsten Zimmermann, and Sebastian Kortmann. „On the relationship between supplier integration and time-to-market“. In: *Journal of Operations Management* 31.3 (2013), pp. 153–167. ISSN: 0272-6963. DOI: https://doi.org/10.1016/j.jom.2012.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0272696312000812.

[48] T. Polzer and A. Steininger. „SET propagation in micropipelines“. In: *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2013 23rd International Workshop on.* Sept. 2013, pp. 126–133. DOI: 10.1109/PATMOS.2013.6662165.

[49] M. Rahman, H. Tennakoon, and C. Sechen. „Library-Based Cell-Size Selection Using Extended Logical Effort“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.7 (July 2013), pp. 1086–1099. ISSN: 0278-0070. DOI: 10.1109/TCAD.2013.2247657.

[50] E. Consoli, G. Giustolisi, and G. Palumbo. „An Accurate Ultra-Compact I-V Model for Nanometer MOS Transistors With Applications on Digital Circuits“. In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 59.1 (Jan. 2012), pp. 159–169. ISSN: 1549-8328. DOI: 10.1109/TCSI.2011.2158704.

[51] Branko L. Dokic. „CMOS and BiCMOS Regenerative Logic Circuits“. In: *Cutting Edge Research in New Technologies.* Ed. by Constantin Volosencu. Rijeka: IntechOpen, 2012. Chap. 2. DOI: 10.5772/34934. URL: https://doi.org/10.5772/34934.

[52] Michael Hofbauer, Kurt Schweiger, Horst Dietrich, Horst Zimmermann, Kay-Obbe Voss, Bruno Merk, Ulrich Schmid, and Andreas Steininger. „Pulse Shape Measurements by On-chip Sense Amplifiers of Single Event Transients Propagating through a 90 nm Bulk CMOS Inverter Chain“. In: *IEEE Transactions on Nuclear Science* 59.6 (Dec. 2012), pp. 2778–2784.

[53] R. Ginosar. „Metastability and Synchronizers: A Tutorial“. In: *IEEE Design & Test of Computers* 28.5 (2011), pp. 23–35.

[54] S. Beer, R. Ginosar, M. Priel, R. Dobkin, and A. Kolodny. „The Devolution of Synchronizers“. In: *IEEE International Symposium on Asynchronous Circuits and Systems.* IEEE, 2010, pp. 94–103.

[55] A. Grafton, G.W. Most, and S. Settis. *The Classical Tradition*. Harvard University Press reference library. Harvard University Press, 2010. ISBN: 9780674035720. URL: https://books.google.at/books?id=LbqF8z2bq3sC.

[56] Z. Huang, A. Kurokawa, M. Hashimoto, T. Sato, M. Jiang, and Y. Inoue. „Modeling the Overshooting Effect for CMOS Inverter Delay Analysis in Nanometer Technologies“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29.2 (Feb. 2010), pp. 250–260. ISSN: 0278-0070. DOI: 10.1109/TCAD.2009.2035539.

[57] A. Morgenshtein, E. G. Friedman, R. Ginosar, and A. Kolodny. „Unified Logical Effort - A Method for Delay Evaluation and Minimization in Logic Paths With RC Interconnect“. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 18.5 (May 2010), pp. 689–696. ISSN: 1063-8210. DOI: 10.1109/TVLSI.2009.2014239.

[58] Tooraj Nikoubin, Poona Bahrebar, Sara Pouri, Keivan Navi, and Vaez Iravani. „Simple Exact Algorithm for Transistor Sizing of Low-Power High-Speed Arithmetic Circuits“. In: *VLSI Design* 2010 (Jan. 2010). ISSN: 1065-514X. DOI: 10.1155/2010/264390. URL: https://doi.org/10.1155/2010/264390.

[59] J. Bhasker and Rakesh Chadha. *Static Timing Analysis for Nanometer Designs: A Practical Approach*. 1st. Springer Publishing Company, Incorporated, 2009. ISBN: 0387938192.

[60] N. Chandra, A. Kumar Yati, and A.B. Bhattacharyya. „Extended-Sakurai-Newton MOSFET Model for Ultra-Deep-Submicrometer CMOS Digital Design“. In: *VLSI Design, 2009 22nd International Conference on*. Jan. 2009, pp. 247–252. DOI: 10.1109/VLSI.Design.2009.48.

[61] W. Ibrahim and V. Beiu. „Reliability of NAND-2 CMOS gates from threshold voltage variations“. In: *2009 International Conference on Innovations in Information Technology (IIT)*. 2009, pp. 135–139. DOI: 10.1109/IIT.2009.5413631.

[62] Ian W. Jones, Suwen Yang, and Mark R. Greenstreet. „Synchronizer Behavior and Analysis“. In: *IEEE International Symposium on Asynchronous Circuits and Systems*. IEEE, 2009, pp. 117–126.

[63] A. Khakifirooz, O.M. Nayfeh, and D. Antoniadis. „A Simple Semiempirical Short-Channel MOSFET Current -Voltage Model Continuous Across All Regions of Operation and Employing Only Physical Parameters“. In: *Electron Devices, IEEE Transactions on* 56.8 (Aug. 2009), pp. 1674–1680. ISSN: 0018-9383. DOI: 10.1109/TED.2009.2024022.

[64] J. Shin, J. Kim, N. Jang, E. Park, and Y. Choi. „A gate delay model considering temporal proximity of Multiple Input Switching“. In: *2009 International SoC Design Conference (ISOCC)*. Nov. 2009, pp. 577–580. DOI: 10.1109/SOCDC.2009.5423815.

