TU WIEN Informatics

η-CIDM: A faithful and composable delay model with adversarial noise

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Informatik

eingereicht von

Daniel Öhlinger, BSc
Matrikelnummer 01525898

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Ulrich Schmid

Wien, 12. Mai 2022

Daniel Öhlinger	Ulrich Schmid

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

TU WIEN Informatics

η-CIDM: A faithful and composable delay model with adversarial noise

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computer Engineering

by

Daniel Öhlinger, BSc
Registration Number 01525898

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Ulrich Schmid

Vienna, 12th May, 2022

Daniel Öhlinger	Ulrich Schmid

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

Erklärung zur Verfassung der Arbeit

Daniel Öhlinger, 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. Mai 2022

Daniel Öhlinger

Danksagung

Ich möchte mich bei Prof. Schmid für seine Unterstützung, nicht nur während meiner Masterarbeit, sondern während meines ganzen Studiums bedanken. Die Chance aktiv teilzunehmen an einem Forschungsgebiet war eine großartige Möglichkeit für mich Einblick in Akademia zu erhalten. Diese Arbeit wurde im Kontext des FWF-Projekts DMAC (P32431) durchgeführt.

Weiters möchte ich mich bei Jürgen Maier bedanken für seine Hilfe. Seine Expertise betreffend Simulation und unsere Diskussionen waren immer sehr hilfreich für mich.

Schlussendlich möchte ich meinen Eltern meine tiefste Dankbarkeit aussprechen für ihre bedingungslose Unterstützung.

Acknowledgements

I would like to thank Prof. Schmid for his continuing support, not just during my master thesis, but throughout my whole studies. Having the chance to actively participate in a field of research was a great opportunity to gain insights into academia. This work was conducted in the context of the FWF project DMAC (P32431).

Moreover, I would like to thank Jürgen Maier for his help. His expertise regarding simulations and our discussions where really fruitful.

Finally, I would like to express my deepest gratitude to my parents for their unconditional support.

Kurzfassung

Die Vorhersage des zeitlichen Verhaltens komplexer digitaler Schaltungen ist ein essentieller Teil in der Design Phase von digitalen Schaltungen. Analoge Simulationen (z.B. SPICE) sind die akkurateste Methode um das Zeitverhalten vorherzusagen. Allerdings sind deren Simulationszeiten exzessiv, selbst für mittelgroße Schaltungen. Eine deutlich weniger kostspielige Alternative ist die digitale dynamische Analyse des Zeitverhaltens, welches die Verzögerung von digitalen Signalen durch eine Schaltung mittels eines digitalen Verzögerungsmodells verfolgt. Diese digitalen Modelle haben allerdings den Nachteil, dass die Genauigkeit der Vorhersage leidet. Die meisten dieser Modelle, auch das populäre *pure delay* Modell und das *inertial delay* Modell, sind unrealistisch, in dem Sinne, dass diese Modelle entweder Verhalten modellieren können, welches in der Realität nicht möglich ist, oder umgekehrt. Das Einzige bis jetzt bekannte Modell, welches die Realität wahrheitsgetreu abbildet, ist das *Involution Delay Model* (IDM).

In dieser Arbeit werden zwei bereits bestehende Erweiterungen des IDM, das η-*Involution Delay Model* (η-IDM) und das *Composable Involution Delay Model* (CIDM), in einer neuen Erweiterung, dem η-*Composable Involution Delay Model* (η-CIDM) kombiniert. Das CIDM fügt transitionsabhängige Verzögerungen, welche die Modellierung von unterschiedlichen Schwellenspannungen ermöglichen, vor dem IDM Kanal ein. Das η-IDM fügt nicht-deterministische Zeitverzögerungen zu den deterministischen Verzögerungsfunktionen hinzu, welche eine substantiell größere Bandbreite haben können, als für das η-IDM. Wir zeigen, dass diese neue Erweiterung eine breitere Anwendbarkeit und größere Toleranz gegenüber *Prozess-, Spannungs- und Temperatur-* (PVT) Variationen besitzt, und die Realität nach wie vor wahrheitsgetreu abbildet.

Um die erhöhte Anwendbarkeit von η-CIDM in der Praxis zu zeigen, wurden umfangreiche Simulationen durchgeführt. Dabei werden die eigentlichen Verzögerungsfunktionen, welche mittels SPICE Simulationen ermittelt werden, mit dem berechneten Zeitverhalten von η-CIDM verglichen. Die Ergebnisse zeigen, dass die neue Erweiterung im Gegensatz zu den originalen Modellen fähig ist, PVT Variationen und die Alterung von Schaltungen abzudecken.

Abstract

Predicting the timing of complex digital circuits is a crucial part in the design phase. While analog simulations (e.g. SPICE) are the golden reference here, their simulation times grow excessively, even for moderately large circuits. A considerably less costly alternative is digital dynamic timing analysis, which traces the propagation of digital signal traces throughout a circuit via some delay model. However, this abstraction comes at the cost of decreased accuracy and, for most existing delay models, including the popular *pure delay* or *inertial delay* model, even unfaithful models. Indeed, as of now, the *Involution Delay Model* (IDM) is the only known candidate for a faithful delay model, i.e., one that allows to model a circuit if and only if it can be built in reality.

In this thesis, two existing extensions for the IDM, namely the *η-Involution Delay Model* (η-IDM) and the *Composable Involution Delay Model* (CIDM), are combined into the *η-Composable Involution Delay Model* (η-CIDM). The CIDM prepends IDM channels by transition-dependent pure delays, which allow to model threshold voltage shifts. The η-CIDM adds adversarial delay variations to the deterministic delay functions, within a range that is substantially larger than for the η-IDM. We prove that this new extension, while providing a considerably better applicability and larger tolerance against *process, voltage and temperature* (PVT) variations and aging, is still faithful.

To demonstrate the increased applicability of the η-CIDM in practice, extensive simulations are performed. By comparing the actual delay functions, obtained by SPICE simulations, and the calculated delay functions of the η-CIDM, the coverage of the new extension is investigated. The results indeed show that, unlike the original models, it covers a wide range of PVT variations, as well as aging effects.

Contents

CHAPTER 1

Introduction

Accurately predicting the delays of signals through a digital circuit is an important task in digital design. While analog simulations, e.g. SPICE, are very accurate, their simulation times are excessive. These simulations are continuous in time and value and use very accurate but also very complex systems of differential equations to describe the behavior of the cells. Using digital delay models, which are discrete in value and continuous in time, makes simulations of large circuits feasible; however, at the cost of accuracy. Prominent examples of digital delay models are the pure delay model and the inertial delay model. These delay models are widely used in simulation suites. While the pure delay model delays all transitions by a constant delay Δ, the inertial delay model removes shorter pulses and delays the other pulses by Δ. The delay values need to be known a-priori and are constant throughout all simulation runs. Tools like *Composite Current Source* (CCS) [Syn16] and *Effective Current Source Model* (ECSM) [Cad15] are used to determine these delay values. More accurate delay models, like the *Degradation Delay Model* (DDM), use the previous-output-to-input delay T in order to determine the input-to-output delay $\delta(T)$.

However, all these state-of-the-art models have a big drawback: They are lacking faithfulness, which is an important property, especially when it comes to formal verification of digital circuits. A model is unfaithful if it can solve problems which cannot be solved in physical reality, or if it cannot solve problems which can be solved in physical reality. In other words: A problem can be solved by a faithful delay model if and only if it can be solved in physical reality. The authors of [FNS16] showed that all existing binary delay models are unfaithful, since these models are not able to correctly predict the behavior of a circuit solving the canonical *Short-Pulse Filtration* (SPF) problem.

The only known candidate for a faithful delay model is the *Involution Delay Model* (IDM) [FNNS20] and its extensions. The authors showed that IDM is indeed able to faithfully model glitch propagation. Its distinguishing property is that the delay functions form

involutions, i.e., $-\delta_\uparrow(-\delta_\downarrow(T)) = T$ and $-\delta_\downarrow(-\delta_\uparrow(T)) = T$, where $\delta_\uparrow$ and $\delta_\downarrow$ are the delay functions for rising resp. falling transitions.

Nevertheless, the IDM has several shortcomings. The first one is concerned with the composability. Characterization of a circuit requires the the output threshold voltage of the preceding gate and the input threshold voltage of the succeeding gate are equal. This makes the characterization process tedious and for certain circuits (with feedback loops) even impossible. The second shortcoming is, that the delays are deterministic, i.e., the IDM cannot cover any kind of process variations and aging.

Thanks to the Involution Tool[1], the initial version of which I have developed as my Bachelor thesis [Öhl18], I was able to contribute to many extensions of the IDM [ÖMFS21; MÖS+21; FMÖS22], developed in the FWF DMAC project in the past, which are not part of the key contributions of this Master thesis. These extensions aim at solving the previously described shortcomings. Most notably are our *Composable Involution Delay Model* (CIDM) [MÖS+21] and the *η-Involution Delay Model* (η-IDM) [FMN+18]. While the first extension aims at simplifying the characterization process, the latter allows adding bounded non-deterministic delay variations in order to extend the modeling power.

1.1 Contributions and Methodology

By combining the extensions *Composable Involution Delay Model* (CIDM) and *η-Involution Delay Model* (η-IDM) into a new extension *η-Composable Involution Delay Model* (η-CIDM), the practical applicability of the IDM shall be increased. In order to do this, the following three challenges will be addressed in this thesis:

(1) Showing that the resulting extension *η-Composable Involution Delay Model* (η-CIDM) still maintains faithfulness. This will be done by means of a reduction proof.

(2) The η-IDM only allows delay variations to be within a small range $[-\eta^-_{min}, \eta^+_{min}]$. By loosening the bounds of the η-IDM, the practical applicability is substantially increased. By mathematical proofs, it will be shown that this less restricted model is still faithful. Moreover, an alternative reduction proof for the impossibility of bounded SPF is presented.

(3) Finally, extensive simulations are performed to compare the new extension with the golden reference SPICE. In order to be able to perform these simulations, the Involution Tool [ÖMFS21] had to be extended significantly. The simulations showed that the new extension is able to cover process, voltage and temperature variations. Furthermore, the effect of aging is also covered.

[1] Publicly available via `https://github.com/oehlinscher/InvolutionTool`.

1.2 Structure of the thesis

Chapter 2 will present related work, especially focusing on state-of-the-art delay models, PVT variations and circuit aging. In Chapter 3, the IDM and its extensions are presented, and necessary prerequisites for the following chapters are introduced. The faithfulness of the new extension η-CIDM is proven in Chapter 4. Extensions for the existing extension η-IDM are discussed in Chapter 5. The extensions for the Involution Tool and the evaluation of the η-CIDM is presented in Chapter 6. Chapter 7 concludes the thesis and presents possible directions for future research.

CHAPTER 2

Related work

In this chapter, the basics of digital delay models are provided. Moreover, the important property of faithfulness is introduced. Finally, PVT variations and aging of circuits are discussed, since these are important targets for the η-CIDM.

2.1 Delay models

Digital delay models are used for predicting the signal delays in digital circuits. The most simplistic model is the *pure delay model*. It delays the input signal by a constant delay Δ. The *inertial delay model* [Ung71] behaves the same for large pulses (width $> A$). Shorter pulses (width $\leq A$) are removed.

Figure 2.1a shows the relation between input pulse width and output pulse width and reveals on big flaw of the inertial delay model: It is discontinuous, which means that pulses with a width $A + \varepsilon, \varepsilon > 0$ are propagated unchanged, whereas a pulse with width $\leq A$ is completely removed. This is of course a contradiction to the physical reality.

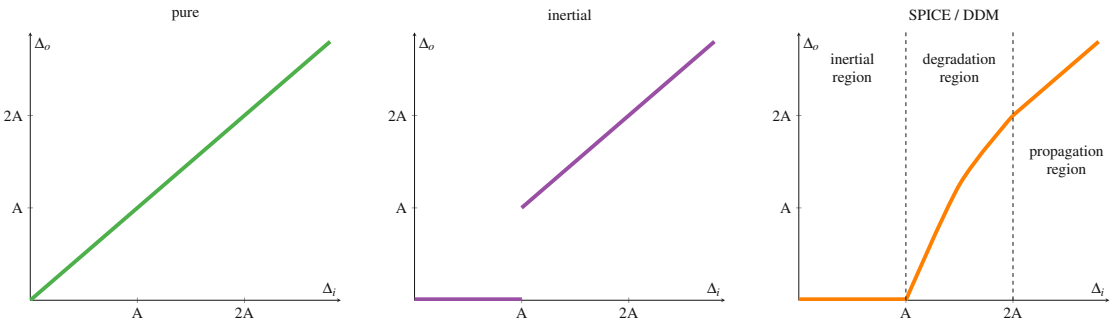
One model that does not exhibit this discontinuity is the DDM by Bellido-Diaz et al. [BDJCA+00; BDJCV06]. Its delay function consists of three parts: The inertial region and the propagation region are already known. To tackle the problem of discontinuity, the degradation region is introduced between the former two. Input pulses in this region are propagated to output; however, their width is degraded. DDM is an instance of a so-called *bounded single history channel* [FNS16]. The delay for each transition is calculated based on a delay function $\delta(T)$, where T is the previous-output-to-input delay (see Figure 3.2a). For a bounded channel, both $\lim_{T\to\infty} \delta(T) < \infty$ and $\lim_{T\to-\infty} |\delta(T)| < \infty$. Chapter 3 will show that this is in stark contrast to the IDM, where the negative delay is unbounded.

Figure 2.1b shows different cases for the presented delay models.

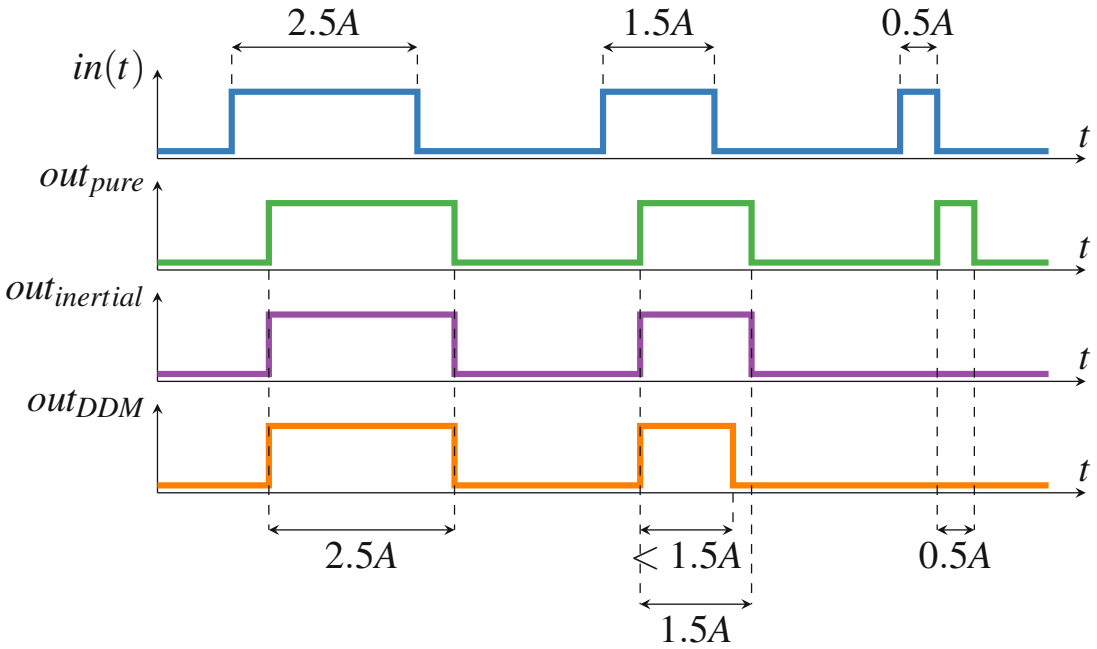
(1) The first pulse is large enough ($2.5A$) and therefore passed unaltered by all three delay models.

(2) The second pulse ($1.5A$) is in the degradation region. While pure and inertial delay pass the pulse unaltered, the pulse width is degraded in the DDM.

(3) The third pulse ($0.5A$) is removed in the inertial delay model and the DDM. Only the pure delay model passes the pulse unaltered.



(a) Input pulse width to output pulse width relation (adapted from [Mai21]).



(b) Example trace (adapted from [Öhl18]).

Figure 2.1: Comparison of different delay models.

These simplistic delay models rely on predetermined gate delays. There are several tools like Synopsis VCS and Cadence NCSim, which use sophisticated techniques to determine these delays. VCS uses an approach called CCS, which characterizes a cell for a table of

different input slopes and output capacities. By storing the current through the load capacitance, the output voltage can be reproduced accurately. NCSim employs the ECSM, which is also a lookup table approach. However, instead of storing the output current it stores the time at which the output crosses certain threshold points. Note that it is possible to retrieve the values for CCS from ECSM and vice versa [NFP11], which essentially makes these two models identical.

2.1.1 Tool support

State-of-the-art tools like Questa Sim, NCSim and VCS support pure delay and inertial delay models out-of-the-box. Moreover, also design languages like VHDL and Verilog support modeling delays: Gate libraries written in VHDL employ VHDL Vital [IEE01, Chapter 9], while Verilog gate libraries us the Verilog delay model [IEE06, Chapter 14]. Both implementations support pure and inertial delay channels.

- When performing simulations with Questa Sim, the Verilog delay model is configured via the arguments `+pulse_e/<percent>` and `+pulse_r/<percent>` (see [Men16, pp. 943 sqq.]). Basically, these two arguments handle at which percentage of the path delay pulse leads to an error on the output resp. is rejected. By setting both values to 100, an inertial delay model can be configured. Setting both values to 0 configures a pure delay.
- VHDL Vital configures the delay model in the gate library. This is less convenient, since the configuration cannot be overridden via command line arguments. Nevertheless, an inertial delay model (`VitalInertial`) and a pure delay model (`VitalTransport`) can be configured directly in the gate library.

2.2 Faithful delay models

One important property of digital delay models is faithfulness. According to [FNS16], a model is called faithful if and only if it is able to solve problems within the model that can be solved in physical reality. Függer et al. showed that no delay model is able to faithfully model glitch propagation, for the canonical SPF problem. As the name indicates, the goal is to remove a pulse if it is too short and otherwise pass the pulse unaltered or extended. Note that SPF only allows a single pulse at the input. While the unbounded version is solvable in physical reality, this is not the case for the bounded version, where the output needs to settle withing in a bounded time $T > 0$. The unsolvability of bounded SPF in physical reality is shown in [Mar77].

As Table 2.1 shows, the presented delay models either contradict the unsolvability of bounded SPF (pure delay), or contradict the solvability of unbounded SPF (inertial delay model, DDM). Chapter 3 will present the IDM, a delay model which is faithful for the SPF problem, and hence the only candidate for a generally faithful delay model known so far.

bounded SPF	✗	✓	✓	✗
SPF	✗	✓	✓	✓
	pure	inertial	DDM	physical

Table 2.1: Solvability of (bounded) SPF for different delay models and the physical reality (taken from [FNS16]).

2.3 PVT variations

Process Voltage and Temperature (PVT) variations cause changes in the timing behavior of *Integrated Circuits* (ICs). The following section introduces important aspects that need to be considered when designing a delay model that aims at covering these variations up to a certain point. Besides standard textbook knowledge, like [CB09] and [Hal20], the exposition primarily follows [YJ14].

2.3.1 Process variations

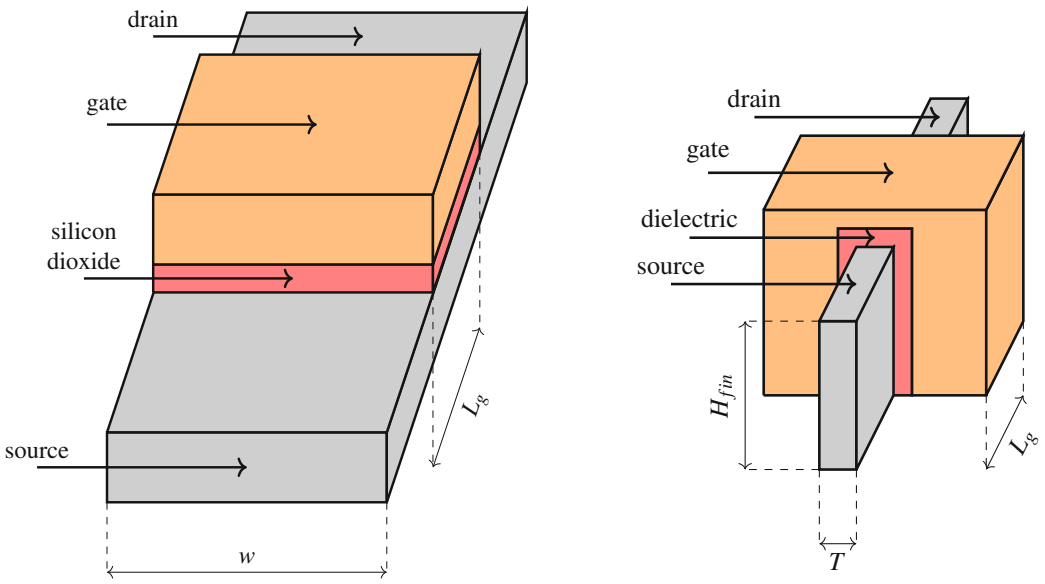
Figure 2.2a shows the typical layout of a planar *Metal-Oxid-Semiconductor Field-Effect Transistor* (MOSFET). With the ever decreasing feature size of such transistors, the process of manufacturing such transistors becomes more difficult. One reason is, that the channel between source and drain becomes smaller, and therefore it needs to be heavily doped, which in turn makes it more susceptible to *Short-Channel Effects* (SCEs).

To tackle the issue of SCEs, *Fin Field-Effect Transistors* (FinFETs) have been introduced. As can be seen in Figure 2.2b, the biggest difference is the three-dimensional structure, which allows the gate to wrap around the channel. Albeit this structure improves the resistance to process variations, FinFETs still experience intra-die and inter-die process variations. The most prominent issues are variations in the gate length, fin thickness and oxide thickness.

Typically, cell libraries provide data for various process corners, including fast, typical and slow. Moreover, some cell libraries even include data to simulate fast n-MOS / slow p-MOS and slow n-MOS / fast p-MOS. Figure 2.3a shows the qualitative impact of the process on the delay.

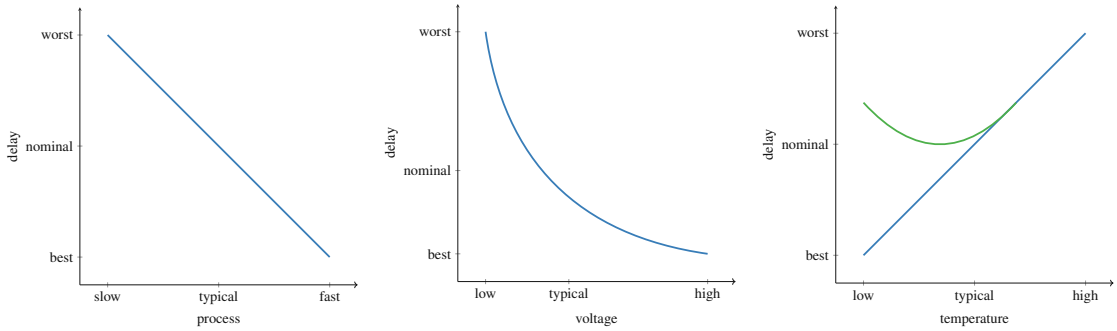
2.3.2 Voltage variation

Another source for delay variations in an IC is the supply voltage. Scaling technology has led to an reduced voltage supply level, and therefore to an increased susceptibility to power supply noise [AR13]. Building a well-balanced *Power Distribution Network* (PDN) is a demanding task. IR drop, which is caused by currents that flow through the PDN, is one of the largest sources for voltage variations in ICs. As shown in [SLD+03], these voltage variations can be as high as 15 %. Moreover, the supply voltage may also rise above its nominal voltage, and therefore variations need to be considered in both directions. Figure 2.3b shows the qualitative impact on the delay.



(a) Planar MOSFET (b) FinFET

Figure 2.2: Comparison of different *Field-Effect Transistor* (FET) layouts (adapted from [Mar20]).



(a) Process vs. delay (b) Voltage vs. delay (c) Temperature vs. delay

Figure 2.3: Impact of PVT variations on the delay (adapted from [CB09]). The green curve in (c) applies for deep sub-micron technologies.

2.3.3 Temperature variations

The junction temperature of the transistor also has an impact on the delay, as Figure 2.3c shows. The junction temperature of a transistor is not only dependent on the ambient temperature, but also on the switching frequency. An interesting effect, called *temperature inversion*, happens for deep sub-micron technologies, where the delay increases for falling temperatures.

2.4 Aging in circuits

Lorenz [Lor12] classifies aging effects into two categories: (i) effects that lead to catastrophic failures and (ii) effects that cause a parameter drift. In the following section, the most important effects of the latter category are presented, since they have an impact on the delay of circuits.

***Negative Bias Temperature Instability* (NBTI)** is one of the most severe aging effects, and only affects PMOS transistors. This effect causes an decrease in the negative threshold voltage V_{th} of the PMOS (i.e., $|V_{th}|$ increases), which decreases the performance. This effect is accelerated by an increased supply voltage and increased temperature. Alam et al. [AM05] report that NBTI is caused by broken Si-H bonds. These broken Si-H bonds can be, at least partly, recovered, which makes the NBTI effect (partly) reversible. There is no consensus yet if the effect can be fully reversed [Mas04].

***Positive Bias Temperature Instability* (PBTI)** is a similar aging effect that affects NMOS transistors. As [ZKN+06] reports, this effect is dependent on the gate dielectric: While it is negligible for SiO2 gate dielectrics, this is not the case any more for FETs with a high-k dielectric metal gate, for which the effect is in the same order of magnitude as NBTI . The PBTI effect is also (partly) reversible.

***Hot Carrier Injection* (HCI)** is an aging effect where carriers are accelerated, leave the channel, and damage the gate oxide. Both, NMOS and PMOS transistors are affected. While the first two effects are reversible, this is not the case for HCI.

