Extracting Exact Time Bounds from Logical Proofs

Mauro Ferrari, Camillo Fiorentini, and Mario Ornaghi

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
{ferram,fiorenti,ornaghi}@dsi.unimi.it

Abstract. Accurate evaluation of delays of combinatorial circuits is crucial in circuit verification and design. In this paper we present a logical approach to timing analysis which allows us to compute exact stabilization bounds while proving the correctness of the boolean behavior.

1 Introduction

Accurate evaluation of delay times of combinatorial logic components is crucial in circuit verification and design [4,8]. For example, it is fundamental in determining the clock rate. Thus, various *timing analysis* methods have been developed in the literature for detecting different kinds of delays, as worst case or minimum delays, as well as for evaluating exact data-dependent delays (see [4, 8] for a discussion). In traditional approaches, timing analysis does not take into account data dependencies. In this way *false paths*, i.e., paths that cannot be activated by any input, may be detected and false path elimination is necessary to ensure accurate timing analysis. Moreover, many approaches model worst case and minimum gate delays as constants, while such delays may depend on the rising and falling times of the input signals and more accurate *gate models* are needed [4].

Among the timing analysis methods that have been developed in the literature, here we consider the logically based ones, see, e.g., [1,2,5,7,8]. Their logical nature automatically excludes false paths, since they are based on sound logical semantics. A semantics represents an abstraction from the physical details, and takes into account only aspects that are relevant for a given kind of analysis. For example, if we are only interested in the functional analysis of a combinatorial logic network, models based on Classical Logic are sufficient, while three valued logic allows us to take into account unstable or unknown signals [2]. Models based on Intuitionistic Logic and on Lax Logic have been proposed in [7,8]; they support in a uniform way both functionality analysis and input-dependent timing analysis with accurate gate models for combinatorial circuits.

In this paper we consider a modification of the approaches based on Intuitionistic and Lax logic. The main differences are in the adopted propositional language and in the formalization of time bounds. We consider optimal (exact) time bound evaluation for any kind of formulas, while the aforementioned approaches consider optimal bounds for restricted classes of formulas. Moreover,

A. Pettorossi (Ed.): LOPSTR 2001, LNCS 2372, pp. 245–265, 2002.

for suitable restrictions on waveforms, we have a completeness result for the logic $\mathbf{F}_{\mathrm{Cl}}$, which is a maximal axiomatizable and decidable intermediate logic [10].

In Section 2 we introduce and briefly discuss waveforms, and we define our propositional language and its waveform interpretation. In Section 3 we introduce time bounds and a constructive semantics based on time bounds. A calculus $\mathcal{ND}$ that is valid with respect to such a semantics is presented in Section 4. In Subsection 4.1 we prove the main result to extract a function calculating exact delays from proofs of $\mathcal{ND}$ and in Subsection 4.2 we apply it to an example; we also show that proofs of Classical Logics are inadequate to accomplish such an analysis. Finally, in Section 5 we briefly discuss the logical aspects of the proposed semantics.

2 Waveforms and Circuits

In the logical approach to circuit analysis a semantics represents an abstraction from the physical details and takes into account aspects that are relevant for a given kind of analysis, disregarding other aspects. To give an example, let us consider the gates INV and NAND of Figure 1; their behavior is specified by the following formulas of Classical Logic

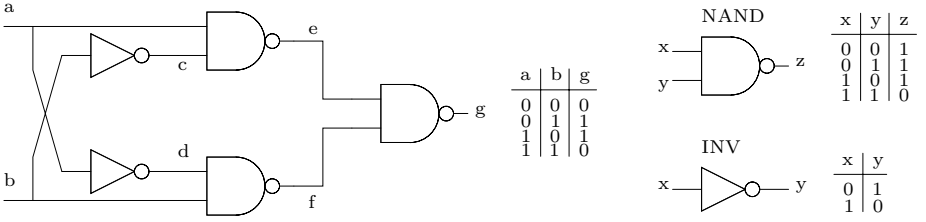
$$\mathrm{INV}(x, y) \equiv (x \to \neg y) \wedge (\neg x \to y) \tag{1}$$

$$\mathrm{NAND}(x, y, z) \equiv (x \wedge y \to \neg z) \wedge (\neg x \vee \neg y \to z) \tag{2}$$

Indeed, the truth table of $\mathrm{INV}(x, y)$ represents the input/output behavior of the INV gate assuming x as input and y as output. Analogously, NAND(x,y,z) represents the NAND gate, where x and y are the inputs and z is the output. Similarly, the classical behavior of the XOR circuit is specified by the formula

$$\mathrm{XOR}(x, y, z) \equiv ((x \wedge \neg y) \vee (\neg x \wedge y) \to z) \wedge ((x \wedge y) \vee (\neg x \wedge \neg y) \to \neg z) \tag{3}$$

where x and y represent the inputs and z the output.



a	b	g
0	0	0
0	1	1
1	0	1
1	1	0

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

x	y
0	1
1	0

Fig. 1. The XOR circuit and its components

Classical semantics allows us to study the input/output behavior of combinatorial circuits, but does not allow us to represent temporal information about the stabilization properties of the circuits. Indeed, a more realistic description of the XOR circuit of Figure 1 should consider the instant at which the signals become stable and the delays in the propagation of signals; e.g., an "informal" characterization of the behavior of the above circuit should be as follows:

$$\boxed{\begin{array}{c}(a \text{ stable to 1 at } t_1) \text{ and } (b \text{ stable to 0 at } t_2)\\ \textbf{or}\\ (a \text{ stable to 0 at } t_1) \text{ and } (b \text{ stable to 1 at } t_2)\end{array}} \Rightarrow (g \text{ stable to 1 at } F(t_1, t_2))$$

$$\boxed{\begin{array}{c}(a \text{ stable to 1 at } t_1) \text{ and } (b \text{ stable to 1 at } t_2)\\ \textbf{or}\\ (a \text{ stable to 0 at } t_1) \text{ and } (b \text{ stable to 0 at } t_2)\end{array}} \Rightarrow (g \text{ stable to 0 at } G(t_1, t_2))$$

where F and G are functions from $\mathbf{N}^2$ in $\mathbf{N}$ and $\mathbf{N}$ represents discrete time.

To formalize this, we need to introduce some notions. As in [8,9], a *signal* is a discrete timed boolean function $\sigma \in \mathbf{N} \to \mathbf{B}$. A *circuit* is characterized by a set $\mathbf{S}$ of *observables* (the atomic formulas of our language) and a *waveform* is a map $V \in \mathbf{S} \to (\mathbf{N} \to \mathbf{B})$ associating with every observable a signal. A waveform represents an observable property of a circuit C, and an observable *behavior* of C is described by a set of waveforms.

As an example, to represent the XOR circuit of Figure 1, we need the set of observables $\{a, b, c, d, e, f, g\}$ representing the connections between the gates of the circuit. A waveform represents a possible behavior of the circuit. For example Figure 2 describes a waveform for the NAND circuit. It puts in evidence that the output z rises to 1 at time t_2 with a certain delay δ_1 with respect to the time t_1 where the input x falls to 0. On the other hand, the output z falls to 0, and stabilizes to 0, at time t_6, with a certain delay δ_2 with respect to the time t_5 where both the inputs are stable to 1. We remark that $\delta_1 \neq \delta_2$; indeed, in a realistic description the delays are input dependent.

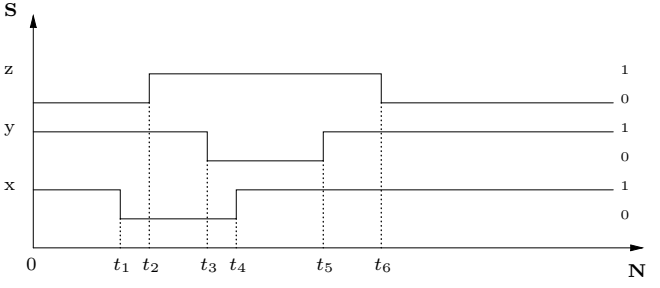


Fig. 2. A waveform for NAND

Since we are interested in studying stabilization properties of a circuit, we consider only waveforms that stabilize at some time. In particular, we introduce the following notions of stabilization for a waveform:

1. V is *stable* iff, for every $a \in \mathbf{S}$ and for every $t \in \mathbf{N}$, $V(a)(t) = V(a)(0)$;
2. V is *eventually stable* iff, for every $a \in \mathbf{S}$, there exists $t \in \mathbf{N}$ such that, for every $k \geq t$, $V(a)(k) = V(a)(t)$.

We denote with STABLE the set of all the stable waveforms and with ESTABLE the set of all the eventually stable waveforms.

To express *stabilization properties* of waveforms and behaviors, we use a propositional language $\mathcal{L}_{\mathbf{S}}$ based on a denumerable set of *observables* $\mathbf{S} = \{a, b, c_1, c_2, \dots\}$. Formulas of $\mathcal{L}_{\mathbf{S}}$ are inductively defined as follows: for every $a \in \mathbf{S}$, a is an *atomic* formula of $\mathcal{L}_{\mathbf{S}}$; if $A, B \in \mathcal{L}_{\mathbf{S}}$, then $A \wedge B$, $A \vee B$, $A \rightarrow B$, $\neg A$ and $\Box A$ belong to $\mathcal{L}_{\mathbf{S}}$.