[65] C. C. Wang and D. Markovic. „Delay Estimation and Sizing of CMOS Logic Using Logical Effort With Slope Correction“. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 56.8 (Aug. 2009), pp. 634–638. ISSN: 1549-7747. DOI: 10.1109/TCSII.2009.2024245.

[66] Y. Wang and M. Zwolinski. „Analytical transient response and propagation delay model for nanoscale CMOS inverter“. In: *2009 IEEE International Symposium on Circuits and Systems*. May 2009, pp. 2998–3001. DOI: 10.1109/ISCAS.2009.5118433.

[67] M. Alioto, M. Poli, and G. Palumbo. „Compact and simple output transition time model in nanometer CMOS gates“. In: *2008 International Conference on Microelectronics*. Dec. 2008, pp. 264–267. DOI: 10.1109/ICM.2008.5393823.

[68] Takayuki Fukuoka, Akira Tsuchiya, and Hidetoshi Onodera. „Statistical gate delay model for Multiple Input Switching“. In: *2008 Asia and South Pacific Design Automation Conference*. Mar. 2008, pp. 286–291. DOI: 10.1109/ASPDAC.2008.4483959.

[69] C. Yan and M. R. Greenstreet. „Verifying an Arbiter Circuit“. In: *Formal Methods in Computer-Aided Design, 2008. FMCAD '08*. Nov. 2008, pp. 1–9. DOI: 10.1109/FMCAD.2008.ECP.11.

[70] C. Galup-Montoro, M.C. Schneider, A.I.A. Cunha, F.Rangel de Sousa, H. Klimach, and O.F. Siebel. „The Advanced Compact MOSFET (ACM) Model for Circuit Analysis and Design“. In: *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*. Sept. 2007, pp. 519–526. DOI: 10.1109/CICC.2007.4405785.

[71] Jin He, Xuemei Xi, Hui Wan, Mohan Dunga, Mansun Chan, and Ali M. Niknejad. „BSIM5: An advanced charge-based {MOSFET} model for nanoscale {VLSI} circuit simulation“. In: *Solid-State Electronics* 51.3 (2007), pp. 433–444. ISSN: 0038-1101. DOI: http://dx.doi.org/10.1016/j.sse.2006.12.006. URL: http://www.sciencedirect.com/science/article/pii/S0038110107000020.

[72] Zhangcai Huang, Hong Yu, Atsushi Kurokawa, and Yasuaki Inoue. „Modeling the Overshooting Effect for CMOS Inverter in Nanometer Technologies“. In: *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*. ASP-DAC '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 565–570. ISBN: 1-4244-0629-3. DOI: 10.1109/ASPDAC.2007.358046. URL: http://dx.doi.org/10.1109/ASPDAC.2007.358046.

[73] S.M. Sze and Kwok K. Ng. *Physics of Semiconductor Devices*. third edition. John Wiley & Sons, Inc., 2007. ISBN: 9780471143239.

[74] S. Yang and M. Greenstreet. „Computing Synchronizer Failure Probabilities“. In: *2007 Design, Automation Test in Europe Conference Exhibition*. Apr. 2007, pp. 1–6. DOI: 10.1109/DATE.2007.364487.

[75] Dipanjan Basu and Aloke K. Dutta. „An explicit surface-potential-based MOSFET model incorporating the quantum mechanical effects“. In: *Solid-State Electronics* 50.78 (2006), pp. 1299–1309. ISSN: 0038-1101. DOI: http://dx.doi.org/10.1016/j.sse.2006.05.022. URL: http://www.sciencedirect.com/science/article/pii/S0038110106002061.

[76] G. Gildenblat, X. Li, W. Wu, H. Wang, A. Jha, R. Van Langevelde, G. D. J. Smit, A. J. Scholten, and D. B. M. Klaassen. „PSP: An Advanced Surface-Potential-Based MOSFET Model for Circuit Simulation“. In: *IEEE Transactions on Electron Devices* 53.9 (Sept. 2006), pp. 1979–1993. ISSN: 0018-9383. DOI: 10.1109/TED.2005.881006.

[77] B. Lasbouygues, S. Engels, R. Wilson, P. Maurine, N. Azemard, and D. Auvergne. „Logical effort model extension to propagation delay representation“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25.9 (Sept. 2006), pp. 1677–1684. ISSN: 0278-0070. DOI: 10.1109/TCAD.2005.857400.

[78] M. Miura-Mattausch, N. Sadachika, D. Navarro, G. Suzuki, Y. Takeda, M. Miyake, T. Warabino, Y. Mizukane, R. Inagaki, T. Ezaki, H. J. Mattausch, T. Ohguro, T. Iizuka, M. Taguchi, S. Kumashiro, and S. Miyamoto. „HiSIM2: Advanced MOSFET Model Valid for RF Circuit Simulation“. In: *IEEE Transactions on Electron Devices* 53.9 (Sept. 2006), pp. 1994–2007. ISSN: 0018-9383. DOI: 10.1109/TED.2006.880374.

[79] J. Sridharan and T. Chen. „Modeling multiple input switching of CMOS gates in DSM technology using HDMR“. In: *Proceedings of the Design Automation Test in Europe Conference*. Vol. 1. Mar. 2006, pp. 6–11. DOI: 10.1109/DATE.2006.244008.

[80] Scott E. Thompson and Srivatsan Parthasarathy. „Moore's law: the future of Si microelectronics“. In: *Materials Today* 9.6 (2006), pp. 20–25. ISSN: 1369-7021. DOI: https://doi.org/10.1016/S1369-7021(06)71539-5. URL: https://www.sciencedirect.com/science/article/pii/S1369702106715395.