2.4.1 Tool support for aging

Several tools allow to simulate the impact of aging effects on devices:

- RelXpert by Cadence [Cad20] takes an existing SPICE circuit and generates, based on the degradation parameters of the transistors, a circuit with degraded transistor models. Each element (i.e., logic gate or sequential logic) of the aged circuit is built from degraded transistors, which means that in general each transistor is now different, depending on the influence of the aging on the transistor. This tool supports several MOSFET models types. However, the reliability parameters are not necessarily specified for all models, which of course limits the applicability of the tool to specific libraries.
- Eldo by Mentor Graphics [Men05] relies on a *User Defined Reliability Model* (UDRM), where the damage that happens to a circuit during operation is modeled. Eldo supports fewer model types than RelXpert, for example BSIM4.

CHAPTER 3

Prerequisites

In this chapter, the necessary prerequisites for the IDM and its existing extensions are provided.

3.1 Involution Delay Model (IDM)

The IDM is the only known candidate for a faithful delay model and has been introduced by Függer et al. in [FNNS20]. The distinguishing property is that the delay functions $\delta_\uparrow(T)$ resp. $\delta_\downarrow(T)$ for rising resp. falling transitions are not bounded from below, i.e., the delay can become $-\infty$ for finite T. The negative delay functions form involutions, which explains the name of the IDM:

Definition 1 ([FNNS20])**.** $-\delta_\uparrow(-\delta_\downarrow(T)) = T$ and $-\delta_\downarrow(-\delta_\uparrow(T)) = T$.

Figure 3.1 shows an example for the delay functions $\delta_\uparrow(T)$ and $\delta_\downarrow(T)$. Both functions meet at the second median at $(-\delta_{min}, \delta_{min})$, which is of course no coincidence, but rather a requirement due to Definition 1, which leads to the next important property of the IDM.

Lemma 2 ([FNNS20, Lemma 2])**.** $\delta_\uparrow(-\delta_{min}) = \delta_{min} = \delta_\downarrow(-\delta_{min})$. *For strictly causal involution channels, we require* $\delta_{min} > 0$.

As Figure 3.1 reveals, the delay functions are defined in the following range:

$$\delta_\uparrow : (-\delta_\infty^\downarrow, \infty) \to (-\infty, \delta_\infty^\uparrow) \text{ and} \tag{3.1}$$

$$\delta_\downarrow : (-\delta_\infty^\uparrow, \infty) \to (-\infty, \delta_\infty^\downarrow), \tag{3.2}$$

where $\delta_\infty^\uparrow = \lim_{T\to\infty} \delta_\uparrow(T)$ and $\delta_\infty^\downarrow = \lim_{T\to\infty} \delta_\downarrow(T)$. Furthermore, the delay functions need to be differentiable and strictly monotonically increasing, which is an important property used throughout several proofs.

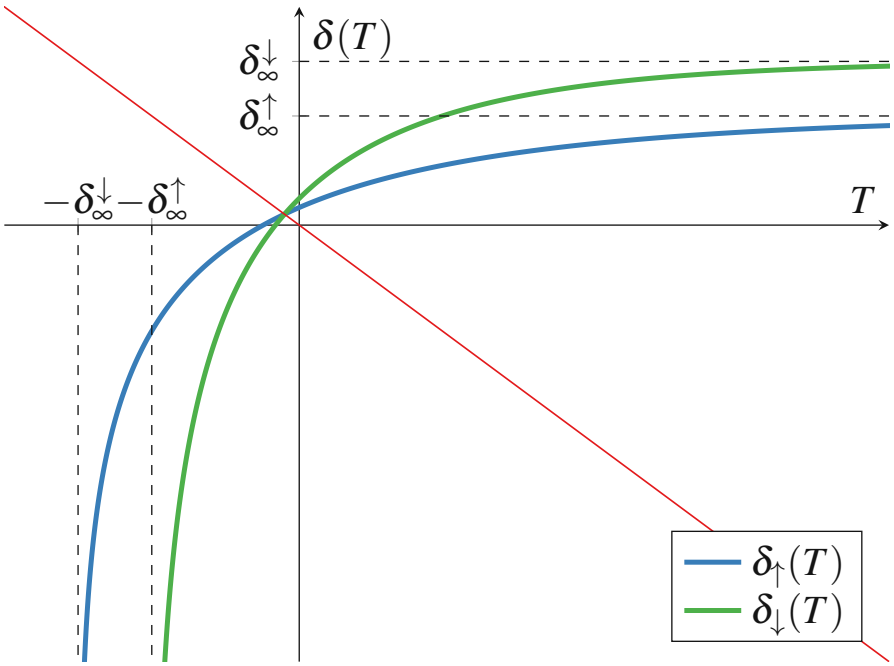


Figure 3.1: Example delay functions $\delta_\uparrow(T)$ and $\delta_\downarrow(T)$.

Note that, in the following chapters, unless otherwise noted, it will always be assumed that $\delta_{min} > 0$, i.e., that the channels are strictly causal. Monotonicity of $\delta_\uparrow$, $\delta_\downarrow$ allows us to rely on the following alternative definition:

Definition 3 ([FNNS20, Definition 1])**.** An involution channel is strictly causal if and only if:

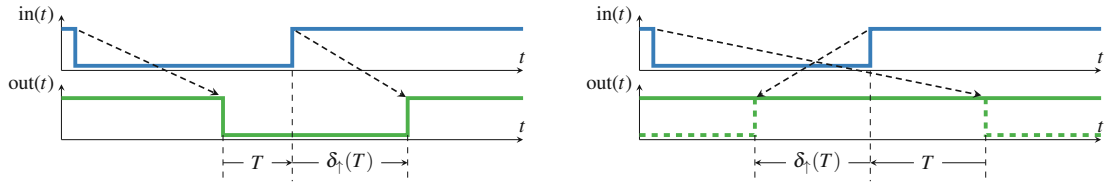
$$\delta_\uparrow(0) > 0 \iff \delta_\downarrow(0) > 0. \tag{3.3}$$

Figure 3.2 shows two examples of how the delay is calculated. In the second example, the output transitions are in the wrong temporal order, i.e., the previous falling transition is predicted after the current rising transition, and therefore these two transitions cancel each other out.

Figure 3.3 shows an analog channel model corresponding to the IDM, which shows that using self-inverse delay functions is indeed reasonable. At first, the input u_i is delayed by a pure delay δ_{min}. The delayed signal u_d is then shaped by a slew rate limiter, which applies the corresponding switching waveform upon a transition ($f^\uparrow/f^\downarrow$). Finally, a comparator is used to digitize the signal again.

3.1.1 Short-Pulse Filtration (SPF) problem

The SPF problem is the task of building a single input single output circuit that filters short pulses. Pulses with a width $\leq \varepsilon$ shall be filtered, whereas longer pulses are passed



(a) Based on the previous-output-to-input time T the delay $\delta_\uparrow(T)$ is calculated.

(b) In this case $\delta_\uparrow(T) \leq -T$, which leads to a cancellation on the output.

Figure 3.2: Two example traces for calculating the delay, based on the previous-output-to-input time (adapted from [ÖMFS21]).

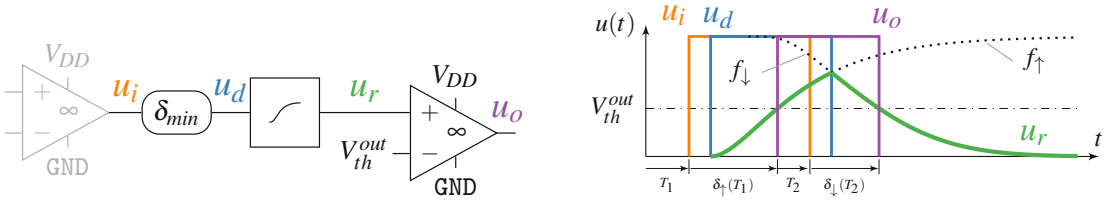


Figure 3.3: Analog channel model (left part) with an example trace (right part) (taken from [MÖS+21])

through. However, longer pulses may be altered, e.g. they might stay at logical one, even if the input goes back to logical zero. Note that the behavior is only specified for the zero input signal and signals with exactly one pulse. The bounded version of SPF has the additional requirement that the output needs to settle within bounded time $K > 0$ after the last input transition.

Definition 4 ([FNNS20])**.** A circuit that solves SPF needs to fulfill the following conditions:

F1) Well-formedness: The circuit has exactly one input port and exactly one output port.

F2) No generation: If the input signal is the zero signal, then so is the output signal.

F3) Nontriviality: There exists an input pulse such that the output signal is not the zero signal.

F4) No short pulses: There exists and $\varepsilon > 0$ such that for every input pulse the output signal never contains a pulse of length less than or equal to ε.

Moreover, a circuit solves unbounded SPF, if the following additional condition is fulfilled:

F5) Bounded stabilization time: There exists a $K > 0$ such that for every input pulse the last output transition is before time $T + K$, where T is the time of the last input transition.

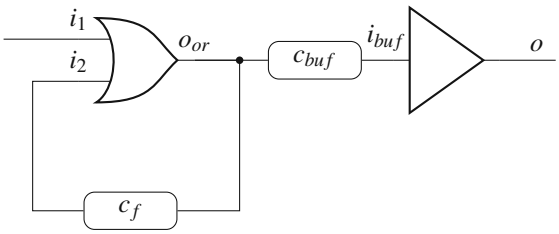


Figure 3.4: A circuit with an OR gate, which is fed-back via an involution channel (c_f). At the output o_{or}, a high-threshold buffer, represented by an IDM-channel (c_{buf}) and a buffer, is located.

Függer et al. [FNNS20] showed that the IDM indeed allows to solve unbounded SPF, while it does not allow to solve the bounded version. Hence, it behaves exactly like physical reality, and is therefore a candidate for a faithful delay model.

Theorem 5 ([FNNS20, Theorem 3])**.** *There is a circuit that solves unbounded SPF.*

Figure 3.4 shows the circuit that is used to implement unbounded SPF in the IDM. The general idea of the proof is to split the input pulse width Δ_0 on i_1 into three ranges, and study the behavior of the circuit in each case. A similar approach will be described in more detail for the η-IDM in Section 3.3.

Since the impossibility proof of bounded SPF is rather lengthy, and not required in detail for the following chapters, the interested reader is referred to [FNNS20].

Theorem 6 ([FNNS20, Theorem 7])**.** *No circuit solves bounded SPF.*

3.2 Composable Involution Delay Model (CIDM)

The CIDM is an extension of the IDM that has been published in [MÖS+21], primarily to make the characterization of the delay functions of the gates in a circuit easier. In general, fixing a threshold voltage V_{th}^{out*} at the output of a gate yields a *unique* corresponding V_{th}^{in*} and δ_{min}. This makes the characterization tedious, since the obtained V_{th}^{in*} automatically determines the output threshold voltage of the previous gate.

The goal of the CIDM is to use a single threshold voltage v_{th} for the characterization of every gate in a circuit. By rearranging the components of the analog channel model from Figure 3.3 and introducing a pure delay shifter $\Delta^{+/-}$, the model, shown in Figure 3.5 can be obtained. Note that the comparator has been split into a thresholder Th and a cancellation unit C that drops out-of-order transitions.

Figure 3.5 reveals that the resulting model can be viewed as the concatenation of a pure delay shifter P ($\Delta^{+/-}$) and an involution channel I with delay functions $\overline{\delta}_{\uparrow}(.)$ and $\overline{\delta}_{\downarrow}(.)$. Therefore, the resulting channel will be called PI channel in the sequel. It turns out that

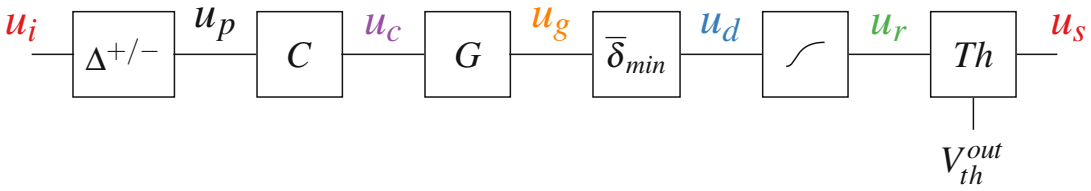


Figure 3.5: Model of a CIDM channel.

the resulting delay functions are not involutions, but are rather defined as:

$$\delta_\uparrow(T) = \Delta^+ + \overline{\delta}_\uparrow(T + \Delta^+), \tag{3.4}$$
$$\delta_\downarrow(T) = \Delta^- + \overline{\delta}_\downarrow(T + \Delta^-). \tag{3.5}$$

However, when concatenating two PI channels, the middle part forms an IP channel, made up by an involution channel and a subsequent pure delay shifter. This channel is, surprisingly, again an involution channel, as the following theorem shows.

Theorem 7 (IP channel properties, [MÖS+21, Theorem 10])**.** *An IP channel formed by an involution channel $\overline{\delta}_\uparrow(.), \overline{\delta}_\downarrow(.)$ followed by a pure delay shifter (Δ^+, Δ^-) can be characterized by an involution channel with the following delay functions:*

$$\delta_\uparrow(T) = \overline{\delta}_\uparrow(T + \Delta^-) + \Delta^+ \tag{3.6}$$
$$\delta_\downarrow(T) = \overline{\delta}_\downarrow(T + \Delta^+) + \Delta^-. \tag{3.7}$$

The IP channel in general has a different δ_{min} than the plain involution channel, which is given by the smallest positive solution of

$$\delta_{min} = \overline{\delta}_\uparrow(-\delta_{min} + \Delta^-) + \Delta^+ = \overline{\delta}_\downarrow(-\delta_{min} + \Delta^+) + \Delta^-. \tag{3.8}$$

Proof. Consider the rising transition of a single negative pulse as shown in Figure 3.6. It can be seen, that the overall delay consists of the delay of the involution channel and the pure delay:

$$\delta_\uparrow(T) = \overline{\delta}_\uparrow(\overline{T}) + \Delta^+ \text{ with } \overline{T} = T + \Delta^-. \tag{3.9}$$

Rearranging yields:

$$\delta_\uparrow(\overline{T} - \Delta^-) = \overline{\delta}_\uparrow(\overline{T}) + \Delta^+ \tag{3.10}$$
$$\overline{\delta}_\uparrow(\overline{T}) = \delta_\uparrow(\overline{T} - \Delta^-) - \Delta^+. \tag{3.11}$$

Performing a similar calculation for a falling transition yields:

$$\delta_\downarrow(T) = \overline{\delta}_\downarrow(\overline{T}) + \Delta^- \text{ with } \overline{T} = T + \Delta^+ \tag{3.12}$$
$$\delta_\downarrow(\overline{T} - \Delta^+) = \overline{\delta}_\downarrow(\overline{T}) + \Delta^- \tag{3.13}$$
$$\overline{\delta}_\downarrow(\overline{T}) = \delta_\downarrow(\overline{T} - \Delta^+) - \Delta^-. \tag{3.14}$$

Next, the involution property of $\overline{\delta}_\uparrow(.)$ and $\overline{\delta}_\downarrow(.)$ is used:

$$\begin{aligned}\overline{T} &= -\overline{\delta}_\uparrow(-\overline{\delta}_\downarrow(\overline{T})) && (3.15)\\ &= -\delta_\uparrow(-\overline{\delta}_\downarrow(\overline{T}) - \Delta^-) + \Delta^+ && (3.16)\\ &= -\delta_\uparrow(-\delta_\downarrow(\overline{T} - \Delta^+) + \Delta^- - \Delta^-) + \Delta^+ && (3.17)\\ &= -\delta_\uparrow(-\delta_\downarrow(\overline{T} - \Delta^+)) + \Delta^+. && (3.18)\end{aligned}$$

By substituting $T = \overline{T} - \Delta^+$, it can be seen that $\delta_\uparrow(.)$ and $\delta_\downarrow(.)$ satisfy the involution property:

$$T = -\delta_\uparrow(-\delta_\downarrow(T)) \tag{3.19}$$

When starting with the reversed involution property of $\overline{\delta}_\uparrow(.)$ and $\overline{\delta}_\downarrow(.)$

$$\begin{aligned}\overline{T} &= -\overline{\delta}_\downarrow(-\overline{\delta}_\uparrow(\overline{T})) && (3.20)\\ &= -\delta_\downarrow(-\overline{\delta}_\uparrow(\overline{T}) - \Delta^+) + \Delta^- && (3.21)\\ &= -\delta_\downarrow(-\delta_\uparrow(\overline{T} - \Delta^-) + \Delta^+ - \Delta^+) + \Delta^- && (3.22)\\ &= -\delta_\downarrow(-\delta_\uparrow(\overline{T} - \Delta^-)) + \Delta^- && (3.23)\end{aligned}$$

and substituting $T = \overline{T} - \Delta^-$ the reversed involution property of $\delta_\uparrow(.)$ and $\delta_\downarrow(.)$ can be shown:

$$T = -\delta_\downarrow(-\delta_\uparrow(T)). \tag{3.24}$$

The resulting δ_{min} of the IP channel can be calculated by inserting δ_{min} in Eq. (3.6) and Eq. (3.7):

$$\begin{aligned}\delta_{min} = \delta_\uparrow(-\delta_{min}) &= \overline{\delta}_\uparrow(-\delta_{min} + \Delta^-) + \Delta^+ && (3.25)\\ \delta_{min} = \delta_\downarrow(-\delta_{min}) &= \overline{\delta}_\downarrow(-\delta_{min} + \Delta^+) + \Delta^-. && (3.26)\end{aligned}$$

□

Using IP channels leaves an I channel at each output and a pure delay shifter at each input. While the former is obviously no issue, it needs to be argued why the pure delay shifter at each input does not impair the applicability of the CIDM. By simply requiring that the outermost input port must have a threshold voltage matching the external input signal, the pure delay shifter reduces to $\Delta^+ = \Delta^- = 0$, and hence does not impair applicability. Alternatively, assuming that an input is actually driven by (the I channel of) some output, we end up with a regular IP channel also here.

Since the focus of this work is on strictly causal channels, the following lemma defines the necessary conditions for strictly causal channels:

Lemma 8 (Causality of PI channels, [MÖS+21])**.** *A PI channel is strictly causal if and only if* Δ^+ *and* Δ^- *satisfy*

$$\delta_\uparrow(0) = \Delta^+ + \overline{\delta}_\uparrow(\Delta^-) > 0 \iff \delta_\downarrow(0) = \Delta^- + \overline{\delta}_\downarrow(\Delta^+) > 0. \tag{3.27}$$

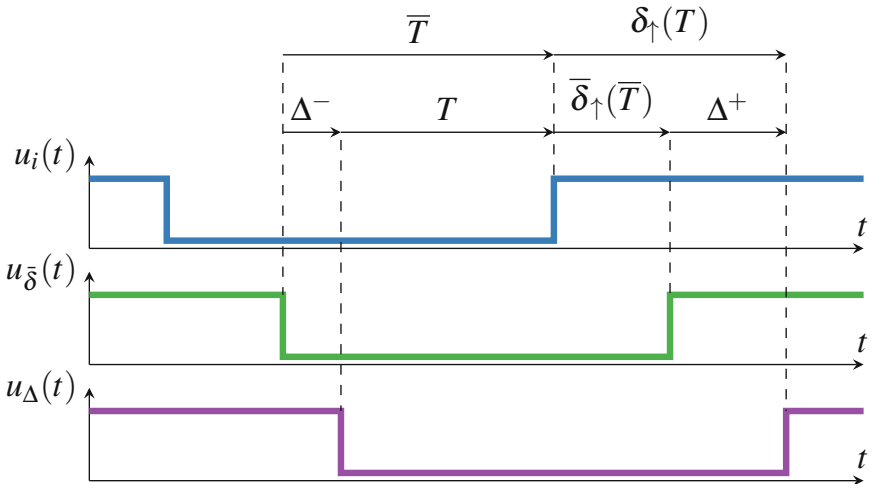


Figure 3.6: Example trace through an IP channel.

Proof. From Definition 3 it is know that an involution channel is causal if and only if $\delta_\uparrow(0) > 0 \iff \delta_\downarrow(0) > 0$. Applying Eqs. (3.6) and (3.7) yields $\delta_\uparrow(0) = \Delta^+ + \overline{\delta}_\uparrow(\Delta^-) > 0 \iff \delta_\downarrow(0) = \Delta^- + \overline{\delta}_\downarrow(\Delta^+) > 0$, which concludes the proof. □

The resulting channel is strictly causal if and only if $\delta_{min} > 0$. Unless otherwise noted, all plots in Chapters 3 to 5 use the delay functions of a representative 28 nm CMOS technology with the following empirically determined parameters: $\overline{\delta}_{min} \approx 193.7\,\text{fs}, \overline{\delta}^{\uparrow}_{\infty} \approx 3162\,\text{fs}, \overline{\delta}^{\downarrow}_{\infty} \approx 2967\,\text{fs}, \Delta^+ \approx 95.31\,\text{fs}$, and $\Delta^- \approx -101.2\,\text{fs}$, which results in overall delay functions with the parameters: $\delta_{min} = 191.0\,\text{fs}, \delta^{\uparrow}_{\infty} = \overline{\delta}^{\uparrow}_{\infty} + \Delta^+ \approx 3258\,\text{fs}$, and $\delta^{\downarrow}_{\infty} = \overline{\delta}^{\downarrow}_{\infty} + \Delta^- \approx 2865\,\text{fs}$. Figure 3.7 shows the value of δ_{min} for different combinations of (Δ^+, Δ^-).

3.3 η-Involution Delay Model (η-IDM)

The η-IDM [FMN+20] is an extension that aims at covering arbitrary delay variations, e.g. caused by PVT variations. On top of the existing delay prediction from the IDM, it allows to add non-deterministic delay variations within the range $[-\eta^-_{min}, \eta^+_{min}]$ for every transition. These delay variations can be random or even adversarial. Figure 3.8 shows an example trace $in(t)$ and two potential output traces $out_1(t)$ and $out_2(t)$. Note that there are infinitely many output traces corresponding to an input trace. The figure also reveals that depending on the choice of the delay variations, pulses might occur on some of the output traces, while they are canceled on other output traces.

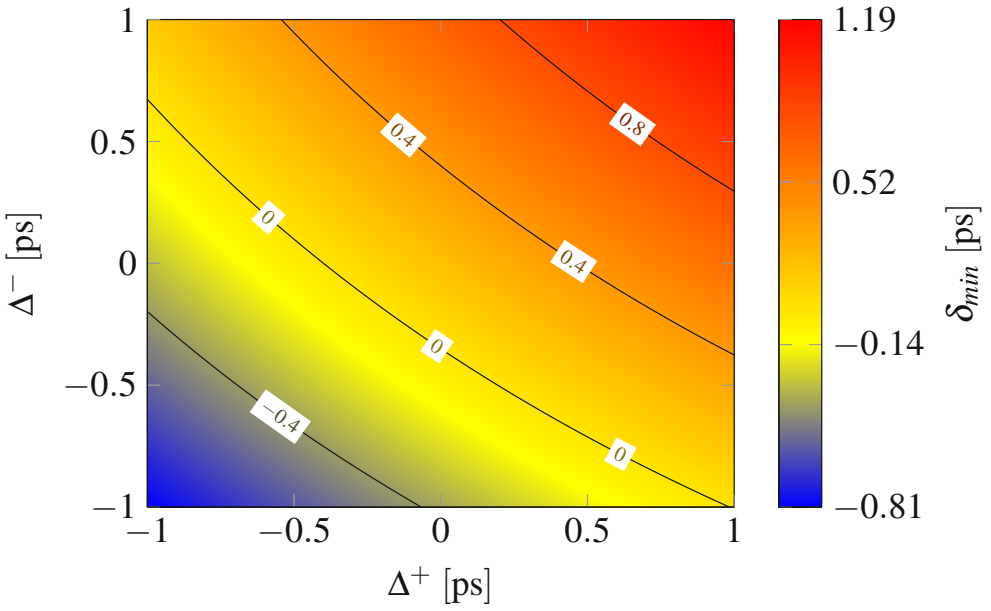


Figure 3.7: δ_{min} for different combinations of (Δ^+, Δ^-).

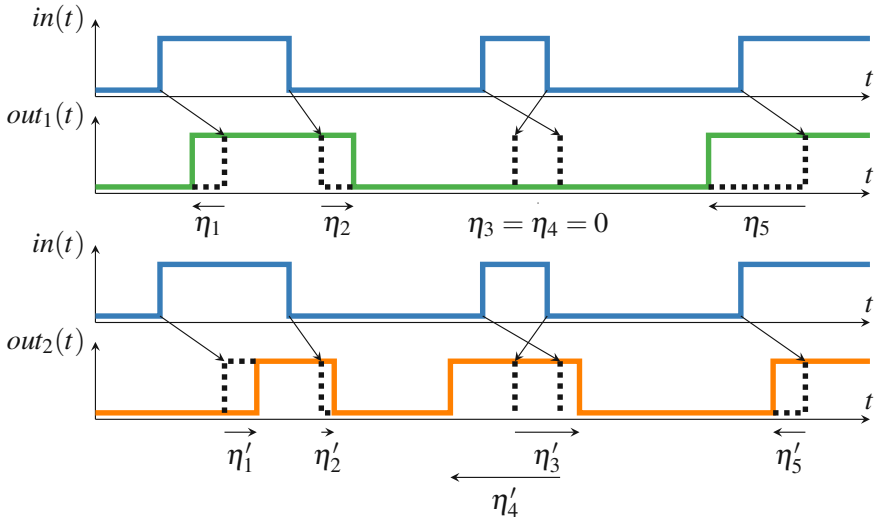


Figure 3.8: Example trace through an η-involution channel (adapted from [FMN+18]).

