Transistor-Level Analysis of Dynamic Delay Models

Jürgen Maier, Matthias Függer, Thomas Nowak, Ulrich Schmid

Transistor-Level Analysis of Dynamic Delay Models

Jürgen Maier*, Matthias Függer†, Thomas Nowak‡, Ulrich Schmid*
*ECS Group, TU Wien
{jmaier, s}@ecs.tuwien.ac.at
†CNRS & LSV, ENS Paris-Saclay, Université Paris-Saclay & Inria
mfuegger@lsv.fr
‡Université Paris-Sud
thomas.nowak@lri.fr

***Abstract*—Delay estimation is a crucial task in digital circuit design as it provides the possibility to assure the desired functionality, but also prevents undesired behavior very early. For this purpose elaborate delay models like the Degradation Delay Model (*DDM*) and the Involution Delay Model (*IDM*) have been proposed in the past, which facilitate accurate dynamic timing analysis: Both use delay functions that determine the delay of the current input transition based on the time difference T to the previous output one. Currently, however, extensive analog simulations are necessary to determine the (parameters of the) delay function, which is a very time-consuming and cumbersome task and thus limits the applicability of these models.**

In this paper, we therefore thoroughly investigate the characterization procedures of a CMOS inverter on the transistor level in order to derive analytical expressions for the delay functions. Based on reasonably simple transistor models we identify three operation regions, each described by a different estimation function. Using simulations with two independent technologies, we show that our predictions are not only accurate but also reasonably robust w.r.t. variations. Our results furthermore indicate that the exponential fitting proposed for *DDM* is actually only partially valid, while our analytic approach can be applied on the whole range. Even the more complex *IDM* is predicted reasonably accurate.

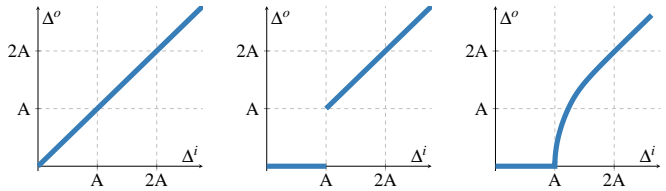
***Index Terms*—Circuit models, glitch propagation, delay models, pulse degradation, model parameterization**

I. Introduction

Delay estimation is a very important task in state-of-the-art design of digital circuits and systems, in particular, asynchronous ones: Already at early development stages, it is crucial to know whether the intended behavior is achieved by the current design or not, as costs to fix increase the later a flaw is detected. While, at early design stages, a coarse and very quick estimation is sufficient, later stages need more accurate and reliable results. Therefore different levels of accuracy are required and desirable when analyzing the timing behavior of a circuit.

The most accurate method currently available is an analog simulation of the systems of differential equations that describe a circuit using e.g. the popular *SPICE* tools. Various circuit models, differing in complexity and accuracy, are available for this purpose, whereat the appropriate model parameters for a given circuit technology are generally provided by the manufacturer. The main drawback of analog simulations is however their huge (time) complexity, which quickly renders them inappropriate with increasing circuit size.

This research was partially funded by the Austrian Science Fund (FWF) projects SIC (P26436) and RiSE (S11405), by the Institut Farman project Dicimus, the CNRS project PEPS DEMO, and the ANR project FREDDA (ANR-17-CE40-0013).



(a) pure (b) inertial (c) Newtonian

Fig. 1: Output (Δ^o) over input (Δ^i) pulse width for different delay models.

Significant speed-ups are achieved by using digital abstractions, namely, a time-continuous model with two discrete values (HI and LO). Such models are best viewed as delay models, which predict the output state transition (= threshold crossing) times based on the input state transition times. Several different instances exist: In the *pure delay* model, the input transitions are simply delayed by a constant amount of time, possibly with differing values for up- and down-transition. The latter causes the pulse width Δ^o of some output to be either bigger or smaller by a constant time compared to the input pulse width Δ^i. An example for equal delays is shown in Fig. 1a. In the *inertial delay* model, output pulse widths are determined like in the pure delay case, but input pulses with width Δ^i smaller than some threshold width A are dropped, i.e., not propagated to the output at all. Fig. 1b gives an example, where one can clearly see the jump at $\Delta^i = A$. Although such pulse cancellations can also be observed in *Newtonian* physics (see Fig. 1c) the output pulse width does not change abruptly in that case but degrades gradually. Whereas there exist delay models, as we will describe later, that are able to model Newtonian physics, the widespread inertial and pure delay models are not. This is a direct consequence of using static delay values, i.e., one that stay the same throughout a complete timing analysis run.

Still, state-of-the-art industry-grade timing analysis tools use inertial and/or pure delay models in conjunction with elaborate delay prediction models like ECSM [1] and CCSM [2], which determine suitable delay values. For accurate predictions,

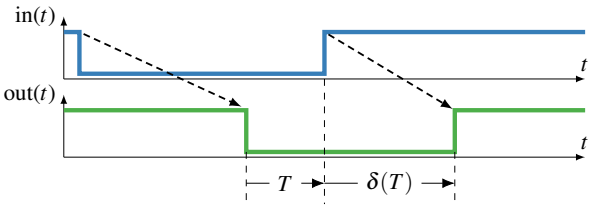


Fig. 2: Principle functionality of a single-history delay model. Based on the input-to-previous output transition time T, the delay $\delta(T)$ is determined [6].

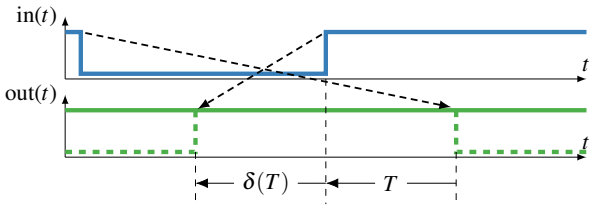


Fig. 3: The input pulse is so short that the transitions at the output appear in reverse order (dashed lines), i.e., cancel. Note that here $T < 0$ and $\delta(T) < 0$.

much effort is invested here: Extensive simulations are carried out to characterize gates for different input slopes and loads.