[81] A. A. Yaroshevsky. „Abundances of chemical elements in the Earth's crust“. In: *Geochemistry International* 44.1 (Jan. 2006), pp. 48–55. ISSN: 1556-1968. DOI: 10.1134/S001670290601006X. URL: https://doi.org/10.1134/S001670290601006X.

[82] Manuel J Bellido, Jorge Juan, and Manuel Valencia. *Logic-Timing Simulation and the Degradation Delay Model*. Imperial College, 2005. DOI: 10.1142/p411. eprint: https://www.worldscientific.com/doi/pdf/10.1142/p411. URL: https://www.worldscientific.com/doi/abs/10.1142/p411.

[83] John R. Hauser. „A new and improved physics-based model for MOS transistors“. In: *Electron Devices, IEEE Transactions on* 52.12 (Dec. 2005), pp. 2640–2647. ISSN: 0018-9383. DOI: 10.1109/TED.2005.859623.

[84] A. Kabbani, D. AlKhalili, and A.J. Al-Khalili. „Technology portable analytical model for DSM CMOS inverter delay estimation“. In: *Circuits, Devices and Systems, IEE Proceedings* - 152.5 (Oct. 2005), pp. 433–440. ISSN: 1350-2409. DOI: 10.1049/ip-cds:20041016.

[85] A. Kabbani, D. Al-Khalili, and A. J. Al-Khalili. „Delay analysis of CMOS gates using modified logical effort model“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24.6 (June 2005), pp. 937–947. ISSN: 0278-0070. DOI: 10.1109/TCAD.2005.847892.

[86] Charles Kittel. *Introduction to Solid State Physics*. 8th ed. Wiley, 2004. ISBN: 9780471415268.

[87] J. L. Rossello and J. Segura. „An analytical charge-based compact delay model for submicrometer CMOS inverters“. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.7 (July 2004), pp. 1301–1311. ISSN: 1549-8328. DOI: 10.1109/TCSI.2004.830692.

[88] J.L. Rossello and J. Segura. „A compact propagation delay model for deep-submicron CMOS gates including crosstalk“. In: *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*. Vol. 2. Feb. 2004, 954–959 Vol.2. DOI: 10.1109/DATE.2004.1269016.

[89] David A. Hodges, Horace G. Jackson, and Resve A. Saleh. *Analysis and design of digital integrated circuits; in deep submicron technology*. 3. ed., international ed. McGraw-Hill series in electrical engineering. Boston, Mass. [u.a.]: McGraw-Hill Higher Education, 2003. ISBN: 0-07-228365-3; 0-07-118164-4.

[90] C. Lallement, J.-M. Sallese, M. Bucher, W. Grabinski, and Pierre C. Fazan. „Accounting for quantum effects and polysilicon depletion from weak to strong inversion in a charge-based design-oriented MOSFET model“. In: *Electron Devices, IEEE Transactions on* 50.2 (Feb. 2003), pp. 406–417. ISSN: 0018-9383. DOI: 10.1109/TED.2003.809040.

[91] W. Bolton. *Control Systems*. Oxford: Newnes, 2002.

[92] P. Maurine, M. Rezzoug, N. Azemard, and D. Auvergne. „Transition time modeling in deep submicron CMOS“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21.11 (Nov. 2002), pp. 1352–1363. ISSN: 0278-0070. DOI: 10.1109/TCAD.2002.804088.

[93] M. H. Na, E. J. Nowak, W. Haensch, and J. Cai. „The effective drive current in CMOS inverters“. In: *Electron Devices Meeting, 2002. IEDM '02. International*. Dec. 2002, pp. 121–124. DOI: 10.1109/IEDM.2002.1175793.

[94] Mika Nyström and Alain J Martin. „Crossing the synchronous-asynchronous divide“. In: *Workshop on Complexity-Effective Design*. 2002.

[95] A. Ortiz-Conde, F.J. GarcıÌ Aa Sánchez, J.J. Liou, A. Cerdeira, M. Estrada, and YY . Yue. „A review of recent MOSFET threshold voltage extraction methods“. In: *Microelectronics Reliability* 42.4 (2002), pp. 583–596. ISSN: 0026-2714. DOI: https://doi.org/10.1016/S0026-2714(02)00027-6. URL: http://www.sciencedirect.com/science/article/pii/S0026271402000276.

[96] Paulino Ruiz-de-Clavijo, Jorge Juan-Chico, Manuel J. Bellido, Alejandro Millán, and David Guerrero. „Efficient and Fast Current Curve Estimation of CMOS Digital Circuits at the Logic Level“. In: *Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*. PATMOS '02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 400–408. ISBN: 3540441433.

[97] M. Miura-Mattausch, H.J. Mattausch, N.D. Arora, and C.Y. Yang. „MOSFET modeling gets physical“. In: *Circuits and Devices Magazine, IEEE* 17.6 (Nov. 2001), pp. 29–36. ISSN: 8755-3996. DOI: 10.1109/101.968914.

[98] John P. Uyemura. *CMOS Logic Circuit Design*. 1st ed. Boston, MA: Springer US, 2001. ISBN: 978-0-306-47529-0. DOI: 10.1007/0-306-47529-4_2. URL: https://doi.org/10.1007/0-306-47529-4_2.

[99] D. Auvergne, J. M. Daga, and M. Rezzoug. „Signal transition time effect on CMOS delay evaluation“. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47.9 (Sept. 2000), pp. 1362–1369. ISSN: 1057-7122. DOI: 10.1109/81.883331.

[100] M. J. Bellido-Díaz, J. Juan-Chico, A. J. Acosta, M. Valencia, and J. L. Huertas. „Logical Modelling of Delay Degradation Effect in Static CMOS Gates“. In: *IEE Proceedings – Circuits, Devices, and Systems* 147.2 (2000), pp. 107–117.