The delay δ_n for the n^{th} transition is calculated as follows:

$$\delta_n = \delta_\uparrow(\max\{t_n - t_{n-1} - \delta_{n-1}, -\delta^\downarrow_\infty\}) + \eta_n \qquad \text{for a rising transition, and} \tag{3.28}$$

$$\delta_n = \delta_\downarrow(\max\{t_n - t_{n-1} - \delta_{n-1}, -\delta^\uparrow_\infty\}) + \eta_n \qquad \text{for a falling transition,} \tag{3.29}$$

where t_n is the time of n^{th} input transition and $\eta_n \in [-\eta^-_{min}, \eta^+_{min}]$ is the delay variation chosen by the adversary.

Note that the max terms are necessary if the adversary shifts a transition such that the resulting previous-output-to-input time T becomes so small that $\delta_\uparrow(.)$ and $\delta_\downarrow(.)$ are not defined anymore. This can happen if there is a very short pulse after a long idle period: Suppose the first (rising) transition gets delayed by $\approx \delta^\uparrow_\infty + \eta^+_{min}$. Then obviously $T \approx -\delta^\uparrow_\infty - \eta^+_{min}$, which might be smaller than $-\delta^\downarrow_\infty$. Basically, the max terms ensure that the delay functions are continued to the left as ∞. However, since they are not necessary for the following proof sketches, they are omitted in the following.

For showing that η-IDM cannot solve bounded SPF, it suffices to find one configuration of the adversary for which this is the case. The most trivial configuration is of course setting all $\eta_n = 0$. By doing so, η-IDM degenerates to plain IDM, for which it is known that bounded SPF cannot be solved [FNNS20]. However, note carefully that it might be the case that there are particular choices for the adversary, for which it is possible to solve bounded SPF.

Theorem 9 (Impossibility of bounded SPF, [FMN+18])**.** *There is no circuit that uses η-IDM channels which solves bounded SPF.*

In order to show that η-IDM allows to solve unbounded SPF, the circuit in Figure 3.9 is employed. It is similar to the circuit in Figure 3.9; however, instead of using IDM channels, η-IDM channels are used. The general idea is to consider the length of the input pulse Δ_0 and distinguish three cases:

(i) $\Delta_0 \geq \delta^\uparrow_\infty + \eta^+_{min}$: For such large pulses, a unique rising transition at time 0 happens at the output (see [FMN+20, Lemma 3])

(ii) $\Delta_0 \leq \delta^\uparrow_\infty - \delta_{min} - \eta^+_{min} - \eta^-_{min}$: For such short pulses, the output only contains the input pulse (see [FMN+20, Lemma 4])

(iii) $\delta^\uparrow_\infty - \delta_{min} - \eta^+_{min} - \eta^-_{min} < \Delta_0 < \delta^\uparrow_\infty + \eta^+_{min}$: This is the most delicate case and therefore presented in more detail in the following.

The general idea for analyzing the behavior of the SPF circuit for input pulses with medium length (iii) is to find an infinite self-repeating "worst-case pulse train". This is similar to the possibility proof for IDM (see [FNNS20]); however, with the difference that, in general, there is not just exactly one "self-repeating" pulse train but rather an infinite number of those, with, in general, varying pulse lengths. In the worst-case pulse train, the adversary is forced to take all rising transitions maximally late (η^+_{min}) and all falling transitions as early as possible (η^-_{min}). This choice minimizes the pulse length of the next pulse Δ_n for a given current pulse Δ_{n-1}. By doing so, the pulse length of the next pulse can be calculated as follows:

$$\Delta_n = f(\Delta_{n-1}) = \Delta_{n-1} + \delta_\downarrow\Big(- \delta_\uparrow(-\Delta_{n-1}) - \eta^+_{min} + \Delta_{n-1}\Big) - \eta^-_{min} - \delta_\uparrow(-\Delta_{n-1}) - \eta^+_{min} \tag{3.30}$$

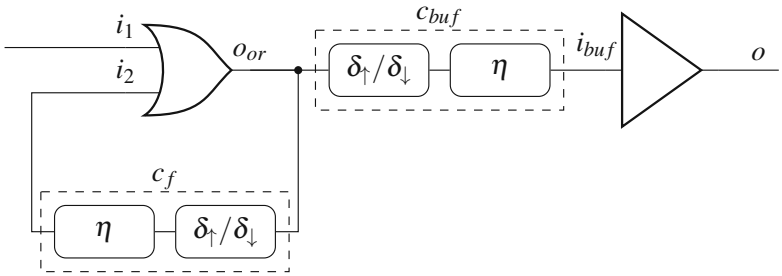


Figure 3.9: A circuit with an OR gate, which is fed-back via an η-IDM channel (c_f). At the output o_{or}, a high-threshold buffer, represented by an η-IDM channel (c_{buf}) and a buffer, is located.

The critical pulse length of the worst-case pulse train can be determined by finding the fixed point Δ of the above equation. Since $f'(\Delta) > 1$, it is guaranteed that there is no larger fixed point. Informally, once the pulse length exceeds Δ, the pulse grows until it reaches $\delta_{min} + \eta^+_{min}$, which locks the storage loop. However, the constraint $\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}$ is required in order to guarantee the existence of a solution.

[FMN+20, Lemma 5] guarantees that there is such a fixed point $0 < \Delta < \delta_{min}$, and that pulse train has a certain period P and a duty cycle Γ. By knowing these parameters of the circuit, the high-threshold buffer can be dimensioned accordingly, see [FNNS20, Lemma 10, Lemma 11]. The high-threshold buffer is responsible for mapping decreasing pulse trains and also infinite pulse trains to constant zero at the output. For increasing pulse trains, the high-threshold buffer generates a single rising transition at the output.

Theorem 10 (Faithfulness of η-IDM, [FMN+18])**.** *Adding non-deterministic delay variations within $\eta \in [-\eta^-_{min}, \eta^+_{min}]$, with $\eta^+_{min}+\eta^-_{min} < \delta_\downarrow(-\eta^+_{min})-\delta_{min}$, to an involution channel $\delta_\uparrow(.), \delta_\downarrow(.)$ again results in a faithful delay model.*

Corollary 11. Due to the constraint from Theorem 10, it holds that $\eta^+_{min} < \delta_{min}$.

Proof. By simply looking at the RHS of the constraint $\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}$, it can be seen that the RHS would become smaller than or equal to 0 if $\eta^+_{min} \geq \delta_{min}$. However, since $\eta^+_{min} \geq 0$ and $\eta^-_{min} \geq 0$, this would be a contradiction □

Corollary 12. Due to the constraint from Theorem 10, it holds that $\eta^-_{min} < \delta_\downarrow(0) - \delta_{min}$.

Proof. By rearranging the constraint and setting $\eta^+_{min} = 0$, we obtain

$$\eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \eta^+_{min} - \delta_{min} \leq \delta_\downarrow(0) - \delta_{min} \tag{3.31}$$

by monotonicity of $\delta_\downarrow$ as asserted. □

The blue line in Figure 3.10 shows an example for the border of all possible pairs η^+_{min} and η^-_{min} and is based on the delay functions for an 28 nm inverter. Although the border looks linear, it is not the case (which can be easily checked by looking at the constraint). The dashed blue area is the region that contains all pairs $(\eta^+_{min}, \eta^-_{min})$ that fulfill the constraint. Moreover, the upper bound δ_{min} for η^+_{min} is depicted by the dashed red pointed line (Corollary 11). The upper bound $\delta_{\downarrow}(0) - \delta_{min}$ for η^-_{min} is shown by the dotted red line (Corollary 12). For this configuration, the maximum value (i.e., the configuration that allows the largest deviations in total) for $\eta^+_{min} + \eta^-_{min}$ is when $\eta^+_{min} = 0$, as indicated by the lilac square. However, note that this is not always the case in general and depends heavily on the delay functions. In Chapter 6, η^+_{min} and η^-_{min} are chosen such that they are approximately equal (indicated by the orange triangle in the figure). While this approach generates a symmetric band around the delay function, it has the the drawback that it does not maximize the sum $\eta^+_{min} + \eta^-_{min}$ in general.

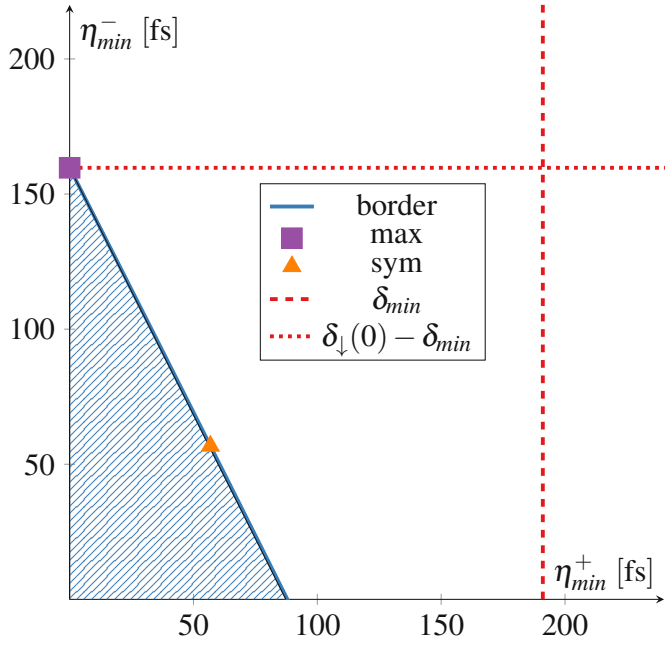


Figure 3.10: Example of allowed combinations of η^+_{min} and η^-_{min} (dashed blue area).

CHAPTER 4

Combining the CIDM and the η-IDM

In this chapter, it is first shown that an η-CIDM channel can be reduced to an η-IDM channel. Next, useful properties are derived which are required to show that η-CIDM is still faithful. Finally, the constraints of an η-CIDM and η-IDM channel are compared.

4.1 Reduction from η-CIDM to η-IDM

The idea of the proof is to again use the fact that an IP channel is an involution channel, as shown in Theorem 7. The only difference is that the resulting channel in general has different parameters δ_{min}, $\delta_{\infty}^{\uparrow}$ and $\delta_{\infty}^{\downarrow}$.

By adding non-deterministic delay variations on top of the newly derived IP channel, an η-IDM channel is obtained, according to Theorem 10. The general structure of the resulting IPη channel and an example trace can be seen in Figure 4.1 and Figure 4.2.

Theorem 13 (IPη channel properties)**.** *An IPη channel formed by an involution channel $\overline{\delta}_{\uparrow}(.), \overline{\delta}_{\downarrow}(.)$ followed by a pure delay shifter (Δ^+, Δ^-) and an adversary that picks delay variations $\eta_n \in [-\eta_{min}^-, \eta_{min}^+], \eta_{min}^- \geq 0, \eta_{min}^+ \geq 0$, can be reduced to an η-IDM channel. The delay for the n-th transition is calculated as follows:*

$$\delta_n = \delta_{\uparrow}(\max\{T_n, -\delta_{\infty}^{\downarrow}\}) + \eta_n \qquad \textit{for a rising transition} \tag{4.1}$$

$$\delta_n = \delta_{\downarrow}(\max\{T_n, -\delta_{\infty}^{\uparrow}\}) + \eta_n \qquad \textit{for a falling transition,} \tag{4.2}$$

where $\delta_{\uparrow}(.)$ and $\delta_{\downarrow}(.)$ are the delay functions for the concatenation of involution channel and pure delay shifter, $T_n = t_n - t_{n-1} - \delta_{n-1}$, t_n is the time of the n-th input transition, and $\eta_{min}^+ + \eta_{min}^- < \delta_{\downarrow}(-\eta_{min}^+) - \delta_{min}$.

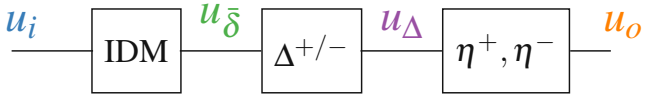


Figure 4.1: Concatenation of an IDM channel and a pure delay shifter with an adversary.

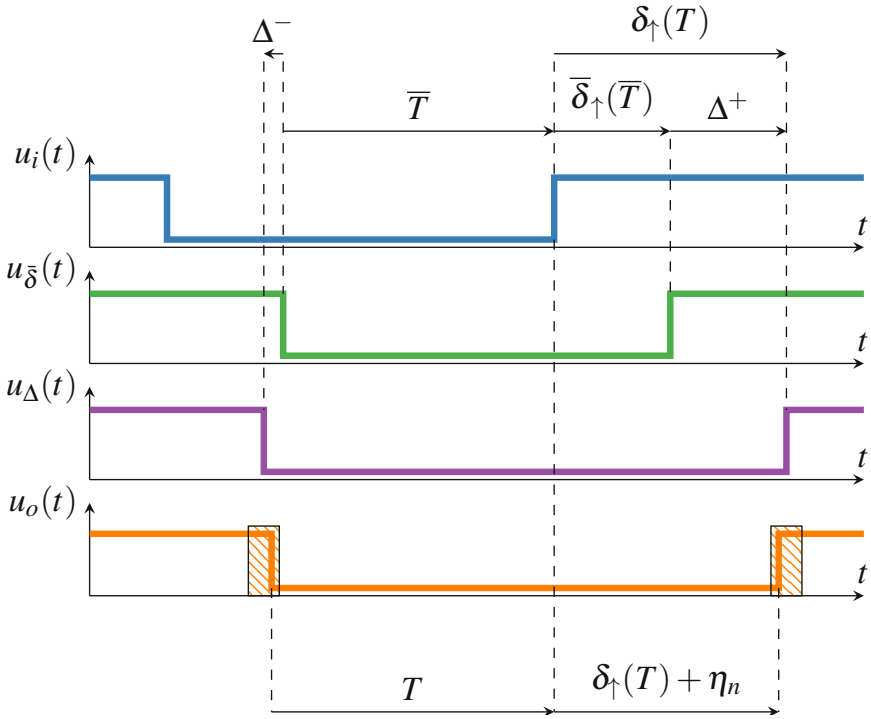


Figure 4.2: Example trace through an IPη channel, as shown in Figure 4.1. The dashed orange rectangle indicates the possible range for delay variations. In this example, $\Delta^+ > 0$ and $\Delta^- < 0$.

Proof. From Theorem 7, it is known that an IP channel can be reduced to an IDM channel.

By using Theorem 10, it is known that an IDM channel with delay variations $\eta_n \in [-\eta^-_{min}, \eta^+_{min}]$, subject to the constraint $\eta^+_{min} + \eta^-_{min} - < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}$, results in an η-IDM channel.

Hence, an IPη channel can be reduced to an η-IDM channel. □

4.2 Interchangeability of delay adversary and pure delay shifter

In this section, it will be shown that the pure delay shifter and the adversary can be interchanged.

Lemma 14. *An IPη channel and an IηP channel are equivalent, if they have the same $\overline{\delta}_\uparrow(.), \overline{\delta}_\downarrow(.)$ and Δ^+, Δ^-.*

Proof. The delay δ_n for the n-th transition of an IPη channel is known from Eqs. (4.1) and (4.2) in Theorem 13. The delay δ_n for a rising transition of an IηP channel is calculated as follows, using the notation of Figures 3.6 and 3.7:

$$\begin{aligned}
\delta_n &= \overline{\delta}_\uparrow(\max\{\overline{T_n}, -\overline{\delta}^\downarrow_\infty\}) + \eta_n + \Delta^+ && (4.3)\\
&= \delta_\uparrow(\max\{\overline{T_n}, -\overline{\delta}^\downarrow_\infty\} - \Delta^-) + \eta_n && (4.4)\\
&= \delta_\uparrow(\max\{T_n + \Delta^-, -\delta^\downarrow_\infty + \Delta^-\} - \Delta^-) + \eta_n && (4.5)\\
&= \delta_\uparrow(\max\{T_n, -\delta^\downarrow_\infty\}) + \eta_n && (4.6)
\end{aligned}$$

Figure 4.3 illustrates the above calculations. The result is the same as Eq. (4.1) in Theorem 13. A similar calculation shows that this is also true for the delay of a falling transition. □

Actually, the interchangeability of the delay adversary and the pure delay shifter is not surprising, since the delay adversary behaves like a generalized version of a pure delay shifter (the delay is different for each transition, whereas for a pure delay shifter the delay is always the same for all rising resp. falling transitions). From this it also immediately follows that multiple pure delay shifters that are in series can be interchanged (as long as there is no cancellation unit between the pure delay shifters).

4.3 Relationship between Δ^+ and Δ^- for perfect matchings

According to [MÖS+21], a matching is called perfect if Δ^+ and Δ^- are chosen such that $\overline{\delta}_{min} = \delta_{min}$. Such a perfect choice is always possible if the actual switching waveform of the predecessor gate is used to characterize the delay function of the current gate. Figure 4.4 shows the pairs of (Δ^+, Δ^-) that are a perfect match for our 28 nm inverter. Note that there are infinitely many such pairs.

According to Eq. (3.8), the values for perfect matchings of Δ^+ and Δ^- are calculated as follows:

$$\begin{aligned}
\Delta^+ &= \overline{\delta}_{min} - \overline{\delta}_\downarrow(-\overline{\delta}_{min} + \Delta^-) \text{ and} && (4.7)\\
\Delta^- &= \overline{\delta}_{min} - \overline{\delta}_\uparrow(-\overline{\delta}_{min} + \Delta^+). && (4.8)
\end{aligned}$$

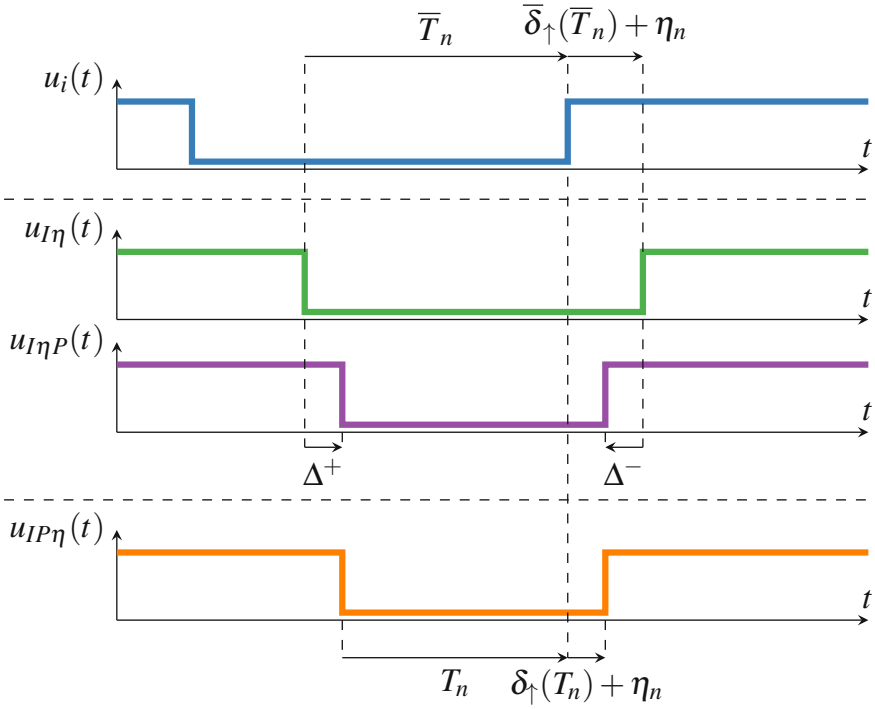


Figure 4.3: Example trace through an IPη channel (orange) and an IηP channel (lilac), which is used to illustrate the proof for the interchangeability of the delay adversary and the pure delay shifter. For simplicity, the max terms are omitted. In this example $\Delta^+ > 0$ and $\Delta^- < 0$.

These equations show that for strictly increasing Δ^+ the value for Δ^- is strictly decreasing (and vice versa), due to the strict monotonicity of $\overline{\delta}_\uparrow$ and $\overline{\delta}_\downarrow$. The values for Δ^+ and Δ^- have no upper bound, and it can be seen that the corresponding value for Δ^- resp. Δ^+ reaches a bounded minimum:

$$\Delta^+_{min} = \lim_{\Delta^-\to\infty} \overline{\delta}_{min} - \overline{\delta}_\downarrow(-\overline{\delta}_{min} + \Delta^-) = \overline{\delta}_{min} - \overline{\delta}^\downarrow_\infty \text{ and} \tag{4.9}$$

$$\Delta^-_{min} = \lim_{\Delta^+\to\infty} \overline{\delta}_{min} - \overline{\delta}_\uparrow(-\overline{\delta}_{min} + \Delta^+) = \overline{\delta}_{min} - \overline{\delta}^\uparrow_\infty. \tag{4.10}$$

These lower bounds are depicted by the dashed green lines in Figure 4.4.

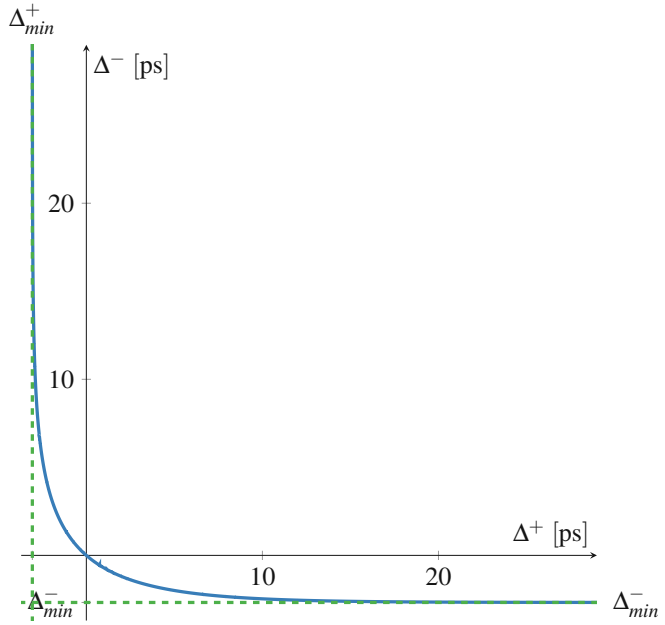


Figure 4.4: Combinations of Δ^+ and Δ^- for which the matching is perfect, i.e., $\overline{\delta}_{min} = \delta_{min}$.

4.4 Relationship between δ_{min} and $\overline{\delta}_{min}$

The goal of this section is to explore the relationship between δ_{min} and $\overline{\delta}_{min}$ in the general case. By setting $X = \overline{\delta}_{min} - \delta_{min}$, we find

$$\begin{aligned} X &= \overline{\delta}_{min} - \delta_{min} && (4.11)\\ &= \overline{\delta}_{min} - \delta_{\uparrow}(-\delta_{min}) && (4.12)\\ &= \overline{\delta}_{min} - \delta_{\uparrow}(-\overline{\delta}_{min} + X) && (4.13)\\ &= \overline{\delta}_{min} - \overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + X + \Delta^-) - \Delta^+ && (4.14) \end{aligned}$$

and hence

$$\overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + X + \Delta^-) + X = \overline{\delta}_{min} - \Delta^+. \quad (4.15)$$

By using the delay functions for the falling transition, a similar equation can be derived:

$$\overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + X + \Delta^+) + X = \overline{\delta}_{min} - \Delta^- \quad (4.16)$$

Unfortunately, the implicit functions Eq. (4.15) and Eq. (4.16) cannot be solved explicitly for X.

However, rearranging yields:

$$\begin{aligned} \delta_{min} &= \overline{\delta}_{min} - X = \overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + X + \Delta^-) + \Delta^+ \text{ and} && (4.17)\\ \delta_{min} &= \overline{\delta}_{min} - X = \overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + X + \Delta^+) + \Delta^-. && (4.18) \end{aligned}$$

Combining Eq. (4.17) and Eq. (4.18) yields

$$\overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + X + \Delta^-) + \Delta^+ = \overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + X + \Delta^+) + \Delta^-, \tag{4.19}$$

which is visualized in Figure 4.5. Moreover, the figure shows how a perfect matching can be obtained for given delay functions $\overline{\delta}_{\uparrow}(.), \overline{\delta}_{\downarrow}(.)$ and Δ^+, Δ^-: Consider the delay channel consisting of $\overline{\delta}_{\uparrow}(.), \overline{\delta}_{\downarrow}(.)$ and a pure delay shifter with $\hat{\Delta}^+$ and $\hat{\Delta}^-$, where

$$\hat{\Delta}^+ = \Delta^+ + X \text{ and} \tag{4.20}$$
$$\hat{\Delta}^- = \Delta^- + X. \tag{4.21}$$

The resulting delay functions are now:

$$\hat{\delta}_{\uparrow}(T) = \hat{\Delta}^+ + \overline{\delta}_{\uparrow}(T + \hat{\Delta}^-) \text{ and} \tag{4.22}$$
$$\hat{\delta}_{\downarrow}(T) = \hat{\Delta}^- + \overline{\delta}_{\downarrow}(T + \hat{\Delta}^+). \tag{4.23}$$

Rearranging Eqs. (4.17) and (4.18) reveals that the pure delay $\hat{\delta}_{min}$ of the newly built channel equals $\overline{\delta}_{min}$; hence, a perfect matching has been found: Since

$$\delta_{min} = \overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + X + \Delta^-) + \Delta^+ = \overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + X + \Delta^+) + \Delta^- \tag{4.24}$$
$$= \overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + \hat{\Delta}^-) + \hat{\Delta}^+ - X = \overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + \hat{\Delta}^+) + \hat{\Delta}^- - X, \tag{4.25}$$

we find

$$\overline{\delta}_{min} = \delta_{min} + X = \overline{\delta}_{\uparrow}(-\overline{\delta}_{min} + \hat{\Delta}^-) + \hat{\Delta}^+ = \overline{\delta}_{\downarrow}(-\overline{\delta}_{min} + \hat{\Delta}^+) + \hat{\Delta}^- \tag{4.26}$$
$$= \hat{\delta}_{\uparrow}(-\overline{\delta}_{min}) = \hat{\delta}_{\downarrow}(-\overline{\delta}_{min}), \tag{4.27}$$

which implies

$$\overline{\delta}_{min} = \hat{\delta}_{min}. \tag{4.28}$$

However, note that the resulting delay functions $\hat{\delta}_{\uparrow}(.)$ and $\hat{\delta}_{\downarrow}(.)$ look in general different, since

$$\hat{\delta}^{\uparrow}_{\infty} = \overline{\delta}^{\uparrow}_{\infty} + \hat{\Delta}^+ \tag{4.29}$$
$$\hat{\delta}^{\downarrow}_{\infty} = \overline{\delta}^{\downarrow}_{\infty} + \hat{\Delta}^- \tag{4.30}$$

are usually different.