We say that a waveform V *validates a stabilization property* A *at time* t, written $t, V \Vdash A$, if one of the following conditions holds:

$t, V \Vdash a$, where $a \in \mathbf{S}$, iff $V(a)(k) = 1$ for all $k \geq t$;
$t, V \Vdash \Box B$ iff $k, V \Vdash B$ for some $k \geq t$;
$t, V \Vdash B \wedge C$ iff $t, V \Vdash B$ and $t, V \Vdash C$;
$t, V \Vdash B \vee C$ iff either $t, V \Vdash B$ or $t, V \Vdash C$;
$t, V \Vdash B \rightarrow C$ iff, for every $k \in \mathbf{N}$, $k, V \Vdash B$ implies $l, V \Vdash C$ for some $l \geq k$;
$t, V \Vdash \neg a$, where $a \in \mathbf{S}$, iff $V(a)(k) = 0$ for all $k \geq t$;
$t, V \Vdash \neg\Box B$ iff $k, V \Vdash \neg B$ for some $k \geq t$;
$t, V \Vdash \neg(B \wedge C)$ iff either $t, V \Vdash \neg B$ or $t, V \Vdash \neg C$;
$t, V \Vdash \neg(B \vee C)$ iff $t, V \Vdash \neg B$ and $t, V \Vdash \neg C$;
$t, V \Vdash \neg(B \rightarrow C)$ iff $t, V \Vdash B$ and $t, V \Vdash \neg C$;
$t, V \Vdash \neg\neg B$ iff $t, V \Vdash B$.

It is easy to check that $t, V \Vdash A$ implies $h, V \Vdash A$, for all $h \geq t$. For a atomic, a and $\neg a$ denote the stability of the observable signal $V(a)$ (at time t, with value 1 and 0 respectively). Indeed $t, V \Vdash a$ ($t, V \Vdash \neg a$) iff the signal $V(a)$ is stable to 1 (to 0 respectively) from t on.

Implication underlies a propagation delay, i.e., $t, V \Vdash A \rightarrow B$ means that, whenever, at some t', A "stabilizes" ($t', V \Vdash A$) then, after a certain amount of time s, B will "stabilize" ($t' + s, V \Vdash B$). E.g., if V is the waveform of Figure 2, we have $0, V \Vdash (x \wedge y) \rightarrow \neg z$; indeed $t, V \Vdash x \wedge y$ iff $t \geq t_5$ and $t \Vdash \neg z$ for all $t \geq t_6$, hence the stabilization delay is at most $t_6 - t_5$. We also remark that, differently from the other connectives, the validity of an implication is independent of t, indeed, $t, V \Vdash (B \rightarrow C)$ iff $0, V \Vdash (B \rightarrow C)$. Intuitively this corresponds to the fact that an implication does not represent a *property observable at a given time*, but a *global property* expressing a behavior invariant with respect to time shift. This is what has to be expected to express delay properties.

The unary modal operator $\Box$ means future stabilization; e.g., for the waveform of Figure 2, $0, V \Vdash \Box x$ since there is a moment in the future where x stabilizes to 1, but $0, V \nVdash x$. Validation of $\wedge, \vee$ is defined as expected. As for the negation, the above semantics defines it as a constructive negation and the validity of $\neg A$ is defined recursively on the structure of A (for a discussion on this negation see [11,13]). We remark that such a constructive understanding of the negation is essential in our approach where $\neg a$ states the *positive* information "a stabilizes to 0" and is different from the usual intuitionistic understanding of negation as "a implies falsehood".

A logical characterization of stable and eventually stable waveforms is the following:

$$V \in \textsc{Stable} \text{ iff } 0, V \Vdash A \vee \neg A \text{ for every } A \in \mathcal{L}_{\mathbf{S}}$$
$$V \in \textsc{EStable} \text{ iff } 0, V \Vdash \Box A \vee \neg\Box A \text{ for every } A \in \mathcal{L}_{\mathbf{S}}$$

Now, to represent the classical input/output behavior of a boolean function in our semantics, we associate with an eventually stable waveform V the classical interpretation V^{CL} done as follows: for every $a \in \mathbf{S}$,

$$V^{\mathrm{CL}}(a) = \begin{cases} 0 & \text{if } 0, V \Vdash \neg\Box a \\ 1 & \text{if } 0, V \Vdash \Box a \end{cases}$$

Definition 1. *Given a boolean function $f : \mathbf{B}^n \to \mathbf{B}$, a formula $F(a_1, \ldots, a_n, b)$ of $\mathcal{L}_{\mathbf{S}}$ represents f iff, for every $V \in$* ESTABLE*,*

$$0, V \Vdash \Box F(a_1, \ldots, a_n, b) \quad \textit{iff} \quad f(V^{\mathrm{CL}}(a_1), \ldots, V^{\mathrm{CL}}(a_n)) = V^{\mathrm{CL}}(b)$$

We remark that the above definition does not work if $0, V \Vdash \Box A \vee \neg\Box A$ does not hold, that is if V is not eventually stable. Indeed, in our approach we do not treat the case of oscillating signals; to treat such signals a different semantics (e.g., a multi-valued semantics) should be considered.

The *formal verification task* of the circuit of Figure 1 consists in exhibiting a formal proof (in the adequate logic) of the formula

$$\mathrm{INV}(b, c) \wedge \mathrm{INV}(a, d) \wedge \mathrm{NAND}(a, c, e) \wedge \mathrm{NAND}(b, d, f) \wedge \mathrm{NAND}(e, f, g) \to \mathrm{XOR}(a, b, g)$$

If our aim is only to prove the correctness of the above circuit Classical Logic is sufficient. But if we aim to extract information about the stabilization delays of the circuit from the correctness proof, we need to introduce an intensional semantics of formulas that takes into account temporal information.

3 Stabilization Bounds

The validation $\Vdash$ provides an interpretation of formulas as stabilization properties, but the information about stabilization delays is not explicit. To extract stabilization delays we need an analysis of all the waveforms of a behavior. To deal with delays in our logic, we use the notion of *stabilization bound* introduced in [7] and inspired by the *evaluation forms* of [11]. Evaluation forms correspond to structural truth evaluations of formulas; stabilization bounds combine both truth and timing analysis.

In [8] the information about delays is linked to the operator $\Box$ (indicated by $\bigcirc$ in [8]). In contrast, we interpret $\Box$ as a "don't care" operator and we do not associate any time information with it (the only possible delay is 0), but we associate a stabilization information with atomic formulas and with every logical connective not in the scope of the $\Box$ operator. A stabilization bound for a and $\neg a$, with $a \in \mathbf{S}$, fixes an upper bound for the stabilization of the signal $V(a)$. The stabilization bounds for $\wedge, \to, \vee$ are defined as in [8], while the interpretation of $\neg$ and $\Box$ is peculiar of our approach.

Formally, we assign to every formula A of $\mathcal{L}_{\mathbf{S}}$ a set of *stabilization bounds* $\lceil A \rceil$ and an equivalence relation $\sim_A$ between elements of $\lceil A \rceil$, inductively on the structure of A:

- If $A = a$ or $A = \neg a$, with $a \in \mathbf{S}$, then $\lceil A \rceil = \mathbf{N}$, and $t \sim_A t'$ for every $t, t' \in \lceil A \rceil$.
- If $A = \Box B$ or $A = \neg\Box B$ then $\lceil A \rceil = \{0\}$, and $0 \sim_A 0$.
- $\lceil B \wedge C \rceil = \lceil B \rceil \times \lceil C \rceil$ and $(\beta, \gamma) \sim_{B \wedge C} (\beta', \gamma')$ iff $\beta \sim_B \beta'$ and $\gamma \sim_C \gamma'$.
- $\lceil A_1 \vee A_2 \rceil = \lceil A_1 \rceil \oplus \lceil A_2 \rceil$ (where $\oplus$ denotes the disjoint sum, that is the set of pairs $(1, \alpha)$ or $(2, \alpha')$ with $\alpha \in \lceil A_1 \rceil$ and $\alpha' \in \lceil A_2 \rceil$) and $(i, \alpha) \sim_{A_1 \vee A_2} (j, \alpha')$ iff $i = j$ and $\alpha \sim_{A_i} \alpha'$.
- $\lceil B \to C \rceil = \{f \mid f : \lceil B \rceil \to \lceil C \rceil$ s.t. $\beta \sim_B \beta'$ implies $f(\beta) \sim_C f(\beta')\}$, and $f \sim_{B \to C} f'$ iff $f(\beta) \sim_C f'(\beta)$ for every $\beta \in \lceil B \rceil$.
- $\lceil \neg(A_1 \wedge A_2) \rceil = \lceil \neg A_1 \rceil \oplus \lceil \neg A_2 \rceil$ and $(i, \alpha) \sim_{\neg(A_1 \wedge A_2)} (j, \alpha')$ iff $i = j$ and $\alpha \sim_{\neg A_i} \alpha'$.
- $\lceil \neg(B \vee C) \rceil = \lceil \neg B \rceil \times \lceil \neg C \rceil$ and $(\beta, \gamma) \sim_{\neg(B \vee C)} (\beta', \gamma')$ iff $\beta \sim_{\neg B} \beta'$ and $\gamma \sim_{\neg C} \gamma'$.
- $\lceil \neg(B \to C) \rceil = \lceil B \rceil \times \lceil \neg C \rceil$ and $(\beta, \gamma) \sim_{\neg(B \to C)} (\beta', \gamma')$ iff $\beta \sim_B \beta'$ and $\gamma \sim_{\neg C} \gamma'$.
- $\lceil \neg\neg B \rceil = \lceil B \rceil$ and $\beta \sim_{\neg\neg B} \beta'$ iff $\beta \sim_B \beta'$.

