Unfaithful Glitch Propagation in Existing Binary Circuit Models

Matthias Függer, Thomas Nowak, Ulrich Schmid

Unfaithful Glitch Propagation in Existing Binary Circuit Models

Matthias Függer
Embedded Computing Systems Group
Institute of Computer Engineering
Vienna University of Technology
Vienna, Austria
Email: fuegger@ecs.tuwien.ac.at

Thomas Nowak
Laboratoire d'Informatique
École polytechnique
Palaiseau, France
Email: nowak@lix.polytechnique.fr

Ulrich Schmid
Embedded Computing Systems Group
Institute of Computer Engineering
Vienna University of Technology
Vienna, Austria
Email: s@ecs.tuwien.ac.at

***Abstract*—We show that no existing continuous-time, binary value-domain model for digital circuits is able to correctly capture glitch propagation. Prominent examples of such models are based on pure delay (P) channels, inertial delay (I) channels, or the elaborate PID channels proposed by Bellido-Díaz et al. We accomplish our goal by considering the solvability/non-solvability border of a simple problem called Short-Pulse Filtration (SPF), which is closely related to arbitration and synchronization. On one hand, we prove that SPF is solvable in bounded time in any such model that provides channels with non-constant delay, like I and PID. However, this is in opposition to the impossibility of solving bounded SPF in real (Newtonian) circuit models, which follows from well-known results on the behavior of bi-stable circuits obtained by Marino. On the other hand, for binary circuit models with pure delay channels, we prove that SPF cannot be solved even in unbounded time. This, however, is in opposition to the fact that one can easily solve the unbounded SPF problem in Newtonian circuit models. Consequently, indeed none of the binary value-domain models proposed so far faithfully captures glitch propagation of real circuits.**

***Index Terms*—circuit models; glitch propagation; binary models; modeling issues;**

I. Introduction

Binary value-domain models that allow to model glitch propagation have always been of interest, especially in asynchronous design [1]: Pure delay channels, which propagate input pulses with some constant delay, and inertial delay channels, which propagate input pulses with some constant delay only when they exceed some minimal duration, are still the basis of most digital timing analysis approaches and tools. The tremendous advances in digital circuit technology, in particular, increased speeds and reduced voltage swings, raised concerns about the accuracy of these models [2]. For example, neither pure nor inertial delay models can express the well-known phenomenon of propagating glitches that decay from stage to stage, which is particularly important for analyzing high-frequency pulse trains or oscillatory metastability [3].

At the same time, the steadily increasing complexity of contemporary digital circuits fuels the need for fast digital timing analysis techniques: Although accurate Spice models, which facilitate very precise analog-level simulations, are usually available for those circuits, the achievable simulation times are prohibitive. Refined digital timing analysis models like the PID model proposed by Bellido-Díaz et al. [2], which is both fast and more accurate, are hence very important from a practical perspective [4].

The interest in binary models that faithfully model glitch propagation and even metastability has also been stimulated recently by the increasing importance of incorporating fault-tolerance in circuit design [5]: Reduced voltage swings and smaller critical charges make circuits more susceptible to particle hits, crosstalk, and electromagnetic interference [6], [7]. Since single-event transients, caused by an ionized particle hitting a reverse-biased transistor, just manifest themselves as short glitches, accurate propagation models are important for assessing soft error rates, in particular, for asynchronous circuits. After all, if system-level fault-tolerance techniques like triple modular redundancy are used for transparently masking value failures, the only remaining issue are timing failures, among which glitches are the most problematic ones.

As a consequence, discrete-value circuit models, analysis techniques and supporting tools for a fast but nevertheless accurate glitch and metastability propagation analysis will be a key issue in the design of future VLSI circuits. In this paper, we rigorously prove that a generalization of the existing binary-value candidate models proposed in the past does not capture glitch propagation adequately. Searching for alternative models is hence an important challenge for future research on asynchronous circuits.

Detailed contributions: In Section III, we present a generic binary value-domain model for digital clocked as well as clockless circuits and introduce the SPF problem and its time-bounded variant. The SPF problem is closely related to glitch propagation, as it is essentially the problem of building a one-shot inertial channel. Our generic model comprises zero-time logical gates interconnected by channels that encapsulate model-specific propagation delays and related decay effects. Non-zero time logical gates can thus be expressed in our model by appending channels with delay at the gate's inputs and outputs.

In Section IV, we prove that (even unbounded) SPF is unsolvable when only pure, i.e., constant, delay channels are available. In Section V, we demonstrate that this is incompatible with what is known for real circuits: We show that a

metastability filter based on a high-threshold inverter allows to solve SPF in the Newtonian circuit model of Marino [3].