Since it is not possible to compute $X = \overline{\delta}_{min} - \delta_{min}$ in general, Figure 4.6 might help to get an intuition how $\overline{\delta}_{min}$ and δ_{min} are related. The contour line for $X = 0$ is of particular interest, since it indicates a perfect matching. Note that Figure 3.7 depicted a similar plot, where instead of X the parameter δ_{min} was shown. Hence, the structure of the plot is the same, only the contour lines are different.

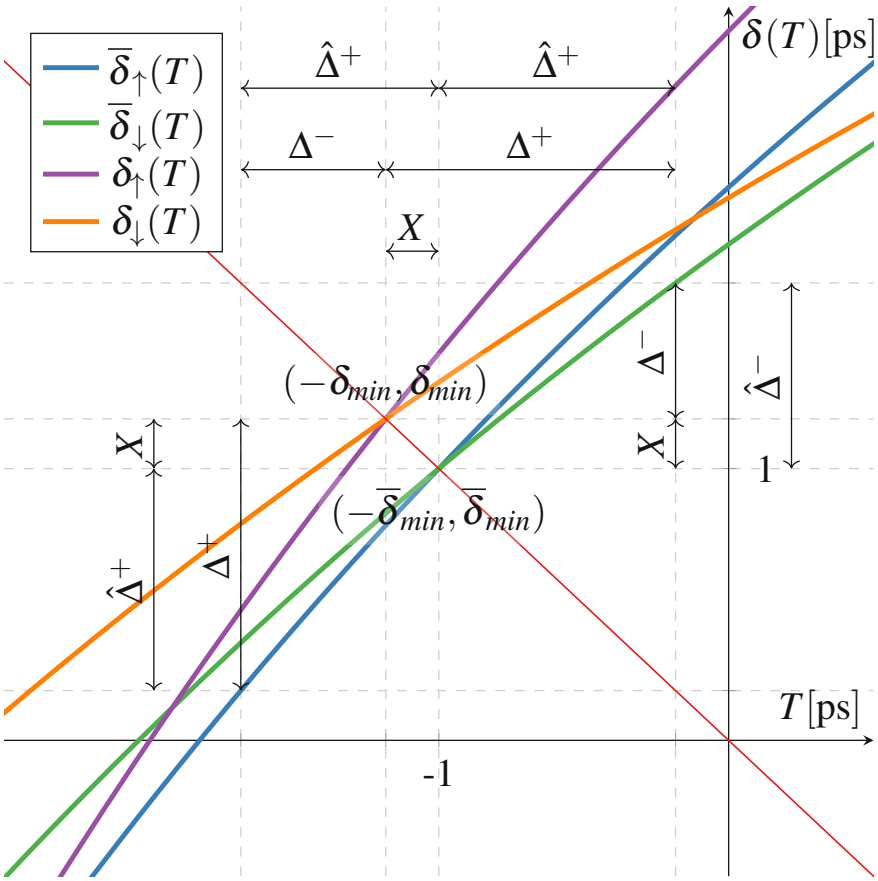


Figure 4.5: Relationship between δ_{min} and $\overline{\delta}_{min}$. Parameters: $\overline{\delta}_{min} = 1\,\text{ps}, \overline{\delta}^{\uparrow}_{\infty} = 10\,\text{ps}, \overline{\delta}^{\downarrow}_{\infty} = 9\,\text{ps}, \Delta^+ = 1\,\text{ps}, \Delta^- = -0.5\,\text{ps}$.

4.5 Faithfulness of η-CIDM

Figure 4.9 shows a circuit which is able to solve unbounded SPF in η-CIDM. Like Figure 3.9, it consists of a fed-back OR gate and a high threshold buffer; however, it uses IηP channels instead of η-IDM channels.

Before starting with the actual reduction proof of the circuit in Figure 4.9 to the circuit in Figure 3.9, it needs to be established how a high-threshold buffer can be built in the CIDM. In the following, it is shown that it is possible build a high-threshold buffer with an arbitrary threshold V_{th}, by choosing perfectly matching (Δ^+, Δ^-) appropriately. Moreover, it is important to note that $\overline{\delta}_{\uparrow}(.)$ and $\overline{\delta}_{\downarrow}(.)$ are assumed to be fixed, since these are parameters of the predecessor gate that cannot be influenced by the high-threshold buffer. Only Δ^+ and Δ^- can be chosen freely. The intuition of the following proof is that Δ^+ can be made arbitrarily large, such that the rising transitions are always canceled by the falling transitions (up to a certain pulse width).

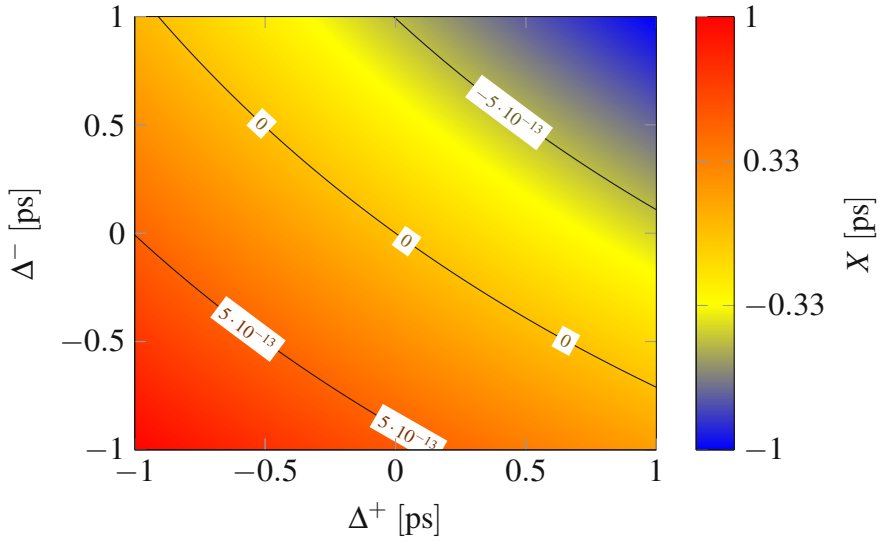


Figure 4.6: Difference $X = \overline{\delta}_{min} - \delta_{min}$ over (Δ^+, Δ^-).

Lemma 15. *Let $\Theta > 0$. For every IDM channel $(\overline{\delta}_\uparrow(.), \overline{\delta}_\downarrow(.))$, there is a perfectly matched pair (Δ^+, Δ^-) such that every finite or infinite pulse train with up-pulse length $\Theta_n \leq \Theta, n \geq 0$, is mapped to the zero signal.*

Proof. At first, an upper bound Θ' on the up-pulse length after the IDM channel, i.e., o'_{or} in Figure 4.9, is derived. Figure 4.7 illustrates the relevant signals. Suppose that the maximum up-pulse length of the signal o_{or} is Θ. By retracing the signal through i_2 and i'_2, the maximum up-pulse length at o'_{or} is derived as:

$$\Theta' = \Theta + \Delta^+_{i_2} + \eta^+_{mini_2} + \eta^-_{mini_2} - \Delta^-_{i_2}. \tag{4.31}$$

Again, like for the proof of Theorem 10, where the worst-case pulse train is considered, the rising transitions are taken maximally late $(\eta^+_{mini_2})$ and the falling transitions are taken maximally early $(-\eta^-_{mini_2})$, since this maximizes Θ'. Note that it is even possible that the pulses at o'_{or} are overlapping, i.e, that the falling transition of the previous transition happens later than the rising transition of the current transition. However, this is not a problem, since these "overlaps" will be resolved by Δ^+ and Δ^- of the high-threshold buffer.

In order to ensure that pulses with a length up to Θ' are always canceled after the pure delay shifter of the high-threshold buffer, (Δ^+, Δ^-) must be chosen such that $\Theta' + \Delta^- + \eta^+_{min} \leq \Delta^+ - \eta^-_{min}$, where $\eta^+_{min}, \eta^-_{min}$ are the maximal adversarial choices in the high-threshold buffer. Since Δ^+ can be arbitrarily large, and Δ^- is bounded by Δ^-_{min}, there are infinitely many pairs which perfectly match and fulfill the requirement of canceling the up-pulses.

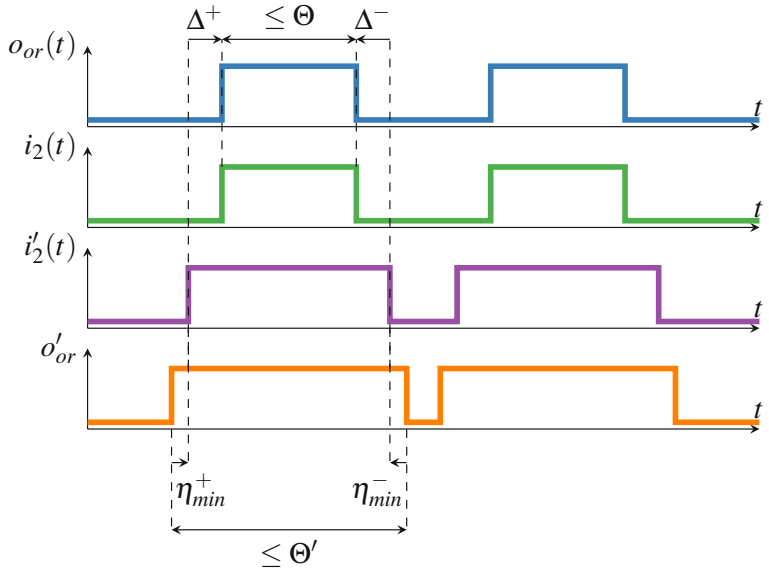


Figure 4.7: Illustration of the relevant signals for Lemma 15. In this example, $\Delta^+ > 0$ and $\Delta^- < 0$.

Note that once the feedback loop locks, i.e, o_{or} is constant one, the last rising transition is not canceled by a falling transition (since there is no more falling transition), and finally, the output o settles at a constant one. After o_{or} has settled, it takes at most $\overline{\delta}^{\uparrow}_{\infty} + \eta^+_{min} + \Delta^+$ for o to also settle to a constant one. □

Another attempt on explaining the high-threshold buffer can be made via switching waveforms and is illustrated in Figure 4.8. Suppose that the OR gate and the high-threshold buffer are characterized with $V^{in}_{th} = V^{out}_{th} = \frac{V_{DD}}{2}$. Obviously, $V^{in*}_{th} > V^{in}_{th}$ for a high-threshold buffer. Δ^+ is the time it takes the up-switching waveform $f_{\uparrow}$ to reach V^{in*}_{th}, when starting from V^{in}_{th}. By increasing V^{in*}_{th}, the required Δ^+ becomes larger, and since $f_{\uparrow}$ must asymptotically approach V_{DD}, Δ^+ can become arbitrarily large. On the other hand, Δ^- is bounded by Δ^-_{min}. Suppose that $V^{in*}_{th} = V_{DD}$, then Δ^- is the time it takes the down-switching waveform $f_{\downarrow}$ to get from V^{in}_{th} to V_{DD}, which is obviously a negative value in this example, namely, $\Delta^-_{min} = \overline{\delta}_{min} - \overline{\delta}^{\uparrow}_{\infty}$.

By using Lemma 15, it can be shown that the circuit in Figure 4.9 can be reduced to the circuit in Figure 3.9.

Theorem 16. *There is a circuit consisting of η-CIDM channels that solves unbounded SPF.*

Proof. By reducing the circuit in Figure 4.9 to the circuit in Figure 3.4, the faithfulness of the η-CIDM is shown.

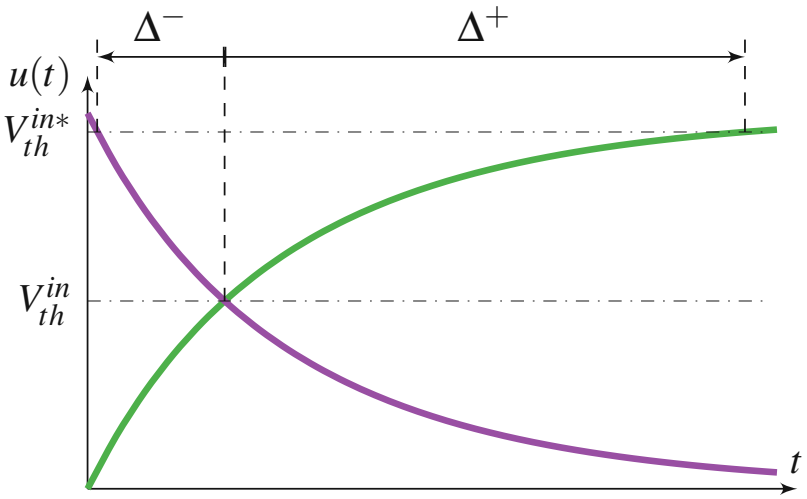


Figure 4.8: Illustration of relationship between V_{th}^{in*}, V_{th}^{in}, Δ^+ and Δ^- for a high-threshold buffer.

In the following, the differences to the original circuit are listed, and it is argued why the new circuit is indeed able to solve unbounded SPF.

- At the input i_1', a pure delay shifter and a cancellation unit can be found. As stated earlier in Section 3.2, we can safely assume matching threshold voltages at the inputs of circuits. Hence, $\Delta^+ = \Delta^- = 0$, and therefore $i_1' = i_1$.
- The first IηP channel starts from o_{or} and goes to i_2, and corresponds to c_f in Figure 3.9. It serves as the feedback channel. As shown in Lemma 14, the delay adversary and the pure delay can be interchanged. Moreover, the resulting $IP\eta$ channel can indeed be reduced to an η channel via Theorem 13. It is worth noting here that this channel can be characterized properly and thus made perfectly matching, i.e, $\overline{\delta}_{min} = \delta_{min}$. Since the constraints on η^+_{min} and η^-_{min} are determined by the parameters of the feedback channel, this fact also has an influence on the constraints, as discussed in Section 4.6.
- The second IηP channel also starts from o_{or} and goes to i_{buf}. It is used to implement the high threshold buffer and corresponds to c_{ht}. While the two channels c_f and c_{buf} were completely disjoint in Figure 3.4, this is not the case here. Both channels share the same I channel and only differ in their pure delay shifter. Nevertheless, as shown in Lemma 15, it is nevertheless possible to build a high threshold buffer that filters all finite and infinite pulse trains that have pulses below a certain maximum pulse length Θ.

  In the case of an infinite pulse train (with a maximum pulse length of $\Theta = \Delta$, where Δ is the fixed point of Eq. (3.30)), the high-threshold buffer maps the output to zero. Otherwise, if the feedback loop locks, and the signal o_{or} goes to a constant

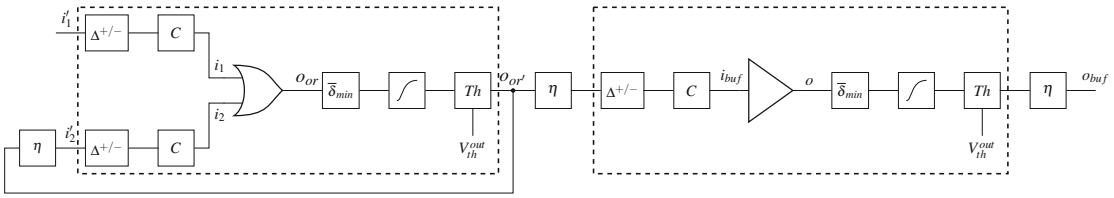


Figure 4.9: Circuit solving the unbounded SPF problem with the CIDM.

one, the high-threshold buffer must ensure that o has exactly one rising transition. This can be ensured by choosing Δ^+ which is larger than the time it takes the feedback loop to lock, i.e, the time when the up-pulse is larger than Δ until it reaches constant one.

For the case of a unique rising transition at time 0 at o_{or}, the high-threshold buffer obviously also generates a single rising transition, whereas a single short pulse at o_{or} is removed.

- At the output of the buffer (o), an additional I channel and a delay adversary are located. This η-IDM channel has no influence on the solvability of the SPF problem, since at the output o either one or zero transitions arrive: In the case where there is no transition at o, there is also no transition on o_{buf}. If there is exactly one transition at o, then this transition is delayed by $\overline{\delta}_{\infty}^{\uparrow}$ of the I channel and some delay η by the delay adversary. This again does not impair solvability.

Hence, the circuit has been reduced to the circuit in Figure 3.9, which concludes the proof. □

4.6 Constraint comparison

Before we can compare the constraints of an η-CIDM channel and its underlying η-IDM channel, another lemma is required that compares the relationship between the delay functions of an I channel and an IP channel.

Lemma 17. *For an IDM channel with delay functions $(\overline{\delta}_{\uparrow}(.), \overline{\delta}_{\downarrow}(.))$ the corresponding IP channel with (Δ^+, Δ^-) given by*

$$\delta_{\uparrow}(T) = \Delta^+ + \overline{\delta}_{\uparrow}(T + \Delta^-) \text{ and} \tag{4.32}$$

$$\delta_{\downarrow}(T) = \Delta^- + \overline{\delta}_{\downarrow}(T + \Delta^+), \tag{4.33}$$

the following cases can be distinguished for the rising delay functions:

$$\overline{\delta}_{\uparrow}(T) - \delta_{\uparrow}(T) = \begin{cases} \text{strictly increasing} & \text{if } \Delta^- > 0 \\ \text{constant} & \text{if } \Delta^- = 0 \\ \text{strictly decreasing} & \text{if } \Delta^- < 0. \end{cases} \tag{4.34}$$

For the falling delay functions, the following cases can be distinguished:

$$\overline{\delta}_{\downarrow}(T) - \delta_{\downarrow}(T) = \begin{cases} \textit{strictly increasing} & \textit{if } \Delta^+ > 0 \\ \textit{constant} & \textit{if } \Delta^+ = 0 \\ \textit{strictly decreasing} & \textit{if } \Delta^+ < 0. \end{cases} \tag{4.35}$$

Moreover, if $\overline{\delta}_{min} = \delta_{min}$, i.e., if the delay functions match perfectly, then

$$\operatorname{sgn}(\overline{\delta}_{\uparrow}(T) - \delta_{\uparrow}(T)) = \begin{cases} -\operatorname{sgn}(\Delta^-) & \textit{if } T > -\overline{\delta}_{min} \\ \operatorname{sgn}(\Delta^-) & \textit{if } T < -\overline{\delta}_{min} \\ 0 & \textit{if } T = -\overline{\delta}_{min} \end{cases}, \tag{4.36}$$

and

$$\operatorname{sgn}(\overline{\delta}_{\downarrow}(T) - \delta_{\downarrow}(T)) = \begin{cases} -\operatorname{sgn}(\Delta^+) & \textit{if } T > -\overline{\delta}_{min} \\ \operatorname{sgn}(\Delta^+) & \textit{if } T < -\overline{\delta}_{min} \\ 0 & \textit{if } T = -\overline{\delta}_{min} \end{cases} \tag{4.37}$$

For Eq. (4.36), the definition range is $T \in [\max\{-\overline{\delta}^{\downarrow}_{\infty}, -\delta^{\downarrow}_{\infty}\}, \infty)$, for Eq. (4.37) it is $T \in [\max\{-\overline{\delta}^{\uparrow}_{\infty}, -\delta^{\uparrow}_{\infty}\}, \infty)$.

Proof. Consider the difference $d(T)$ of the delay functions for falling transitions:

$$d(T) = \overline{\delta}_{\downarrow}(T) - \delta_{\downarrow}(T) = \delta_{\downarrow}(T - \Delta^+) - \Delta^- - \delta_{\downarrow}(T). \tag{4.38}$$

Taking the derivative yields:

$$d'(T) = \delta'_{\downarrow}(T - \Delta^+) - \delta'_{\downarrow}(T). \tag{4.39}$$

Using the fact that $\delta'_{\downarrow}(T)$ is strictly decreasing, it can be seen that $\operatorname{sgn}(d'(T)) = \operatorname{sgn}(\Delta^+)$. Hence, depending on the sign of Δ^+ the difference $d(T)$ is either strictly increasing, constant, or strictly decreasing.

To show the second part of the lemma, perfectly matched delay functions are assumed, i.e., $\overline{\delta}_{min} = \delta_{min}$, hence, $d(-\overline{\delta}_{min}) = 0$. Again, consider the falling delay functions: Depending on Δ^+, three cases can be distinguished:

- $\Delta^+ > 0$: The difference $d(T)$ is strictly increasing. Since $d(-\overline{\delta}_{min}) = 0$, it must hold that $d(T) > 0$ for $T > -\overline{\delta}_{min}$ and $d(T) < 0$ for $T < -\overline{\delta}_{min}$.
- $\Delta^+ < 0$: The difference $d(T)$ is strictly decreasing. Since $d(-\overline{\delta}_{min}) = 0$, it must hold that $d(T) < 0$ for $T > -\overline{\delta}_{min}$ and $d(T) > 0$ for $T < -\overline{\delta}_{min}$.
- $\Delta^+ = 0$: In this case, the difference $d(T)$ is constant. Since $d(-\overline{\delta}_{min}) = 0$, $d(T) = 0$ for the complete definition range.

Combining the three cases concludes the proof for falling delay functions. For rising delay functions, the proof is symmetric. □

We are now ready to address the following question: Is one of the implementations Figure 3.9 resp. Figure 4.9 preferable in terms of the constraint on the adversary? We assume here that the same I channel is used in both.

The adversary for the involution channel $(\overline{\delta}_\uparrow(.), \overline{\delta}_\downarrow(.))$ in Figure 3.9 can pick delay variations from the range $[-\overline{\eta}^-_{min}, \overline{\eta}^+_{min}]$, subject to the following constraint:

$$\overline{\eta}^+_{min} + \overline{\eta}^-_{min} < \overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \overline{\delta}_{min}. \tag{4.40}$$

Adding the pure delay shifter in the PI channel of Figure 4.9 of course has an influence on the applicable range for the delay variation (see Theorem 13). The resulting IP channel can pick delay variations $[-\eta^-_{min}, \eta^+_{min}]$, subject to the constraint:

$$\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}. \tag{4.41}$$

However, as shown in Section 4.5, the delay functions match perfectly, i.e., $\overline{\delta}_{min} = \delta_{min}$, and hence the constraint can be rewritten as

$$\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \overline{\delta}_{min}. \tag{4.42}$$

To answer our questions we investigate how the border (see Figure 3.10) is affected by the pure delay shifter. Suppose that $\overline{\eta}^+_{min}$ and η^+_{min} are chosen to be equal, and that $\overline{\eta}^-_{min}$ and η^-_{min} are chosen as large as possible while obeying the constraints, then

$$\begin{aligned} \overline{\eta}^-_{min} - \eta^-_{min} &= (\overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \overline{\delta}_{min}) - (\delta_\downarrow(-\eta^+_{min}) - \delta_{min}) && (4.43)\\ &= \overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \delta_\downarrow(-\overline{\eta}^+_{min}). && (4.44) \end{aligned}$$

Of course, a similar equation can be derived when setting $\overline{\eta}^-_{min} = \eta^-_{min}$ and maximizing $\overline{\eta}^+_{min}$ and η^+_{min}:

$$\begin{aligned} \overline{\eta}^+_{min} - \eta^+_{min} &= (\overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \overline{\delta}_{min}) - (\delta_\downarrow(-\eta^+_{min}) - \delta_{min}) && (4.45)\\ &= \overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \delta_\downarrow(-\eta^+_{min}). && (4.46) \end{aligned}$$

By Corollary 11, it is known that $0 \leq \overline{\eta}^+_{min} < \overline{\delta}_{min}$. Applying Lemma 17 yields that $\mathrm{sgn}(\overline{\eta}^+_{min} - \eta^+_{min}) = \mathrm{sgn}(\overline{\delta}_\downarrow(-\overline{\eta}^+_{min}) - \delta_\downarrow(-\overline{\eta}^+_{min}))$, hence, the border of the IP channel is either strictly larger, strictly smaller or exactly the same as the border of the original channel.

Figure 4.10 shows two examples for perfectly matched delay functions (dashed green and dashed orange line), indicated by *. Note that there are also pairs (Δ^+, Δ^-) which are no perfect matches (lilac and red dashed line). Depending on the actual values of Δ^+

and Δ^-,the borders might cross the original border in this case. Note that all channels in Figure 4.10 are strictly causal.

Since in Chapter 6 the values for η^+_{min} and η^-_{min} are chosen such that $\eta^+_{min} \approx \eta^-_{min}$, it is of particular interest how different pairs (Δ^+, Δ^-) influence this value. The point where $\eta^+_{min} = \eta^-_{min}$ is at the intersection of the dashed blue line (1st median) and the respective border. This shows that the values for $\eta^+_{min} = \eta^-_{min}$ are different for different choices of (Δ^+, Δ^-). It would be nice if we were able to allow a larger delay variation by choosing an appropriate pair (Δ^+, Δ^-); however, Δ^+ and Δ^- are determined during the characterization, so there is no chance to select these two values to allow larger delay variations.