The equivalence relation $\sim_A$ is needed to cut undesired functions in the definition of $\lceil B \to C \rceil$. Intuitively, a stabilization bound $\alpha \in \lceil A \rceil$ intensionally represents a set of waveforms V that validate A for the "same reasons" and with the "same delay bounds". Formally, let us denote with V^t the waveform obtained by shifting V of t, i.e.,

$$V^t(a)(k) = V(a)(t + k) \text{ for all } a \in \mathbf{S}, k \in \mathbf{N}$$

A waveform V *validates* A *with stabilization bound* α, and we write $\alpha, V \models A$, if one of the following conditions holds.

- $t, V \models a$, with $a \in \mathbf{S}$, iff $t, V \Vdash a$;
- $t, V \models \neg a$, with $a \in \mathbf{S}$, iff $t, V \Vdash \neg a$;
- $0, V \models \Box B$ iff $t, V \Vdash B$ for some $t \in \mathbf{N}$;
- $(\beta, \gamma), V \models B \wedge C$ iff $\beta, V \models B$ and $\gamma, V \models C$;
- $(i, \alpha), V \models A_1 \vee A_2$ iff $\alpha, V \models A_i$, where $i \in \{1, 2\}$;
- $f, V \models B \to C$ iff, for every $t \in \mathbf{N}$ and $\beta \in \lceil B \rceil$, $\beta, V^t \models B$ implies $f(\beta), V^t \models C$;
- $0, V \models \neg\Box B$ iff $t, V \Vdash \neg B$ for some $t \in \mathbf{N}$;
- $(i, \alpha), V \models \neg(A_1 \wedge A_2)$ iff $\alpha, V \models \neg A_i$, where $i \in \{1, 2\}$;
- $(\beta, \gamma), V \models \neg(B \vee C)$ iff $\beta, V \models \neg B$ and $\gamma, V \models \neg C$;
- $(\beta, \gamma), V \models \neg(B \to C)$ iff $\beta, V \models B$ and $\gamma, V \models \neg C$;
- $\beta, V \models \neg\neg B$ iff $\beta, V \models B$.

To give an example, the INV and NAND gates of the previous section have the following sets of stabilization bounds:

$$\lceil \mathrm{INV}(x, y) \rceil = (\mathbf{N} \to \mathbf{N}) \times (\mathbf{N} \to \mathbf{N})$$
$$\lceil \mathrm{NAND}(x, y, z) \rceil = (\mathbf{N} \times \mathbf{N} \to \mathbf{N}) \times (\mathbf{N} \oplus \mathbf{N} \to \mathbf{N})$$

A stabilization bound for $\mathrm{INV}(x, y)$ is, for example, the pair of identical functions $(f_{\mathrm{INV}}, f_{\mathrm{INV}})$ where

$$f_{\mathrm{INV}}(t) = t + \delta_I \tag{4}$$

$(f_{\mathrm{INV}}, f_{\mathrm{INV}})$ provides an example of a data-independent stabilization bound for the INV gate. It represents the set of valuations V such that $V(y)$ stabilizes at time $t + \delta_I$ if $V(x)$ stabilizes at time t with constant delay δ_I. Indeed

$$(f_{\mathrm{INV}}, f_{\mathrm{INV}}), V \Vdash \mathrm{INV}(x, y) \text{ iff } \begin{cases} x \text{ stable to 1 at } t \;\Rightarrow\; y \text{ stable to 0 at } t + \delta_I \\ x \text{ stable to 0 at } t \;\Rightarrow\; y \text{ stable to 1 at } t + \delta_I \end{cases}$$

Analogously, the pair $(f^-_{\mathrm{NAND}}, f^+_{\mathrm{NAND}}) \in \lceil \mathrm{NAND}(x, y, z) \rceil$, with

$$f^-_{\mathrm{NAND}}((t_1, t_2)) = \max\{t_1, t_2\} + \delta_N \quad \text{and} \quad f^+_{\mathrm{NAND}}((i, t)) = t + \delta_N \tag{5}$$

provides an example of a data-independent stabilization bound for the NAND gate. Indeed in f^-_{NAND}, δ_N is independent of t_1 (the time at which x stabilizes to 1) and of t_2 (the time at which y stabilizes to 1); in f^+_{NAND}, δ_N is independent of the pair (i, t).

We point out that, in the general case, stabilization bounds represent data-dependent information; e.g., a stabilization bound for $\mathrm{NAND}(x, y, z)$ may consist of a pair of functions (η^-, η^+), where η^- calculates the stabilization bound for output stable to 0 and η^+ for output stable to 1.

It is easy to prove that validity is preserved by time shifting, i.e., $\alpha, V \models A$ implies $\alpha, V^t \models A$ for every $t \in \mathbf{N}$.

Moreover, it is easy to check the following result:

Proposition 1. *Let T be the following time evaluation function:*

- *$T(t) = t$, for $t \in \mathbf{N}$;*
- *$T((\alpha, \beta)) = \max\{T(\alpha), T(\beta)\}$;*
- *$T((i, \alpha)) = T(\alpha)$, for $i = 1, 2$;*
- *$T(f) = 0$, with f any function.*

Let V be a waveform and let A be a formula. For every $t \in \mathbf{N}$, $t, V \Vdash A$ if and only if there is $\alpha \in \lceil A \rceil$ such that $T(\alpha) \leq t$ and $\alpha, V \models A$.

Proposition 1 links the intensional semantics based on stabilization bounds and the extensional semantics of Section 2. Stabilization bounds convey a detailed temporal information allowing us to model exact temporal bounds of the kind considered in [8]. In our setting exact bounds are formalized as follows. Let A be a formula, let $\alpha \in \lceil A \rceil$ and let V be a waveform; α is *exact for V and A* if $\alpha, V \models A$ and one of the following conditions holds:

- $A = \Box B$ or $A = \neg\Box B$;
- $A = a$ or $A = \neg a$, with $a \in \mathbf{S}$, and, for all $t \in \mathbf{N}$, $t, V \models A$ implies $\alpha \leq t$;
- $A = B \wedge C$, $\alpha = (\beta, \gamma)$, β is exact for B and V, and γ is exact for C and V;
- $A = B_1 \vee B_2$, $\alpha = (k, \beta_k)$, with $k \in \{1, 2\}$, and β_k is exact for V and B_k;
- $A = B \rightarrow C$ and, for all $\beta \in \lceil B \rceil$, if β is exact for V and B, then $\alpha(\beta)$ is exact for V and C;
- $A = \neg\neg B$ and α is exact for V and B;
- $A = \neg(B_1 \wedge B_2)$, $\alpha = (k, \beta_k)$, with $k \in \{1, 2\}$, and β_k is exact for V and $\neg B_k$;
- $A = \neg(B \vee C)$, $\alpha = (\beta, \gamma)$, β is exact for V and $\neg B$, γ is exact for V and $\neg C$;

- $A = \neg(B \rightarrow C)$, $\alpha = (\beta, \gamma)$, β is exact for V and B, γ is exact for V and $\neg C$.

For instance, let A be the formula $(x \wedge y \rightarrow \neg z) \wedge (\neg x \vee y \rightarrow z)$ describing the NAND gate, and let V be the waveform of Figure 2. Let $\beta = (\beta^-, \beta^+)$ where $\beta^- : \mathbf{N}^2 \rightarrow \mathbf{N}$ and $\beta^+ : \mathbf{N} \oplus \mathbf{N} \rightarrow \mathbf{N}$; $\beta, V \models A$ iff $\beta^-((t_4, t_5)) = t$ with $t \geq t_6$. An exact stabilization bound for V and A is given by $t = t_6$, which is the "exact instant" where z stabilizes to 0. We remark that such a detailed analysis cannot be accomplished using the approach of [8,9].

4 Timing Analysis of a Circuit

Let us consider the problem to compute the stabilization delays of the XOR circuit of Figure 1. Firstly we have to provide a complete description of the components of the circuit. This means that, for every component of the circuit, we have to provide a formula A representing the component and a time bound $\alpha \in \lceil A \rceil$ which is exact for the set $\mathcal{V}$ of observed behaviors (the set of waveforms resulting from an experimental analysis of the component).

In our example the description is given by the formulas:

- $\mathrm{INV}(b, c)$ and $\mathrm{INV}(a, d)$ obtained by instantiating the formula $\mathrm{INV}(x, y)$ of (1);
- $\mathrm{NAND}(a, c, e)$, $\mathrm{NAND}(b, d, f)$ and $\mathrm{NAND}(e, f, g)$ obtained by instantiating the formula $\mathrm{NAND}(x, y, z)$ of (2).

We remark that such formulas uniquely characterize the structure of the circuit of Figure 1. Indeed, $\mathrm{NAND}(e, f, g)$ describes the NAND gate occurring in the XOR circuit, having as inputs e and f and as output g; in turn, e is the output of the NAND gate having as input a and c, described by the formula $\mathrm{NAND}(a, c, e)$, and f is the output of the NAND gate having as input b and d, described by the formula $\mathrm{NAND}(b, d, f)$ and so on. Hence

$$\mathcal{C}_{\mathrm{XOR}} = \{\mathrm{INV}(a, d), \mathrm{INV}(b, c), \mathrm{NAND}(a, c, e), \mathrm{NAND}(b, d, f), \mathrm{NAND}(e, f, g)\} \qquad (6)$$

is the description of the circuit of Figure 1.

As for the stabilization bounds, in our example we assume that:

- All the instances of the INV gate occurring in the circuit have the same stabilization bound $(f_{\mathrm{INV}}, f_{\mathrm{INV}})$ described in (4);
- All the instances of the NAND gate occurring in the circuit have the same stabilization bound $(f^-_{\mathrm{NAND}}, f^+_{\mathrm{NAND}})$ described in (5).