To model Newtonian delays, one obviously needs a dynamic delay value described by a function that depends on a parameter like the input pulse width Δ^i. Below some threshold input pulse width, pulses should get canceled, and with growing value, the output pulse width Δ^o should gradually increase. In that vein, Bellido-Díaz et al. proposed the *Degradation Delay Model* (*DDM*) [3], [4], which allows the delay $\delta(T)$ of the current transition to depend on the parameter T, defined as the time difference between the current input and the most recent previous output transition, shown in Fig. 2. However, Függer et al. showed in [5], [6] that *DDM* is not faithful, in the sense that one can prove circuits correct in *DDM* that are not implementable in reality.

In [7], Függer et al. therefore proposed the *Involution Delay Model* (*IDM*) as the only candidate for a faithful model known so far. It requires $\delta(T)$ to satisfy an *involution property*, namely, $-\delta(-\delta(T)) = T$. In case the delays for rising and falling input transitions differ, and denoting them by $\delta_\uparrow$ and $\delta_\downarrow$ respectively, the involution property reads $-\delta_\uparrow(-\delta_\downarrow(T)) = -\delta_\downarrow(-\delta_\uparrow(T)) = T$. The authors showed that this guarantees a continuity property of the output, in the sense that, if some input pulse-width goes to zero, then the output behaves as if the pulse was not there at all.

For a given circuit, the delay functions $\delta(T)$ for *DDM* and *IDM* can be acquired numerically by using extensive *SPICE* circuit simulations, digitizing them and then extracting the values for T and $\delta(T)$. Very little is known yet about analytic expressions of the delay functions that can/shall be used. This is also true for *DDM*, whose exponential delay function approximation has not been derived analytically, but rather determined as a good fit experimentally. Moreover, any delay function that is of practical use has to be parameterized in order to accommodate different circuit technologies and operating conditions. Understanding the underlying processes that govern the parameters can reduce the effort considerably and is hence of utmost practical importance.

Main contributions: In this paper, we address the latter task: We carry out a transistor-level analysis on the characterization procedure of an inverter and derive an analytic formula for the delay function that depends on technological and operational parameters only. The main goal is not to replace the currently used delay functions, but to gain a deeper understanding of the underlying processes and thus pave the way to easier, faster and better characterization methods.

Note that we constrain ourselves to inverters here as (1) *IDM* is currently limited to single-input single-output gates and (2) the approach for multi-input gates in *DDM* still lacks a proper consideration of significant degradation effects such as the Charlie effect [8]. A proper extension to multi-input gates is a very important avenue of future research, however, which will hopefully be supported by the results of this paper.

We start our considerations with a detailed look at *DDM* and recreate the characterization procedure used in [3] using transistor-level modeling. In order to get an accurate description, which turned out to be quite robust w.r.t. different technologies, several different operating regions of the delay function need to be distinguished and investigated separately. Despite the simple models and the (over-)simplifications used throughout the analysis, the results nicely match reality even for different technology nodes. Finally, we show that the results can be extended to *IDM* reasonably well.

Related work: Given the very few dynamic delay models that have been proposed so far, there is not much work devoted to characterization. Besides the well-known state-of-the-art ECSM [1] and CCSM [2], we are only aware of some related work on *DDM* [9], [3], [10], [11]. Albeit all these studies relate the delay function to technology parameters like threshold voltages and load capacitances, they share the common weakness that certain weights need to be determined via fitting to simulation data.

Our paper is organized as follows: Section II gives a short introduction to *DDM* and *IDM*, followed by a quick overview of the used transistor model in Section III. The refined characterization procedure, our physically guided abstraction and the resulting delay functions are provided in Section IV. Our paper is concluded in Section V, where we also provide a short outlook to future research.

II. Dynamic Delay Models for Digital Circuits

The dynamic delay models *DDM* and *IDM* are, like pure and inertial delay channels, instances of *single-history channels*. Their characteristic property is that their delays depend only on the time difference T between the current input transition and the previous output transition. A simple example is shown in Fig. 2.

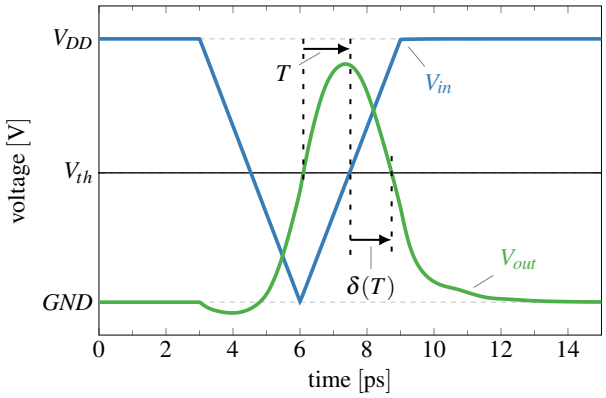


Fig. 4: Characterization procedure of *DDM*, showing the input slopes (V_{in}) and output trajectory (V_{out}) of an inverter gathered from *SPICE* simulations in 15 nm FinFET technology ($V_{DD} = 0.8\,V$).

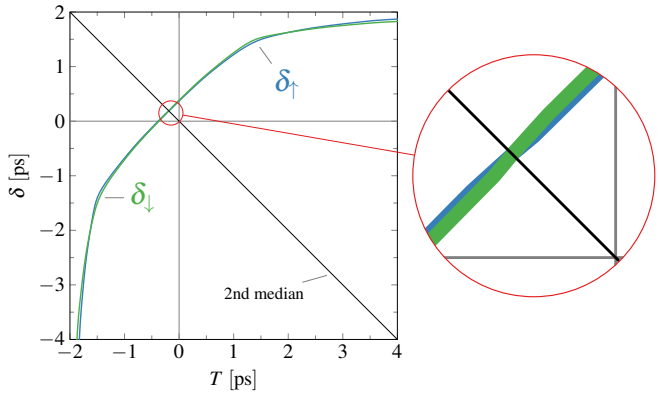


Fig. 5: Involution delay function of an inverter in 15 nm FinFET technology. Note that $\delta_\uparrow$ and $\delta_\downarrow$ have to meet at the second median to fulfill the involution property.