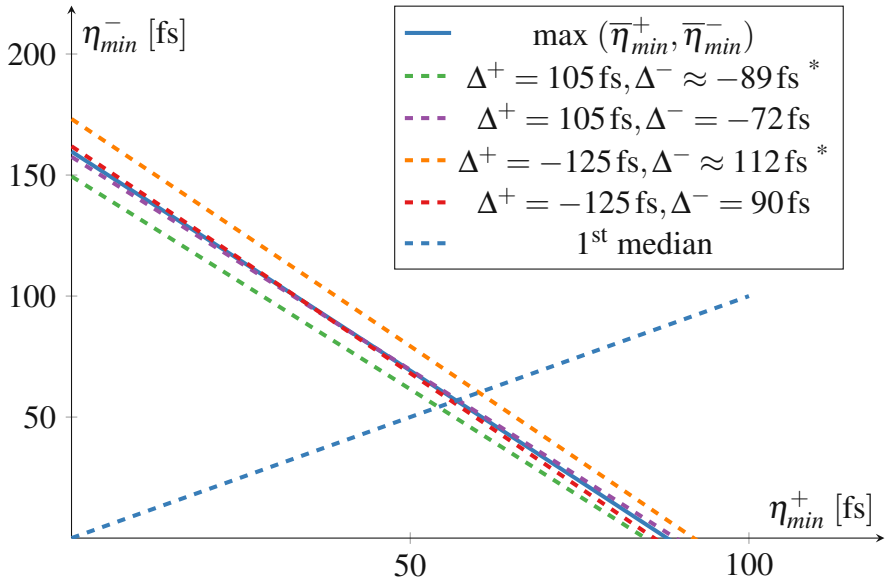


Figure 4.10: Pairs of $(\eta^+_{min}, \eta^-_{min})$ for different pairs of (Δ^+, Δ^-), starting from an involution channel with the same parameters as in Figure 3.10. The * indicates perfectly matched delay functions.

CHAPTER 5

Extensions for the η-IDM

The goal of this chapter is to extend the bounds $[-\eta^-_{min}, \eta^+_{min}]$ of the already existing η-IDM. By doing so, a wider applicability can be achieved, which also improves the applicability of η-CIDM. Moreover, by providing a generalized reduction proof for the impossibility of SPF, it can be shown that η-IDM is indeed suitable to cover delay variations due to threshold voltage variations.

5.1 Loosening constraints on delay variations

The goal in this section is to loosen the bounds on the adversarial delay. Figure 5.1 shows the current situation (blue lines) and the desired improvements (green lines). There are two main objectives:

(A) Introduce a T-dependent linear bound for certain ranges of T.

(B) Introduce a constant and even looser bound for the remaining ranges.

The above objectives result in the following definition of the adversarial delays:

Definition 18. For $\overline{\Delta} \geq \delta_\uparrow(-\overline{\Delta}) + \rho^+ \cdot (\overline{\Delta} - \Delta) + \eta^+_{min}$, where Δ denotes the fixed point of $f(.)$ in Eq. (3.30), let

$$\eta^+(T) = \begin{cases} \eta^+_\infty & \text{for } T < -\overline{\Delta} \\ \rho^+ \cdot (-T - \Delta) + \eta^+_{min} & \text{for } -\overline{\Delta} \leq T \leq -\Delta \\ \eta^+_\infty & \text{for } T > -\Delta \end{cases} \tag{5.1}$$

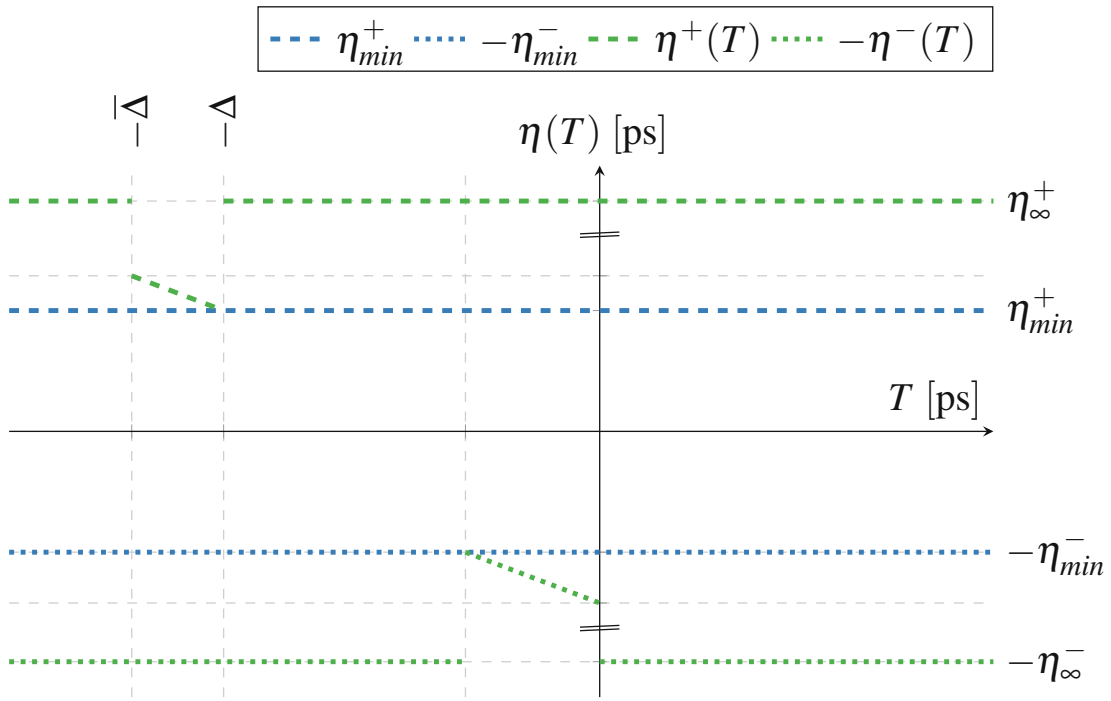


Figure 5.1: In [FMN+20] a constant boundary η^+_{min} and η^-_{min} is introduced (blue dashed and dotted lines). The idea of this section is to introduce a T-dependent bound in the critical regions, and introduce an even looser bound for the remaining ranges. Note the discontinuity of the y-axis: For our used inverters, the values for η^+_∞ and η^-_∞ were $\eta^+_\infty \approx 20\eta^+_{min}$ and $\eta^-_\infty \approx 20\eta^-_{min}$.

$$\eta^-(T) = \begin{cases} \eta^-_\infty & \text{for } T < \Delta - \delta_\uparrow(-\Delta) - \eta^+_{min} \\ \rho^- \cdot (-T - \Delta + \delta_\uparrow(-\Delta) + \eta^+_{min}) + \eta^-_{min} & \text{for } \Delta - \delta_\uparrow(-\Delta) - \eta^+_{min} \leq T < 0 \\ \eta^-_\infty & \text{for } T \geq 0 \end{cases} \tag{5.2}$$

The parameters $\eta^{+/-}_{min}, \rho^{+/-}$, and $\eta^{+/-}_\infty$ are subject to the following constraints:

(1) $\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}$

(2) $\eta^+_\infty + \eta^-_\infty < \delta^\uparrow_\infty - \delta_{min}$

(3) $\rho^+ \geq 0$, $\rho^- \geq 0$

(4) $(1 - \rho^-)(\delta'_\uparrow(-\Delta) - \rho^+ + 1) > 1$

Lemma 19 (Adapted Lemma 3 from [FMN+20], cp. (i) on Page 19)**.** *If the input pulse's length* Δ_0 *satisfies* $\Delta_0 \geq \delta^{\uparrow}_{\infty} + \eta^{+}_{\infty}$*, then the output of the OR gate* o_{or} *in Figure 3.9 has a unique rising transition at time* 0 *and no falling transition.*

Proof. Since the gate is modeled as a zero-time gate, the output o_{or} of the OR gate immediately follows the rising transition at the input i_1. This rising transition is then propagated through the feedback channel and arrives at latest at $\delta^{\uparrow}_{\infty} + \eta^{+}_{\infty}$ at the other input i_2. Therefore, the falling transition on input i_1 at time Δ_0 has no influence on the output, since i_2 is already at 1. T Therefore, the storage loop locks at constant 1. Figure 5.2a illustrates the proof. □

Lemma 20 (Adapted Lemma 4 from [FMN+20], cp. (ii) on Page 19)**.** *If the length of the input pulse* Δ_0 *on* i_1 *satisfies* $\Delta_0 \leq \delta^{\uparrow}_{\infty} - \delta_{min} - \eta^{+}_{\infty} - \eta^{-}_{\infty}$*, then the output of the OR gate* o_{or} *in Figure 3.9 only contains the input pulse. Moreover, to ensure that* $\Delta_0 > 0$*, the following constraint must hold:* $\eta^{+}_{\infty} + \eta^{-}_{\infty} < \delta^{\uparrow}_{\infty} - \delta_{min}$*.*

Proof. The earliest time that the rising transition from i_1 can happen on i_2 is $t'_1 \geq \delta^{\uparrow}_{\infty} - \eta^{-}_{\infty}$. Therefore, for the falling transition we get $T = \Delta_0 - t'_1 \leq \Delta_0 - \delta^{\uparrow}_{\infty} + \eta^{-}_{\infty}$ and hence the corresponding falling output transition cannot occur later than $t'_2 \leq \Delta_0 + \delta_{\downarrow}(T) + \eta^{+}(T)$. The two transitions on i_2 cancel each other out iff $t'_2 \leq t'_1$, i.e., if

$$X = \Delta_0 + \delta_{\downarrow}(T) + \eta^{+}(T) - \delta^{\uparrow}_{\infty} + \eta^{-}_{\infty} \leq 0 \tag{5.3}$$

holds, we get a cancellation since

$$t'_2 \leq \Delta_0 + \delta_{\downarrow}(T) + \eta^{+}(T) \leq \delta^{\uparrow}_{\infty} - \eta^{-}_{\infty} \leq t'_1. \tag{5.4}$$

Since $\eta^{+}(T)$ is upper bounded by η^{+}_{∞}, Δ_0 can be substituted with the upper bound $\delta^{\uparrow}_{\infty} - \delta_{min} - \eta^{+}_{\infty} - \eta^{-}_{\infty}$ from the lemma, and finally $\delta_{\downarrow}(T) \leq \delta_{\downarrow}(-\delta_{min} - \eta^{+}_{\infty})$ by monotonicity, we get

$$X \leq -\delta_{min} + \delta_{\downarrow}(-\delta_{min} - \eta^{+}_{\infty}) \leq 0 \tag{5.5}$$

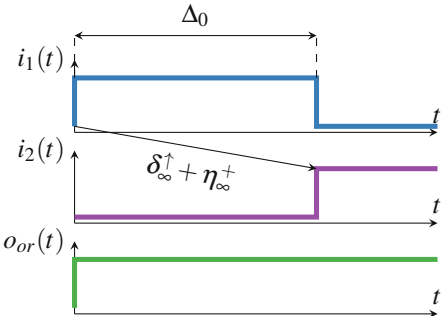
Since $\eta^{+}_{\infty} \geq 0$, we end up with $X \leq -\delta_{min} + \delta_{\downarrow}(-\delta_{min}) \leq 0$ by monotonicity of $\delta_{\downarrow}(.)$ and Lemma 2. Hence, Eq. (5.3) holds. Figure 5.2b illustrates this proof.

Rearranging the upper bound $0 < \Delta_0 \leq \delta^{\uparrow}_{\infty} - \delta_{min} - \eta^{+}_{\infty} - \eta^{-}_{\infty}$ also yields the constraint on η^{+}_{∞} and η^{-}_{∞}:

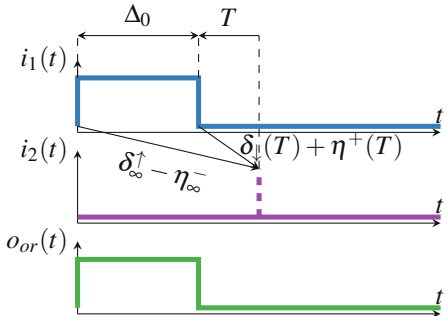
$$\eta^{+}_{\infty} + \eta^{-}_{\infty} < \delta^{\uparrow}_{\infty} - \delta_{min} \tag{5.6}$$

□

So it remains to deal with the most delicate case (iii) on Page 19, namely, input pulses with a length of $\delta^{\uparrow}_{\infty} - \delta_{min} - \eta^{+}_{\infty} - \eta^{-}_{\infty} < \Delta_0 < \delta^{\uparrow}_{\infty} + \eta^{+}_{\infty}$. This requires a generalization of [FMN+20, Lemma 5], which we describe in the sequel. The original lemma states the following:



(a) Waveforms which illustrate the proof of Lemma 19.



(b) Waveforms which illustrate the proof of Lemma 20.

Figure 5.2: Waveforms illustrating the proofs of Lemma 19 and Lemma 20

.

Lemma 21 (Lemma 5, [FMN+20])**.** *Consider the circuit in Figure 3.9 subject to the constraint of Theorem 10. Assume that the input pulse length 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, $\Delta = \delta_\downarrow(\eta^+_{min} - \tau)$ with $\Delta < \delta_{min}$ is the up-time of an infinite self-repeating pulse train with period $P = \tau = \delta_\uparrow(-\Delta) + \eta^+_{min}$ and duty cycle $\gamma = \Delta/P$, with $\tau > 0$, denoting the smallest positive fixed point of the equation $\delta_\downarrow(\eta^+_{min} - \tau) + \delta_\uparrow(-\eta^-_{min} - \tau) = \tau$, which is guaranteed to exist and satisfies $\eta^+_{min} + \delta_{min} < \tau < \min\{-\eta^+_{min} + \delta^\downarrow_\infty, \eta^+_{min} + \delta^\uparrow_\infty\}$.*

The idea of the loosened constraint in Definition 18 is that the further away the adversary gets from the self repeating "worst-case pulse train", i.e., the larger the pulses get, the more it is allowed to add deviations. The important property that needs to be maintained is that once the pulse is longer than the fixed point Δ, it needs to grow strictly monotonically until the feedback loop locks.

For the "worst-case pulse train", it is known that the up-time of the pulse is Δ, whereas the down-time is $\Delta' = \delta_\uparrow(-\Delta) + \eta^+_{min} - \Delta$. Signal $w_1(t)$ in Figure 5.3 depicts this case.

Now suppose we have a pulse with a larger up-time $\Delta_{n-1} > \Delta$ and therefore $\delta_\uparrow(-\Delta_{n-1}) < \delta_\uparrow(-\Delta)$. If the original constraint on the delay variations is used, this results in $\Delta'_{n,2} = \delta_\uparrow(-\Delta_{n-1}) + \eta^+_{min} - \Delta_{n-1} < \Delta'$, and hence the delay of the falling transition $\delta_\downarrow(-\Delta'_{n,2}) > \delta_\downarrow(\Delta)$. From Section 3.3, we know that the resulting $\Delta_{n,2} > \Delta_{n-1} > \Delta$ is even larger. This case is illustrated by the signal $w_2(t)$ in Figure 5.3

Signal $w_3(t)$ in Figure 5.3 now illustrates the case with the new proposed bounds on the delay variations Eq. (5.1) and Eq. (5.2): By allowing a delay variation $\eta^+(-\Delta_{n-1}) > \eta^+_{min}$, we obtain the down-time $\Delta'_{n,3}$. However, it is important to choose $\eta^+(-\Delta_{n-1})$ such that $\Delta'_{n,3} < \Delta'$ is ensured. Furthermore, $\delta_\downarrow(-\Delta'_{n,3})) > \delta_\downarrow(-\Delta')$ since $\Delta'_{n,3} < \Delta'$, which allows us to choose a delay variation $\eta^-(-\Delta'_{n,3}) > \eta^-_{min}$. Again, the delay variation $\eta^-(-\Delta'_{n,3})$

needs to be chosen such that $\Delta_{n,2} > \Delta_{n-1}$ is guaranteed. This is necessary, since we still require that pulses which are larger than Δ grow strictly monotonically.

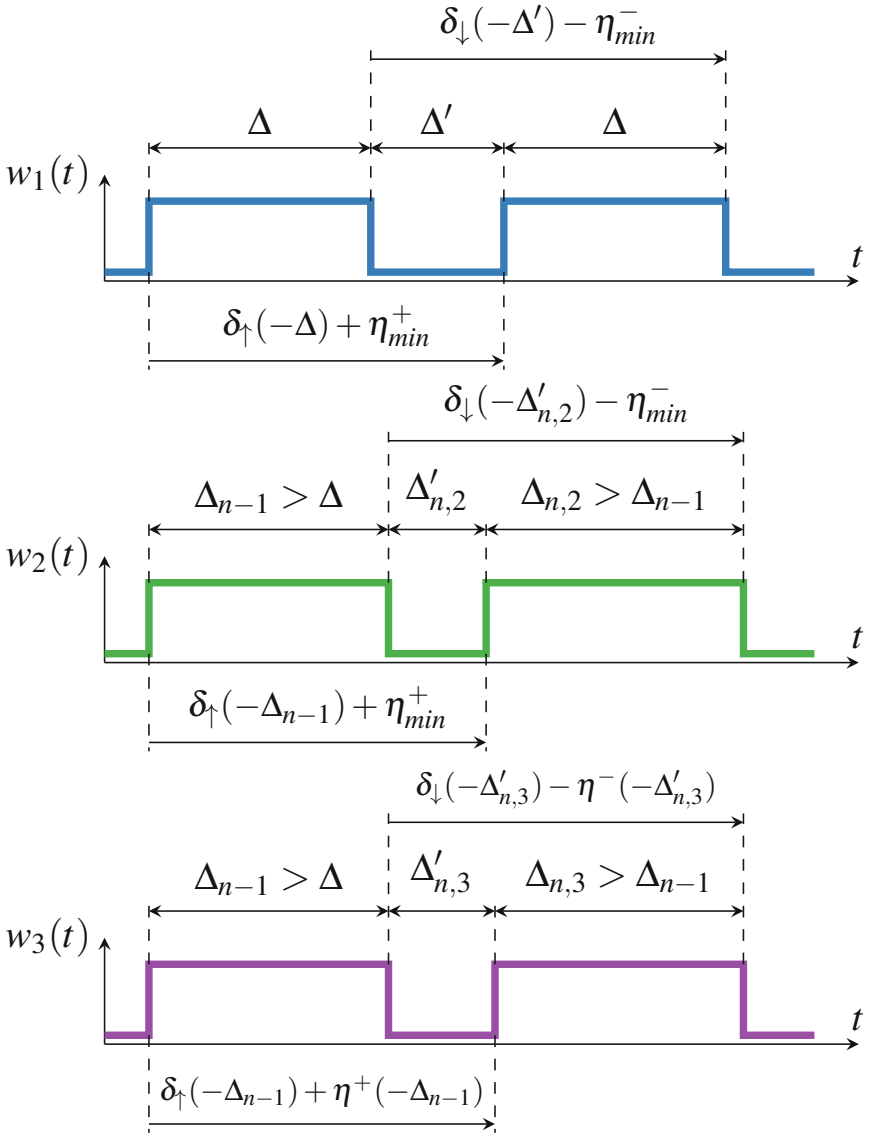


Figure 5.3: Waveforms illustrating the idea of the loosened bounds.

With the notation of Lemma 21, the length of the next pulse can be calculated from the

previous pulse length Δ_{n-1} as follows:

$$\begin{aligned}\Delta_n = g(\Delta_{n-1}) =& \Delta_{n-1} \\ &+ \delta_\downarrow(-\delta_\uparrow(-\Delta_{n-1}) - \eta^+(-\Delta_{n-1}) + \Delta_{n-1}) \\ &- \eta^-(-\delta_\uparrow(-\Delta_{n-1}) - \eta^+(-\Delta_{n-1}) + \Delta_{n-1}) \\ &- \delta_\uparrow(-\Delta_{n-1}) - \eta^+(-\Delta_{n-1}),\end{aligned} \tag{5.7}$$

where $\eta^+(T)$ and $\eta^-(T)$ are defined as in Definition 18.

The parameters ρ^+ and ρ^- will be chosen such that it is not possible for the adversary to bring the pulse durations back into the range $[0, \Delta]$ if it has been left before, which is done in the following steps:

(i) Show that the fixed point of $g(.)$ satisfies $\Delta' = g(\Delta') = f(\Delta) = \Delta$.

(ii) Show that the derivative of $g(x)$ is greater than 1 for $\overline{\Delta} \geq x \geq \Delta$ under certain constraints.

(iii) Show that for $\Delta_n > \Delta$, the pulse length is strictly monotonically increasing, i.e., $\Delta_{n+1} > \Delta_n > \Delta$ for $n \geq 1$, where $\Delta_{n+1} = g(\Delta_n)$.

(iv) Show that it suffices to enforce a constraint only for the linear ranges of T given in Definition 18.

Step (i): In the first step, it is shown that $g(.)$, as defined in Eq. (5.7), has a fixed point $\Delta' = g(\Delta')$, which is equal to the fixed point Δ of $f(.)$, as defined in Eq. (3.30).

Lemma 22. *There is a fixed point $\Delta' = g(\Delta')$ that is equal to the fixed point $\Delta = f(\Delta)$, where $f(.)$ is as defined in Eq.* (3.30)*.*

Proof. Plugging in Δ in Eq. (5.7) yields:

$$\begin{aligned}g(\Delta) =& \delta_\downarrow\Big(- \delta_\uparrow(-\Delta) - \eta^+(-\Delta) + \Delta\Big) \\ &- \eta^-\Big(- \delta_\uparrow(-\Delta) - \eta^+(-\Delta) + \Delta\Big) \\ &- \delta_\uparrow(-\Delta) - \eta^+(-\Delta) + \Delta.\end{aligned} \tag{5.8}$$

By using the definition of $\eta^+(.)$ in Definition 18, the above equation can be further simplified:

$$\begin{aligned}g(\Delta) =& \delta_\downarrow\Big(- \delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta\Big) \\ &- \eta^-\Big(- \delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta\Big) \\ &- \delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta.\end{aligned} \tag{5.9}$$

A close look at $\eta^-(.)$ in Definition 18 reveals that also the term $\eta^-(.)$ can be simplified:

$$g(\Delta) = \delta_\downarrow\Big(-\delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta\Big) - \eta^-_{min} - \delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta. \tag{5.10}$$

Since Eq. (3.30) shows

$$\Delta = f(\Delta) = \delta_\downarrow\Big(-\delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta\Big) - \eta^-_{min} - \delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta, \tag{5.11}$$

a term-wise comparison from Eq. (5.10) and Eq. (5.11) reveals $g(\Delta) = \Delta$, i.e., Δ is indeed a fixed point of $g(.)$, which concludes the proof. □

Note that there is also an intuitive explanation why $g(.)$ and $f(.)$ have the same fixed point: For the "worst-case pulse train", the relevant values of T for the delay variation are $-\Delta$ for $\eta^+(.)$ and $\Delta - \delta_\uparrow(-\Delta) - \eta^+_{min}$ for $\eta^-(.)$. In these two points, the original definition of the delay variations from η-IDM is equal to the definition from Definition 18, namely:

$$\eta^+(-\Delta) = \eta^+_{min} \text{ and} \tag{5.12}$$
$$\eta^-(\Delta - \delta_\uparrow(-\Delta) - \eta^+_{min}) = \eta^-_{min}. \tag{5.13}$$

Since $\Delta' = \Delta$, we are only going to use Δ to indicate the fixed point of $f(.)$ and $g(.)$ in the sequel.

Another interesting observation about the fixed point Δ is the following:

Observation 23. The fixed point Δ is approximately δ_{min} and upper bounded by δ_{min}.

Proof. Investigate the fixed point equation obtained from Eq. (3.30):

$$\Delta = f(\Delta) = \Delta + \delta_\downarrow(-\delta_\uparrow(-\Delta) - \eta^+_{min} + \Delta) - \eta^-_{min} - \delta_\uparrow(-\Delta) - \eta^+_{min} \tag{5.14}$$

Plugging in δ_{min} into $f(.)$ gives

$$f(\delta_{min}) = \delta_{min} + \delta_\downarrow(-\delta_\uparrow(-\delta_{min}) - \eta^+_{min} + \delta_{min}) - \eta^-_{min} - \delta_\uparrow(-\delta_{min}) - \eta^+_{min} \tag{5.15}$$
$$= \delta_{min} + \delta_\downarrow(-\eta^+_{min}) - \eta^-_{min} - \delta_{min} - \eta^+_{min} \tag{5.16}$$

Due to the constraint $\eta^+_{min} + \eta^-_{min} < \delta_\downarrow(-\eta^+_{min}) - \delta_{min}$ in Theorem 10, it is known that $f(\delta_{min})$ approaches Δ if η^+_{min} and η^-_{min} are chosen as large as possible. Therefore, $\Delta \approx \delta_{min}$, and $\Delta < \delta_{min}$ (see also [FMN+20, (9)]). □

Step (ii): Derive constraints for which $g'(x) > 1$.

Lemma 24. *$g'(x) > 1$ if $(1-\rho^-)(\delta'_\uparrow(-\Delta) - \rho^+ + 1) > 1$ and $\overline{\Delta} \geq x \geq \Delta$.*

Proof. Calculating the derivative of $g(x)$ yields:

$$\begin{aligned} g'(x) =& 1 + \Big[\delta'_\downarrow\Big(- \delta_\uparrow(-x) - \eta^+(-x) + x\Big) - \eta^{-'}\Big(- \delta_\uparrow(-x) - \eta^+(-x) + x\Big)\Big] \\ & \cdot \Big(\delta'_\uparrow(-x) + \eta^{+'}(-x) + 1\Big) + \delta'_\uparrow(-x) + \eta^{+'}(-x) && (5.17) \\ =& \Big[1 + \delta'_\downarrow\Big(- \delta_\uparrow(-x) - \eta^+(-x) + x\Big) - \eta^{-'}\Big(- \delta_\uparrow(-x) - \eta^+(-x) + x\Big)\Big] \\ & \cdot \Big(\delta'_\uparrow(-x) + \eta^{+'}(-x) + 1\Big) && (5.18) \\ \geq& \Big[1 + \delta'_\downarrow\Big(- \delta_\uparrow(-x) - \eta^+(-x) + x\Big) - \rho^-\Big] \cdot \Big(\delta'_\uparrow(-x) - \rho^+ + 1\Big) && (5.19) \\ >& (1-\rho^-)(\delta'_\uparrow(-\Delta) - \rho^+ + 1) && (5.20) \end{aligned}$$

Since $\delta_\uparrow(.)$ and $\delta_\downarrow(.)$ are strictly increasing and concave, $\delta'_\uparrow(.) > 0$ and $\delta'_\downarrow(.) > 0$. Moreover, by definition $\rho^+ \geq 0$ and $\rho^- \geq 0$. Since $x \geq \Delta$, $\delta'_\uparrow(-x)$ can be lower bounded by $\delta'_\uparrow(-\Delta)$. Hence, a sufficient condition to ensure that $g(x) > 1$ can be computed from Eq. (5.20), by choosing ρ^+ and ρ^- appropriately. □

Note that it is always possible to find a pair (ρ^+, ρ^-) that fulfills the above constraint. In the worst-case, we can set $\rho^+ = \rho^- = 0$:

$$(1-\rho^-)(\delta'_\uparrow(-\Delta) - \rho^+ + 1) = \delta'_\uparrow(-\Delta) + 1 > 1. \tag{5.21}$$

This condition is always fulfilled, since $\delta'_\uparrow(.) > 0$.