Starting from this information we would like to compute an exact stabilization bound for the whole circuit. In this section we prove that such a stabilization bound can be extracted from a formal correctness proof of the circuit in a constructive calculus. Here, we use the natural deduction calculus $\mathcal{ND}$ described in Table 1. $\mathcal{ND}$ is obtained by adding to the natural calculus for Intuitionistic Logic (see [12]) the rules for constructive negation $\neg$, the rules for the modal operator $\Box$ and the rule $\mathrm{KP}_{\Box}$. In Table 1 we put between square brackets the assumptions

Table 1. The calculus $\mathcal{ND}$

$$A \ \text{IAx} \qquad \frac{A \quad B}{A \wedge B}\ \text{I}\wedge \qquad \frac{A_1 \wedge A_2}{A_i}\ \text{E}\wedge_i \quad i\in\{1,2\}$$

$$\frac{A_i}{A_1 \vee A_2}\ \text{I}\vee_i \quad i\in\{1,2\} \qquad \frac{A\vee B \qquad \begin{array}{c}[A]\\ \vdots\ \pi_1\\ C\end{array} \qquad \begin{array}{c}[B]\\ \vdots\ \pi_2\\ C\end{array}}{C}\ \text{E}\vee$$

$$\frac{\begin{array}{c}[A]\\ \vdots\ \pi\\ B\end{array}}{A\to B}\ \text{I}\mapsto \qquad \frac{A \quad A\to B}{B}\ \text{E}\to \qquad \frac{A \quad \neg A}{B}\ \text{Contr} \quad \begin{array}{l}\text{where } B = p \text{ or}\\ B = \neg p \text{ with } p \in \mathbf{S}\end{array}$$

$$\frac{\neg A_i}{\neg(A_1 \wedge A_2)}\ \text{I}\neg\wedge_i \ i\in\{1,2\} \qquad \frac{\neg(A\wedge B) \qquad \begin{array}{c}[\neg A]\\ \vdots\ \pi_1\\ C\end{array} \qquad \begin{array}{c}[\neg B]\\ \vdots\ \pi_2\\ C\end{array}}{C}\ \text{E}\neg\wedge$$

$$\frac{A}{\neg\neg A}\ \text{I}\neg\neg \qquad \frac{\neg\neg A}{A}\ \text{E}\neg\neg \qquad \frac{\neg A \quad \neg B}{\neg(A\vee B)}\ \text{I}\neg\vee \qquad \frac{\neg(A_1\vee A_2)}{\neg A_i}\ \text{E}\neg\vee_i \ i\in\{1,2\}$$

$$\frac{A \quad \neg B}{\neg(A\to B)}\ \text{I}\neg\to \qquad \frac{\neg(A\to B)}{A}\ \text{E}\neg\to_1 \qquad \frac{\neg(A\to B)}{\neg B}\ \text{E}\neg\to_2$$

$$\frac{\begin{array}{c}[\neg A]\\ \vdots\ \pi_1\\ B\wedge\neg B\end{array}}{\Box A}\ \text{I}\Box \qquad \frac{\begin{array}{c}[A]\\ \vdots\ \pi_1\\ B\wedge\neg B\end{array}}{\neg\Box A}\ \text{I}\neg\Box \qquad \frac{\begin{array}{c}[\Box A]\\ \vdots\ \pi_1\\ B\vee C\end{array}}{(\Box A\to B)\vee(\Box A\to C)}\ \text{KP}\Box$$

of the proof *discharged* by the application of the rule; $\pi : \{A_1, \ldots, A_n\} \vdash B$ denotes the fact that π is a proof with *undischarged* assumptions $A_1, \ldots, A_n$ and consequence B (for a detailed presentation of such notions see [12]).

It is easy to check that the calculus $\mathcal{ND}$ formalizes a fragment of Classical Logic according to the following translation. Let $\widetilde{H}$ be the formula obtained by deleting from $H \in \mathcal{L}_{\mathbf{S}}$ all the occurrences of $\Box$. If $\{A_1, \ldots, A_n\} \vdash B$ is provable in $\mathcal{ND}$, then $\{\widetilde{A}_1, \ldots, \widetilde{A}_n\} \vdash \widetilde{B}$ is provable in the natural deduction calculus $\mathcal{ND}_{\mathbf{Cl}}$ for Classical Logic.

Coming back to our example, if there exists a proof

$$\Pi \ : \ \mathcal{C}_{\mathrm{XOR}} \vdash \mathrm{XOR}(a, b, g)$$

in $\mathcal{ND}$, then, since $\mathrm{INV}(x, y)$, $\mathrm{NAND}(x, y, z)$ and $\mathrm{XOR}(x, y, z)$ represent the corresponding boolean functions *inv*, *nand* and *xor* according to Definition 1, the input/output behavior of the XOR circuit of Figure 1 is proved to be correct. Obviously, this holds also if Π is a proof of Classical Logic. But, how we are going to show, from proofs of $\mathcal{ND}$ we can also extract information about the stabilization delays. In the following subsection we present the main result about the extraction of stabilization bounds form proofs of $\mathcal{ND}$, and in Subsection 4.2 we apply such a result to compute the propagation delays of the XOR circuit; finally, we also show that proofs of Classical Logics are inadequate to accomplish such an analysis.

4.1 Computing Stabilization Delays

Here we describe how to associate with every proof $\pi : \{A_1, \ldots, A_n\} \vdash B$ of $\mathcal{ND}$ a function $F_\pi : \lceil A_1 \rceil \times \cdots \times \lceil A_n \rceil \to \lceil B \rceil$. Here we denote with $\underline{\alpha}$ an element of $\lceil A_1 \rceil \times \cdots \times \lceil A_n \rceil$. The function is defined by induction on the structure of π as follows.

Assumption Introduction:

$$\begin{gathered} \pi \equiv A \\ F_\pi \text{ is the identity function} \end{gathered} \tag{7}$$

Conjunction Introduction: in this case π is the proof

$$\dfrac{\begin{array}{cc} A_1, \ldots, A_k & A_{k+1}, \ldots, A_n \\ \vdots\ \pi_1 & \vdots\ \pi_2 \\ B & C \end{array}}{B \wedge C}\ \mathrm{I}\wedge \tag{8}$$

$$F_\pi(\underline{\alpha}) = (F_{\pi_1}(\alpha_1, \ldots, \alpha_k), F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_n))$$

The function F_π is defined similarly for the cases corresponding to the rules $\mathrm{I}\neg\vee$, $\mathrm{I}\neg\to$.

Conjunction Elimination: in this case π is the proof

$$\dfrac{\begin{array}{c} A_1, \ldots, A_n \\ \vdots\ \pi_1 \\ B_1 \wedge B_2 \end{array}}{B_i}\ \mathrm{E}\wedge_i \tag{9}$$

$$F_\pi(\underline{\alpha}) = (F_{\pi_1}(\underline{\alpha}))_i$$

The function F_π is defined similarly for the cases corresponding to the rules $\mathrm{E}\neg\vee$, $\mathrm{E}\neg\to$.

Disjunction Introduction: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n\\ \vdots\ \pi_1\\ B_i\end{array}}{B_1 \vee B_2}\ \mathrm{I}\vee_{\mathrm{i}} \tag{10}$$

$$F_\pi(\underline{\alpha}) = (i, F_{\pi_1}(\underline{\alpha}))$$

The function F_π is defined similarly for the case corresponding to the rule I$\neg\wedge$.

Disjunction Elimination: in this case π is the proof

$$\dfrac{\begin{array}{ccc}\begin{array}{c}A_1,\ldots,A_k\\ \vdots\ \pi_1\\ B\vee C\end{array} & \begin{array}{c}A_{k+1},\ldots,A_l,[B]\\ \vdots\ \pi_2\\ D\end{array} & \begin{array}{c}A_{l+1},\ldots,A_n,[C]\\ \vdots\ \pi_3\\ D\end{array}\end{array}}{D}\ \vee\mathrm{E} \tag{11}$$

$$F_\pi(\underline{\alpha}) = \begin{cases} F_{\pi_2}(\alpha_{k+1},\ldots,\alpha_l,\beta) & \text{if } F_{\pi_1}(\alpha_1,\ldots,\alpha_k) = (1,\beta) \\ F_{\pi_3}(\alpha_{l+1},\ldots,\alpha_n,\gamma) & \text{if } F_{\pi_1}(\alpha_1,\ldots,\alpha_k) = (2,\gamma) \end{cases}$$

The function F_π is defined similarly for the case corresponding to the rule E$\neg\wedge$.

Implication Introduction: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n,[B]\\ \vdots\ \pi_1\\ C\end{array}}{B\to C}\ \to\mathrm{I} \tag{12}$$

$F_\pi(\underline{\alpha})$ is the function $f : \lceil B\rceil \to \lceil C\rceil$ such that $f(\beta) = F_{\pi_1}(\underline{\alpha},\beta)$

Implication Elimination: in this case π is the proof

$$\dfrac{\begin{array}{cc}\begin{array}{c}A_1,\ldots,A_k\\ \vdots\ \pi_1\\ B\end{array} & \begin{array}{c}A_{k+1},\ldots,A_n\\ \vdots\ \pi_2\\ B\to C\end{array}\end{array}}{C}\ \to\mathrm{E} \tag{13}$$

$$F_\pi(\underline{\alpha}) = F_{\pi_2}(\alpha_{k+1},\ldots,\alpha_n)(F_{\pi_1}(\alpha_1,\ldots,\alpha_k))$$

Contr: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n\\ \vdots\ \pi_1\\ B\wedge\neg B\end{array}}{C}\ \mathrm{Contr} \tag{14}$$

$$F_\pi(\underline{\alpha}) = \gamma \text{ where } \gamma \text{ is any element in } \lceil C\rceil$$

$\neg\neg$-Introduction: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n\\ \vdots\ \pi_1\\ B\end{array}}{\neg\neg B}\ \mathrm{I}\neg\neg \tag{15}$$

$$F_\pi(\underline{\alpha}) = F_{\pi_1}(\underline{\alpha})$$

The function F_π is defined similarly for the case corresponding to the rule E$\neg\neg$.