[101] A. A. Hamoui and N. C. Rumin. „An analytical model for current, delay, and power analysis of submicron CMOS logic circuits“. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.10 (Oct. 2000), pp. 999–1007. ISSN: 1057-7130. DOI: 10.1109/82.877142.

[102] Jorge Juan-Chico, Manuel J. Bellido, Paulino Ruiz-de-Clavijo, Antonio J. Acosta, and Manuel Valencia. „Degradation Delay Model Extension to CMOS Gates“. English. In: *Integrated Circuit Design*. LNCS 1918. Springer, 2000, pp. 149–158. ISBN: 978-3-540-41068-3. DOI: 10.1007/3-540-45373-3_15.

[103] Alexander Klös and Arno Kostka. „PREDICTMOS - a predictive compact model of small-geometry MOSFETs for circuit simulation and device scaling calculations“. In: *Solid-State Electronics* 44.7 (2000), pp. 1145–1156. ISSN: 0038-1101. DOI: http://dx.doi.org/10.1016/S0038-1101(00)00045-9. URL: http://www.sciencedirect.com/science/article/pii/S0038110100000459.

[104] K.A. Bowman, B.L. Austin, J.C. Eble, X. Tang, and J.D. Meindl. „A physical alpha-power law MOSFET model“. In: *Solid-State Circuits, IEEE Journal of* 34.10 (Oct. 1999), pp. 1410–1414. ISSN: 0018-9200. DOI: 10.1109/4.792617.

[105] A. Chatzigeorgiou, S. Nikolaidis, and I. Tsoukalas. „A modeling technique for CMOS gates“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18.5 (May 1999), pp. 557–575. ISSN: 0278-0070. DOI: 10.1109/43.759070.

[106] Ana Isabela Araújo Cunha, Márcio Cherem Schneider, and Carlos Galup-Montoro. „Derivation of the unified charge control model and parameter extraction procedure“. In: *Solid-State Electronics* 43.3 (1999), pp. 481–485. ISSN: 0038-1101. DOI: http://dx.doi.org/10.1016/S0038-1101(98)00285-8. URL: http://www.sciencedirect.com/science/article/pii/S0038110198002858.

[107] J. M. Daga and D. Auvergne. „A comprehensive delay macro modeling for submicrometer CMOS logics“. In: *IEEE Journal of Solid-State Circuits* 34.1 (Jan. 1999), pp. 42–55. ISSN: 0018-9200. DOI: 10.1109/4.736655.

[108] Mark R Greenstreet. „Real-time merging“. In: *Proceedings. Fifth International Symposium on Advanced Research in Asynchronous Circuits and Systems.* IEEE. 1999, pp. 186–198.

[109] Ivan Sutherland, Bob Sproull, and David Harris. *Logical Effort; Designing fast CMOS circuits.* Hier auch später erschienene, unveränderte Nachdrucke. San Francisco, Calif. [u.a.]: Morgan Kaufmann, 1999. ISBN: 1-55860-557-6; 978-1-55860-557-2.

[110] A. Asenov. „Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 mu MOSFET's: A 3-D "atomistic" simulation study“. In: *IEEE Transactions on Electron Devices* 45.12 (1998), pp. 2505–2513. DOI: 10.1109/16.735728.

[111] L. Bisdounis, S. Nikolaidis, and O. Koufopavlou. „Analytical transient response and propagation delay evaluation of the CMOS inverter for short-channel devices“. In: *IEEE Journal of Solid-State Circuits* 33.2 (Feb. 1998), pp. 302–306. ISSN: 0018-9200. DOI: 10.1109/4.658636.

[112] Erik Plahte, Thomas Mestl, and Stig W. Omholt. „A methodological basis for description and analysis of systems with complex switch-like interactions“. In: *Journal of Mathematical Biology* 36.4 (Mar. 1998), pp. 321–348. ISSN: 1432-1416. DOI: 10.1007/s002850050103. URL: https://doi.org/10.1007/s002850050103.

[113] J. Segura, J. L. Rossello, J. Morra, and H. Sigg. „A variable threshold voltage inverter for CMOS programmable logic circuits“. In: *IEEE Journal of Solid-State Circuits* 33.8 (1998), pp. 1262–1265. DOI: 10.1109/4.705367.

[114] Roger T. Howe and Charles G. Sodini. *Microelectronics; an integrated approach.* Prentice Hall electronics and VLSI series. London: Prentice-Hall Internat., 1997. ISBN: 0-13-271131-1.

[115] J. Juan-Chico, M. J. Bellido, A. J. Acosta, A. Barriga, and M. Valencia. „Delay degradation effect in submicronic CMOS inverters“. In: *Proc. Seventh International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'97)*. Louvain la Neuve, Belgium, 1997, pp. 215–224. ISBN: 2-87200-025-9.

[116] V. Chandramouli and K. A. Sakallah. „Modeling the effects of temporal proximity of input transitions on gate propagation delay and transition time“. In: *33rd Design Automation Conference Proceedings, 1996*. June 1996, pp. 617–622. DOI: 10.1109/DAC.1996.545649.

[117] S. Dutta, S.S.M. Shetti, and S.L. Lusky. „A comprehensive delay model for CMOS inverters“. In: *Solid-State Circuits, IEEE Journal of* 30.8 (Aug. 1995), pp. 864–871. ISSN: 0018-9200. DOI: 10.1109/4.400428.

[118] Christian C. Enz, François Krummenacher, and Eric A. Vittoz. „An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications“. In: *Analog Integrated Circuits and Signal Processing* 8.1 (1995), pp. 83–114.