In Section VI, we turn our attention to a generalization of constant delay channels, termed *single-history channels*, which are FIFO channels with a generalized delay function that also takes into consideration the last output transition. We distinguish between *forgetful* and *non-forgetful* single-history channels, depending on their behavior when a pulse disappears at the output due to decay effects. All existing binary models we are aware of can be expressed as single-history channels with specific delay functions: A pure delay channel (P) as either a forgetful or non-forgetful single-history channel, a classical inertial delay channel (I) as a forgetful single-history channel, and the channel model proposed by Bellido-Díaz et al. [2] that additionally has a decay component (PID) as a non-forgetful single-history channel.

In Section VII, we prove that bounded SPF is solvable if just a single forgetful or non-forgetful single-history channel with non-constant delay is available: We present circuits solving SPF for all non-constant delay channels and prove their correctness. However, by using a reduction of bounded SPF to the well-known impossibility of building a bistable circuit that stabilizes in bounded-time in the Newtonian circuit model of Marino [3], we again show the inadequacy of any of these models for modeling glitch propagation in real circuits.

Fig. 1 summarizes our possibility and impossibility results, showing the discrepancy of each model with real circuits.

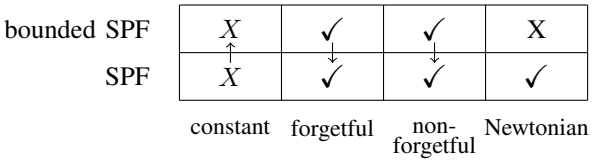

Fig. 1. Summarizing Possibility (✓) and Impossibility (X) results for constant, non-constant forgetful, non-constant non-forgetful and Newtonian physical channels. Arrows represent implications, e.g., resulting from the fact that a circuit that solves bounded SPF also solves SPF.

II. Related Work

Unger [1] proposed a general technique for deriving asynchronous sequential switching circuits that can cope with unrelated input signals. It assumes signals to be binary valued, and requires the availability of combinational circuit elements, as well as pure and inertial delay channels.

Bellido-Díaz et al. [2] propose the PID model, and justify its appropriateness both analytically and by comparing the model predictions against Spice simulation results. The results confirm very good accuracy even for such challenging scenarios as long chains of gates and ring oscillators.

Marino [8] showed that the problem of building a synchronizer can be reduced to the problem of building an inertial delay channel. The reduction circuit only makes use of combinational gates and pure delay channels in addition to inertial delay channels. Marino further shows, in a continuous value signal model, that for a set of standard designs of inertial delay channels, input pulses exist that produce outputs violating the requirements of inertial delay channels. Barros and Johnson [9] extended this work, by showing the equivalence of arbiter, synchronizer, latch, and inertial delay channels.

Marino [3] developed a general theory of metastable operation, and provided impossibility proofs for metastability-free synchronizers and arbiter circuits for several continuous-value circuit models. Branicky [10] proved the impossibility of time-unbounded deterministic and time-invariant arbiters modeled as ordinary differential equations. Mendler and Stroup [11] considered the same problem in the context of continuous automata.

Brzozowski and Ebergen [12] formally proved that, in a model that uses only binary values, it is impossible to implement Muller C-Elements (among other basic state-holding components used in (quasi) delay-insensitive designs) using only zero-time logical gates interconnected by wires without timing restrictions.

Függer and Schmid [13] use a binary-value modeling framework for analyzing the Byzantine fault-tolerant distributed DARTS clocking approach. They proposed a system of interconnected Tick Generation (TG) components, and showed that clock transitions at each non-faulty TG component are generated in synchrony (i.e., within bounded skew and with bounded minimum and maximum delay between successive ticks), despite at most a third of its TG components being Byzantine faulty. They use pure delay channels and assume the existence of SPF modules to circumvent glitch propagation induced by faults. The same modeling and analysis framework is also used in the self-stabilizing Byzantine fault-tolerant FATAL protocol for distributed clock generation in SoCs [14].

III. Binary System Model

In this section, we introduce our binary value-domain circuit model and the SPF problem.

A. Signals, Events

We consider circuits processing binary-valued signals with continuous time, i.e., signal values are from $\mathbb{B} = \{0, 1\}$ and they evolve over time $\mathcal{T} = [0, \infty)$. A *signal* is a function $\mathcal{T} \to \mathbb{B}$ that does not change an infinite number of times during a finite time interval and that already has its new value at a time instant of a value transition.[1] A signal transition is modeled by an *event*. Formally an event is a pair $e = (t, x)$ in $(\mathcal{T} \cup \{-\infty\}) \times \mathbb{B}$. We call t the event's *time* and x the event's *value*. We use "virtual events" at $t = -\infty$ to simplify notation when specifying initial values. An *event list* is a finite or infinite sequence of events.

To every signal, there corresponds an event list $(e_n) = (t_n, x_n)$ with the following properties:

S1) There is always an initial event at time $-\infty$.

[1]The requirement that a signal already has its new value when changing values is merely a convention. On the other hand, the requirement that it only changes a finite number of times during a finite time interval is more fundamental to our model and thus, our results.