$\Box$-Introduction: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n,[\neg B]\\ \vdots\ \pi_1\\ C\wedge\neg C\end{array}}{\Box B}\ \Box\mathrm{I} \tag{16}$$

$$F_\pi(\underline{\alpha}) = 0.$$

The function F_π is defined similarly for the case corresponding to the rule I$\neg\Box$.

Rule $\mathrm{KP}_\Box$: in this case π is the proof

$$\dfrac{\begin{array}{c}A_1,\ldots,A_n,[\Box B]\\ \vdots\ \pi_1\\ C\vee D\end{array}}{(\Box B\to C)\vee(\Box B\to D)}\ \mathrm{KP}_\Box \tag{17}$$

$$F_{\pi_1}(\underline{\alpha}) = \begin{cases}(1,\lambda x.\beta) & \text{if } F_{\pi_1}(\underline{\alpha},0) = (1,\beta)\\ (2,\lambda x.\gamma) & \text{if } F_{\pi_1}(\underline{\alpha},0) = (2,\gamma)\end{cases}$$

The main properties of the function F_π associated with a proof $\pi \in \mathcal{ND}$ are given by the following result.

Theorem 1. *Let $\pi : \{A_1,\ldots,A_n\} \vdash B$ be a proof of the calculus $\mathcal{ND}$ and let*

$$F_\pi : \lceil A_1\rceil \times\cdots\times \lceil A_n\rceil \to \lceil B\rceil$$

be the function associated with π. For all $\alpha_1 \in \lceil A_1\rceil$, ... , $\alpha_n \in \lceil A_n\rceil$, and for every eventually stable waveform V:

(i). $\alpha_1, V \models A_1$, ... , $\alpha_n, V \models A_n$ *implies* $F_\pi(\alpha_1,\ldots,\alpha_n), V \models B$.
(ii). $\alpha'_1 \sim_{A_1} \alpha_1$, ... , $\alpha'_n \sim_{A_n} \alpha_n$ *implies* $F_\pi(\alpha'_1,\ldots,\alpha'_n) \sim_B F_\pi(\alpha_1,\ldots,\alpha_n)$.
(iii). α_1 *exact for* V *and* A_1, ... , α_n *exact for* V *and* A_n *implies* $F_\pi(\alpha_1,\ldots,\alpha_n)$ *exact for* V *and* B.

Proof. We prove the assertion by induction on the structure of the proof π. If π only consists of an assumption introduction (the base case), then F_π is the identity on $\lceil A \rceil$ and the assertions trivially follow. The induction step goes by cases according to the last rule applied in π; here, we only consider some representative cases.

$\vee$-*elimination.* If the last rule applied in π is a $\vee$-elimination, π has the structure described in Point (11).
(i). Let us suppose that $\alpha_1, V \models A_1, \ldots, \alpha_n, V \models A_n$. By induction hypothesis on π_1, $F_{\pi_1}(\alpha_1, \ldots, \alpha_k), V \models B \vee C$. Let us assume that $F_{\pi_1}(\alpha_1, \ldots, \alpha_k) = (1, \beta)$; then $\beta, V \models B$. Now, let us consider the subproof $\pi_2 : \{A_{k+1}, \ldots, A_l, B\} \vdash D$ of π; by induction hypothesis, $F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_l, \beta), V \models D$, from which (i) follows. The case $F_{\pi_1}(\alpha_1, \ldots, \alpha_k) = (2, \gamma)$ is similar.
(ii). Suppose that $\alpha'_1 \sim_{A_1} \alpha_1, \ldots, \alpha'_n \sim_{A_n} \alpha_n$ and $F_{\pi_1}(\alpha_1, \ldots, \alpha_k) = (1, \beta)$. By induction hypothesis on π_1, $F_{\pi_1}(\alpha'_1, \ldots, \alpha'_k) = (1, \beta')$ with $\beta' \sim_B \beta$. By induction hypothesis on π_2, $F_{\pi_2}(\alpha'_{k+1}, \ldots, \alpha'_l, \beta') \sim_D F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_l, \beta)$, hence $F_\pi(\alpha'_1, \ldots, \alpha'_n) \sim_D F_\pi(\alpha_1, \ldots, \alpha_n)$.
(iii). Let α_1 be exact for V and $A_1, \ldots$, let α_n be exact for V and A_n. Let us suppose that $F_{\pi_1}(\alpha_1, \ldots, \alpha_k) = (1, \beta)$; by induction hypothesis, $(1, \beta)$ is exact for V and $B \vee C$, therefore, by definition, β is exact for V and B. By the induction hypothesis on π_2, $F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_l, \beta)$ is exact for V and D, and this concludes the proof.

$\rightarrow$*introduction.* In this case π has the structure described in Point (12). First of all we must check that f is well-defined, i.e., that $\beta \sim_B \beta'$ implies $f(\beta) \sim_C f(\beta')$. If $\beta \sim_B \beta'$, by the induction hypothesis (ii) applied on π_1, $F_{\pi_1}(\alpha_1, \ldots, \alpha_n, \beta) \sim_C F_{\pi_1}(\alpha_1, \ldots, \alpha_n, \beta')$, which implies $f(\beta) \sim_C f(\beta')$.
(i). Let us suppose that $\alpha_1, V \models A_1, \ldots, \alpha_n, V \models A_n$ and let $f = F_\pi(\alpha_1, \ldots, \alpha_n)$; we prove that $f, V \models B \rightarrow C$. Let us take $\beta \in \lceil B \rceil$ and $t \in \mathbf{N}$ such that $\beta, V^t \models B$. We also have $\alpha_1, V^t \models A_1, \ldots, \alpha_n, V^t \models A_n$; since V^t is eventually stable, by induction hypothesis on π_1 we can conclude that $f(\beta), V^t \models C$.
(ii). Let $\alpha'_1 \sim_{A_1} \alpha_1, \ldots, \alpha'_n \sim_{A_n} \alpha_n$, $f = F_\pi(\alpha_1, \ldots, \alpha_n)$, $f' = F_\pi(\alpha'_1, \ldots, \alpha'_n)$. Suppose $\beta \sim_B \beta'$; by induction hypothesis $f(\beta) \sim_C f(\beta')$, hence $f \sim_{B \rightarrow C} f'$.
(iii). Let α_1 be exact for V and $A_1, \ldots, \alpha_n$ be exact for V and A_n; we prove that $f = F_\pi(\alpha_1, \ldots, \alpha_n)$ is exact for V and $B \rightarrow C$. To this aim, let us take $\beta \in \lceil B \rceil$ such that β is exact for V and B. By induction hypothesis, it follows that $f(\beta)$ is exact for V and C, and this concludes the proof.

$\rightarrow$*elimination.* In this case π has the structure described in Point (13).
(i). Suppose that $\alpha_1, V \models A_1, \ldots, \alpha_n, V \models A_n$; let $\beta = F_{\pi_1}(\alpha_1, \ldots, \alpha_k)$ and $f = F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_n)$. By induction hypothesis, we have both $\beta, V \models B$ and $f, V \models B \rightarrow C$, hence $f(\beta), V \models C$ and (i) is proved.
(ii). Let us suppose $\alpha'_1 \sim_{A_1} \alpha_1, \ldots, \alpha'_n \sim_{A_n} \alpha_n$ and let $\beta = F_{\pi_1}(\alpha_1, \ldots, \alpha_k)$, $\beta' = F_{\pi_1}(\alpha'_1, \ldots, \alpha'_k)$, $f = F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_n)$, $f' = F_{\pi_2}(\alpha'_{k+1}, \ldots, \alpha'_n)$. By induction hypothesis we have both $\beta \sim_B \beta'$ and $f \sim_{B \rightarrow C} f'$, and this implies that $f(\beta) \sim_C f'(\beta')$.
(iii). Suppose α_1 to be exact for V and $A_1, \ldots, \alpha_n$ to be exact for V and A_n;

let $\beta = F_{\pi_1}(\alpha_1, \ldots, \alpha_k)$ and $f = F_{\pi_2}(\alpha_{k+1}, \ldots, \alpha_n)$. By induction hypothesis, β is exact for V and B, f is exact for V and $B \to C$; this implies that $f(\beta)$ is exact for V and C.

$\Box$-*introduction*. In this case π has the structure described in Point (16).
(i). Suppose $\alpha_1, V \models A_1, \ldots, \alpha_n, V \models A_n$. Since V is eventually stable, there is $t \in \mathbf{N}$ such that either $t, V \Vdash B$ or $t, V \Vdash \neg B$. Suppose that $t, V \Vdash \neg B$. By Proposition 1, there is $\beta \in \lceil \neg B \rceil$ such that $\beta, V \models \neg B$; by induction hypothesis on π_1, $F_{\pi_1}(\alpha_1, \ldots, \alpha_n, \beta), V \models C \wedge \neg C$, which implies (by Proposition 1) $t', V \Vdash C \wedge \neg C$, where $t' \in \mathbf{N}$, a contradiction. It follows that $t, V \Vdash B$, therefore $0, V \models \Box B$.
The proof of Points (ii) and (iii) is trivial. $\Box$

We remark that the assumption that V is eventually stable is essential to treat the cases of $\Box$-introduction and $\neg\Box$-introduction.