Note that certain values of parameter T result in negative values of $\delta(T)$, which corresponds to the case that the input pulse width was too short to reach the threshold. We refer to this situation (depicted in Fig. 3), where the next scheduled output transition actually happens before the previous one, as *pulse cancellation*, i.e., when $\delta(T) < -T$.

A. Degradation Delay Model (DDM)

DDM has been introduced by Bellido-Díaz et al. in [10], based on some preceding versions [9], and was later extended several times. A comprehensive overview of the model and all of its features is given in [4].

To determine the shape of the delay function,[1] the authors used extensive *SPICE* simulations with input ramps. The values for T resp. δ are measured as the time difference from the input's second crossing of the threshold voltage ($V_{th} = V_{DD}/2$) to the output's first threshold crossing (T) resp. the second input to the second output crossing (δ) as shown in Fig. 4. Note that we will refer to such threshold crossings as *transitions*, i.e., Heaviside jumps. By varying the input pulse width, the delay function $\delta(T)$ can be determined numerically.

Careful analysis allowed Bellido-Díaz et al. [3] to fit their delay function to a decaying exponential function, i.e.,

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

where t_{p0} denotes the maximum delay, T_0 the value for which $\delta(T) = 0$, and τ the rate by which t_{p0} is approached. Moreover, the authors also provided qualitative physical explanations for them as well as characterization methods for the parameters T_0 and τ. Note carefully, however, that extensive *SPICE* simulations are still required for this purpose.

[1] In the papers from Bellido-Díaz et al., the delay function is called $t_p(T)$. For the sake of uniformity, however, we will use $\delta(T)$ throughout this paper.

B. Involution Delay Model (IDM)

Later, Függer et al. proved [5] that all bounded single-history delay models, including all existing models and thus also *DDM*, are not faithful: circuits that cannot be built in reality can be proved correct in the model, or vice versa. Herein, *bounded* means that $\delta(T)$ is bounded from below for all finite values of T, i.e., it never approaches $-\infty$. In the sequel, Függer et al. proposed the unbounded single-history *IDM* [7], which requires a delay function that satisfies $-\delta_\uparrow(-\delta_\downarrow(T)) = T$. As already mentioned, this property assures that input pulses with a pulse width approaching zero have diminishing effects on the output, such that a zero-time glitch cannot be distinguished from no pulse at all.

Clearly, *IDM* can be numerically characterized by the same approach as *DDM*. To somehow mask the fact that the former does not consider input slopes at all, but only transition times, properly shaped input signals were used in the characterization process. Although T and δ are extracted in the same fashion as before, it was not possible to find an analytic function with a good fit so far.

Fig. 5 shows a fully assembled delay function for *IDM*. Only the parts above the second median ($T = -\delta(T)$) can be characterized by *SPICE* simulations. Below this line, we experience $\delta(T) < -T$ and thus cancellation, i.e., no output transitions. To extrapolate the values in this region, the simulated delay functions were mirrored along the second median: the upper part of $\delta_\uparrow$ becomes the lower part of $\delta_\downarrow$ and vice versa. This of course guarantees the required involution property by construction.

III. CMOS Circuits

Most digital circuits today are manufactured using complementary metal-oxide-semiconductor (CMOS) technology, where two (complementary) transistor types (nMOS and pMOS) are used (see Fig. 6). While their overall structure is very similar, their internal composition differs, which results in deviating physical and, hence, electrical properties. Both,

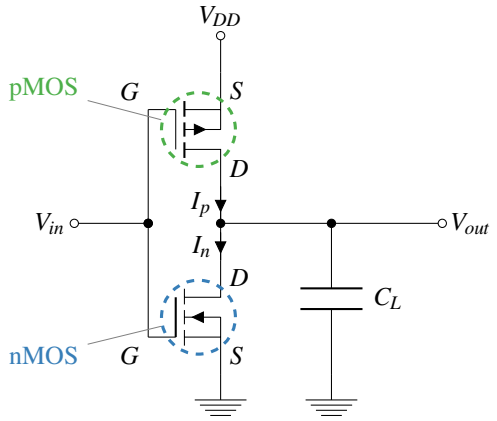


Fig. 6: Transistor level implementation of a CMOS inverter.

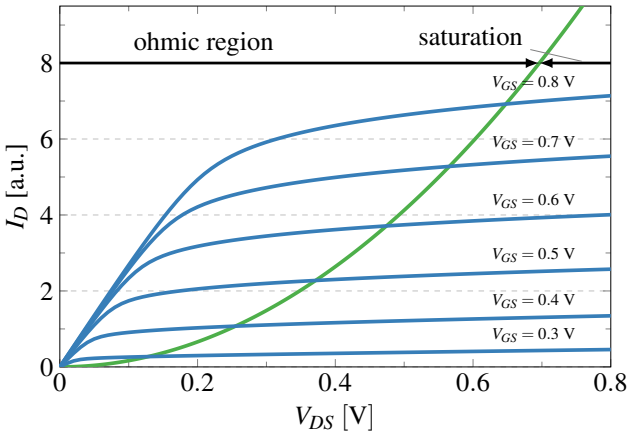


Fig. 7: Current through the 15 nm FinFET nMOS over V_{DS} for different values of V_{GS}. The higher the latter the more current is delivered.

however, can be seen as switches with three[2] terminals (source S, gate G and drain D) that propagate electric current between S and D, whereat the conductivity depends on the voltage balance between the single terminals. nMOS and pMOS mainly differ by the direction the important differential voltages are measured: for nMOS $V_G - V_S$ (V_{GS}), $V_D - V_S$ (V_{DS}) and $V_G - V_D$ (V_{GD}) have to be used, while for pMOS the direction is reversed, i.e., $V_S - V_G$, $V_S - V_D$ and $V_D - V_G$. To increase readability, we solely use the notation for the nMOS in the following. Expressions for the pMOS can, however, simply be achieved by switching the indices.