S2) The sequence (t_n) of event times is strictly increasing and discrete, with $\lim_{n\to\infty} t_n = \infty$ if the event list is infinite.

S3) Values are alternating: $x_n \neq x_{n+1}$

Conversely, every such event list corresponds to a unique signal.

B. Channels, Constant Delay Channels

A *channel* c is a function mapping an input signal s to an output signal $c(s)$. The simplest class of channels is the class of constant delay channels. A *constant delay* channel c with delay parameter δ and initial value $x \in \mathbb{B}$ produces at its output the input signal delayed by δ, i.e.,

$$c(s)(t) = \begin{cases} x & \text{if } t < \delta \\ s(t-\delta) & \text{if } t \geqslant \delta \end{cases} .$$

C. Circuits, Executions

Circuits are obtained by interconnecting a set of input ports and a set of output ports, forming the external interface of a circuit, and a set of combinational gates via channels. We constrain the way components are interconnected in a natural way, by requiring that input ports are attached to one or more channel inputs only (C4), and that both output ports and gate inputs are attached to just one channel's output (C5, C6); the latter prevents channel outputs driving against each other.

Formally, a *circuit* is a tuple $C = (G, I, O, c, n)$ such that

C1) G is a directed graph whose vertex set can be partitioned as $I \cup O \cup B$.

C2) Every vertex b in B (*(Boolean) gate*) is assigned a Boolean function $\mathbb{B}^d \to \mathbb{B}$ where d is the in-degree (i.e., the number of incoming neighbors) of b. By slight abuse of notation, b also denotes the function assigned to b.

C3) c is a function that maps every edge (u, v) in G to its corresponding channel $c_{u,v}$.

C4) Every vertex in I (*input ports*) has in-degree 0.

C5) Every vertex in O (*output ports*) has in-degree 1.

C6) n is a function that maps every vertex v in G to a linearly ordered subset $n_v = \{v_1, \ldots, v_{d_v}\}$ of its in-neighbor vertices in G, i.e., where edge (v_i, v) for $i = 1$ up to v's in-degree d_v is in G.

An execution of circuit C is an assignment of signals to vertices that respects the channel functions and Boolean gate functions.

Formally, an *execution* of circuit C is a collection of signals s_v for all vertices v of C such that the following properties holds: If i is an input port, then there are no restrictions on s_i. If o is an output port, then $s_o = c_{v,o}(s_v)$ where v is the unique incoming neighbor of o. Let now b be a Boolean gate with d incoming neighbors $v_1, v_2, \ldots, v_d$, ordered according to n_b. We then apply, for each incoming edge (v_k, b), the channel $c_{v_k,b}$ to signal s_{v_k} and check that the signal value $s_b(t)$ is the gate's Boolean combination of these incoming signals at time t. That is, for all $t \in \mathcal{T}$,

$$s_b(t) = b\big(c_{v_1,b}(s_{v_1})(t)\,,\ \ldots\ ,\ c_{v_d,b}(s_{v_d})(t)\big)\ .$$

Not all circuits necessarily do have executions. For example, the circuit comprising a single inverter gate whose output is fed back to its input via a zero-delay channel does not have an execution. Whenever we introduce a circuit (for a possibility result), we will thus make sure that it allows for a unique execution once the input signals are fixed. In case of positive constant delay channels, this is always the case. In particular, using inverter gates with non-zero constant delay feedback channels, it is possible to produce perfect clock signals with arbitrary rational duty cycles. Thus, synchronous circuits with multiple clock sources whose phase shift can be precisely defined by the circuit designer can be specified within our circuit model. This obviously strengthens the impossibility result of Section IV.

D. Short-Pulse Filtration

A *pulse* p of length $\Delta > 0$ at time T is a signal of the form

$$p(t) = \begin{cases} 0 & \text{if } t < T \text{ or } t \geqslant T + \Delta \\ 1 & \text{if } T \leqslant t < T + \Delta \end{cases} .$$

A signal *contains a pulse* of length $\Delta > 0$ at time T if its event list contains the subsequent events $(T, 1)$ and $(T+\Delta, 0)$.

A circuit *solves Short-Pulse Filtration (SPF)* if it fulfills the following conditions:

F1) It has exactly one input port i and exactly one output port o.

F2) For every pulse p, there exists an execution that has p as the input signal (i.e., $s_i = p$). *(Well-formedness)*

F3) In all executions, if the input signal is zero, then so is the output signal. *(No generation)*

F4) There exist a pulse p such that, in all executions with p as the input signal, the output signal is not the zero signal. *(Nontriviality)*

F5) There exists an $\varepsilon > 0$ such that, in all executions, the output signal does not contain a pulse of length less than ε. *(No short pulses)*

A circuit *solves bounded SPF* if additionally:

F6) There exists a $K > 0$ such that, in all executions with a pulse of length Δ at time T as the input signal, the output signal does not change anymore after time $T + \Delta + K$. *(Bounded stabilization time)*

IV. Unsolvability of Short-Pulse Filtration with Constant-Delay Channels

In this section, we present our first major result, namely that no circuit whose channels are all positive constant-delay channels solves SPF. The idea of the proof is to exploit the fact that the value of the output signal of the circuit at each time t only depends on a *finite* number of values of the input signal at times t' between 0 and t.