Summarizing, we have shown how to define, for every proof

$$\pi : \{A_1, \ldots, A_n\} \vdash B$$

a function F_π associating with $V \in$ ESTABLE and every n-upla of stabilization bounds $\alpha_1, \ldots, \alpha_n$ for $A_1, \ldots, A_n$ such that α_i is exact for A_i and V, an exact stabilization bound for B and V. We remark that the main advantage of our approach is given by Point (iii) of Theorem 1; indeed the logical approaches to timing analysis of [1,2,5,7,8] do not allow us to compute exact time bounds.

4.2 Application to the XOR Circuit

In this subsection we apply Theorem 1 to compute the exact stabilization bounds for the XOR circuit of Figure 1. To this aim, firstly we describe the formal correctness proof

$$\Pi \ : \ \mathcal{C}_{\mathrm{XOR}} \vdash \mathrm{XOR}(a, b, g)$$

in the calculus $\mathcal{ND}$, then we show how to construct the function F_Π.

The proof can be constructed as follows:

$$\Pi \equiv \dfrac{\begin{array}{cc} \Gamma_3 & \Gamma_6 \\ \vdots\, \pi_3 & \vdots\, \pi_6 \\ (a \wedge \neg b) \vee (\neg a \wedge b) \to g & (a \wedge b) \vee (\neg a \wedge \neg b) \to \neg g \end{array}}{\mathrm{XOR}(a, b, g)}\ \mathrm{I}\wedge$$

where the structure of the proofs π_3 and π_6 is described below.

$$\pi_3 \equiv \dfrac{\dfrac{\dfrac{[(a \wedge \neg b) \vee (\neg a \wedge b)] \quad \begin{array}{cc} [a \wedge \neg b], \Gamma_1 & [\neg a \wedge b], \Gamma_2 \\ \vdots\, \pi_1 & \vdots\, \pi_2 \\ \neg e \vee \neg f & \neg e \vee \neg f \end{array}}{\neg e \vee \neg f}\ \mathrm{E}\vee \quad \dfrac{\mathrm{NAND}(e, f, g)}{\neg e \vee \neg f \to g}\ \mathrm{E}\wedge_2}{g}\ \mathrm{E}\to}{(a \wedge \neg b) \vee (\neg a \wedge b) \to g}\ \mathrm{I}\mapsto$$

where $\Gamma_3 = \Gamma_1 \cup \Gamma_2 \cup \{\mathrm{NAND}(e, f, g)\}$.

$$\pi_6 \equiv \dfrac{\dfrac{[(a \wedge b) \vee (\neg a \wedge \neg b)] \quad \begin{matrix}[a \wedge b], \Gamma_4 \\ \vdots\, \pi_4 \\ e \wedge f\end{matrix} \quad \begin{matrix}[\neg a \wedge \neg b], \Gamma_5 \\ \vdots\, \pi_5 \\ e \wedge f\end{matrix}}{e \wedge f}\,\mathrm{E}\vee \qquad \dfrac{\mathrm{NAND}(e, f, g)}{e \wedge f \to \neg g}\,\mathrm{E}\wedge_1}{\dfrac{\neg g}{(a \wedge b) \vee (\neg a \wedge \neg b) \to \neg g}\,\mathrm{I}\to}\,\mathrm{E}\to$$

where $\Gamma_6 = \Gamma_4 \cup \Gamma_5 \cup \{\mathrm{NAND}(e, f, g)\}$.

$$\pi_1 \equiv \dfrac{\dfrac{\dfrac{a \wedge \neg b}{a}\,\mathrm{E}\wedge_1 \qquad \dfrac{\dfrac{a \wedge \neg b}{\neg b}\,\mathrm{E}\wedge_2 \quad \dfrac{\mathrm{INV}(b, c)}{\neg b \to c}\,\mathrm{E}\wedge_2}{c}\,\mathrm{E}\to}{a \wedge c}\,\mathrm{I}\wedge \qquad \dfrac{\mathrm{NAND}(a, c, e)}{a \wedge c \to \neg e}\,\mathrm{E}\wedge_1}{\dfrac{\neg e}{\neg e \vee \neg f}\,\mathrm{I}\vee_1}\,\mathrm{E}\to$$

where $\Gamma_1 = \{\mathrm{INV}(b, c), \mathrm{NAND}(a, c, e)\}$.

$$\pi_2 \equiv \dfrac{\dfrac{\dfrac{\neg a \wedge b}{b}\,\mathrm{E}\wedge_2 \qquad \dfrac{\dfrac{\neg a \wedge b}{\neg a}\,\mathrm{E}\wedge_1 \quad \dfrac{\mathrm{INV}(a, d)}{\neg a \to d}\,\mathrm{E}\wedge_2}{d}\,\mathrm{E}\to}{b \wedge d}\,\mathrm{I}\wedge \qquad \dfrac{\mathrm{NAND}(b, d, f)}{b \wedge d \to \neg f}\,\mathrm{E}\wedge_1}{\dfrac{\neg f}{\neg e \vee \neg f}\,\mathrm{I}\vee_2}\,\mathrm{E}\to$$

where $\Gamma_2 = \{\mathrm{INV}(a, d), \mathrm{NAND}(b, d, f)\}$.

$$\pi_4 \equiv \dfrac{\dfrac{\dfrac{\dfrac{a \wedge b}{b}\,\mathrm{E}\wedge_2 \quad \dfrac{\mathrm{INV}(b, c)}{b \to \neg c}\,\mathrm{E}\wedge_1}{\neg c}\,\mathrm{E}\to}{\neg a \vee \neg c}\,\mathrm{I}\vee_2 \qquad \dfrac{\mathrm{NAND}(a, c, e)}{\neg a \vee \neg c \to e}\,\mathrm{E}\wedge_2}{e}\,\mathrm{E}\to \qquad \dfrac{\dfrac{\dfrac{a \wedge b}{a}\,\mathrm{E}\wedge_1 \quad \dfrac{\mathrm{INV}(a, d)}{a \to \neg d}\,\mathrm{E}\wedge_1}{\neg d}\,\mathrm{E}\to}{\neg b \vee \neg d}\,\mathrm{I}\vee_2 \qquad \dfrac{\mathrm{NAND}(b, d, f)}{\neg b \vee \neg d \to f}\,\mathrm{E}\wedge_2}{f}\,\mathrm{E}\to$$

$$\dfrac{e \qquad f}{e \wedge f}\,\mathrm{I}\wedge$$

where $\Gamma_4 = \{\mathrm{INV}(a,d), \mathrm{INV}(b,c), \mathrm{NAND}(a,c,e), \mathrm{NAND}(b,d,f)\}$.

$$\pi_5 \equiv \dfrac{\dfrac{\dfrac{\dfrac{\neg a \wedge \neg b}{\neg a}\,\mathrm{E}\wedge_1}{\neg a \vee \neg c}\,\mathrm{I}\vee_1 \quad \dfrac{\mathrm{NAND}(a,c,e)}{\neg a \vee \neg c \to e}\,\mathrm{E}\wedge_2}{e}\,\mathrm{E}\to \quad \dfrac{\dfrac{\dfrac{\neg a \wedge \neg b}{\neg b}\,\mathrm{E}\wedge_2}{\neg b \vee \neg d}\,\mathrm{I}\vee_1 \quad \dfrac{\mathrm{NAND}(b,d,f)}{\neg b \vee \neg d \to f}\,\mathrm{E}\wedge_2}{f}\,\mathrm{E}\to}{e \wedge f}\,\mathrm{I}\wedge$$

where $\Gamma_5 = \{\mathrm{NAND}(a,c,e), \mathrm{NAND}(b,d,f)\}$.

Now, by definition the function associated with Π has the following form:

$$F_\Pi : \lceil \mathrm{INV}(b,c) \rceil \times \lceil \mathrm{INV}(a,d) \rceil \times \lceil \mathrm{NAND}(a,c,e) \rceil \times \lceil \mathrm{NAND}(b,d,f) \rceil \times \lceil \mathrm{NAND}(e,f,g) \rceil \to \lceil \mathrm{XOR}(a,b,g) \rceil$$

In general we can associate with every formula in $\mathcal{C}_{\mathrm{XOR}}$ a different stabilization bound, however, we assume that:

- All the instances of the formula $\mathrm{INV}(x,y)$ have the same stabilization bound $\iota = (\iota^-, \iota^+)$;
- All the instances of the formula $\mathrm{NAND}(x,y,z)$ have the same stabilization bound $\eta = (\eta^-, \eta^+)$.

With these assumptions, we can simply write $F_\Pi(\iota, \eta)$ instead of $F_\Pi(\iota, \iota, \eta, \eta, \eta)$; we do the same for the other functions defined hereafter.