A transistor shows significantly different output behavior based on its terminal voltages. One can distinguish three operation regions, with rather smooth boundaries in between:

Sub-threshold (ST): When V_{GS} is below the *threshold voltage* ($V_{th,n}$), the transistor is open, i.e., only conducts poorly, for the purpose of this research, even negligibly.

Ohmic region (OHM): As soon as V_{GS} exceeds $V_{th,n}$, the device starts to close, i.e., to conduct, whereat V_{GD} also has an impact on conductivity. In the ohmic region ($V_{GD} \geq V_{th,n}$), the current changes quickly with varying values of V_{DS} and V_{GS}. Fig. 7 shows the drain current (I_D from D to S) over V_{DS}. The single lines represent different values of V_{GS}, with the highest value at the top. For the purpose of this paper, we use a simplistic expression to describe the behavior in this ohmic region[3] according to [12], namely,

$$I_D = S_n \cdot V_{DS}(V_{GS} - V_{th,n} - V_{DS}/2),$$

with the scaling factor S_n, which is in general mainly a function of the transistor width. Note that the current depends *quadratically* on V_{DS} but linearly on V_{GS}.

Saturation (SAT): In the case of $V_{GD} < V_{th,n}$, the transistor is said to be in saturation, i.e., the current only changes moderately with V_{DS}, as can be seen in Fig. 7. For the purpose of this paper, we use an even coarser abstraction [12] by neglecting V_{DS} altogether:

$$I_D = \frac{S_n}{2}(V_{GS} - V_{th,n})^2$$

Even for very elaborate technologies, e.g., 15 nm FinFET, this way of modeling is still a very good fit, as the simulations in Fig. 8 show. The latter were also used to determine the threshold voltages of n- and pMOS ($V_{th,n}$ and $V_{th,p}$).

Note that, in this simple model, I_D stays constant when varying V_{DS}, which seems a very rough approximation, especially when looking at Fig. 7. Furthermore, all the short channel effects that can be observed in modern technologies are not considered at all. We stuck to this simple model, however, as (1) it facilitated analytic calculations and (2) is actually capable of providing reasonably accurate predictions (see Section IV-A) for the quantities we are aiming at. Let us recall at this point that our major goal is to explain the general shape of the delay functions. Whereas more accurate equations would lead to more accurate results, we estimate the differences to be minor. In fact, considering that digital timing simulations are inherently inaccurate, we do not see this as a major concern.

IV. Characterizing Delay Functions

An easy-to-compute closed form of the delay function $\delta(T)$, which maps the current-input-to-previous-output time T to the delay δ, offers several advantages over tabulated numerical results: a) less storage requirements, b) higher accuracy due to no need for interpolating intermediate values, c) analytic circuit/delay composition, and last but not least d) additional insights into physical/electrical processes governing circuit delays. Especially the latter is important when arguing about the applicability for future technologies.

While $\delta(T)$ of *IDM* for real circuits is only available numerically so far, the delay function of *DDM* was successfully fitted to an exponential in [3]. However, to the best of our

[2]We deliberately neglect the bulk here.

[3]There are more elaborate methods available, which are however not necessary to achieve the qualitative results we are aiming for.

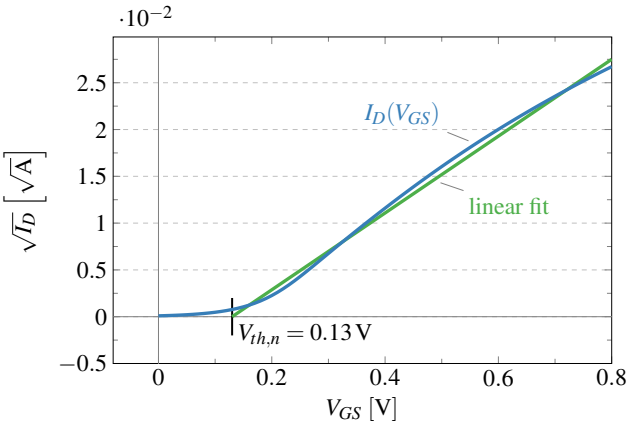


Fig. 8: Squared root of the current through the 15 nm FinFET nMOS over V_{GS} ($V_{DS} = V_{DD}$) with a linear approximation, which is also used to determine the threshold voltage $V_{th,n}$.

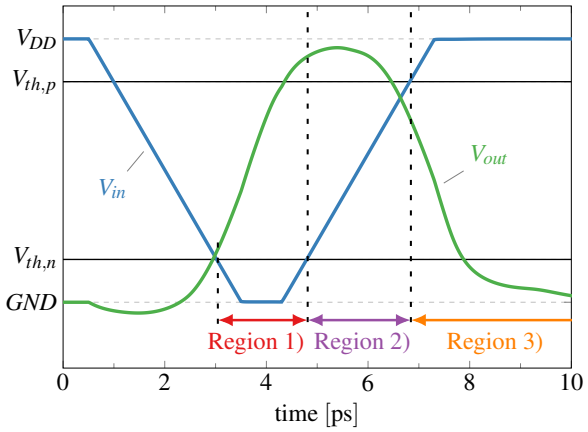


Fig. 9: Overview of 15 nm inverter operation regions during switching. $V_{th,n}$ respectively $V_{th,p}$ represent the threshold voltages for n- and pMOS.

knowledge, the authors did not investigate why this is the case and whether it can be expected to hold also for future technologies. In this section, we will therefore address these questions and explore the delay functions for *DDM* in more detail. In the course of this research, we will be able to answer the question of where and why the exponential fit is reasonable. Moreover, relying on physical considerations, we will also develop appropriate abstractions that eventually lead to closed form analytic results. Finally, we will argue whether and how our approach can be adapted to *IDM*.