Calling each such time t' a *measure point* for time t, we show that indeed only a finite number of measure points exists for time t, i.e., the circuit cannot distinguish two different input signals that do not differ in the input signal values at the measure points for time t: For both such input signals, the

output signal must have the same value at time t. Combining this indistinguishability result with a shifting argument of the input signal allows us to construct an arbitrary short pulse at the output of the circuit, a contradiction to property (F5) of Short-Pulse Filtration.

For each constant delay circuit with a single input port and a single output port, we define its *dependence graph*, which describes the way output signals depend on input signals:

Let $C = (G, I, O, c, m)$ be a circuit with constant delay channels, a single input port i, and a single output port o. For every channel $c_{u,v}$ of C, denote by $\delta(u, v)$ its delay parameter δ and by $x(u, v)$ its initial value. The *dependence graph* $DG(t)$ of C at time t is a directed graph with vertices (v, τ), where v is a vertex in G and τ a time. It is defined as follows:

- The pair $(o, 0)$ is a vertex of $DG(t)$.
- If (v, τ) is a vertex of $DG(t)$ and (u, v) is an edge in G with $\tau + \delta(u, v) \leqslant t$, then the pair $\big(u, \tau + \delta(u, v)\big)$ is also a vertex of $DG(t)$ and there is an edge in $DG(t)$ from $\big(u, \tau + \delta(u, v)\big)$ to (v, τ).
- If (v, τ) is a vertex of $DG(t)$ and (u, v) is an edge in G with $\tau + \delta(u, v) > t$, then $x(u, v)$ is a vertex of $DG(t)$ and there is an edge in $DG(t)$ from $x(u, v)$ to (v, τ).

Because all $\delta(u, v)$ are strictly positive, the dependence graphs are finite and acyclic. A vertex of $DG(t)$ without incoming neighbors is a *leaf*, all others *intermediate vertices*. A vertex of the form (i, τ), with $i \in I$, is an *input leaf* and we call the time $t - \tau$ the corresponding *measure point* for time t. If $DG(t) = DG(\tilde{t})$, then the measure points for t are exactly the measure points for $\tilde{t}$ shifted by the difference $t - \tilde{t}$. All leaves of $DG(t)$ are either input leaves or elements of $\mathbb{B}$ (initial values of channels).

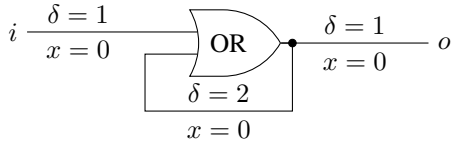

Fig. 2. Example circuit

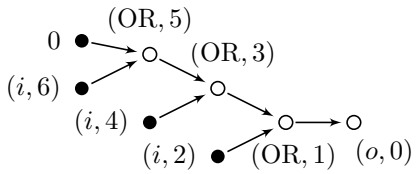

Fig. 3. Example dependence graph $DG(6)$

As an example, consider the circuit shown in Fig. 2. The dependence graph $DG(6)$ is shown in Fig. 3. Leaves are depicted as filled nodes, while intermediate nodes are empty. From the construction of the graph, we immediately see that in each execution the output signal value $s_o(6)$ only depends on the three input signal values $s_i(4)$, $s_i(2)$, and $s_i(0)$.

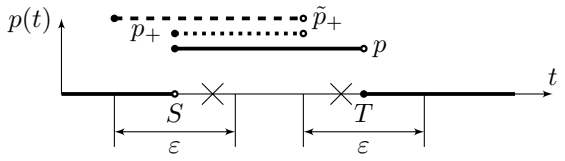

Fig. 4. Input pulse p, together with its derived pulses p_+ and $\tilde{p}_+$, and measure points for time $\tilde{t}$

Generalizing this observation, we obtain:

Lemma 1. *The value of the output signal at time t only depends on the values of the input signal at the measure points for time t, according to $DG(t)$.*

Furthermore, if $DG(t) = DG(\tilde{t})$ and the values of input signals s_i and $\tilde{s}_i$ coincide at the respective measure points for t and $\tilde{t}$, then the respective output signals fulfill $s_o(t) = \tilde{s}_o(\tilde{t})$.

Due to the fact that there are only finitely many measure points for a given time t, they are discrete and hence there is always a small margin until a new measure point appears:

Lemma 2. *For every $t \in \mathcal{T}$ there exists an $\varepsilon > 0$ such that $DG(t) = DG(t + \varepsilon')$ for all $0 \leq \varepsilon' \leq \varepsilon$.*

In the rest of this section, we prove the following impossibility result.