Step (iii): We now show that pulses which are larger than Δ cannot come back, i.e., that once the pulse width is larger than Δ it is strictly monotonically increasing until the feedback loop is locked, provided that ρ^+ and ρ^- are suitably chosen.

Lemma 25. *For $g(.)$ given in Eq. (5.7) with fixed point Δ, it holds that $g(\Delta_1) - \Delta > (\Delta_1 - \Delta)$ if $\Delta_1 > \Delta$, provided ρ^+, ρ^- are chosen such that $(1-\rho^-)(\delta'_\uparrow(-\Delta) + 1 - \rho^+) > 1$.*

Proof. From the mean value theorem of calculus, it is known that

$$\exists \xi \in (\Delta, \Delta_1) \text{ s.t. } g'(\xi) = \frac{g(\Delta_1) - g(\Delta)}{\Delta_1 - \Delta}. \tag{5.22}$$

Rearranging and applying the fact that Δ is the fixed point of $g(.)$ yields

$$g(\Delta_1) - \Delta = g'(\xi)(\Delta_1 - \Delta). \tag{5.23}$$

From Eq. (5.20) we know that $g'(\xi) > (1-\rho^-)(\delta'_\uparrow(-\Delta)+1-\rho^+)$, which provides

$$g(\Delta_1) - \Delta' \geq (1-\rho^-)(\delta'_\uparrow(-\Delta)+1-\rho^+)(\Delta_1-\Delta) > (\Delta_1-\Delta), \tag{5.24}$$

if ρ^+, ρ^- are chosen accordingly. □

Lemma 25 shows that the number of pulses is bounded until the feedback loop locks. The stabilization time is in the order of $\log_a(\frac{1}{\Delta_1-\Delta})$, where $a = (1-\rho^-)(\delta'_\uparrow(-\Delta)+1-\rho^+)$.

Step (iv): It still remains to show for which choice of $\overline{\Delta}$ the feedback loop locks. This is the case if the rising transition is scheduled before or at the same time as the previous falling transition:

$$\delta_\uparrow(-\overline{\Delta}) + \eta^+(-\overline{\Delta}) \leq \overline{\Delta} \tag{5.25}$$

$$\delta_\uparrow(-\overline{\Delta}) + \rho^+(\overline{\Delta}-\Delta) + \eta^+_{min} \leq \overline{\Delta}. \tag{5.26}$$

Let us choose $\overline{\Delta} = \delta_{min} + \eta^+_{min}$. We need to find a suitable choice for ρ^+ for which Eq. (5.26) is fulfilled, i.e.,

$$\delta_\uparrow(-\delta_{min}-\eta^+_{min}) - \delta_{min}(1-\rho^+) + \rho^+\eta^+_{min} - \rho^+\Delta \leq 0, \tag{5.27}$$

which results in

$$\rho^+ \leq \frac{\delta_{min} - \delta_\uparrow(-\delta_{min}-\eta^+_{min})}{\delta_{min}+\eta^+_{min}-\Delta}. \tag{5.28}$$

If we choose

$$\rho^+ = \frac{\delta_{min} - \delta_\uparrow(-\delta_{min}-\eta^+_{min})}{\delta_{min}+\eta^+_{min}-\Delta}, \tag{5.29}$$

we can immediately express ρ^- by using constraint (4) from Page 38 (resp. Lemma 24):

$$\rho^- < 1 - \frac{1}{\delta'_\uparrow(-\Delta)-\rho^++1}. \tag{5.30}$$

Note than the choices for ρ^+ and ρ^- from Eqs. (5.29) and (5.30) can be simplified. By the mean value theorem of calculus it is known that

$$\exists \xi \in (-\delta_{min}-\eta^+_{min}, -\Delta) \; s.t. \; \delta'_\uparrow(\xi) = \frac{\delta_\uparrow(-\delta_{min}-\eta^+_{min}) - \delta_\uparrow(-\Delta)}{-\delta_{min}-\eta^+_{min}+\Delta}. \tag{5.31}$$

Since $\delta'_\uparrow(.)$ is strictly monotonically decreasing, we obtain

$$\delta'_\uparrow(-\Delta) \leq \delta'_\uparrow(\xi), \tag{5.32}$$

and hence

$$\delta_\uparrow(-\delta_{min} - \eta^+_{min}) \leq \delta_\uparrow(-\Delta) - \delta'_\uparrow(-\Delta)(\delta_{min} + \eta^+_{min} - \Delta). \tag{5.33}$$

If we choose

$$\rho^+ = \frac{\delta_{min} - \delta_\uparrow(-\Delta) + \delta'_\uparrow(-\Delta)(\delta_{min} + \eta^+_{min} - \Delta)}{\delta_{min} + \eta^+_{min} - \Delta} = \frac{\delta_{min} - \delta_\uparrow(-\Delta)}{\delta_{min} + \eta^+_{min} - \Delta} + \delta'_\uparrow(-\Delta), \tag{5.34}$$

then Eq. (5.28) is also satisfied. By using Lemma 24 and the value for ρ^+ from Eq. (5.34), a condition for ρ^- can be obtained:

$$1 - \rho^- > \frac{1}{\delta'_\uparrow(-\Delta) - \rho^+ + 1} = \frac{1}{1 - \frac{\delta_{min} - \delta_\uparrow(-\Delta)}{\delta_{min} + \eta^+_{min} - \Delta}} = \frac{\delta_{min} + \eta^+_{min} - \Delta}{\eta^+_{min} - \Delta - \delta_\uparrow(-\Delta)}, \tag{5.35}$$

and hence

$$\rho^- < \frac{\delta_{min} - \delta_\uparrow(-\Delta)}{\eta^+_{min} - \Delta - \delta_\uparrow(-\Delta)} = \frac{\delta_\uparrow(-\Delta) - \delta_{min}}{\delta_\uparrow(-\Delta) + \Delta - \eta^+_{min}} < 1, \tag{5.36}$$

since $\eta^+_{min} < \delta_{min}$.

We note, however, that our choice $\overline{\Delta} = \delta_{min} + \eta^+_{min}$ is actually somewhat arbitrary; we can allow any choice that fulfills Eq. (5.26). Since a smaller $\overline{\Delta}$ is in general better, cp. Eq. (5.1) and Eq. (5.2), we investigated the relation between $\overline{\Delta}$ and ρ^+ numerically via Eq. (5.26); the results are shown in Figure 5.4. If we choose a large ρ^+, the adversary can add more delay variations to the rising transitions. However, this comes at the cost that it looses power for falling transitions (smaller ρ^-), see Figure 5.5. Moreover, a larger value of ρ^+ results in a larger value of $\overline{\Delta}$, which means that the linear range $[\Delta, \overline{\Delta}]$ gets larger

Overall, the best choice, in terms of gained range for delay variation, for the parameters $\overline{\Delta}, \rho^+$ and ρ^- is heavily dependent on the actual delay functions. For simplicity, we will hence choose $\rho^+ = \rho^-$ in Chapter 6. Note that fixing these two parameters automatically fixes $\overline{\Delta}$ as well.

Figure 5.6 shows an overview of the functions used in the above proofs. In this example, the values for ρ^+ and ρ^- are ≈ 0.42. It can be clearly seen that $f(\hat{\Delta}) - \hat{\Delta}$ and $g(\hat{\Delta}) - \hat{\Delta}$ are strictly increasing (for $\hat{\Delta} \geq \Delta$), which illustrates that there is no fixed point that is greater than Δ. It can be also seen that the blue curves are steeper, which shows that the feedback loop locks faster for the original η-IDM bounds. While the stabilization time for the η-IDM was in the order of $\log_a(\frac{1}{\Delta_1 - \Delta})$ where $a = 1 + \delta'_\uparrow(-\Delta)$, the value for a for the new bounds is $a = (1 - \rho^-)(\delta'_\uparrow(-\Delta) + 1 - \rho^+)$, which is smaller than the original value (for $(\rho^+, \rho^-) \neq (0, 0)$). However, a larger stabilization time is not an issue, as we only need to adapt the high-threshold buffer accordingly, so that the output only switches after the input has locked.

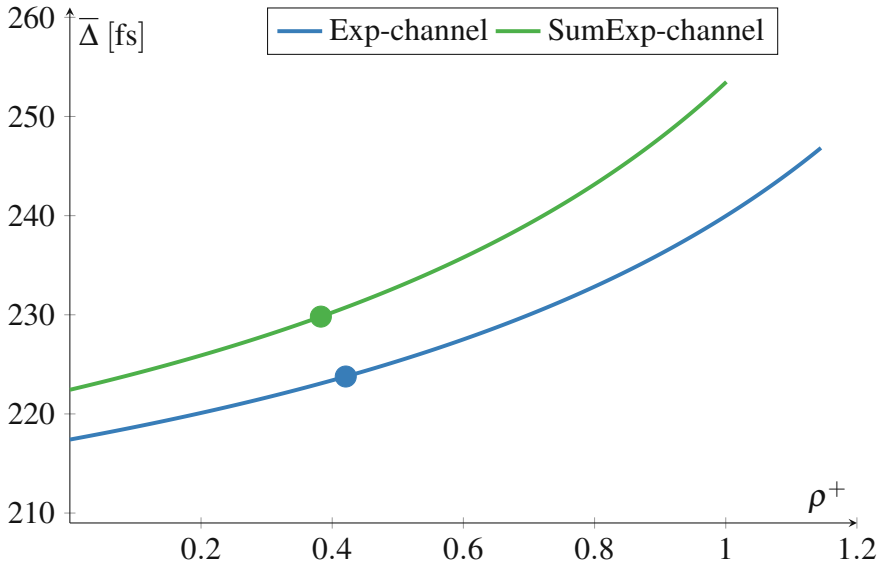


Figure 5.4: Influence of ρ^+ on $\overline{\Delta}$ for an 28 nm inverter. The circles indicate the choice $\rho^+ = \rho^-$. In this case, $\Delta \approx 191$ fs, for both, Exp-channel and SumExp-channel. Moreover, if we choose ρ^+ according to Eq. (5.29), this would result in $\rho^- = 0$ for both channel types, which is obviously not a good choice. The value for $\overline{\Delta}$ is larger when using a SumExp-channel. For the sizes of the linear range $[\Delta - \delta_\uparrow(-\Delta) - \eta^+_{min}, 0]$ we get ≈ 57 fs for the Exp-channel, and ≈ 63 fs for the SumExp-channel. The influence of the value of $\overline{\Delta}$ will be investigated in more detail in Chapter 6.

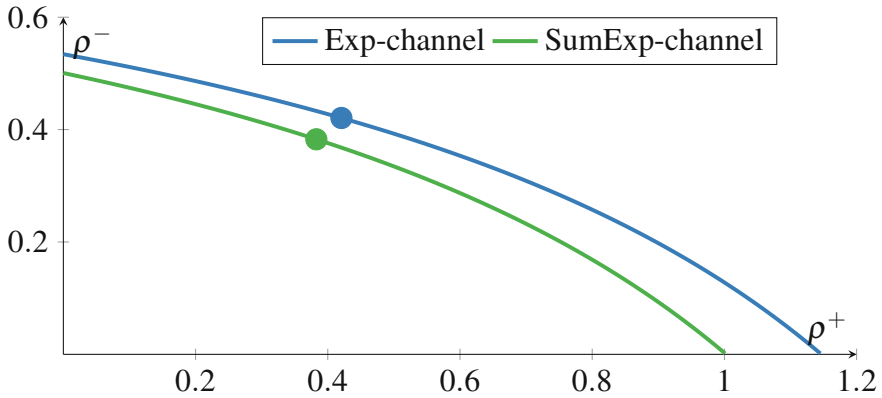


Figure 5.5: Possible combinations for ρ^+ and ρ^- for an 28 nm inverter (depending on the channel type). The circles indicate the case where $\rho^+ = \rho^-$. In this case, it can be seen that the adversary is weaker when using a SumExp-channel. We will investigate this in more detail in Chapter 6.

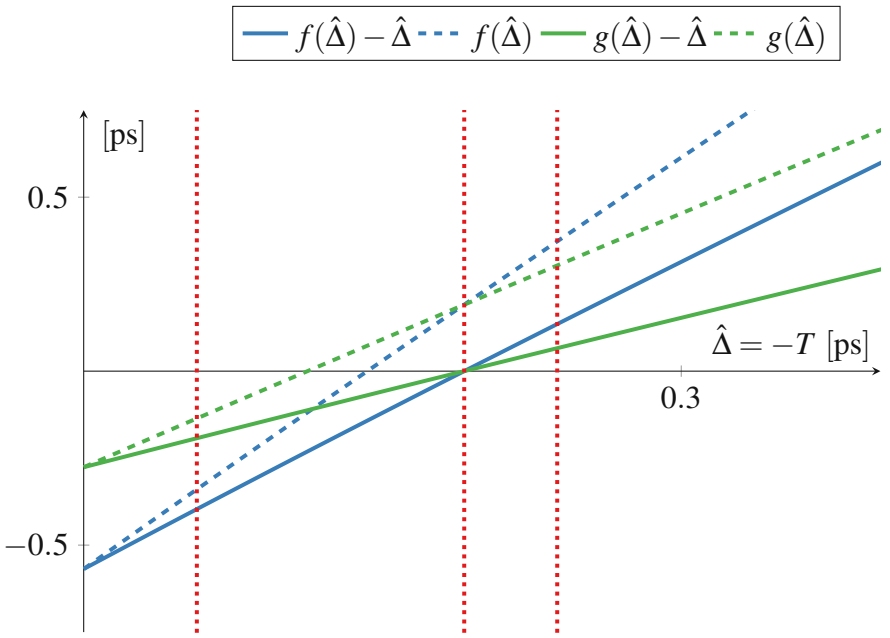


Figure 5.6: Examples for the various functions used throughout this section. The actual values are taken from the characterization of the same 28 nm inverter as in Figure 3.10. The red dotted lines indicate the critical values $\delta_\uparrow(-\Delta)+\eta^+_{min}-\Delta$, Δ, and $\overline{\Delta}$.

5.2 (Im)perfect cancellation

Figure 5.7 shows an example for perfect cancellation for an IDM and an η-IDM channel. For the IDM, the rising transition of the zero-time glitch is delayed by $\delta_\uparrow(T_1)$. The subsequent falling transition on $i(t)$, which happens at the same time instant as the rising transition, is then delayed by $\delta_\downarrow(T_2)$, where $T_2=-\delta_\uparrow(T_1)$. By applying the involution property, it can be seen that the final delay is $\delta_\downarrow(-\delta_\uparrow(T_1))=-T_1$. Hence the term perfect cancellation, since the state of the output is now as if the zero-time glitch never happened, i.e., the zero-time glitch is perfectly canceled. However, for the η-IDM channel, the behavior is different: The rising transition is delayed by $\delta_\uparrow(T_1)+\eta_n$, since the adversary is free to add a delay on top. In this example $\eta_n>0$. The falling transition is delayed by $\delta_\downarrow(T_2)+\eta_{n+1}$, where $T_2=-\delta_\uparrow(T_1)-\eta_n$. Again, it is assumed that $\eta_{n+1}>0$. In this case, the involution property cannot be applied and the final delay is $\delta_\downarrow(-\delta_\uparrow(T_1)-\eta_n)+\eta_{n+1}$. In this example, some interesting effects can be observed:

- While only the last output transition can be canceled in the IDM, this is not the case any more for the η-IDM. In fact, an arbitrary number of previous output transitions can be canceled by the adversary.
- The importance of the max terms in Eq. (3.28) and Eq. (3.29) becomes evident. Suppose that $T_1=\delta^\uparrow_\infty$: If the adversary adds a delay variation $\eta_n>0$, this would

result in $T_2 = -T_1 - \eta_n < -\delta^{\uparrow}_{\infty}$, which would be out of the definition range of $\delta_{\downarrow}(.)$ The max terms ensure that the delay functions are defined for all $T \in \mathbb{R}$.

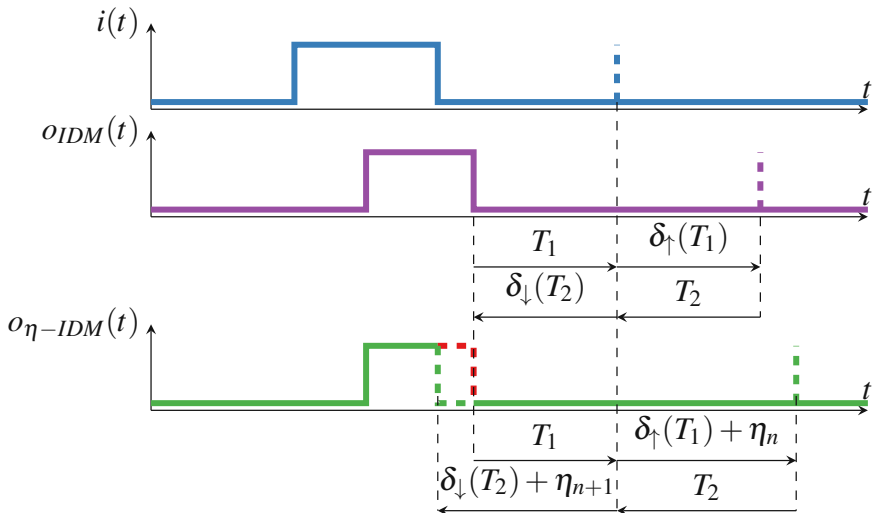


Figure 5.7: Illustration for perfect cancellation for IDM and possible issues in η-IDM.

5.3 Alternative reduction proof for impossibility of bounded SPF

In [FMN+18], the authors prove the impossibility of bounded SPF of η-IDM with a simple reduction to IDM. By setting all delay deviations to $\eta_n = 0$, the η-IDM directly degenerates to the IDM (see Theorem 9). They also state that η-IDM is able to cover deterministic effects like slightly different thresholds. In this section, a generalized reduction proof is presented, which shall resemble threshold variations.

Lemma 26. *There is no circuit that uses η-IDM channels, with delay functions as specified in Eqs.* (3.28) *and* (3.29) *that solves bounded SPF, if the adversary chooses a fixed deterministic delay offset*

$$\eta_n = \eta^{\uparrow} \text{ for all rising transitions, and} \tag{5.37}$$
$$\eta_n = \eta^{\downarrow} \text{ for all falling transitions,} \tag{5.38}$$

where $\eta^{\uparrow}, \eta^{\downarrow} \in [-\eta^{-}_{min}, \eta^{+}_{min}]$, with

$$\eta^{\uparrow} + \delta_{\uparrow}(\eta^{\downarrow}) > 0 \iff \eta^{\downarrow} + \delta_{\downarrow}(\eta^{\uparrow}) > 0 \tag{5.39}$$

for strict causality.

Proof. By setting $\Delta^+ = \eta^\uparrow$ and $\Delta^- = \eta^\downarrow$ and using Theorem 7, the η-IDM channel (which is an IP channel in this case) can be reduced to an IDM channel. From Theorem 5 it is known that IDM channels are unable to solve bounded SPF.

In order to prove strict causality, Lemma 8 can be employed. □

Since the proof for this reduction is based on the same insight as the one for IP channels (see Theorem 7), one might wonder what the added value is, since the CIDM already covers threshold variations. However, CIDM can only account for static threshold variations (determined during simulation and extraction of the delay functions). By contrast, the η-IDM is able to account for process variations, which show up only at run-time.

CHAPTER 6

Evaluation of the η-CIDM

In this chapter, some adaptions of the Involution Tool [ÖMFS21] are described. These improvements are mainly concerned with the characterization of circuits. Finally, extensive simulations have been performed to validate the claim that the η-CIDM (with the newly derived bounds) is indeed able to cover variations better than the η-IDM (with the old bounds).

6.1 Goals

The goals of the evaluation are to compare the actual delay functions of the individual cells of each circuit, with the derived delay function of our delay model. This requires the following steps:

1. We characterize the actual delay function $\delta^*_{\uparrow def}(T)$ and $\delta^*_{\downarrow def}(T)$ via SPICE-simulations under the default environment (25 °C, $V_{DD} = 0.9\,\mathrm{V}$). These delay functions serve as our baseline.

2. We repeat the SPICE-simulations under PVT variations and aging. By this, we obtain more sets of delay functions, e.g. $(\delta^*_{\uparrow x}(T), \delta^*_{\downarrow x}(T))$, where x identifies the environment conditions, e.g. $x \in \{2a, 85\,°\mathrm{C}, ff, \dots\}$.

3. Based on the delay functions under the default environment $\delta^*_{\uparrow def}(T)$ and $\delta^*_{\downarrow def}(T)$, we obtain the parameters $\overline{\delta}_{min}, \Delta^+, \Delta^-, \delta^{\uparrow}_{\infty}, \delta^{\downarrow}_{\infty}$. These parameters are combined with an actual switching waveform (e.g. Exp-channel, SumExp-channel) and result in the delay functions $\delta_{\uparrow}(T)$ and $\delta_{\downarrow}(T)$. This is how the CIDM models the cell.

4. Furthermore, η-CIDM adds a corridor for the delay variations on top of the delay functions. We want to check if the characterized delay functions, e.g. $\delta^*_{\uparrow def}(T)$,

$\delta^*_{\downarrow def}(T)$, $\delta^*_{\uparrow x}(T)$, $\delta^*_{\downarrow x}(T)$, where x indicates again the environment conditions, are within this corridor. If they are, our model is able to cover the behavior of the cell under the specific conditions. If not, we determine the deviation area in order to get a metric for comparing the old and the new bounds.

6.2 Characterization for different IDM variants

The characterization of gates in a circuit is a tedious task. However, it is a necessary prerequisite before running simulations with CIDM or IDM. In order to speed up the process, Python scripts have been added to the Involution Tool that automate this process.

6.2.1 Characterization variants

As described in [MÖS+21], there are three different ways how a circuit can be characterized:

- IDM*: The output threshold voltage V_{th}^{out*} of the last gate(s) is fixed (e.g. to $V_{DD}/2$) and the matching input threshold voltage V_{th}^{in*} is determined, which in term serves as the output threshold voltage for the previous gate(s). This process is repeated until all gates are characterized. Since we start characterizing the output gate(s), and go to the front, we call this backwards characterization. Note that it would also be possible to fix the input voltage V_{th}^{in*} of the gate(s) at the input and characterize the circuit in forward direction. However, since gates usually tend to have an amplification $\gg 1$, already the slightest deviations push the resulting V_{th}^{out*} towards GND or V_{DD}, which makes the characterization of the following gates hard. Moreover, IDM* has a serious drawback when it comes to forks: It requires that all gates fed by the fork have the same V_{th}^{in*}, which is certainly not the case in general.
- IDM+: Another approach is to fix the output threshold voltage for each gate (e.g. to $V_{DD}/2$) and to determine the matching V_{th}^{in*}. This has the obvious drawback that V_{th}^{out*} of the previous gate in general does not match V_{th}^{in*} of the successor gate(s). While these deviations might be negligible for small circuits, they become an issue for deep circuits (e.g. a clock tree), since the deviations might add up. Again, it is also possible to fix V_{th}^{in*} and to determine V_{th}^{out*} for each gate, which however might lead to very asymmetric V_{th}^{out*} because of the amplification of the gates. This approach allows to characterize all gates in parallel, since the threshold voltages of a gate are not depending on the threshold voltages of other gates.
- CIDM: This approach fixes the threshold voltages of all inputs and outputs a-priori to a fixed value, e.g. $V_{th} = V_{DD}/2$. In addition to δ_{min}, the resulting values for Δ^+ and Δ^- need to be determined. Like IDM+, all gates can be characterized in parallel. For CIDM, there is no distinction between forward and backward characterization, since both threshold voltages are fixed a-priori.

Note that the characterization script currently only supports *Single-Input-Single-Output* (SISO) gates. While it should be easily possible to extend the script for CIDM, IDM* would face another issue here: With *Multi-Input-Single-Output* (MISO) gates, circuits with feedback loops can be built. If an output of a gate is directly fed back to one of its inputs, it is required that $V_{th}^{in*} = V_{th}^{out*}$. Even if such a pair can be found for the gate in the feedback loop, it is highly unlikely that the preceding and succeeding gates match V_{th}^{in*} resp. V_{th}^{out*}.

Figure 6.1 shows the threshold voltages for a 15 nm inverter chain with alternating high- and low-threshold inverters determined by the above characterization methods. It can be clearly seen that the the error introduced by IDM+ is significant. For IDM*, there is no error introduced; however, the threshold voltages are quite asymmetric. This could make characterization difficult when the output threshold voltages are too close to V_{DD} or GND.