To verify our modeling assumptions, we resorted to *SPICE* simulations as a golden reference; ten stage inverter chains were synthesized and parasitics extracted with Cadence, once using a standard 65 nm UMC library ($V_{DD} = 1.2\,\text{V}, V_{th,n} = 0.4\,\text{V}$, $V_{th,p} = 0.73\,\text{V}$) and once for 15 nm FinFET technology using the 15 nm Nangate Open Cell Library with FreePDK15™ FinFET models [13] ($V_{DD} = 0.8\,\text{V}, V_{th,n} = 0.13\,\text{V}$, $V_{th,p} = 0.67\,\text{V}$). We did this to investigate whether our results are actually technology independent, which indeed turned out to be the case. The 65 nm inverter chain was further modified to include large 72 fF load capacitances instead of the relatively small parasitics between each inverter. This allowed to pronounce effects that where otherwise too small and too fast to be observed; demonstrating that our approach works also in the presence of large parasitics. All our calibrations use the input and output signals of the (i) first inverter in the chain, if trapezoidal input signals are required, and the (ii) seventh inverter if shaped input signals are required.

A. General Remarks

For a start, we take a closer look at the linear input shape modeling originally used for *DDM* in [3], which simplifies calculations and analysis considerably. For *IDM*, we will later extend our results to the more general case of realistically shaped inputs. Our model is an optimal inverter (single nMOS and pMOS transistor with a load capacitance and no parasitics as shown in Fig. 6). To get comparable results for *DDM* channels, we used the same settings as described in [4]: linear ramps as input signals, and V_{in} and V_{out} digitized at $V_{DD}/2$. The linear input slope at the first inverter stage is chosen to have about the same rise/fall time as the shaped output signal (see Fig. 4).

Fig. 9 shows the *SPICE* results of an up-pulse at the output, i.e., starting and ending at *GND*. As mentioned earlier, each transistor of the inverter can operate in one of three operation regions. According to [14], for an inverter only seven of the possible nine are reachable, while for our considerations of the up-pulse only the subset shown in Fig. 10 is important. We distinguish three regions in the switching process (see Fig. 9):

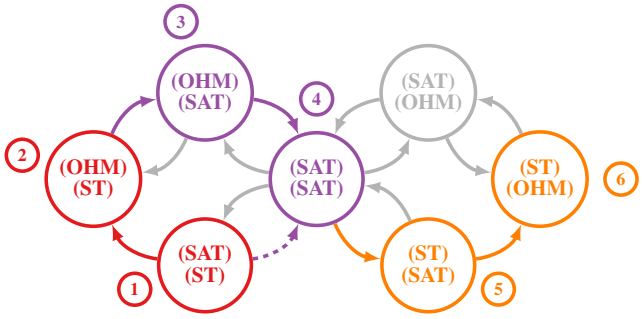


Fig. 10: Transition graph of the transistor operation regions in an inverter (inspired by [14]). The first line in a node shows the pMOS, the second one the nMOS. The colors correspond to Region 1) [red], 2) [purple] and 3) [orange].

Region 1): We start our considerations in state ① of Fig. 10, i.e., when V_{in} drops below $V_{th,n}$ and thus opens the nMOS (non-conducting) while the pMOS is still in (SAT). As V_{out} increases, eventually the pMOS enters (OHM) ②, which reduces the current and thus the speed by which V_{out} increases. Only after V_{in} has exceeded the threshold $V_{th,n}$ of the nMOS in its rising transition, the latter starts to conduct again, causing a transition to ③. Note that quick input changes make it possible to transition from ① directly to ④.

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

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

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

B. Region 2), state (SAT)–(SAT)

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

According to the transistor-level implementation of the inverter (see Fig. 6), the derivative of the output is proportional to the difference between the current flowing through n- and pMOS, i.e.,

$$\frac{\mathrm{d}V_{out}}{\mathrm{d}t} = C_L \cdot I_{out} = C_L \cdot (I_p - I_n).$$

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

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

with some integration constant C_0 and

$$C_3 = S_p - S_n, \qquad C_2 = -\frac{3}{k}(A \cdot S_p + B \cdot S_n)$$
$$A = V_{DD} - V_{th,p} - V_{th}, \qquad B = V_{th} - V_{th,n}$$
$$I = \frac{A^3 \cdot S_p}{3 \cdot k} + \frac{B^3 \cdot S_n}{3 \cdot k}.$$

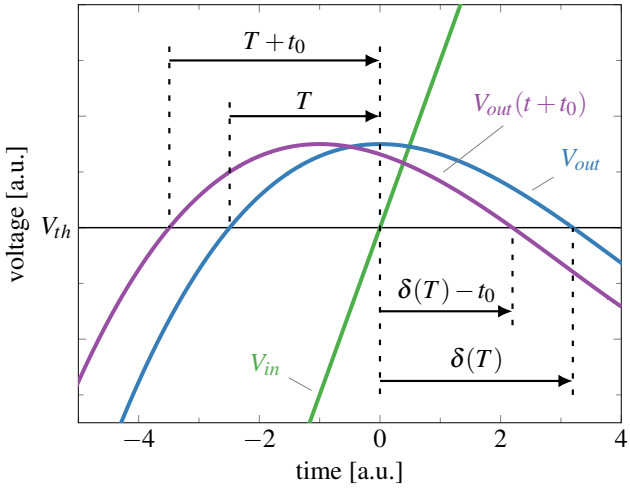


Fig. 11: Cubic approximation of V_{out}. The time shift t_0 causes an increase in T and a decrease in δ. The slope of the input signal V_{in} is approximated.