Theorem 1. *No circuit with positive constant delay channels solves SPF.*

Assume by contradiction that circuit C solves SPF. By the nontriviality property (F4), there exists an input pulse such that the corresponding output signal is non-zero, i.e., there exists an input pulse and a time t such that the corresponding output signal's value at time t is 1.

By Lemma 2, there exists an $\varepsilon > 0$ such that $DG(t) = DG(t + \varepsilon)$. We may choose ε arbitrarily small, in particular strictly smaller than all differences of distinct measure points for time t. Choose $\tilde{t} = t + \varepsilon/2$ to be the midpoint of the interval $[t, t + \varepsilon]$ and denote by Δ the infimum of input pulse lengths such that the corresponding output signal's value at time $\tilde{t}$ is 1. This infimum is finite by the choice of t and $\tilde{t}$.

Now let p be a pulse of length slightly larger than Δ, with an upward transition at time S and a downward transition at time T. We can choose p's length in such a way that the output value at time $\tilde{t}$ is 0 whenever we shorten p's length by $\varepsilon/2$. This implies that there exists one measure point for time $\tilde{t}$ between $T - \varepsilon/2$ and T. (See shortened pulse p_+ and the marked measure point on the right in Fig. 4.) Because ε was chosen to be smaller than the distance between any two measure points for $\tilde{t}$, there is no measure point for $\tilde{t}$ between T and $T + \varepsilon/2$.

Similarly, there is one measure point for time $\tilde{t}$ between S and $S + \varepsilon/2$, and none between $S - \varepsilon/2$ and S (see Fig. 4).

Now consider the pulse $\tilde{p}_+$ generated by shifting pulse p into the past by $\varepsilon/2$, i.e., $\tilde{p}_+$'s upwards transition is at time $S - \varepsilon/2$ and its downwards transition at $T - \varepsilon/2$. Because $\tilde{p}_+$

coincides with p_+ at all measure points for $\tilde{t}$, the output signal corresponding to $\tilde{p}_+$ has value 0 at time $\tilde{t}$. Because $DG(\tilde{t}) = DG(\tilde{t}+\varepsilon/2)$, part two of Lemma 1 shows that $s_p(\tilde{t}+\varepsilon/2) = 0$.

Likewise, by considering p shifted into the future by $\varepsilon/2$, we see that also $s_p(\tilde{t} - \varepsilon/2) = 0$. But because $s_p(\tilde{t}) = 1$, this shows that the output signal s_p contains a pulse of length strictly less than ε. Since ε can be chosen arbitrarily small, we have proved Theorem 1.

V. Possibility of Short-Pulse Filtration in Physical Systems

In this section, we will reconsider the SPF problem in the model of [3], which matches physical circuits. In sharp contrast to the *impossibility* of implementing unbounded SPF in our binary model with constant delay channels established in Theorem 1, we will show in Theorem 2 that it is *possible* to build such a circuit in reality.

The model of Marino [3] considers circuits, which process signals with both continuous value-domain and continuous time-domain. Accordingly, we assume (normalized) signal voltages to be within $[0, 1]$, and denote by $L_0 = [0, l_0]$ resp. $L_1 = [l_1, 1]$, with $0 < l_0 < l_1 < 1$, the signal ranges that are interpreted as logical 0 resp. logical 1 by a circuit.

A *physical* circuit with one input signal i and one output signal o, reset at time 0 to a predefined state, solves Short-Pulse Filtration (SPF) if it satisfies the natural generalizations of Properties (F1) – (F5) defined in Section III-D. In particular, (F5) [which prohibits output pulse with duration $< \varepsilon$] says that if the output signal is not interpreted as logical 1 at two points in time t and t' with $t' - t < \varepsilon$, then it is not logical 1 at any time in between t and t'.

A physical circuit that solves SPF can be implemented by combining a simple storage loop (like the one shown in Fig. 2) and a high-threshold buffer acting as a metastability filter (see, e.g., [15, p. 40]): It is easy to see that the properties of SPF are fulfilled for input signals that are either constant 0 or pulses of a duration longer than the delay of the feedback loop. According to [3], short pulses may drive the storage loop into a metastable internal state for an unbounded time, however. It may hence produce an output signal within some region of metastable output values $[v_M^-, v_M^+] \subset [0, 1]$ during an unbounded time; the values v_M^-, and v_M^+ depend on technology parameters. However, since it is possible to compute safe bounds V_M^-, and V_M^+ such that $[v_M^-, v_M^+] \subset [V_M^-, V_M^+] \subset [0, 1]$, a subsequent high-threshold buffer with threshold larger than V_M^+ can be used to map any metastable internal state to logical 0 at the output, which effectively prohibits short pulses at the output. Hence, we obtain:

Theorem 2. *There is a physical circuit that solves SPF.*

VI. Single-History Channels

This section formally introduces the notion of a single-history channel in the binary circuit model. They are a generalization of constant-delay channels that cover all existing channel models for binary circuit models we are aware of.