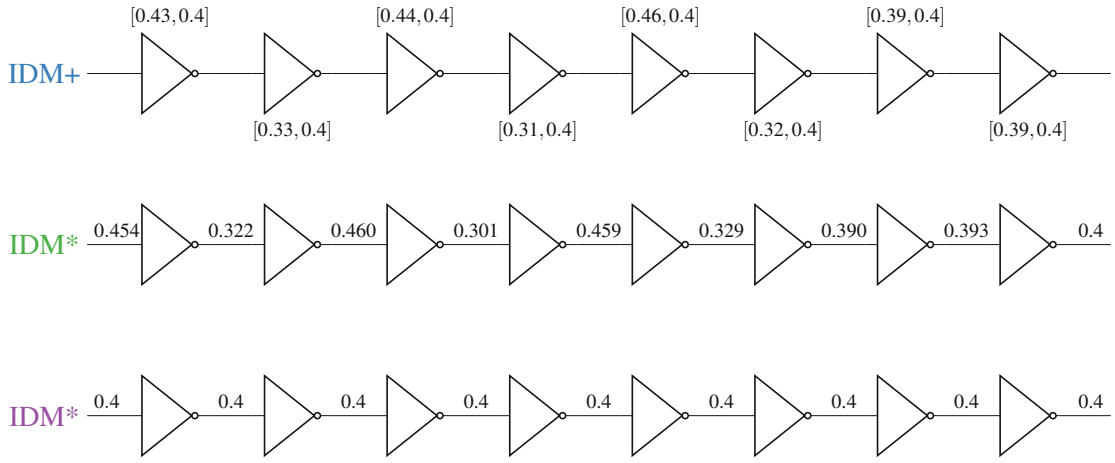


Figure 6.1: Different thresholds for a 15 nm high/low threshold inverter chain for the characterization methods IDM+, IDM* and CIDM (adapted from [Mai21]). The values $[x, y]$ for IDM+ are the matching input threshold voltage V_{th}^{in*} for our chosen output threshold voltage $V_{th}^{out*} = 0.4\,\mathrm{V}$. It can be clearly seen, that there is an error introduced for each interconnect. IDM* and CIDM do not introduce this error.

6.2.2 Principle of the characterization

The general idea of the characterization algorithm is to determine $\delta_{\uparrow}^{*}(T)$ and $\delta_{\downarrow}^{*}(T)$ for each cell of a circuit. The characterization is the same for all environment conditions, hence we neglect the subscript for the delay functions. In order to perform the characterization for a single cell, we run simulations where we apply a single up or down pulse with different pulse widths at the overall input of the circuit. By doing so, we can determine the delay function $\delta_{\uparrow}^{*}(T)$ and $\delta_{\downarrow}^{*}(T)$ for an individual cell. The characterization process for a single cell is divided into two main steps:

1. Determine δ_{min} for IDM+ and IDM* resp. $\overline{\delta}_{min}, \Delta^+$ and Δ^- for CIDM.
2. Characterize the complete delay function.

By repeating this process for all cells of a circuit, we can characterize the complete circuit.

Determination of the parameters: The idea of this step is illustrated in Figure 6.2. The input pulse V_{in}^{up} (lilac) width is varied in a binary search manner, until the output waveform $V_{out}(V_{in}^{up})$ (orange) at the cell that should be characterized exactly hits the desired V_{th}^{out*} (in our case $V_{th}^{out*} = V_{DD}/2$) at time t_o. The same is done for the opposite direction, until $V_{out}(V_{in}^{do})$ (green) exactly touches V_{th}^{out*}. In general, this happens at a time different from t_o. Hence, we need to shift the output trajectory $V_{out}(V_{in}^{do})$ and the corresponding input trajectory (V_{in}^{do}), such that the former touches V_{th}^{out*} at time t_o.

The distance between the crossing of V_{in}^{up} and V_{in}^{do} at the time t_i to the time t_o is $\overline{\delta}_{min}$ (resp. δ_{min} for IDM+ and IDM*). For IDM* and IDM+, the crossing point of the input waveforms also automatically determines V_{th}^{in*}. However, for CIDM, we still need to determine Δ^+ and Δ^-. These values are determined as the time it takes the input waveforms to get from V_{th}^{in} to V_{th}^{in*}. In Figure 6.2, we chose a $V_{th}^{in} \neq V_{th}^{out*}$ for illustration purposes. Δ^+ and Δ^- are determined as the distance between t_i and the V_{th}^{in}-crossing of respective waveform, i.e., the time it takes the input waveform to get from V_{th}^{in} to V_{th}^{in*}. Our illustration also shows that Δ^+ and Δ^- must have distinct signs (or are both 0 in the case of $V_{th}^{in} = V_{th}^{in*}$).

For the forward characterization, the approach is similar; however, it involves another binary search: First, we need to find two corresponding output waveforms that touch each other at some V_{th}^{out}. However, we do not a-priori know the value V_{th}^{out*} at which they will be touching, since this value is determined by the fixed V_{th}^{in*}. Hence, we start with $V_{th}^{out} = V_{DD}/2$. In a second step, we determine the matching V_{th}^{in} for our current V_{th}^{out}. In case the resulting V_{th}^{in} is too large, we need to find output waveforms that touch each other at a smaller V_{th}^{out} (in case of a buffer, for an inverter, we need a larger V_{th}^{out}). These two steps are repeated until V_{th}^{in} is reasonable close to our desired V_{th}^{in*}. Because of this second binary search, the forward characterization is more costly than the backward characterization. Moreover, it is more susceptible to error amplification and propagation. Therefore, we discourage using the forward characterization, since there is no real benefit compared to the backward characterization.

Characterization of the complete delay function: In the second step, we need to determine the complete delay function. In the first step we have determined the minimum input pulse length for which we get a threshold crossing on the output. To be more precise, we determined the length of the input pulse that is required to get a zero-time output pulse. The obtained values determine the leftmost part of the delay function that can be characterized. In order to determine now the delay for larger T, we apply larger pulses on the input. Figure 6.3 illustrates the idea: We determine the time between

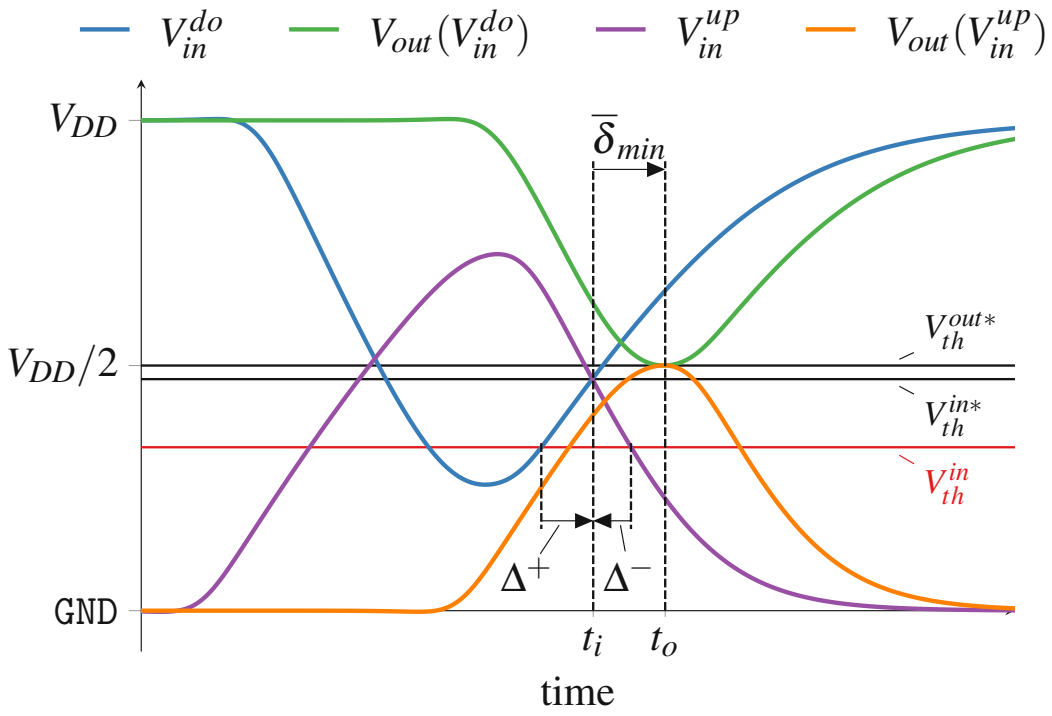


Figure 6.2: Characterization idea for a buffer (taken from [MÖS+21]).

the previous output transition t_{po} and the current input transition t_i, which results in $T = t_i - t_{po}$. In the case of the example, the value of T is still negative. Furthermore, the time of the output transition t_o is determined, and the distance between $t_o - t_i = \delta_{\downarrow}(T)$. Imagine now that the input pulse width is further increased. This also results in a larger input pulse V_{in} at the cell that is currently characterized. Hence, T gets larger (less negative). Moreover, also V_{out} gets larger, which results in a larger $\delta_{\downarrow}(T)$. This process is repeated for several values of T, until the delay function is reasonable well determined. Note that we are not able to determine the delay function for values of T which are smaller than $-\overline{\delta}_{min}$, since there is no output threshold crossing any more (this region corresponds to the region of sub-threshold pulses).

Performance of the characterization algorithm

The main cost of the characterization algorithm are caused by the SPICE simulations. Therefore, we store each simulation and reuse the results throughout the whole circuit, whenever possible. This saves a lot of time, especially since at the beginning of the binary search, the same values are encountered again and again for all cells. However, this speedup comes at the price of requiring additional space. For the circuit used in the sequel, the result files required approximately 64 MB. For larger circuits, e.g. the clock tree of a MIPS, used for example in [ÖMFS21], the necessary space was ≈ 2.23 GB.

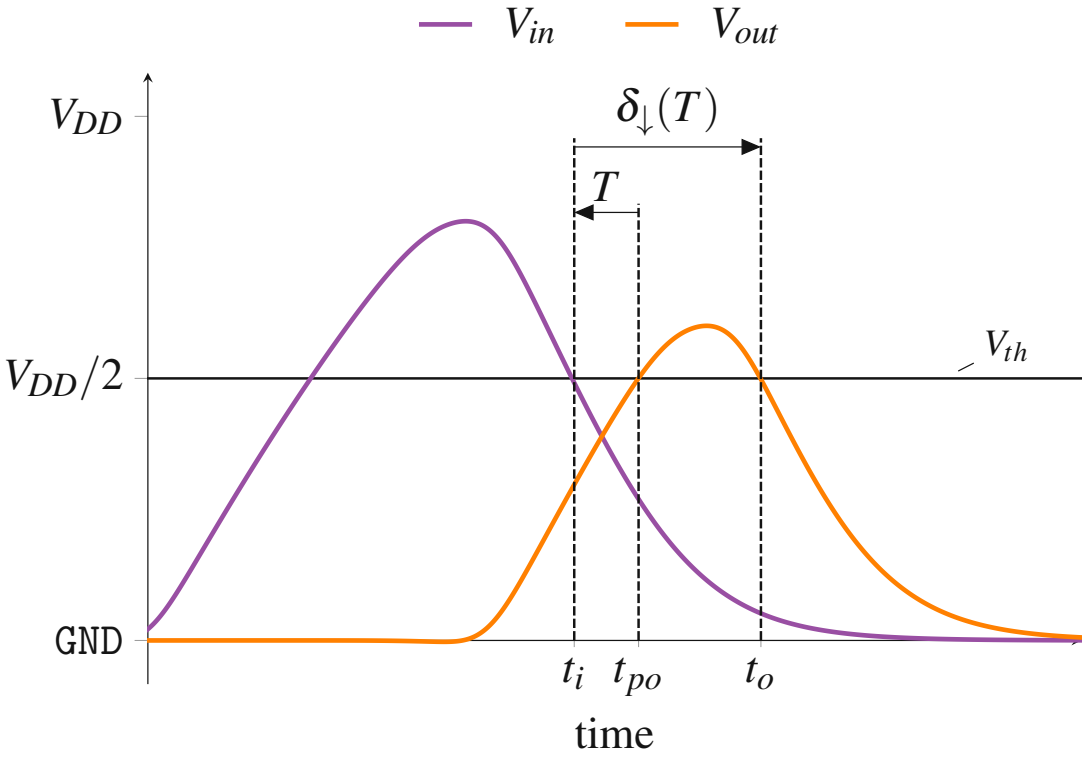


Figure 6.3: Determination of $\delta_{\downarrow}(T)$ for a single value of T for a buffer (adapted from [Mai21]).

6.2.3 Ideas for improving the characterization

Despite reusing the simulation result files, the biggest bottleneck of the characterization are still the SPICE simulations. Moreover, the result files are specific for this exact circuit and cannot usually be reused for other circuits. Even if we only make small changes to the circuit, the result files become worthless.

Hence, an important target for future work would be to come up with a more efficient characterization algorithm. One idea might be to perform extensive simulations on all kinds of cells and generate a lookup table for each cell. These tables could consider the input slope, fan in, fan out, and the load capacity. Based on these lookup tables, one could interpolate between the best fitting entries in the table and determine the parameters $\delta_{min}, \delta_{\infty}^{\uparrow}$, and $\delta_{\infty}^{\downarrow}$ for each cell. Taking this one step further, one could even interpolate between the delay functions themselves, rather than between the parameters, in order to obtain delay functions for each cell. This would also allow to accommodate more sophisticated channel types (e.g. the SumExp-channel). However, this approach trades accuracy for performance, and it needs to be investigated whether the obtained results were still reasonably accurate.

6.3 Results

The following results were obtained by using the adapted Involution Tool [ÖMFS21]. The focus of the evaluation is comparing the characterized delay functions ($\delta^*_{\uparrow x}(T)$ and $\delta^*_{\downarrow x}(T)$) under different environmental conditions with the calculated delay functions $\delta_\uparrow(T)$ and $\delta_\downarrow(T)$ of the η-CIDM. The calculated delay functions are based on the parameters from the characterized delay functions $\delta^*_{\uparrow def}(T)$ and $\delta^*_{\downarrow def}(T)$ under default environmental conditions (25 °C, $V_{DD} = 0.9$ V, typical process, no aging). The overall goal is to see how well the η-CIDM is able to cover the different characterized delay functions, i.e., if these delay functions are within the allowed corridor for the delay variations. If this is the case, we have shown that η-CIDM is indeed able to cover PVT variations and aging.

For the analog simulations, which are required to characterize the delay functions, Spectre (version 20.1) is used. The circuit under test is an inverter chain, consisting of 7 inverters (see Figure 6.4), which is synthesized with 28 nm SPICE models from UMC (G-05-LOGIC/MIXED_MODE28N-HPC-SPICE). The reason for choosing these is that the SPICE model cards contain parameters for simulating *Negative Bias Temperature Instability* (NBTI), *Positive Bias Temperature Instability* (PBTI) and *Hot Carrier Injection* (HCI). The first two effects cause a shift in the threshold voltage of nMOS transistors resp. pMOS transistors. The latter effect is caused by carriers which damage the gate oxid. A detailed explanation of these aging effects can be found in Section 2.4. These additional parameters can be used by tools like RelXpert to calculate aged versions of the transistors, and hence aged versions of the original circuit under test can be obtained. More recent 15 nm FinFET models, which have been used for example in [ÖMFS21], do not offer these parameters and are therefore unsuitable this thesis.

Unfortunately, the 28 nm library comes with no cell library; hence, the required cells had to be built from the existing SPICE models. According to the documentation, the model is applicable for lengths L_{DES} and widths W_{DES}, which are within

$$0.03\,\mu\text{m} \leq L_{DES} \leq 1\,\mu\text{m} \text{ and} \tag{6.1}$$

$$0.1\,\mu\text{m} \leq W_{DES} \leq 3\,\mu\text{m}. \tag{6.2}$$

Since these are large ranges for the parameters, the geometry of comparable 15 nm and 45 nm cells has been considered as well. We decided to choose the parameters $W_{DES} = 0.35\,\mu$m and $L_{DES} = 0.035\,\mu$m, since these values are between the values of the smaller and larger technologies, and fulfill the constraints of the library.

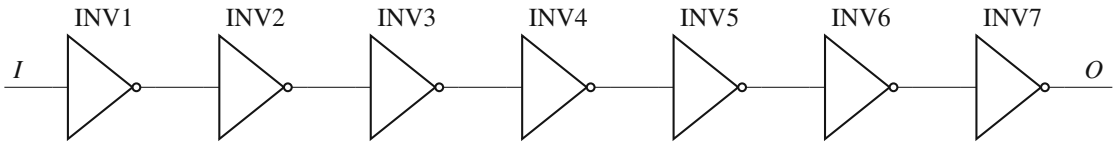


Figure 6.4: The circuit used for evaluation, a simple inverter chain consisting of 7 inverters.

For the characterization, we employed the CIDM characterization approach as described in Section 6.2. For each cell, we obtain the characteristic parameters δ_{min}, Δ^+, and Δ^- and the delay functions. Since the η-CIDM is based on logical channels, i.e., IDM channels, we first need to transform the characterized delay functions into IDM channels. This is done in the following two steps: Suppose we have two cells, c_1 followed by c_2, with the parameters $\delta_{min1}, \Delta_1^+, \Delta_1^-$ resp. $\delta_{min2}, \Delta_2^+, \Delta_2^-$. The characterized delay functions of c_1 are $\delta^*_{\uparrow_1}(.)$ and $\delta^*_{\downarrow_1}(.)$ with $\delta^{\uparrow}_{\infty 1}$ and $\delta^{\downarrow}_{\infty 1}$. Then, the delay functions of the logical channel between c_1 and c_2 are:

$$\delta^*_{\uparrow}(T) = \delta^*_{\uparrow_1}(T - \Delta_1^+ + \Delta_2^-) - \Delta_1^+ + \Delta_2^+ \text{ and} \tag{6.3}$$

$$\delta^*_{\downarrow}(T) = \delta^*_{\downarrow_1}(T - \Delta_1^- + \Delta_2^+) - \Delta_1^- + \Delta_2^-, \tag{6.4}$$

and

$$\delta^{\uparrow}_{\infty} = \delta^{\uparrow}_{\infty 1} - \Delta_1^+ + \Delta_2^+ \text{ and} \tag{6.5}$$

$$\delta^{\downarrow}_{\infty} = \delta^{\downarrow}_{\infty 1} - \Delta_1^- + \Delta_2^-. \tag{6.6}$$

The idea behind this transformation is to first remove the influence of pure delay shifter of the first cell c_1 (Δ_1^+, Δ_1^-). In a second step, the pure delay shifter (Δ_2^+, Δ_2^-) of the succeeding cell c_2 is added to the delay function.

Since there is no formula for calculating δ_{min} analytically, as described in Section 4.4, the resulting δ_{min} is calculated numerically.

All characterized delay functions $\delta^*_{\uparrow_x}(T), \delta^*_{\downarrow_x}(T)$ are transformed like this, before they are compared to the calculated delay functions $\delta_{\uparrow}(T), \delta_{\downarrow}(T)$, which are in turn based on the parameters $\delta_{min}, \delta^{\uparrow}_{\infty}, \delta^{\downarrow}_{\infty}$ of the characterized delay function under the default environment. Moreover, for the calculated delay function, a switching waveform model needs to be chosen. Possible waveform models are for example switching waveforms based on an exponential function (Exp-channel) or a sum of exponential functions (SumExp-channel) (see [ÖMFS21] for a detailed description of the switching waveforms).

For the delay functions which are not characterized under the default environment $(\delta^*_{\uparrow_x}(T), \delta^*_{\downarrow_x}(\mathrm{T}))$, a second step is necessary: In general, the resulting δ_{minx} of these delay functions is different from the one from the default delay functions (δ_{min}). Note that the characterization is only able to characterize the delay functions for $T \geq \delta_{minx}$. In some cases $\delta_{minx} < \delta_{min}$, and hence some of these delay functions are not fully characterized for $T \in [-\delta_{min}, \infty)$. To tackle this issue, we continue the delay functions to the left, such that they fulfill the involution property (see Definition 1). By doing so, all characterized delay functions are now defined for $[-\delta_{min}, \infty)$, which allows us to compare all delay functions for the same range. Note that this continuation is just an assumption on how the delay functions actually look like, since this range corresponds to the range of sub-threshold pulses. However, we assume that for $T \approx -\delta_{min}$ this is a reasonably good approximation of the delay functions.

6.3.1 Coverage under the default environment

The CIDM models the delay functions of each cell as perfect involutions. The extensions η-CIDM adds a corridor for allowed variations on top of this perfect involutions, and hence also allows "imperfect" delay functions, while still maintaining the important property of faithfulness.

In a first step, the deviation between the actual characterized delay functions $\delta^*_{\uparrow_{def}}(T)$, $\delta^*_{\downarrow_{def}}(T)$ under a default environment (temperature = 25 °C and $V_{DD} = 0.9$ V, typical process) and the calculated delay function $\delta_\uparrow(T), \delta_\downarrow(T)$ needs to be investigated. Figure 6.5 shows the characterized delay function $\delta^*_{\uparrow_{def}}(T)$ (blue) between the fourth (INV4) and fifth inverter (INV5) of the circuit under test (see Figure 6.4). The calculated delay function $\delta_\uparrow(T)$ (red), based on an exponential switching waveform (Exp-channel), is calculated from the obtained parameters for $\delta_{min}, \delta^\uparrow_\infty, \delta^\downarrow_\infty$. The η_{min} corridor shows the allowed (narrow) corridor for deviations when using the original borders. It can be seen that $\delta^*_{\uparrow_{def}}(T)$ is not covered for large ranges of T. When employing the newly derived border ($\eta(T)$ corridor), large portions of the applicable range for T are covered, which shows that the derived extensions of the bounds in Chapter 5 are indeed useful.

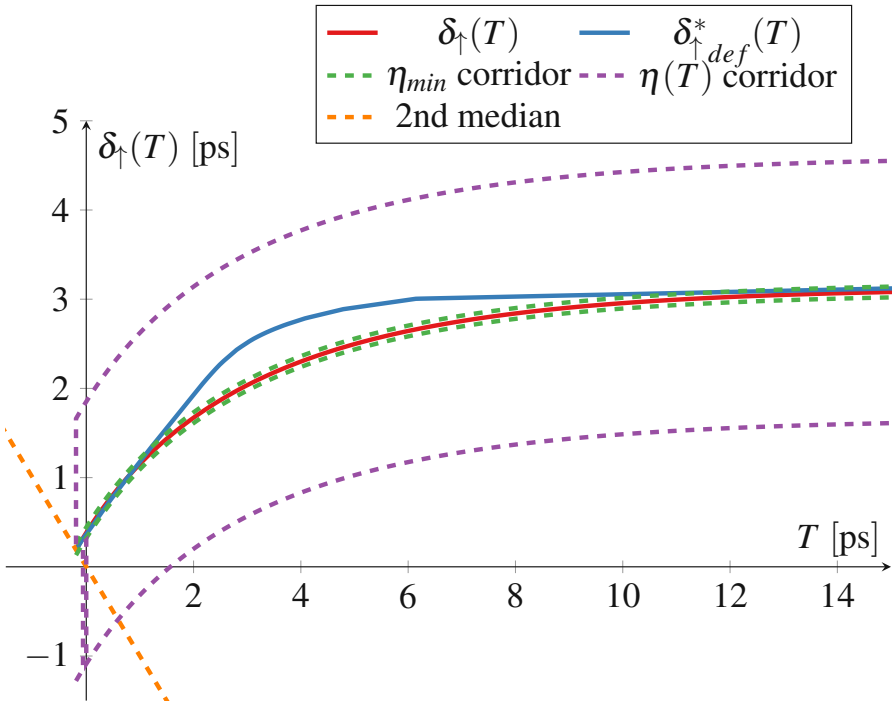


Figure 6.5: Actual (measured) delay function $\delta^*_{\uparrow_{def}}(T)$ between the fourth and fifth inverter of the simulated inverter chain. It is compared with the calculated delay function $\delta_\uparrow(T)$, based on an Exp-channel.

In [ÖMFS21], a more sophisticated delay channel, based on switching waveforms that

are a sum of exponential functions, has been introduced. This so-called SumExp-channel has more parameters (x_1, $\frac{\tau_1}{\tau_2}$) that can be used to fit the delay function of η-CIDM to the characterized delay functions $\delta^*_{\uparrow def}(T)$, $\delta^*_{\downarrow def}(T)$. Figure 6.6 shows the resulting delay function $\delta_\uparrow(T)$. It can be clearly seen that the delay function is covered better by the corridor. Of course, this is not a surprise, since the possible delay functions with SumExp-channel are a superset of the delay functions possible with an Exp-channel. By setting the weight factor (x_1) to 1, the SumExp-channel degenerates to an Exp-channel. Note that the resulting corridors are based on the delay function $\delta_\uparrow(T)$, and are therefore in general different for the SumExp-channel and the Exp-channel. Moreover, η_{min}, $\overline{\Delta}$, and Δ are in general different as well. The comparison between Figure 6.5 and Figure 6.6 shows that the choice of the underlying switching waveform is of utmost importance for the coverage of the delay function.

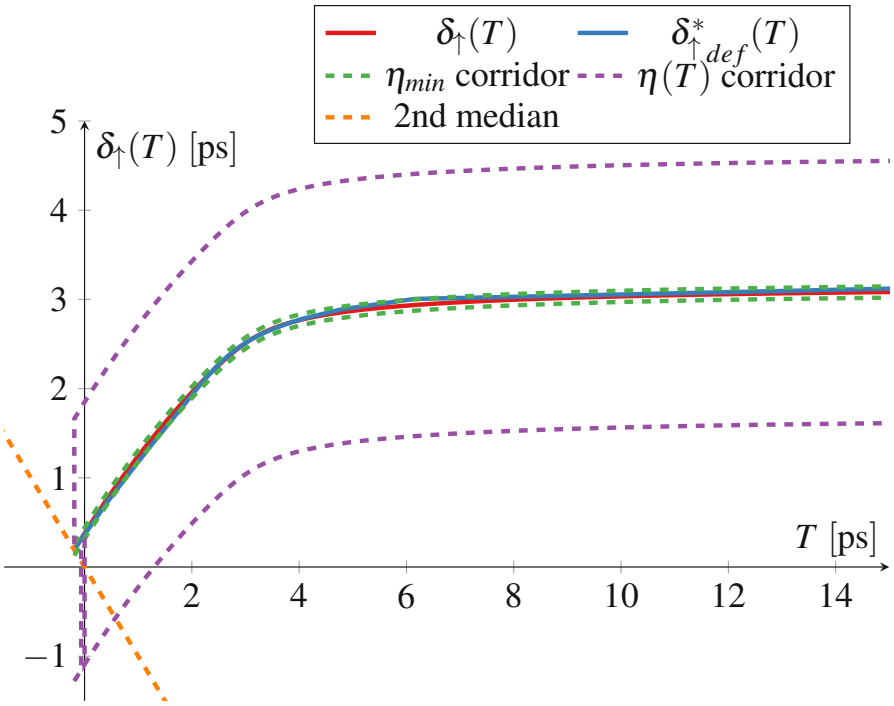


Figure 6.6: Actual (measured) delay function $\delta^*_{\uparrow def}(T)$ between the fourth and fifth inverter of the simulated inverter chain. It is compared with the calculated delay function $\delta_\uparrow(T)$, based on a SumExp-channel.