To construct the function F_π we have to consider the case of Conjunction Introduction in Point (8). We get:

$$F_\Pi(\iota, \eta) = (\ F_{\pi_3}(\iota, \eta),\ F_{\pi_6}(\iota, \eta))\ ) \in (\mathbf{N}^2 \oplus \mathbf{N}^2 \to \mathbf{N})^2$$

where F_{π_3} and F_{π_6} are the functions associated with the subproofs π_3 and π_6. The construction goes on as follows:

- $F_{\pi_3}(\iota, \eta)$ is a function $f : \mathbf{N}^2 \oplus \mathbf{N}^2 \to \mathbf{N}$ such that:

$$f((1,(t_1,t_2))) = \eta^+(\ F_{\pi_1}((t_1,t_2),\iota,\eta)\ )$$
$$f((2,(t_1,t_2))) = \eta^+(\ F_{\pi_2}((t_1,t_2),\iota,\eta)\ )$$

- $F_{\pi_1}((t_1,t_2),\iota,\eta) = (\ 1,\ \eta^-((t_1,\iota^+(t_2)))\ ) \in \mathbf{N} \oplus \mathbf{N}$.
- $F_{\pi_2}((t_1,t_2),\iota,\eta) = (\ 2,\ \eta^-((t_2,\iota^+(t_1)))\ ) \in \mathbf{N} \oplus \mathbf{N}$.
- $F_{\pi_6}(\iota, \eta)$ is a function $g : \mathbf{N}^2 \oplus \mathbf{N}^2 \to \mathbf{N}$ such that:

$$g((1,(t_1,t_2))) = \eta^-(\ F_{\pi_4}((t_1,t_2),\iota,\eta)\ )$$
$$g((2,(t_1,t_2))) = \eta^-(\ F_{\pi_5}((t_1,t_2),\iota,\eta)\ )$$

- $F_{\pi_4}((t_1,t_2),\iota,\eta) = (\ \eta^+((2,\iota^-(t_2))),\ \eta^+((2,\iota^-(t_1)))\ ) \in \mathbf{N} \times \mathbf{N}$.
- $F_{\pi_5}((t_1,t_2),\iota,\eta) = (\ \eta^+((1,t_1)),\ \eta^+((1,t_2))\ ) \in \mathbf{N} \times \mathbf{N}$.

Now, given a concrete stabilization bound for the INV and the NAND gates we can compute the resulting stabilization bound for the XOR circuit. Here we consider the stabilization bounds for INV and NAND given in Points (4) and (5); hence $\iota = (f_{\mathrm{INV}}, f_{\mathrm{INV}})$ and $\eta = (f^{-}_{\mathrm{NAND}}, f^{+}_{\mathrm{NAND}})$ where

$$f_{\mathrm{INV}}(t) = t + \delta_I$$
$$f^{-}_{\mathrm{NAND}}((t_1, t_2)) = \max\{t_1, t_2\} + \delta_N \quad \text{and} \quad f^{+}_{\mathrm{NAND}}((i, t)) = t + \delta_N$$

We get:

$$F_{\Pi}(\iota, \eta) = (F_1, F_2)$$
$$F_1((i, (t_1, t_2))) = \begin{cases} \max\{t_1, t_2 + \delta_I\} + 2\delta_N \text{ if } i = 1 \\ \max\{t_2, t_1 + \delta_I\} + 2\delta_N \text{ if } i = 2 \end{cases}$$

$$F_2((i, (t_1, t_2))) = \begin{cases} \max\{t_1, t_2\} + \delta_I + 2\delta_N \text{ if } i = 1 \\ \max\{t_1, t_2\} + 2\delta_N \qquad\quad \text{if } i = 2 \end{cases}$$

As an example, let us suppose that $V(a)$ stabilizes to 1 at time 10 and $V(b)$ stabilizes to 0 at time 20 (see Figure 3). The formula $(a \wedge \neg b) \vee (\neg a \wedge b) \to g$ states that $V(g)$ must stabilize to 1, and the stabilization time is given by the exact stabilization bound t for V and g. By Theorem 1, t corresponds to the value of F_1 on the exact stabilization bound $(1, (10, 20))$ for V and $(a \wedge \neg b) \vee (\neg a \wedge b)$; therefore $t = F_1((1, (10, 20))) = 20 + \delta_I + 2\delta_N$.

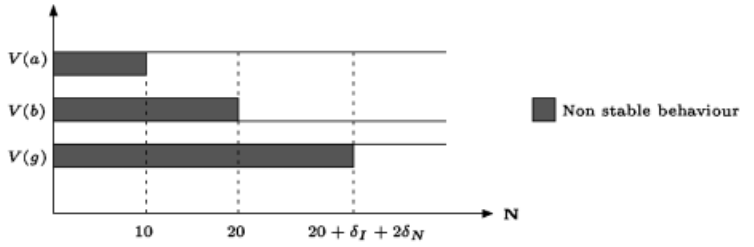


Fig. 3. A possible behavior of the XOR circuit

To conclude this section, we show that Theorem 1 essentially depends on the calculus $\mathcal{ND}$ and does not hold for proofs of Classical Logic. Indeed, let us consider the formulas $\mathrm{XOR}(x, y, z)$ of (3) and its disjunctive normal form

$$\mathrm{XOR}'(x, y, z) = (\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z)$$

Clearly, $\mathrm{XOR}'(x, y, z)$ is classically equivalent to $\mathrm{XOR}(x, y, z)$ and $\mathrm{XOR}'(x, y, z)$ represents the boolean function $xor : \mathbf{N}^2 \to \mathbf{N}$. Moreover, it is easy to find a proof

$$\Pi' \ : \ \mathcal{C}_{\mathrm{XOR}} \vdash \mathrm{XOR}'(a, b, g)$$

in the natural deduction calculus $\mathcal{ND}_{\mathbf{Cl}}$ for Classical Logic (see [12] for the description of such a calculus). On the other hand, as we show hereafter, there is no $\gamma \in \lceil \mathrm{XOR}'(x,y,z) \rceil$ satisfying Point (i) of Theorem 1.

First of all, we remark that the set of stabilization bounds for $\mathrm{XOR}'(x,y,z)$ is isomorphic to $(\mathbf{N}^3 \oplus \mathbf{N}^3 \oplus \mathbf{N}^3 \oplus \mathbf{N}^3)$, hence a stabilization bound of this set can be written as $(i,(t_1,t_2,t_3))$ with $i \in \{1,\ldots,4\}$ and $t_1,t_2,t_3 \in \mathbf{N}$. Now, let us consider the following stabilization bounds for the formulas of $\mathcal{C}_{\mathrm{XOR}}$:

- Let $\iota = (\iota^-,\iota^+)$ be the stabilization bound for all the instances of the formula INV(x, y), where $\iota^-(t) = \iota^+(t) = 0$ for every $t \in \mathbf{N}$;
- Let $\eta = (\eta^-,\eta^+)$ be the stabilization bound for all the instances of the formula NAND(x, y, z), where $\eta^-((t_1,t_2)) = 0$ for every $t_1,t_2 \in \mathbf{N}$ and $\eta^+((i,t)) = 0$ for $i = 1,2$ and for every $t \in \mathbf{N}$.

Now, let us assume that $F_{\Pi'}(\iota,\eta) = (1,(c_1,c_2,c_3))$ for some $c_1,c_2,c_3 \in \mathbf{N}$. Let us consider the stable waveform V such that $V(a) = V(b) = V(e) = V(f) = 1$ and $V(c) = V(d) = V(g) = 0$. It is easy to check that

$$\iota, V \models \mathrm{INV}(a,d) \qquad \iota, V \models \mathrm{INV}(b,c)$$
$$\eta, V \models \mathrm{NAND}(a,c,e) \quad \eta, V \models \mathrm{NAND}(b,d,f) \quad \eta, V \models \mathrm{NAND}(e,f,g)$$

while

$$(1,(c_1,c_2,c_3)), V \not\models \mathrm{XOR}'(a,b,g)$$

since $(c_1,c_2,c_3), V \not\models \neg a \wedge \neg b \wedge \neg g$. Similar conclusions can be obtained considering $F_{\Pi'}(\iota,\eta) = (j,(c_1,c_2,c_3))$ with $j = 2,3,4$ and $c_1,c_2,c_3 \in \mathbf{N}$. Hence, there in no function $F_{\Pi'}$ satisfying Point (i) of Theorem 1. Finally, we point out that there is no proof in $\mathcal{ND}$ of $\mathcal{C}_{\mathrm{XOR}} \vdash \mathrm{XOR}'(a,b,g)$.

5 The Logics $\mathbf{L_{Ef}}$ and $\mathbf{F_{Cl}}$

In this section we briefly discuss the logical properties of our semantics. Let $\mathbf{L}_{\mathrm{Ef}}$ be the logic semantically defined as follows:

$$\mathbf{L}_{\mathrm{Ef}} = \{ A \ : \ \exists \alpha \in \lceil A \rceil \ \forall V \in \textsc{Estable} \quad \alpha, V \models A \ \}$$

where we recall that ESTABLE is the set of all the eventually stable waveforms. It can be shown that $\mathbf{L}_{\mathrm{Ef}}$ is a *non-standard intermediate logic*, that is $\mathbf{Int} \subseteq \mathbf{L}_{\mathrm{Ef}} \subseteq \mathbf{Cl}$, where **Int** (**Cl**) denotes the set of the intuitionistically (classically) valid formulas of $\mathcal{L}_{\mathbf{S}}$ and $\mathbf{L}_{\mathrm{Ef}}$ is closed under *modus ponens*. We emphasize that, differently from *standard* intermediate logics, $\mathbf{L}_{\mathrm{Ef}}$ is not closed under arbitrary substitutions of propositional variables with formulas, but only under substitutions associating a formula of the kind $\Box A$ with every propositional variable. Moreover, $\mathbf{L}_{\mathrm{Ef}}$ has the *disjunction property*, that is $A \vee B \in \mathbf{L}_{\mathrm{Ef}}$ implies $A \in \mathbf{L}_{\mathrm{Ef}}$ or $B \in \mathbf{L}_{\mathrm{Ef}}$.

As a consequence of Theorem 1, every formula provable in the calculus $\mathcal{ND}$ (of Table 1) belongs to $\mathbf{L}_{\mathrm{Ef}}$. This means that $\mathcal{ND}$ is a correct calculus for $\mathbf{L}_{\mathrm{Ef}}$;

on the other hand we do not know if it is complete for $\mathbf{L}_{\mathrm{Ef}}$ and, as far as we know, no axiomatization for $\mathbf{L}_{\mathrm{Ef}}$ is known.