[119] S.H.K. Embabi and R. Damodaran. „Delay models for CMOS, BiCMOS and BiNMOS circuits and their applications for timing simulations“. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 13.9 (Sept. 1994), pp. 1132–1142. ISSN: 0278-0070. DOI: 10.1109/43.310902.

[120] I.M. Filanovsky and H. Baltes. „CMOS Schmitt trigger design“. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 41.1 (Jan. 1994), pp. 46–49. ISSN: 1057-7122. DOI: 10.1109/81.260219.

[121] K.O. Jeppson. „Modeling the influence of the transistor gain ratio and the input-to-output coupling capacitance on the CMOS inverter delay“. In: *Solid-State Circuits, IEEE Journal of* 29.6 (June 1994), pp. 646–654. ISSN: 0018-9200. DOI: 10.1109/4.293109.

[122] A. Nabavi-Lishi and N.C. Rumin. „Inverter models of CMOS gates for supply current and delay evaluation“. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 13.10 (Oct. 1994), pp. 1271–1279. ISSN: 0278-0070. DOI: 10.1109/43.317470.

[123] Narain Arora. *MOSFET models for VLSI circuit simulation; theory and practice.* Computational microelectronics. Wien [u.a.]: Springer, 1993. ISBN: 3-211-82395-6; 0-387-82395-6.

[124] J.H. Huang, Z.H. Liu, M. Jeng, P.K. Ko, and C. Hu. „A robust physical and predictive model for deep-submicrometer MOS circuit simulation“. In: *Custom Integrated Circuits Conference, 1993., Proceedings of the IEEE 1993*. May 1993, pp. 14.2.1–14.2.4. DOI: 10.1109/CICC.1993.590711.

[125] Elmar Melcher, Wolfgang Röthig, and Michel Dana. „Multiple input transitions in CMOS gates“. In: *Microprocessing and Microprogramming* 35.1 (1992). Software and Hardware: Specification and Design, pp. 683–690. ISSN: 0165-6074. DOI: http://dx.doi.org/10.1016/0165-6074(92)90387-M. URL: http://www.sciencedirect.com/science/article/pii/016560749290387M.

[126] A. Pfister. „Novel CMOS Schmitt trigger with controllable hysteresis“. In: *Electronics Letters* 28.7 (Mar. 1992), pp. 639–641. ISSN: 0013-5194. DOI: 10.1049/el:19920404.

[127] Chan-Kwang Park, Chang-Yeol Lee, Kwyro Lee, Byung-Jong Moon, Young Hee Byun, and M. Shur. „A unified current-voltage model for long-channel nMOSFETs“. In: *IEEE Transactions on Electron Devices* 38.2 (Feb. 1991), pp. 399–406. ISSN: 0018-9383. DOI: 10.1109/16.69923.

[128] T. Sakurai and A.R. Newton. „A simple MOSFET model for circuit analysis“. In: *Electron Devices, IEEE Transactions on* 38.4 (Apr. 1991), pp. 887–894. ISSN: 0018-9383. DOI: 10.1109/16.75219.

[129] D. Auvergne, N. Azemard, D. Deschacht, and M. Robert. „Input waveform slope effects in CMOS delays“. In: *Solid-State Circuits, IEEE Journal of* 25.6 (Dec. 1990), pp. 1588–1590. ISSN: 0018-9200. DOI: 10.1109/4.62196.

[130] T. Sakurai and A.R. Newton. „Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas“. In: *Solid-State Circuits, IEEE Journal of* 25.2 (Apr. 1990), pp. 584–594. ISSN: 0018-9200. DOI: 10.1109/4.52187.

[131] M.-C. Shiau and Ph.D. Chung-Yu Wu. „The signal delay in interconnection lines considering the effects of small-geometry CMOS inverters“. In: *Circuits and Systems, IEEE Transactions on* 37.3 (Mar. 1990), pp. 420–425. ISSN: 0098-4094. DOI: 10.1109/31.52736.

[132] R. W. Brockett. „Smooth dynamical systems which realize arithmetical and logical operations“. In: *Three Decades of Mathematical System Theory: A Collection of Surveys at the Occasion of the 50th Birthday of Jan C. Willems*. Ed. by Hendrik Nijmeijer and Johannes M. Schumacher. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 19–30. ISBN: 978-3-540-46709-0. DOI: 10.1007/BFb0008457. URL: https://doi.org/10.1007/BFb0008457.

[133] L.M. Brocco, S.P. McCormick, and J. Allen. „Macromodeling CMOS circuits for timing simulation“. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 7.12 (Dec. 1988), pp. 1237–1249. ISSN: 0278-0070. DOI: 10.1109/43.16802.

[134] Masakazu Shoji. *CMOS digital circuit technology*. Prentice-Hall International editions. London: Prentice-Hall Internat., 1988. ISBN: 0-13-138843-6.

[135] R. E. Bryant, D. Beatty, K. Brace, Kyeongsoon Cho, and T. Sheffler. „Simulator for MOS Circuits“. In: *24th ACM/IEEE Design Automation Conference*. June 1987, pp. 9–16. DOI: 10.1109/DAC.1987.203215.

[136] Tomasz Kacprzak and Alexander Albicki. „Analysis of Metastable Operation in RS CMOS Flip-Flops“. In: *Journal of Solid-State Circuits* 22.1 (1987), pp. 57–64.

[137] L. Kleeman and Antonio Cantoni. „Metastable Behavior in Digital Systems“. In: *IEEE Design Test of Computers* 4.6 (Dec. 1987), pp. 4–19. ISSN: 0740-7475. DOI: 10.1109/MDT.1987.295189.