Intuitively, a single-history channel propagates each event at time t of the input signal to an event at the output happening after some *output-to-input* delay $\delta(T)$, which depends on the *input-to-previous-output* delay $T = t - t'$. Note that T is positive if the channel delay is short compared to the input signal transition times, and negative otherwise. Fig. 5 illustrates this relation and the involved delays. In case FIFO order would be invalidated, i.e., $t + \delta(T) \leqslant t'$, such that the next output event would not occur after the previous one, both events annihilate.

There exist two variants of single-history channels in the literature, depending on whether the time of an annihilated event is remembered or not. We dub these two variants *forgetful* and *non-forgetful* single-history channels, which we both formally define below. At the end of this section, we give a list of channel models that are special cases of our definition of a single-history channel.

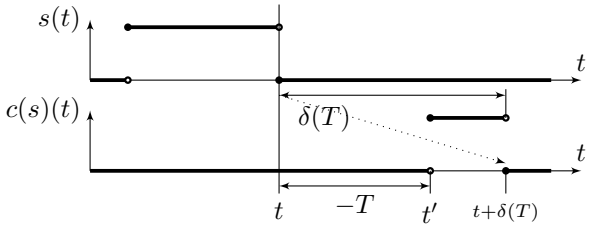

Fig. 5. Input/output signal of a single-history channel, involving the input-to-previous-output delay T and the resulting output-to-input delay $\delta(T)$

Formally, a *single-history channel* c is characterized by an *initial value* $x \in \mathbb{B}$, a nondecreasing *delay function* $\delta : \mathbb{R} \to \mathbb{R}$ such that $\delta(\infty) = \lim_{T\to\infty} \delta(T)$ is finite, and the fact whether it is forgetful or not. We detail the channel behavior in the next two subsections.

A. Forgetful Single-History Channels

This class of channels includes the classical inertial delay channels as used, for example, in VHDL simulators [16].

Their behavior is defined by the following algorithm: Let s be a signal and let $\big((t_n, x_n)\big)_n$ be its event list, the *input list*. The algorithm iterates the input list and updates the *output list*, which will define the channel's output signal $c(s)$.

Initially, let $(-\infty, x)$ be the sole element of the output list. In its nth iteration, the algorithm considers input event (t_n, x_n) and modifies the output list accordingly:

1) Denote by (t'_n, x'_n) be the last event in the output list. If $x_n = x'_n$, then input event (t_n, x_n) has no effect: Proceed to the $(n + 1)$th iteration.
2) Otherwise, let $T_n = t_n - t'_n$ be the difference of input and output event times. (Note that $T_n = \infty$ is possible. In this case $\delta(T_n) = \delta(\infty) = \lim_{T\to\infty} \delta(T)$, which is finite by assumption.) If $t_n + \delta(T_n) > t'_n$, then add the event $\big(t_n + \delta(T_n), x_n\big)$ to the output list. If $t_n + \delta(T_n) \leqslant t'_n$, then delete the event (t'_n, x'_n) from the output list.

The output sequence's first event is always $(-\infty, x)$, its sequence of event times is strictly increasing, and its sequence of values is alternating.

If the input list is finite, the algorithm halts. If not, the output sequence nonetheless stabilizes in the sense that, for every time t, there exists some N such that all iterations with $n \geqslant N$ make no changes to the output sequence at times $\leqslant t$. This property makes the limit output list as n tends to infinity well-defined, and one may define the output signal by:

Definition 1. For input signal s, the output signal $c(s)$ of the forgetful single-history channel c is the signal whose event list is S after deleting all events with finite negative times and the first non-negative time event if its value is equal to the channel's initial value x.

B. Non-Forgetful Single-History Channels

The PID channel introduced by Bellido-Díaz et al. [2] is not covered by the above forgetful single-history channels, since it has been designed to reasonably match analog RC waveforms. Since the resulting exponential functions do not "forget" sub-threshold pulses, they cannot be modeled via delay functions $\delta(T)$ that depend on the input-to-previous output delay T. To also cover the PID model, we hence introduce non-forgetful single-history channels, the delay function of which may also depend on the last annihilated event.