Another logic that emerges from our semantical setting is $\mathbf{F}_{\mathrm{Cl}}$, a well-known axiomatizable non-standard intermediate logic which has strong connections with $\mathbf{L}_{\mathrm{Ef}}$. In [10,11] $\mathbf{F}_{\mathrm{Cl}}$ has been characterized as the smallest set of formulas closed under modus ponens containing **Int**, all the instances of $\mathrm{KP}_\Box$ and all the instances of $\mathrm{At}_\Box$, where:

- $\mathrm{KP}_\Box = (\Box A \to B \vee C) \to (\Box A \to B) \vee (\Box A \to C)$ is the axiom schema obtained by translating the well-known Kreisel and Putnam Principle [3] into the language $\mathcal{L}_{\mathbf{S}}$, and corresponds to the rule $\mathrm{KP}_\Box$ of Table 1;
- $\mathrm{At}_\Box = \Box a \to a$ with $a \in \mathbf{S}$.

A valid and complete natural calculus for $\mathbf{F}_{\mathrm{Cl}}$ is the calculus $\mathcal{ND}_{\mathbf{F}_{\mathrm{Cl}}}$ obtained by adding to $\mathcal{ND}$ the rule:

$$\frac{\Box a}{a}\ \mathrm{E}\Box\mathrm{AT} \quad \text{with } a \in \mathbf{S}$$

In [10,11] it is proved that $\mathbf{F}_{\mathrm{Cl}}$ is a non-standard intermediate logic with the disjunction property. Moreover, $\mathbf{F}_{\mathrm{Cl}}$ meets some important proof-theoretical properties; indeed it is interpolable and enjoys a Normal Form Theorem that can be used to reduce provability in $\mathbf{F}_{\mathrm{Cl}}$ to provability in Classical Logic. In the above quoted papers it is also illustrated the relationship between $\mathbf{F}_{\mathrm{Cl}}$ and *Medvedev Logic of Finite Problems* [6,10].

To characterize $\mathbf{F}_{\mathrm{Cl}}$ in our semantical setting, let

$$\textsc{StableAt}_t = \{V \mid V \text{ is a waveform such that } V^t \text{ is stable}\}$$

It can be shown that

$$\mathbf{F}_{\mathrm{Cl}} = \{A \,:\, \exists \alpha \in \lceil A \rceil\ \forall V \in \textsc{StableAt}_t \quad \alpha, V \models A\ \}$$

From the above semantical characterization it is immediate to check that $\mathbf{L}_{\mathrm{Ef}} \subseteq \mathbf{F}_{\mathrm{Cl}}$; on the other hand, $\mathbf{L}_{\mathrm{Ef}} \neq \mathbf{F}_{\mathrm{Cl}}$ since $\mathrm{At}_\Box$ does not hold in $\mathbf{L}_{\mathrm{Ef}}$.

Also from proofs of $\mathcal{ND}_{\mathbf{F}_{\mathrm{Cl}}}$ we can extract exact stabilization bounds. Here we describe how to associate with every proof $\pi : \{A_1, \ldots, A_n\} \vdash B$ of $\mathcal{ND}_{\mathbf{F}_{\mathrm{Cl}}}$ a function $F^\tau_\pi : \lceil A_1 \rceil \times \cdots \times \lceil A_n \rceil \to \lceil B \rceil$ where τ is a parameter in $\mathbf{N}$ needed to treat the rule E□AT. The function is defined by induction on the structure of π. For the rules occurring in $\mathcal{ND}$, the function is defined according to Points (7)-(17) (the parameter τ plays no role), while the rule E□AT is treated as follows:

Rule E□AT: in this case π is the proof

$$\frac{\begin{array}{c} A_1, \ldots, A_n \\ \vdots\ \pi_1 \\ \Box a \end{array}}{a}\ \mathrm{E}\Box\mathrm{AT} \tag{18}$$

$$F_\pi(\underline{\alpha}) = \tau$$

The main properties of the function F^t_π associated with a proof $\pi \in \mathcal{ND}_{\mathbf{F}_{\mathrm{Cl}}}$ and with $t \in \mathbf{N}$ are given by the following result.

Theorem 2. *Let $\pi : \{A_1, \ldots, A_n\} \vdash B$ be a proof of the calculus $\mathcal{ND}_{\mathbf{F}_{\mathrm{Cl}}}$, let $t \in \mathbf{N}$ and let*

$$F^t_\pi : \lceil A_1 \rceil \times \cdots \times \lceil A_n \rceil \to \lceil B \rceil$$

be the function associated with π and t. For all $\alpha_1 \in \lceil A_1 \rceil$, ... , $\alpha_n \in \lceil A_n \rceil$, and for every waveform $V \in \textsc{StableAt}_t$:

(i). $\alpha_1, V \models A_1$, ... , $\alpha_n, V \models A_n$ *implies* $F^t_\pi(\alpha_1, \ldots, \alpha_n), V \models B$.
(ii). $\alpha'_1 \sim_{A_1} \alpha_1$, ... , $\alpha'_n \sim_{A_n} \alpha_n$ *implies* $F^t_\pi(\alpha'_1, \ldots, \alpha'_n) \sim_B F^t_\pi(\alpha_1, \ldots, \alpha_n)$.
(iii). α_1 *exact for* V *and* A_1, ... , α_n *exact for* V *and* A_n *implies* $F^t_\pi(\alpha_1, \ldots, \alpha_n)$ *exact for* V *and* B.

6 Conclusion

In this paper we have shown how we can get a timing analysis with data-dependent valuation of exact delays by a specialization of evaluation forms semantics [11]. There are several interesting aspects we aim to investigate in our future work.

As for the semantics here considered, we want to examine thoroughly the kind of delay information related to different formulas representing the same boolean function. As an example we remark that the $\Box$ operator can be used to avoid the timing analysis of the subformulas to which it applies. It is easy to see that there exists a proof

$$\Pi' \ : \ \mathcal{C}_{\mathrm{XOR}} \vdash \Box\mathrm{XOR}(a, b, g)$$

in $\mathcal{ND}$, and this proof guarantees (according to Definition 1) the correctness of the input/output behavior of the XOR circuit of Figure 1. On the other hand the stabilization bounds for $\Box$-formulas give no information about the delays. At the same way, using $\Box$ in front of a formula representing a component of the circuit, we can abstract from the temporal behavior of such a component.

As for the expressiveness of our language, we observe that the *nand* function, we described by means of the formula $\mathrm{NAND}(x, y, z)$ of Point (2), can also be represented by different formulas, e.g.,

$$\mathrm{NAND}'(x, y, z) \equiv (x \wedge y \to \neg z) \wedge (\neg x \to z) \wedge (\neg y \to z)$$

Actually, $\lceil \mathrm{NAND}(x, y, z) \rceil \neq \lceil \mathrm{NAND}'(x, y, z) \rceil$, however accomplishing the analysis of the XOR circuit using $\mathrm{NAND}'(x, y, z)$ we obtain essentially the same results, e.g., also in this case we obtain the diagram of Figure 3.

Another aspect we aim to investigate is the extension of our language by other modal operators as the Lax operator of [8].

Finally, we remark that the semantical setting of evaluation forms supports a variety of specializations that preserve the Soundness Theorem, that is the

fundamental result to compute stabilization bounds with proofs. In this paper we have studied a specialization of evaluation forms semantics directly inspired by [7,8]; our aim is to investigate other specializations of evaluation forms semantics and their relation with timing analysis models.

References

1. D.A. Basin and N. Klarlund. Automata based symbolic reasoning in hardware verification. *Formal Methods in Systems Design*, 13(3):255–288, 1998.
2. J. Brzozowski and M. Yoeli. Ternary simulation of binary gate networks. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 41–50. D. Reidel, 1977.
3. A. Chagrov and M. Zakharyaschev. *Modal Logic*. Oxford University Press, 1997.
4. C.T. Gray, W. Liu, R.K. Cavin III, and H.-Y. Hsieh. Circuit delay calculation considering data dependent delays. *INTEGRATION, The VLSI Journal*, 17:1–23, 1994.
5. S. Malik. Analysis of Cyclic Combinational Circuits. In *IEEE /ACM International Conference on CAD*, pages 618–627. ACM/IEEE, IEEE Computer Society Press, 1993.
6. Ju.T. Medvedev. Interpretation of logical formulas by means of finite problems and its relation to the realizability theory. *Soviet Mathematics Doklady*, 4:180–183, 1963.
7. M. Mendler. A timing refinement of intuitionistic proofs and its application to the timing analysis of combinational circuits. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings of the 5th International Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 261–277. Springer, LNAI 1071, 1996.
8. M. Mendler. Characterising combinational timing analyses in intuitionistic modal logic. *Logic Journal of the IGPL*, 8(6):821–852, 2000.
9. M. Mendler. Timing analysis of combinational circuits in intuitionistic propositional logic. *Formal Methods in System Design*, 17(1):5–37, 2000.
10. P. Miglioli, U. Moscato, M. Ornaghi, S. Quazza, and G. Usberti. Some results on intermediate constructive logics. *Notre Dame Journal of Formal Logic*, 30(4):543–562, 1989.
11. P. Miglioli, U. Moscato, M. Ornaghi, and G. Usberti. A constructivism based on classical truth. *Notre Dame Journal of Formal Logic*, 30(1):67–90, 1989.
12. D. Prawitz. *Natural Deduction*. Almquist and Winksell, 1965.
13. R.H. Thomason. A semantical study of constructible falsity. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 15:247–257, 1969.