[138] Zlatko V. Bundalo and Branko L. Dokić. „Non-inverting regenerative CMOS logic circuits“. In: *Microelectronics Journal* 16.5 (1985), pp. 5–17. ISSN: 0026-2692. DOI: https://doi.org/10.1016/S0026-2692(85)80002-1. URL: http://www.sciencedirect.com/science/article/pii/S0026269285800021.

[139] J.B. Lasky, S.R. Stiffler, F.R. White, and J.R. Abernathey. „Silicon-on-insulator (SOI) by bonding and ETCH-back“. In: *1985 International Electron Devices Meeting*. 1985, pp. 684–687. DOI: 10.1109/IEDM.1985.191067.

[140] Y. P. Li and W. Y. Ching. „Band structures of all polycrystalline forms of silicon dioxide“. In: *Phys. Rev. B* 31 (4 Feb. 1985), pp. 2172–2179. DOI: 10.1103/PhysRevB.31.2172. URL: https://link.aps.org/doi/10.1103/PhysRevB.31.2172.

[141] T. Shibata, R. Nakayama, K. Kurosawa, S. Onga, M. Konaka, and H. Iizuka. „A simplified box (buried-oxide) isolation technology for megabit dynamic memories“. In: *1983 International Electron Devices Meeting*. 1983, pp. 27–30. DOI: 10.1109/IEDM.1983.190432.

[142] Leslie Lamport, Robert Shostak, and Marshall Pease. „The Byzantine generals problem“. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982), pp. 382–401.

[143] L R Marino. „General Theory of Metastable Operation“. In: *IEEE Transactions on Computers* C-30.2 (Feb. 1981), pp. 107–115.

[144] Harry J.M. Veendrick. „The behaviour of flip-flops used as synchronizers and prediction of their failure rate“. In: *IEEE Journal of Solid-State Circuits* 15.2 (Apr. 1980), pp. 169–176. ISSN: 0018-9200. DOI: 10.1109/JSSC.1980.1051359.

[145] T.J. Chaney. „Comments on "A Note on Synchronizer or Interlock Maloperation"“. In: *IEEE Transactions on Computers* C-28.10 (Oct. 1979), pp. 802–804. ISSN: 0018-9340. DOI: 10.1109/TC.1979.1675252.

[146] L. R. Marino. „The Effect of Asynchronous Inputs on Sequential Network Reliability“. In: *IEEE Transactions on Computers* 26.11 (1977), pp. 1082–1090.

[147] E.G. Wormald. „A Note on Synchronizer or Interlock Maloperation“. In: *IEEE Transactions on Computers* C-26.3 (Mar. 1977), pp. 317–318. ISSN: 0018-9340. DOI: 10.1109/TC.1977.1674833.

[148] Yoshio Nishi. „Silicon on Sapphire technology“. In: *ESSCIRC 76: 2nd European Solid State Circuits Conference*. 1976, pp. XII–XIII. DOI: 10.1109/ESSCIRC.1976.5469255.

[149] D.J. Kinniment and D.B.G. Edwards. „Circuit Technology in a Large Computer System“. In: *Conference on Computers - Systems and Technology.* Oct. 1972, pp. 441–450.

[150] Stephen H. Unger. „Asynchronous Sequential Switching Circuits with Unrestricted Input Changes“. In: *IEEE Transaction on Computers* 20.12 (1971), pp. 1437–1444.

[151] H. Shichman and D. A. Hodges. „Modeling and simulation of insulated-gate field-effect transistor switching circuits“. In: *IEEE Journal of Solid-State Circuits* 3.3 (Sept. 1968), pp. 285–289. ISSN: 0018-9200. DOI: 10.1109/JSSC.1968.1049902.

[152] B Dawson. „The Covalent Bond in Silicon“. In: *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences* 298 (May 1967), pp. 379–394. DOI: 10.1098/rspa.1967.0110.

[153] Archibald Vivian Hill. „The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves“. In: *Journal of Physiology (London)* 40 (1910), pp. 4–7.

Curriculum Vitae

Jürgen Maier

November 6, 2021

Personal Data

Surname	Maier
Given Name	Jürgen
ORCID	0000-0002-0965-5746
Title	Dipl.-Ing. Dipl.-Ing. BSc
Email	juergen.maier@tuwien.ac.at
Homepage	`http://ti.tuwien.ac.at/ecs/people/jmaier`
Nationality	Austria
Memberships	IEEE, ACM
Languages	German (native), English, Spanish (basics)
Interests	Member of Voluntary Fire Brigade, physics, woodworking, languages

Education

since 2015/10
PhD in Computer Sciences, TU Wien, Austria
Thesis: "Proper Abstractions for Digital Electronic Circuits: A Physically Guided Approach"

2013/10 - 2016/04
MSc in Microelectronics and Photonics, TU Wien, Austria
graduated with distinction
Thesis: "Modeling III-V Semiconductor Interfaces at an Atomistic Level using Empirical Potentials"

2011/09 - 2014/10
MSc in Computer Engineering, TU Wien, Austria
graduated with distinction
Thesis "Online Test Vector Insertion – A Concurrent Built-In Self-Testing (CBIST) Approach for Asynchronous Logic"

2008/10 - 2011/09
BSc Computer Engineering, TU Wien, Austria
graduated with distinction
Thesis: "Powerline in Building Automation"

2005/07
HTL Donaustadt, 1220 Vienna
Departement for Electronic Data Processing and Organisation

Employment

2015/09 - 2021/12
TU Wien, Faculty of Informatics, Austria
Project/University Assistant, Embedded Computing Systems Group, Univ.-Prof. Ulrich Schmid

2014/01 - 2015/02
TU Wien, Faculty of Electrical Engineering and Information Technology, Austria
Project Employee, Energy Economics Group (EEG)