For the up-pulse we get a negative quadratic part, the peak value at $t = 0$ is primarily determined by the integration constant C_0. Fig. 11 shows an example trace, where the cubic part is clearly visible. Note that an arbitrary slope was chosen for V_{in} in this figure, as k is hard to derive from the cubic function used for drawing the output curve.

To calculate $\delta(T)$, we could vary C_0, i.e., the peak value, and observe the appropriate V_{th} crossing times. This tedious process can however be simplified significantly by analytically determining at which points in time the function given in (2) has the same value. Out of the three solutions, we are only interested in the ones closest to 0 on the negative ($-T$) and positive (δ) side. To get an explicit form, i.e., an expression for $\delta(T)$, however, we have to derive δ as a function of T. Note that the specific values are actually of no concern for this analysis; just knowing the shape is sufficient.

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

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

i.e., the delay function is a ramp with slope 1. Since it is almost impossible that both transistors are absolutely identical, however, we are more interested in the cases where $S_n \neq S_p$. Recall that we are looking for an explicit formula, so we need to find an expression that determines δ based on the knowledge of $T > 0$. As already mentioned, we need a positive value δ with $V_{out}(\delta) = V_{out}(-T)$ for this purpose, i.e., by using Equation (2) we need to solve

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

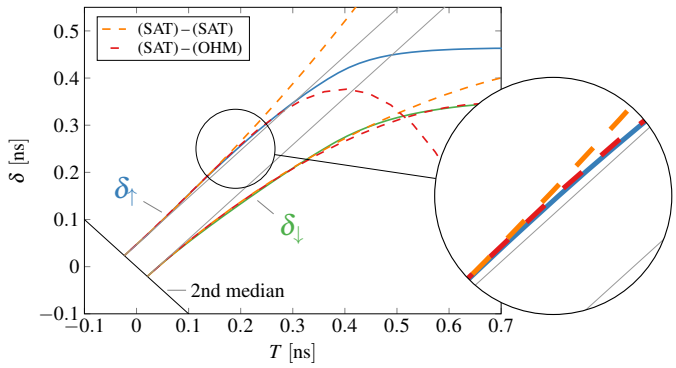


Fig. 12: *DDM* delay function of the first inverter in the 65 nm inverter chain (solid lines) vs. predictions based on our simplifications (dashed lines). Clearly visible is the super-linear growth of $\delta_\uparrow$ for $T < 0.2\,\mathrm{ns}$.

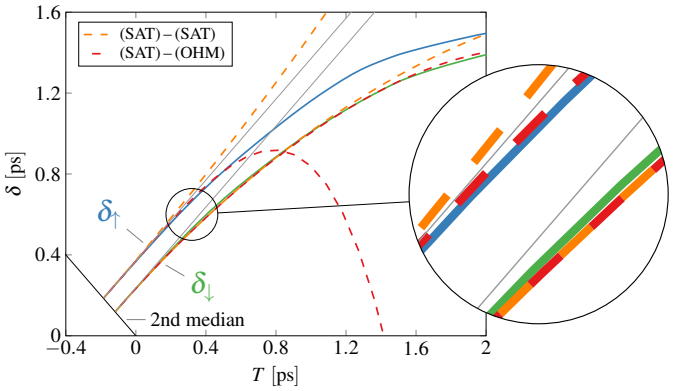


Fig. 13: *DDM* delay function of the first inverter in the 15 nm inverter chain (solid lines) vs. predictions based on our simplifications (dashed lines).

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

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

One of those solutions is the desired result, provided the constants C_2, C_3 and T do not cause the argument of the square root to become negative: Depending on the sign of C_2, the negative branch ($C_2 < 0$) or the positive branch ($C_2 \geq 0$) must be used. Comparing this estimation to delay functions simulated in *SPICE* (see Fig. 12) we observe good agreement for small values of T.[4] Note carefully that both delay functions initially have a slope of 1 (cp. the gray lines). Whereas the derivative of $\delta_\downarrow$ continuously decreases from there onward, the one of $\delta_\uparrow$ rises initially. This is in stark contrast to *DDM*, which demands sub-linear growth at all times. The 15 nm technology shown in Fig. 13 appears better balanced, as no super-linear growth can be observed, implying a small cubic part.

C. Region 2), state (SAT)−(OHM)

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

What actually happens when the pMOS operates in (OHM) is that it delivers less current than before. This implies that the peak of the output pulse shifts to a lower value of V_{in} as the nMOS has to close less to reach the current equilibrium $I_n = I_p$, which is the key property of the peak value ($I_{out} = V'_{out} = 0$). With respect to our cubic fitting of V_{out}, this means that the peak is now at some time $t < 0$ instead of $t = 0$. We can approximate this behavior by artificially shifting the whole pulse to the left. As a consequence, the time T between the first output V_{th} crossing to the input V_{th} crossing increases, while δ decreases by the same amount, see Fig. 11. Note that this decreases the derivative of the resulting $\delta(T)$, and also guarantees a continuous transition between the low T situation of Section IV-B and the higher T situation analyzed later.

However, we still need to answer the question by how much the peak shall be shifted: Since I_D changes in (OHM) only linearly with V_{in} but quadratically with V_{out}, we chose to carry out a time shift that depends quadratically on the peak value V_p. The resulting changes to our process of determining a closed-form expression for $\delta(T)$ are straightforward: We just reduce T by $k \cdot V_p^2$ and increase δ by the same amount. The peak value can be computed from (2) as $V_p = V_{out}(0) - V_{out}(-T) = -C_3 \cdot T^3 + C_2 \cdot T^2$. Actually, the latter is only an approximation, as we would have to replace T by $T - k \cdot V_p^2$ here to get the correct result. As this would unnecessarily complicate the expression for V_p, however, we omit this improvement. And indeed, the predictions obtained with this approximation fits actual delay simulations, see Figures 12 and 13. Qualitatively, the results look similar for both technologies, where a strong curvature in the approximation for $\delta_\uparrow$ can be observed. This forces us to investigate the region for big T separately.