The algorithm for non-forgetful channels thus maintains an additional variable r, which, in each iteration, contains the time of the *potential output event* considered in the last iteration. It was first presented by Bellido-Díaz et al. [2, Fig. 13]. Similar to the forgetful case, it determines the output signal $c(s)$ of a non-forgetful single-history channel c, given input signal s with corresponding input event list $\big((t_n, x_n)\big)_n$, as follows:

Initially, the output list contains $(-\infty, x)$ and $r = r_{-1} = -\infty$. In its nth iteration, the algorithm considers input event (t_n, x_n) and modifies the output list accordingly:

1) Equivalent to rule (1) of forgetful channels.
2) Otherwise, let $T_n = t_n - r_{n-1}$ be the difference of input and most recent potential output event times, and set $r_n = t_n + \delta(T_n)$. If $t_n + \delta(T_n) > r_{n-1}$, then add the event $\big(t_n + \delta(T_n), x_n\big)$ to the output list. If $t_n + \delta(T_n) \leqslant r_{n-1}$, then delete the event (t'_n, x'_n) from the output list.

Again, the output sequence's first event is always $(-\infty, x)$, its sequence of event times is strictly increasing, and its sequence of values is alternating. Moreover, the algorithm's final output list S is eventually stabilizing and hence well-defined, in the same sense as for forgetful channels, which finally allows to also carry over Definition 1 for the resulting output signal.

C. Examples of Single-History Channels

Below, we summarize how the existing binary-value models are mapped to our single-history channels:

A classical *pure-delay channel* is a single-history channel whose delay function δ is constant and positive. The behavior of a pure-delay channel does not depend on the fact whether it is forgetful or not.

An *inertial channel* is a forgetful single-history channel whose delay function δ is of the form

$$\delta(T) = \begin{cases} \delta_0 & \text{if } T > T_0 \\ -T_0 & \text{if } T \leqslant T_0 \end{cases}$$

for parameters $\delta_0 > 0$ and $T_0 > -\delta_0$. An inertial channel filters an incoming pulse if and only if its pulse length is less or equal to $T_0 + \delta_0$; otherwise, it is forwarded with delay δ_0.

The *PID-channels* of Bellido-Díaz et al. [2] are non-forgetful with delay function

$$\delta(T) = t_{p0} \cdot \left(1 - e^{-(T - T_0)/\tau}\right)$$

for certain (measured) positive parameters t_{p0}, τ, and T_0.

VII. Bounded Short-Pulse Filtration with a Single Non-Constant Delay Channel

In this section, we present our second major result, namely that bounded SPF is solvable as soon as there is a single non-constant delay single-history channel available. More specifically, we prove that, given a single-history channel with non-constant delay, there exists a circuit that uses only constant delay channels apart from the given non-constant channel and that solves bounded SPF. The proof is split into two parts, depending on whether the given non-constant channel is forgetful or not. The forgetful case allows an easier proof.

In the remainder of this section, let c^* be a single-history channel that is not a constant delay channel. What we actually require is that its delay function δ is non-constant for $T > -\delta(\infty)$, since smaller arguments cannot occur due to $T_n > -\delta(\infty)$ in every step of the channel algorithm. Without loss of generality, we can assume that the initial value of c^* is 0, as we could modify the circuits used in the subsequent proofs by negating signals at appropriate places otherwise.

A. Forgetful Channels

Consider circuit C_{ff}, obtained from the circuit shown in Fig. 2 by replacing the feedback channel with a constant delay channel with delay $\delta = \varepsilon$ and the channel connnecting the OR gate output and o with a forgetful channel c^*.

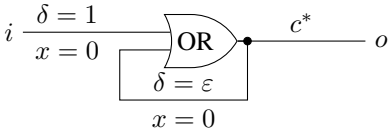

Fig. 6. Circuit C_{ff}

We will show that the storage loop C_{ff} shown in Fig. 6 solves bounded SPF. It remains to describe how to choose the delay parameter $\varepsilon > 0$. First, one can observe that for each forgetful single-history channel c, there exists some $\gamma(c) > 0$ such that $c(s)$ is the zero signal whenever s is a pulse of length less than $\gamma(c)$. Specifically, we can choose

$$\gamma(c) = \inf\big\{\Delta > 0 \mid \Delta - \delta(\infty) + \delta\big(\Delta - \delta(\infty)\big) > 0\big\} \quad .$$

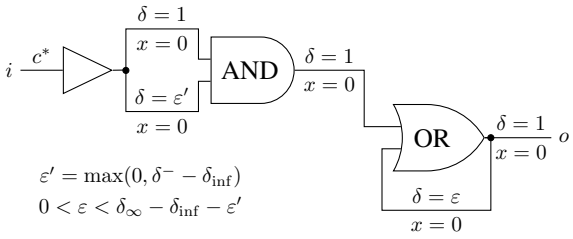

Fig. 7. Circuit C_{NC} used in Case 1

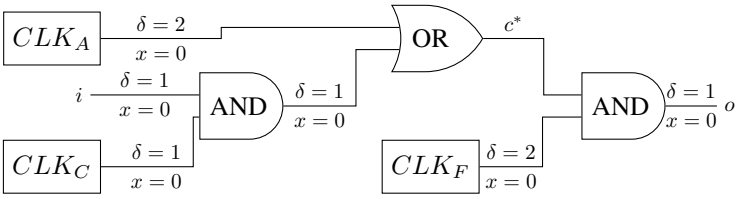

Fig. 8. Circuit C_{NF} used in Case 2

More generally, if signal s does not contain pulses of length greater or equal to $\gamma(c)$, then $c(s)$ is the zero signal, according to the following Lemma 3.