2006/10 - 2010/12
CSPmed GmbH., later Compugroup AG, Vienna, Austria
Programming and Support

2006/01 - 2006/07
Compulsory Military Service, Mistelbach, Austria

Awards and Scholarships

2021/10	Best Distance Lecture Award, TU Wien
2019/02	Christiana Hoerbiger scholarship, TU Wien
2016/05	Excellence scholarship, Windhag foundation for Lower Austria, Austria
2014/12	Nominee for “Distinguished Young Alumna/Alumnus”-Award, Faculty of Informatics, TU Wien
2014/04	Best paper award at the 17th International Symposium on Design and Diagnostics of Electronic Circuits Systems, Warsaw, Poland
2011-2015	Merit-based scholarship, TU Wien (per annum)

Further Education

Information Technology:

Expert: C, HSPICE, Python, LaTeX, Linux (Shell), VHDL, Quartus, QuestaSim, Git

Advanced: Bash scripting, Makefiles, C++, Assembler, SQL, Matlab, R, Spectre

Basics: Genus, Innovus, Mathematica, Verilog, Ada

Courses on low level microelectronics:

Microelectronic Reliability: Devices

Selected chapters of semiconductor physics for PhD students, topic: simulation of carrier density in a resonant-tunnelling diode

Books read in spare time:

Mark Thomson. *Modern particle physics.* eng. 1. publ.. Cambridge [u.a.]: Cambridge Univ. Press, 2013. ISBN: 1107034264

Wolfgang Demtröder. *Experimental Physics 1: Mechanics and Heat.* ger. 8. Aufl. 2018. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 9783662548462

Wolfgang Demtröder. *Experimental Physics 2: Electricity and Optics.* ger. 7. Aufl. 2017. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 9783662557891

Wolfgang Demtröder. *Experimental Physics 3: Atoms, Molecules and Solids.* ger. 5. Aufl. 2016. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 3662490935

Wolfgang Demtröder. *Experimental Physics 4: Nuclear, Particle and Astro Physics.* ger ; eng. 5. Aufl. 2017. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. ISBN: 9783662528839

Didactic Training

An efficient transfer of knowledge is crucial to tackle future challenges, e.g., by educating personal or discussing topics in a team. To improve my skills in this regard, especially my teaching capabilities, I attended the following workshops, which were part of the Internal Continuing Education program at TU Wien:

Each generation wants to cooperate! : How to motivate people of different age to cooperate (one day).

Lecture planning! : How to plan and develop a lecture with a focus on didactics, i.e., how can the content be transported in a suitable way (two days).

Lots of content, little time! : Mainly focused on selecting the proper content for a lecture with clearly defined goals and timing constraints (one day).

Inverted Classroom! : Pros, cons and realization of the inverted classroom teaching technique (one day).

Inspire others with your own enthusiasm! : Methods to inspire your audience and thus optimize the knowledge transfer (two days).

Cognitive activating methods for teaching! : Multiple methods to activate the listeners and thus improve the learning experience (one day).

others : Several online workshops on online-teaching, assessment techniques and proper video setups.

Teaching and Lecturing

I was able to contribute to several lectures, which cover a wide range of topics and modes of execution. In fact I collected experience in giving lectures in front of a class (remote and in person), lab supervision and flipped (inverted) classroom setups by contributing to the following lectures:

Scientific Writing : Early course on how to properly compose a scientific article (BSc level).

Hardware Modeling : Lecture on hardware design (languages) (BSc level).

Digital Design and Computer Architecture : Lab course to implement hardware on FPGAs (BSc level).

Advanced Digital Design : Lecture with exercises on more sophisticated properties of digital designs and asynchronous logic (MSc level).

HW/SW Codesign : Lab course to implement a complete design, requiring proper splitting between hard- and software, on an FPGA (MSc level).

Co-Supervised Bachelor Theses

2021	L. Graussam: Automatic Verification Framework of VHDL Code examples J. Salzmann: Approximating Analog Waveforms by adding Arbitrary Functions
2018	D. Öhlinger: Involution Tool M. Houzar: Comparison of the metastable characteristics of Schmitt Trigger Implementations

Academic Services

Faculty Services	Member of the Scientific Staff Council (2018-2020) Member of the study commission and curriculum coordinator regarding the interdisciplinary master course “Computational Science and Engineering”
Local Co-Chair	ASYNC’18, DDECS’21
Conference Reviews	ASYNC (2016, 2019), DATE (2018), DDECS (2016, 2019-2021), DSD (2018, 2019), ICCD (2016, 2018), ISCAS (2020), Austrochip (2018, 2021)
Journal Reviews	Journal of Circuits, Systems, and Computers (JCSC) (2018, 2021), IEEE Transactions on Circuits and Systems I: Regular Papers (2021)

Publications

Key Publications

J. Maier and A. Steininger. “Efficient Metastability Characterization for Schmitt-Triggers”. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019. DOI: 10.1109/ASYNC.2019.00024

J. Maier, D. Öhlinger, U. Schmid, M. Függer, and T. Nowak. “A Composable Glitch-Aware Delay Model”. In: *Proceedings of the 2021 on Great Lakes Symposium on VLSI*. GLSVLSI ’21. Virtual Event, USA: Association for Computing Machinery, 2021. ISBN: 9781450383936. DOI: 10.1145/3453688.3461519. URL: https://doi.org/10.1145/3453688.3461519

A. Steininger, **J. Maier**, and R. Najvirt. “The Metastable Behavior of a Schmitt-Trigger”. In: *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2016. DOI: 10.1109/ASYNC.2016.19