D. Regions 1) & 3)

If the output pulse, and hence T, grows further, V_{out} is well above V_{th} when the rising input exceeds $V_{th,n}$ of the (open) nMOS, i.e., the inverter is in Region 1) here. When the rising input eventually also exceeds $V_{th,p}$, the inverter is in Region 3). This actually allows us to make radical reductions and thus simplifications. First of all, we assume that the part of the trajectory that lies in Region 2) is fixed, meaning that its

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

shape and thus the contribution to T (T_2) and δ (δ_2) is constant (cp. Fig. 9). This is reasonable, as we assume a linear input signal which will therefore be the same for all pulses.[5] This also implies that the voltage gained in Region 1) has to be completely compensated in Region 3), which simplifies our calculations even further.

After the V_{th} crossing of the rising output transition, the pMOS operates in (OHM). We can hence represent it as a simple resistor, leaving the overall inverter in Region 1) as shown in Figure 14. Consequently, the capacitance C_L will be charged according to an exponential function, with a time constant $\tau = R \cdot C_L$. During most of the falling output transition, the nMOS is in (SAT), which causes the current in Region 3) to only change moderately with V_{out} (see Fig. 7). We repeat our assumptions of constant current in (SAT) here and replace the transistor by a constant current source, as shown in Fig. 15. Fig. 16 shows these simplifications as fittings to a simulated *SPICE* trace. In Region 1) & 3) (outside dashed lines) very good agreement can be observed.

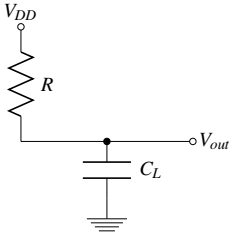


Fig. 14: Inverter in Region 1).

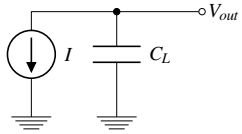


Fig. 15: Inverter Region 3).

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

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

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

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

Similarly, the delay function for the rising output transition reads

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

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

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

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

It can be seen clearly that the overall shape of the delay function for large T is determined by the RC constant of the transistor active in the first part, i.e., $\tau = R \cdot C_L$ from Fig. 14. To determine R, one has to investigate the slope of I_D shown in Fig. 7 for $V_{GS} = V_{DD}$ and low values of V_{DS}. As there are different fittings possible, finding an appropriate value might be a challenging task.

Figure 17a shows the resulting delay functions in logarithmic space, e.g. $\log\big(1 - \delta_\uparrow(T)/\delta_\uparrow(\infty)\big)$. For large values of T, we get a linear dependency, i.e., an exponential behavior. In this region, the *DDM* delay function given in [4] is indeed correct. Unlike the cubic fitting established in the previous subsections, it can, however, not explain the significant curvature for T towards 0. Simulations on the 15 nm technology show a quite different picture (see Fig. 17b), as no curvature is observable there. Instead the complete delay function may be fitted using a single exponential more or less accurately.

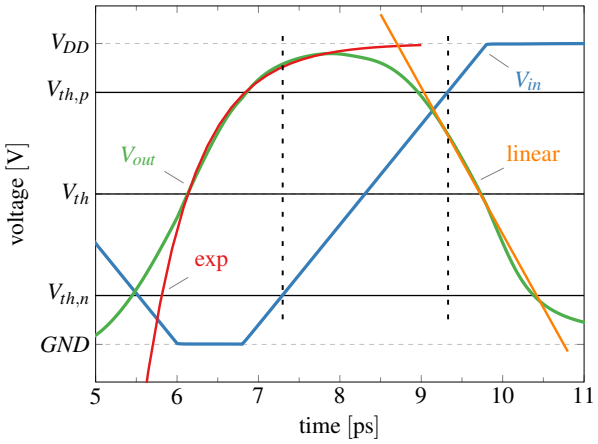
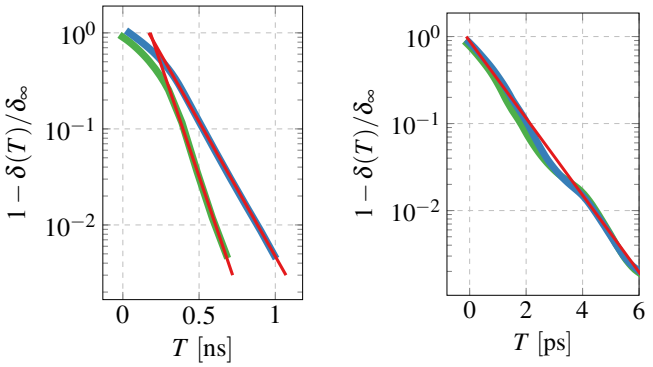


Fig. 16: Simplification of V_{out} for Region 1) & 3) for 15 nm technology. The exponential increase is followed by a linear drop.

E. Extension to IDM

As pointed out earlier, the main difference when switching to *IDM* are the shaped input signals used for characterization, instead of the linear ones used by *DDM*. We simulated this by picking the seventh inverter in our ten inverter chain. This "minor" change increases not only the overall complexity, as the changing input derivative induces varying currents via

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



(a) 65 nm (b) 15 nm

Fig. 17: *DDM* delay function of the first inverter for different technologies in logarithmic space with linear fitting.