Figure 6.7 shows the difference between the characterized delay function $\delta^*_{\uparrow def}(T)$ and the calculated delay function $\delta_\uparrow(T)$. It can be clearly seen that the new bounds (in lilac) are large enough to cover the deviation. This figure also shows that the old bounds (in green) are quite restrictive.

One possible metric to compare the various channel types, and more importantly the old and new delay variation corridor, is to integrate over the area outside of the bounds. If

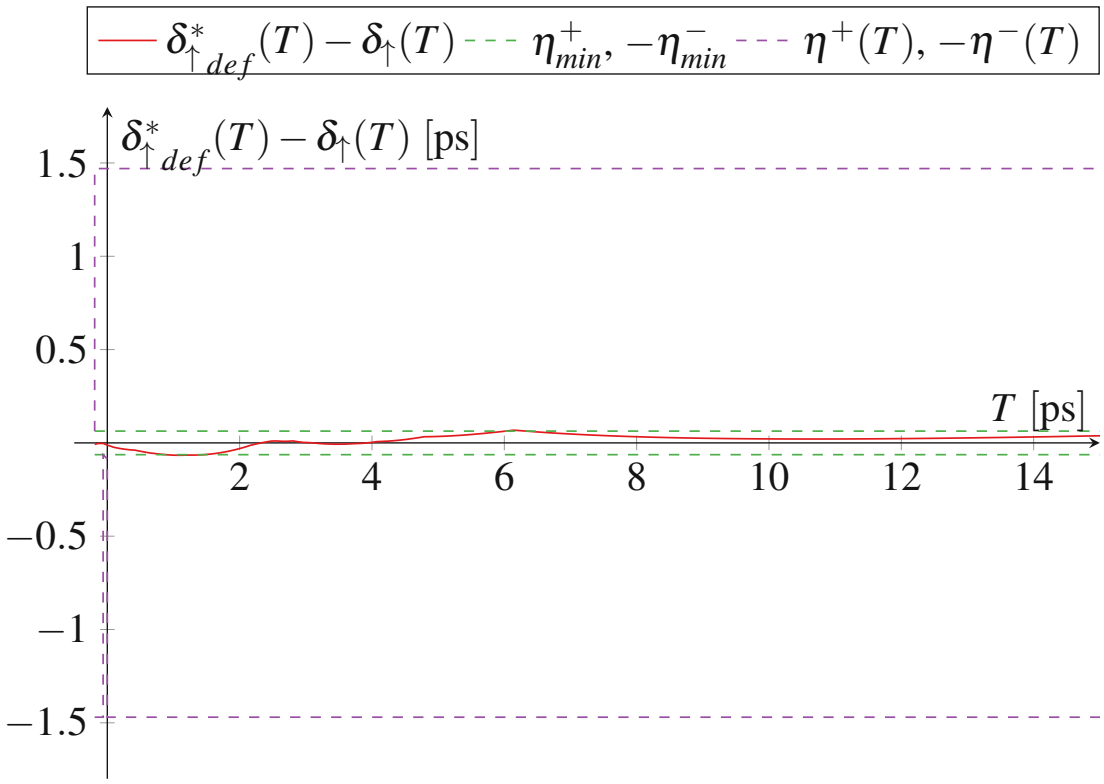


Figure 6.7: Difference between the measured delay function $\delta^*_{\uparrow def}(T)$ and the calculated delay function $\delta_\uparrow(T)$. Moreover, the old and new bounds are shown.

the value of $\delta^*(T) - \delta(T)$ is inside the corridor, the value is 0, and otherwise we take the absolute distance to the corridor. Moreover, since absolute values are meaningless, we normalize this value to the distance over which has been integrated, i.e., we consider the average deviation.

Figure 6.8 shows the average deviation for different channel types and compares the results for the old and new delay variation corridor. It can be seen that the choice of the switching waveform (i.e., Exp-channel vs. SumExp-channel) plays a vital role when using the old bounds. The SumExp-channel clearly outperforms the Exp-channel. However, when using the new bounds, the results can be improved even further, since now both channels are perfectly covered (as can be seen in Figures 6.5 and 6.6, where it can be clearly seen that the delay function $\delta^*_{\uparrow def}(T)$ is inside the new bounds).

Moreover, Figure 6.8 shows the deviation for certain ranges of the calculated delay function. We assumed that 0 % is the value of $\delta_\uparrow(-\delta_{min}) = \delta_\downarrow(-\delta_{min}) = \delta_{min}$, whereas 100 % is $\delta^\uparrow_\infty$ (resp. $\delta^\downarrow_\infty$ for falling delay functions). It can be seen, that the biggest issue of the Exp-channel is between 50 % and 75 % of the value of the delay function.

Until now, only the deviation for the rising delay function between INV4 and INV5

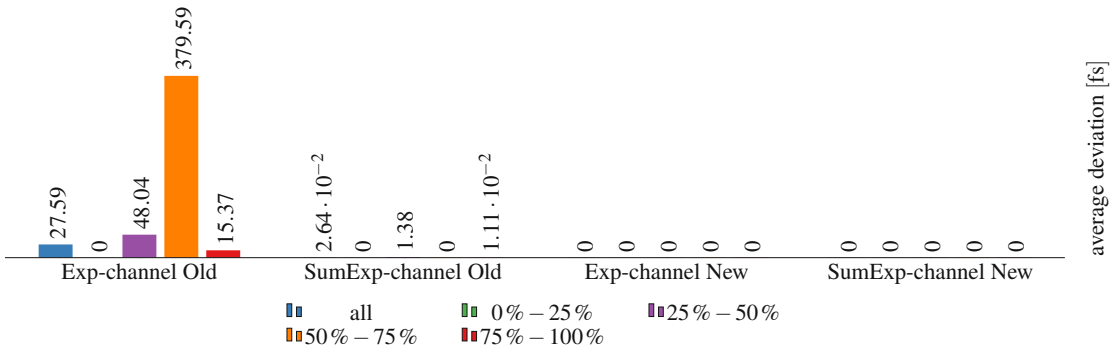


Figure 6.8: Average deviation for different channel types (Exp-channel, SumExp-channel) and old and new delay variation corridor. Deviation for the delay function for rising transitions for the channel between INV4 and INV5.

has been considered as a representative example. In the next step, we want to consider coverage for the complete circuit: Figure 6.9 shows the results. One thing that can be noticed is that the overall results for the old bounds are worse than for the single delay function as in Figure 6.8. This is primarily due to the fact that the delay functions between the first (INV1) and second (INV2) inverter are badly behaved, i.e., cannot be approximated properly by a SumExp-channel. Nevertheless, with the new bounds, this does not impact the coverage. Once again, it can be seen that the biggest issue of the Exp-channel is between 50 % and 75 %, which is actually no surprise: The values for 0 % and 100 % are fixed, and there is no deviation between the characterized delay function under the default environment and the calculated delay function; hence, the largest deviation must be in the middle part of the delay function.

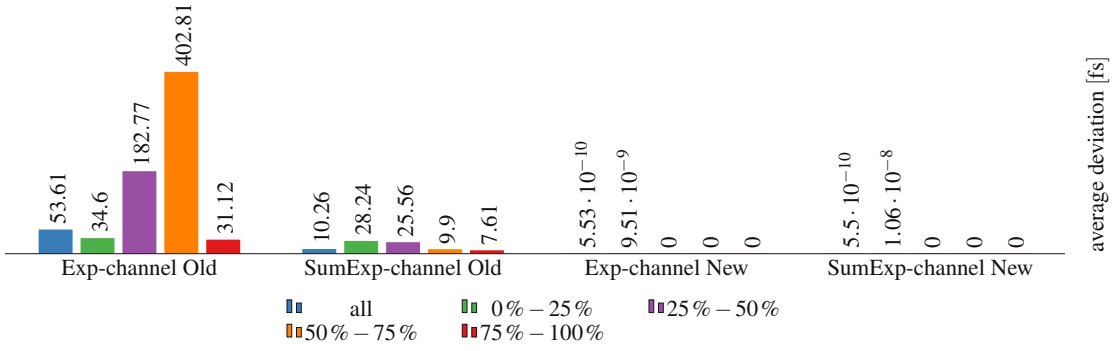


Figure 6.9: Average deviation for different channel types (Exp-channel, SumExp-channel) and old and new delay variation corridor.

Since the SumExp-channel is clearly superior to the Exp-channel, the following sections only consider SumExp-channels.

6.3.2 Process variations

In this subsection, we evaluate the influence of process variations. Two different types of process variations have been considered: (i) We used the different process corners, which are already supported by the library itself, and (ii) we varied the width of the transistors like it has been done in [FMN+18].

Figure 6.10 shows the influence of different process corners on the coverage. With the old borders, especially the corner with a slow nMOS transistor and a slow pMOS transistor (ss) caused huge deviations. However, with the new border, all simulated corners are now covered perfectly.

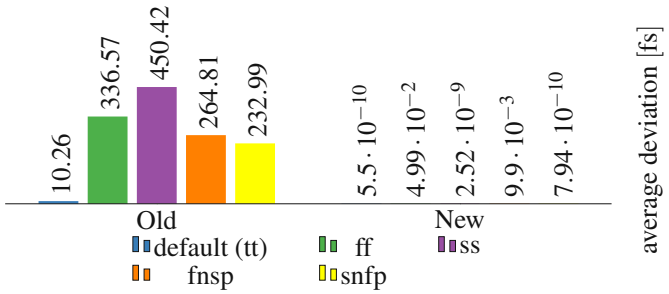


Figure 6.10: Average deviation for different process corners.

For the second set of simulations the width of the transistors has been varied, in accordance with the simulations performed in [FMN+18]. Figure 6.11 shows the results. Again, the new bounds achieve a perfect coverage. Moreover, it seems that the influence of variations of the transistor width is negligible, compared to the variations caused by different process corners.

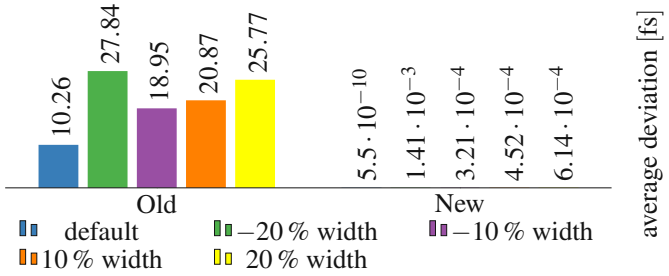


Figure 6.11: Average deviation for different values of W_{DES}.

6.3.3 Voltage variations

Figure 6.12 shows the influence of variations on the supply voltage. Obviously, a decrease in the supply voltage results in slower delay functions, i.e., δ_∞ becomes larger. Especially for the old bounds, this is a huge issue, since there are strict bounds ($[-\eta^-_{min}, \eta^+_{min}]$) throughout the complete range of T. The new model applies a less restrictive bound

$([-\eta_\infty^-, \eta_\infty^+])$ on the ranges where $T > 0$, and hence supply voltage variations can be covered better. Nevertheless, it seems that a decreased voltage supply by 20 % is still not covered well, even by the new model. However, a supply voltage drop by 20 % is a large deviation, which should not occur in modern ICs. On the other hand, an increased supply voltage (which results in a smaller δ_∞) is covered well by the new model. This is also in accordance with the results from Figure 6.10, where the corner (ss) caused the largest deviations, whereas the corner (ff) was covered better.

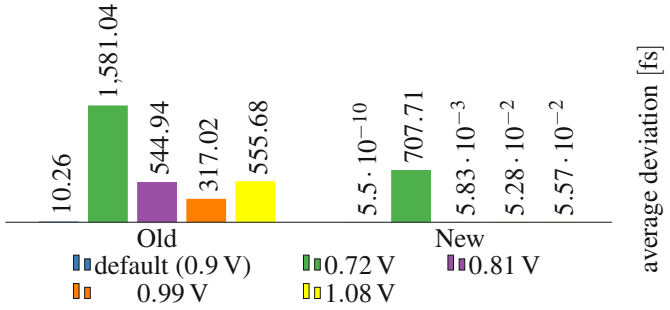


Figure 6.12: Average deviation for different supply voltages (V_{DD}).

6.3.4 Temperature variation

The influence of the ambient temperature on the coverage is shown in Figure 6.13. It can be seen, that for the old bounds, the deviation grows larger with increasing ambient temperature. However, the new bounds are able to perfectly cover these variations.

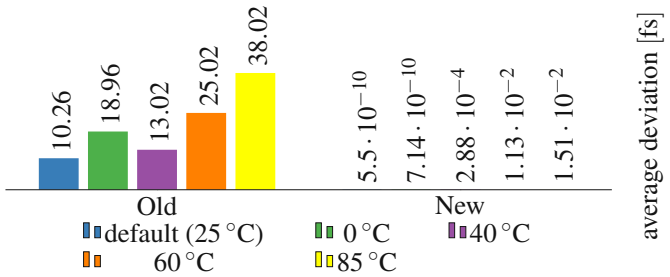


Figure 6.13: Average deviation for different ambient temperatures.

6.3.5 Age variations

For examining the influence of aging on the delay functions, we aged the circuit by using the tool RelXpert [Cad20]. This tool is able to use the aging parameters from the SPICE model card of each transistor, and obtain an aged model. While in the original circuit, each nMOS transistor and each pMOS transistor had the same parameters, this is not the case anymore in an aged circuit. The tool derives new parameters for each transistor, depending on the age that should be simulated and on the location of the

transistor. While some transistor are in paths with lots of switching activity (e.g. cells in the clock tree), others might be experience less switching activity. The switching activity of course has an influence on the aging of the transistor. In our inverter chain, each transistor experiences a similar switching activity, and hence the aged transistors had similar parameters.

One drawback of this approach is that the circuit files get large, even for moderately sized circuit like the inverter chain (from 2 KB of the original circuit to 23 MB of the aged circuit). The reason for this is that the complete model card is copied and adapted for each transistor. Unfortunately, we did not find and option for RelXpert, that would allow to re-use the default model card and only overrides adapted parameters. Especially for larger circuits, this might become an issue.

Figure 6.14 shows the influence of aging on the coverage of our model. Unsurprisingly, the deviation grows with increasing circuit age (since the circuit gets slower over time) for the old bounds. The new bounds are, however, able to perfectly cover these variations.

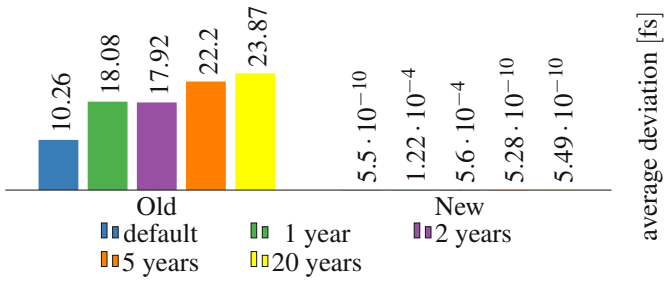


Figure 6.14: Average deviation for different circuit ages.

CHAPTER 7

Conclusion

7.1 Conclusion

In this Master thesis an extension for a digital delay model was proposed. This new extension, called η-CIDM is based on the IDM [FNNS20], which is the only known candidate for a faithful delay model. The goal of this extension is to combine the reduced characterization effort of the CIDM [MÖS+21], while maintaining the resistance against delay variations of the η-IDM [FMN+20].

In a first step, it has been proved that the new extension is indeed faithful. Since the original proof of faithfulness of the IDM is quite tedious, the faithfulness of the new extension has been proved by a reduction from η-CIDM to η-IDM (where CIDM in turn is reduced to IDM as well).

The next step was to further improve the applicability of the η-IDM. In the original version of the paper [FMN+20], the constraints on the constant bounds of the delay variation where quite strict. We have loosened these constraints by introducing a linear bound in the critical range of the delay function, and allowing an even looser constant bound outside this small critical range. We proved this by employing a similar technique as in the original paper [FMN+20].

The final step was to perform simulations with our newly derived extension. By comparing the delay functions of a simple inverter chain circuit under various conditions with our calculated delay function, we showed that the new bounds are indeed able to cover PVT variations and aging.

As a side-effect of this thesis, the Involution Tool [ÖMFS21] has been significantly extended. We added scripts for the characterization of circuits and improved the extensibility by allowing new delay channels to be implemented in Python (instead of VHDL).

7.2 Future work

We have shown that the new extension is able to cover delay variations. Nevertheless, there are still several open topics which might be of interest for future work.

Application in larger circuits It still remains to be shown how the new extension is able to cover variations in larger circuits. Currently, only a small inverter chain has been used. One logical next step could be to use the clock tree of a MIPS, as in [ÖMFS21], to show that the new extension is indeed also applicable for larger circuits.

Characterization However, in order to be able to apply this extension for larger circuits, the characterization script needs to be improved. Currently, we characterize each cell individually, which is already a quite computing time intensive task. One possible approach would be to implement a lookup table based approach, where each cell is characterized by interpolating the closest entries in a lookup table. This would also require an significant amount of computing time; however, these lookup tables could be used throughout all circuits.

Moreover, we were only able to compare the coverage of our extension for $T \geq -\delta_{min}$, i.e., for pulses which cross the threshold. For sub-threshold pulses, we are not able to obtain the delay function, and hence we cannot check the coverage of our extension.

Extension to multi-input gates Extending the η-IDM to multi-input gates is another open topic. One possible starting point could be to introduce non-deterministic delays in the simple hybrid model from [FMÖS22].

List of Figures

List of Tables

Glossary

η-CIDM η-Composable Involution Delay Model xi, xiii, xv, xvi, 2, 3, 5, 23, 29, 31, 33, 37, 51, 52, 54, 56–60, 62, 64, 67

η-IDM η-Involution Delay Model xi, xiii, xv, 2, 3, 14, 17, 19–21, 23, 24, 26, 28–30, 32–34, 36–38, 40, 42–44, 46, 48–51, 67–70

CCS Composite Current Source 1, 6, 7

CIDM Composable Involution Delay Model xi, xiii, xv, 2, 14–16, 23, 24, 26, 28–30, 32–34, 36, 50–54, 58, 59, 67, 69, 70

DDM Degradation Delay Model 1, 5–7

ECSM Effective Current Source Model 1, 7

FET Field-Effect Transistor 9, 10, 69

FinFET Fin Field-Effect Transistor 8, 9, 57

HCI Hot Carrier Injection 10, 57

IC Integrated Circuit 8, 64

IDM Involution Delay Model xi, xiii, xv, 1–3, 5, 7, 11–14, 17, 19, 24, 30, 33, 48–50, 52, 58, 67, 69, 70

MISO Multi-Input-Single-Output 53

MOSFET Metal-Oxid-Semiconductor Field-Effect Transistor 8–10

NBTI Negative Bias Temperature Instability 10, 57

NCSim A set of tools related to the design and verification of ASICs, published by Cadence Design Systems 6, 7

PBTI Positive Bias Temperature Instability 10, 57

PDN Power Distribution Network 8

PVT Process Voltage and Temperature xi, xiii, xv, 8, 9, 17, 51, 57, 67, 69

Questa Sim A program by Mentor Graphics for simulating hardware description languages like VHDL, Verilog and SystemC 7

SCE Short-Channel Effect 8

SISO Single-Input-Single-Output 53

Spectre Simulation program for SPICE simulations 57

SPF Short-Pulse Filtration 1, 2, 7, 8, 12–14, 19, 29, 31–33, 37, 49, 50, 70, 73

SPICE Simulation program with integrated circuit emphasis is a software which is used to simulate digital, analog and mixed electrical circuits xi, xiii, 1, 57, 64, 76

UDRM User Defined Reliability Model 10

VCS A suite of tools for functional verification and simulation, published by Synopsis 6, 7

Bibliography

[AM05] M. Alam and S. Mahapatra. "A comprehensive model of PMOS NBTI degradation". In: *Microelectronics Reliability* 45.1 (2005), pp. 71–81. ISSN: 0026-2714. DOI: https://doi.org/10.1016/j.microrel.2004.03.019. URL: https://www.sciencedirect.com/science/article/pii/S0026271404001751.

[AR13] M. Aparicio Rodriguez. "Modelling and Simulation of the IR-Drop phenomenon in integrated circuits". Theses. Université Montpellier II - Sciences et Techniques du Languedoc, Dec. 2013. URL: https://tel.archives-ouvertes.fr/tel-00998547.

[BDJCA+00] 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.

[BDJCV06] M. J. Bellido-Díaz, J. Juan-Chico, and M. Valencia. *Logic-Timing Simulation and the Degradation Delay Model*. London: Imperial College Press, 2006.

[Cad15] Cadence Design Systems. *Effective Current Source Model (ECSM) Timing and Power Specification*. Version 2.1.2. Cadence Design Systems. Jan. 2015.

[Cad20] Cadence Design Systems. *Spectre® RelXpert Reliability Simulator User Guide*. Product Version 20.1. Cadence Design Systems. Sept. 2020.

[CB09] R. Chadha and J. Bhasker. *Static Timing Analysis for Nanometer Designs: A Practical Approach*. Springer, 2009.

[FMN+18] 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)*. 2018, pp. 1327–1332. DOI: 10.23919/DATE.2018.8342219. URL: https://ieeexplore.ieee.org/document/8342219/.

[FMN+20] M. Függer, J. Maier, R. Najvirt, T. Nowak, and U. Schmid. *A Faithful Binary Circuit Model with Adversarial Noise*. 2020. arXiv: 2006.08485 [cs.OH].

[FMÖS22] A. Ferdowsi, J. Maier, D. Öhlinger, and U. Schmid. “A Simple Hybrid Model for Accurate Delay Modeling of a Multi-Input Gate”. Accepted for DATE’2022. 2022.

[FNNS20] M. Függer, R. Najvirt, T. Nowak, and U. Schmid. “A Faithful Binary Circuit Model”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.10 (2020), pp. 2784–2797. DOI: 10.1109/TCAD.2019.2937748.

[FNS16] M. Függer, T. Nowak, and U. Schmid. “Unfaithful Glitch Propagation in Existing Binary Circuit Models”. In: *IEEE Transactions on Computers* 65.3 (2016), pp. 964–978. DOI: 10.1109/TC.2015.2435791.

[Hal20] B. Halak, ed. *Ageing of Integrated Circuits*. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-23781-3. URL: https://doi.org/10.1007/978-3-030-23781-3.

[IEE01] IEEE Computer Society. *IEEE Standard for VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification*. IEEE Std 1076.4-2000. IEEE Computer Society. Sept. 2001.

[IEE06] IEEE Computer Society. “IEEE Standard for Verilog Hardware Description Language”. In: *IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001)* (2006), pp. 1–590. DOI: 10.1109/IEEESTD.2006.99495.

[Lor12] D. Lorenz. “Aging Analysis of Digital Integrated Circuits”. https://d-nb.info/1023128659. PhD thesis. Technische Universität München, 2012.

[Mai21] J. Maier. *A Composable Glitch-Aware Delay Model*. https://ti.tuwien.ac.at/institute/teaching/ti-research-presentations/composable_involution_delay_model.pdf. Accessed: 2021-11-12. Nov. 2021.

[Mar20] L. Mari. *A Comparison of FinFET Configurations*. https://eepower.com/technical-articles/a-comparison-of-finfet-configurations/. Accessed: 2021-06-12. Oct. 2020.

[Mar77] L. R. Marino. “The Effect of Asynchronous Inputs on Sequential Network Reliability”. In: *IEEE Transactions on Computers* C-26.11 (1977), pp. 1082–1090. DOI: 10.1109/TC.1977.1674754.

[Mas04] J. Massey. “NBTI: what we know and what we need to know - a tutorial addressing the current understanding and challenges for the future”. In: *IEEE International Integrated Reliability Workshop Final Report, 2004*. 2004, pp. 199–211. DOI: 10.1109/IRWS.2004.1422784.

[Men05] Mentor Graphics Corporation. *Eldo User’s Manual*. Software Version 6.6_1 Release 2005.3. 2005.

[Men16] Mentor Graphics Corporation. *ModelSim® SE Command Reference Manual*. Software Version 10.5c. 2016.

[MÖS+21] 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*. New York, NY, USA: Association for Computing Machinery, 2021, 147–154. ISBN: 9781450383936. URL: https://doi.org/10.1145/3453688.3461519.

[NFP11] S. Nazarian, H. Fatemi, and M. Pedram. “Accurate Timing and Noise Analysis of Combinational and Sequential Logic Cells Using Current Source Modeling”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 19.1 (2011), pp. 92–103. DOI: 10.1109/TVLSI.2009.2024945.

[Öhl18] D. 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.

[ÖMFS21] D. Öhlinger, J. Maier, M. Függer, and U. 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.

[SLD+03] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif. “Full chip leakage-estimation considering power supply and temperature variations”. In: *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED ’03.* 2003, pp. 78–83. DOI: 10.1109/LPE.2003.1231839.

[Syn16] Synopsis Inc. *CCS Timing Library Characterization Guidelines*. Version 3.4. Synopsis Inc. 2016.

[Ung71] S. H. Unger. “Asynchronous Sequential Switching Circuits with Unrestricted Input Changes”. In: *IEEE Transaction on Computers* 20.12 (1971), pp. 1437–1444.

[YJ14] Y. Yang and N. K. Jha. “FinPrin: FinFET Logic Circuit Analysis and Optimization Under PVT Variations”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.12 (2014), pp. 2462–2475. DOI: 10.1109/TVLSI.2013.2293886.

[ZKN+06] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik. “A Comparative Study of NBTI and PBTI (Charge Trapping) in SiO2/HfO2 Stacks with FUSI, TiN, Re Gates”. In: *2006 Symposium on VLSI Technology, 2006. Digest of Technical Papers.* 2006, pp. 23–25. DOI: 10.1109/VLSIT.2006.1705198.