J. Maier, M. Függer, T. Nowak, and U. Schmid. “Transistor-Level Analysis of Dynamic Delay Models”. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019. DOI: 10.1109/ASYNC.2019.00019

M. Függer, **J. Maier**, R. Najvirt, T. Nowak, and U. Schmid. “A faithful binary circuit model with adversarial noise”. In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. Nominee for Best Paper Award. Mar. 2018. DOI: 10.23919/DATE.2018.8342219

Peer-reviewed Conference Articles

Jürgen Maier. “Gain and Pain of a Reliable Delay Model”. In: *2021 24th Euromicro Conference on Digital System Design (DSD)*. 2021. DOI: 10.1109/DSD53832.2021.00046

J. Maier, D. Öhlinger, U. Schmid, M. Függer, and T. Nowak. “A Composable Glitch-Aware Delay Model”. In: *Proceedings of the 2021 on Great Lakes Symposium on VLSI*. GLSVLSI ’21. Virtual Event, USA: Association for Computing Machinery, 2021. ISBN: 9781450383936. DOI: 10.1145/3453688.3461519. URL: https://doi.org/10.1145/3453688.3461519

D. Öhlinger, **J. Maier**, M. Függer, and U. Schmid. “The Involution Tool for Accurate Digital Timing and Power Analysis”. In: *2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. July 2019. DOI: 10.1109/PATMOS.2019.8862165

J. Maier and A. Steininger. “Efficient Metastability Characterization for Schmitt-Triggers”. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019. DOI: 10.1109/ASYNC.2019.00024

J. Maier, M. Függer, T. Nowak, and U. Schmid. “Transistor-Level Analysis of Dynamic Delay Models”. In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2019. DOI: 10.1109/ASYNC.2019.00019

Chuchu Fan, Yu Meng, **Jürgen Maier**, Ezio Bartocci, Sayan Mitra, and Ulrich Schmid. “Verifying nonlinear analog and mixed-signal circuits with inputs”. In: *IFAC-PapersOnLine* 51.16 (2018). 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.08.041

M. Függer, **J. Maier**, R. Najvirt, T. Nowak, and U. Schmid. “A faithful binary circuit model with adversarial noise”. In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. Nominee for Best Paper Award. Mar. 2018. DOI: 10.23919/DATE.2018.8342219

A. Steininger, R. Najvirt, and **J. Maier**. “Does Cascading Schmitt-Trigger Stages Improve the Metastable Behavior?” In: *2016 Euromicro Conference on Digital System Design (DSD)*. Aug. 2016. DOI: 10.1109/DSD.2016.56

A. Steininger, **J. Maier**, and R. Najvirt. “The Metastable Behavior of a Schmitt-Trigger”. In: *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. May 2016. DOI: 10.1109/ASYNC.2016.19

J. Maier and A. Steininger. “Online Test Vector Insertion: A Concurrent Built-In Self-Testing (CBIST) Approach for Asynchronous Logic”. In: *Design and Diagnostics of Electronic Circuits Systems, 17th International Symposium on*. Apr. 2014. DOI: 10.1109/DDECS.2014.6868759

Peer-reviewed Journal Articles

D. Öhlinger, **J. Maier**, M. Függer, and U. Schmid. “The Involution Tool for Accurate Digital Timing and Power Analysis”. In: *Integration* 76 (2021). ISSN: 0167-9260. DOI: https://doi.org/10.1016/j.vlsi.2020.09.007. URL: http://www.sciencedirect.com/science/article/pii/S0167926020302777

Jürgen Maier and Hermann Detz. “Atomistic modeling of interfaces in III-V semiconductor superlattices”. In: *physica status solidi (b)* 253.4 (2016). ISSN: 1521-3951. DOI: 10.1002/pssb.201552496

Report

Jürgen Maier. *Modeling the CMOS Inverter using Hybrid Systems.* Tech. rep. TUW-259633. E182 - Institut für Technische Informatik; Technische Universität Wien, 2017

Theses

Jürgen Maier. "Modeling III-V Semiconductor Interfaces at an Atomistic Level using Empirical Potentials". Master Thesis, Institute of Solid State Electronics, TU Wien, Vienna, Austria. MA thesis. TU Wien, Vienna, Austria, Apr. 2016. URL: http://catalogplus.tuwien.ac.at/UTW:UTW:UTW_alma2150179390003336

Jürgen Maier. "Online Test Vector Insertion: A Concurrent Built-In Self-Testing (CBIST) Approach for Asynchronous Logic". Master Thesis, Institute of Computer Engineering, TU Wien, Vienna, Austria. MA thesis. TU Wien, Vienna, Austria, Oct. 2014. URL: http://catalogplus.tuwien.ac.at/UTW:UTW:UTW_alma2139475450003336

Fresh-Ideas

P. Paulweber, **J. Maier**, and J. Cortadella. "Unified (A)Synchronous Circuit Development". In: *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. Fresh idea accepted for ASYNC'19. May 2019

Selected Presentations

Gain and Pain of a Reliable Delay Model, DSD, 2021

A Composable Glitch-Aware Delay Model, GLSVLSI, 2021

Transistor-Level Analysis of Dynamic Delay Models, ASYNC, 2019

Efficient Metastability Characterization for Schmitt-Triggers, ASYNC, 2019

Unified (A)Synchronous Circuit Development, ASYNC, 2019

A Faithful Binary Circuit Model with Adversarial Noise, DATE, 2018

Involution Model: Faithful Delay Prediction in Digital Circuits, Université Paris-Sud, Laboratoire de Recherche en Informatique (LRI), Paris, France, July 2017, (**Invited Talk**)

Online Test Vector Insertion: A Concurrent Built in Self-Testing (CBIST) Approach for Asynchronous Logic, DDECS, 2014