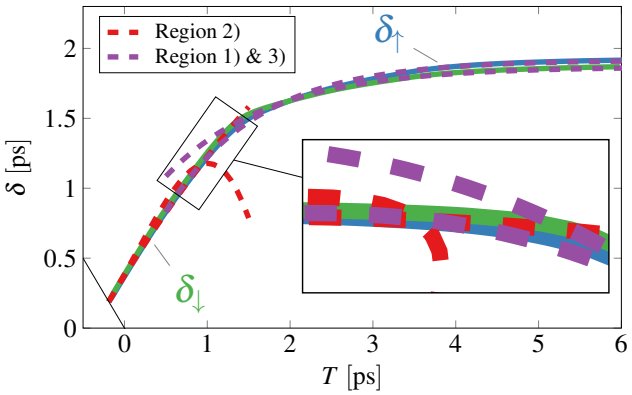


Fig. 18: Fitting of the involution delay function for the seventh inverter in the 15 nm chain.

coupling capacitances, but also has an impact on the shape of the delay function. In general, an increased bending of the delay function can be observed. Nevertheless, our assumptions still seem valid as the projected trace is very close to the simulated one (see Fig. 18). Solely in the transition region, where we have to switch between the different approximations, the accuracy slightly decreases.

F. Summary

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

Unfortunately, the transition to *IDM* turned out to be more challenging than expected. Albeit we observe quite good agreement also here, the inaccuracies during the transitions between the different regions is bigger.

Despite using the very simple models described in Section III, which neglect a lot of phenomena present in modern technology, the fitting to accurate *SPICE* simulations is generally very good. A comparison between the predictions of our models and the real delays observed in different technologies allow us to conjecture that we have identified a set of equations that is sufficiently parametrizable to properly model the delay functions both for *DDM* and *IDM*. Our research also revealed that the exponential fitting of the DDM can also be justified based on physical consideration, albeit only for a one specific operation region; for the other regions, it does not describe the real behavior well. A key question left to future research is how to determine the boundaries between the different regions and of course the parameters for different technologies.

V. Conclusion and Future Work

In this article, we studied delay functions arising in dynamic delay models, namely, *DDM* and *IDM*, with a special focus on their parameterization. Using the characterization method proposed for *DDM*, i.e., linear inputs and common threshold voltages, in conjunction with transistor-level analysis, we derived analytic expressions for the delay function $\delta(T)$ for three main operating regions. By comparing predicted and simulated delays for inverters in both 65 nm and 15 nm technology, we can claim that our results are reasonably robust w.r.t. technology. Moreover, in sharp contrast to *DDM*, our characterization does not involve weight parameters that need to be determined by time-consuming simulation runs. Our analyses show that the exponential fitting used in *DDM* is only appropriate for larger values of T, whereas there is a significantly different behavior for lower values.

Although our results are very encouraging, there is still a long way to go towards a practical delay characterization method. In particular, we are still experiencing inaccuracies when switching between the different regions. Part of our current and future work is hence devoted to determine and better characterize these borders. Furthermore we are looking for methods that can be used to efficiently determine the particular parameterization for a given circuit technology. Ideally, costly *SPICE* simulations should be replaced by deriving the delay function parameters from available transistor parameters.

References

[1] *Effective Current Source Model (ECSM) Timing and Power Specification*, Cadence Design Systems, January 2015, version 2.1.2.

[2] *CCS Timing Library Characterization Guidelines*, Synopsis Inc., October 2016, version 3.4.

[3] 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," *IEE Proceedings – Circuits, Devices, and Systems*, vol. 147, no. 2, pp. 107–117, 2000.

[4] M. J. Bellido, J. Juan, and M. Valencia, *Logic-Timing Simulation and the Degradation Delay Model*. Imperial College, 2005. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/p411

[5] M. Függer, T. Nowak, and U. Schmid, "Unfaithful glitch propagation in existing binary circuit models," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 964–978, March 2016.

[6] M. Függer, T. Nowak, and U. Schmid, "Unfaithful glitch propagation in existing binary circuit models," in *Proceedings 19th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'13)*. IEEE Computer Society, 2013, pp. 191–199.

[7] M. Függer, R. Najvirt, T. Nowak, and U. Schmid, "Towards binary circuit models that faithfully capture physical solvability," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 1455–1460.

[8] A. J. Winstanley, A. Garivier, and M. R. Greenstreet, "An Event Spacing Experiment," in *Proceedings of the 8th International Symposium on Asynchronous Circuits and Systems (ASYNC)*, April 2002, pp. 47–56.

[9] J.-C. B. Acosta, J. Juan-chico, M. J. Bellido, A. J. Acosta, A. B. riga, M. Valencia, S. Centro, and N. Microelectrnica, "Delay degradation effect in submicronic CMOS inverters," in *Proc. FODO'93*. Springer, 1997, pp. 215–224.

[10] J. Juan-Chico, M. J. Bellido, P. Ruiz-de Clavijo, A. J. Acosta, and M. Valencia, "Degradation delay model extension to CMOS gates," in *Integrated Circuit Design*, ser. LNCS 1918. Springer, 2000, pp. 149–158.

[11] A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de Clavijo, and D. Guerrero, "Characterization of normal propagation delay for delay degradation model (DDM)," in *Integrated Circuit Design*, ser. LNCS 2451. Springer, 2002, pp. 477–486.

[12] S. Sze and K. K. Ng, *Physics of Semiconductor Devices*, 3rd ed. John Wiley & Sons, Inc., 2007.

[13] M. Martins, J. M. Matos, R. P. Ribas, A. Reis, G. Schlinker, L. Rech, and J. Michelsen, "Open cell library in 15nm freepdk technology," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, ser. ISPD '15. New York, NY, USA: ACM, 2015, pp. 171–178. [Online]. Available: http://doi.acm.org/10.1145/2717764.2717783

[14] J. Maier, "Modeling the cmos inverter using hybrid systems," E182 - Institut für Technische Informatik; Technische Universität Wien, Tech. Rep. TUW-259633, 2017. [Online]. Available: http://publik.tuwien.ac.at/files/publik_259633.pdf