Lemma 3. *Let c be a non-constant delay forgetful single-history channel with initial value 0. Let s be a signal not containing pulses of length greater or equal to $\gamma(c)$ and that is not eventually continuously at 1. Then, $c(s)$ is the zero signal.*

Note that Lemma 3 does fundamentally *not* hold for general *non-forgetful* channels.

For the delay parameter ε in circuit C_{ff}, we choose $0 < \varepsilon < \gamma(c^*)$. If the input signal of C_{ff} is a pulse of length at least ε, then the signal s_{OR} at the output of the OR gate, and hence the circuit's output o, is eventually stable at 1 because of the ε-delay feedback loop. If the circuit's input signal is a pulse of length $\Delta < \varepsilon$, then s_{OR} only contains pulses of length $\Delta < \gamma(c^*)$, from which it follows by Lemma 3 that the circuit's output signal is always zero. Hence, we obtain:

Theorem 3. *Let c^* be a non-constant delay forgetful single-history channel. Then there exists a circuit solving bounded SPF whose channels are all constant delay channels or c^*.*

B. Non-Forgetful Channels

Theorem 4 reveals that a single non-constant delay *non-forgetful* single-history channel c^* also allows to solve bounded SPF:

Theorem 4. *Let c^* be a non-constant delay non-forgetful single-history channel. Then there exists a circuit solving SPF whose channels are all either constant delay channels or c^*.*

The proof idea is as follows: Denote by $\delta_\infty = \delta(\infty) = \lim_{t\to\infty} \delta(t)$ and by $\delta_{\inf} = \lim_{t\to 0^+} \delta(-\delta_\infty + t)$ the right-sided limit of δ at $-\delta_\infty$. Since c^* has non-constant delay, we have $\delta_{inf} < \delta_\infty$. We distinguish two cases, depending on the behavior of δ at $-\delta_{\inf}$:

1) $\delta(t) = \delta_\infty$ for all $t > -\delta_{\inf}$ and $\delta^- = \lim_{t\to 0^-} \delta(-\delta_{\inf} + t) < \delta_\infty$, i.e., δ is non-continuous at $-\delta_{\inf}$.
2) All other δ.

For Case (1), we can prove that circuit C_{NC} depicted in Fig. 7 solves bounded SPF. It is based on the following idea: Since one can show that c^* does not produce pulses of length within the non-empty interval $[\max(0, \delta^- - \delta_{\inf}), \delta_\infty - \delta_{\inf})$, it suffices to filters out all pulses with duration less than $\max(0, \delta^- - \delta_{\inf})$ (ensured by the AND gate) and continuously hold all pulses of length at least $\delta_\infty - \delta_{\inf}$ (done by the OR gate).

For Case (2), the more involved circuit C_{NF} depicted in Fig. 8 can be proved to solve bounded SPF. It uses three periodic clocks $CLK_{A/C/F}$, running at the same frequency but with different duty cycles, which can easily be built from constant delay channels and inverters. Their purpose is to separate time into consecutive *phases* A–F (where E and F are actually overlapping with A). The durations A–F of the corresponding phases are chosen in accordance with δ.

The clock period of all clocks is $A + B + C + D$, and the duty cycle of CLK_C is designed such that its output signal is 0 during $[\tau_k, \tau_k + A + B) \cup [\tau_k + A + B + C, \tau_{k+1})$ and 1 during $[\tau_k + A + B, \tau_k + A + B + C)$. Similarly, CLK_A's output is 1 during $[\tau_k, \tau_k + A)$ and 0 during $[\tau_k + A, \tau_{k+1})$, whereas CLK_F's output is 0 during $[\tau_k, \tau_k + E) \cup [\tau_k + E + F, \tau_{k+1})$ and 1 during $[\tau_k + E, \tau_k + E + F)$.

Time is divided into consecutive *rounds* $[\tau_k, \tau_{k+1})$ with $\tau_k = k(A + B + C + D)$. Setting $t_k = \tau_k + 2$, the phases of round k are $[t_k, t_k + A)$ (phase A), $[t_k + A, t_k + A + B)$ (phase B), $[t_k + A + B, t_k + A + B + C)$ (phase C), and $[t_k + A + B + C, t_k + A + B + C + D)$ (phase D); phase F is the interval $[t_k + E, t_k + E + F)$. The value of the output s_{OR} of the OR gate during phase A is always 1, and during phases B and D it is always 0. During phase C, it is either 0 or contains a pulse, depending on the input signal i. Note carefully that pulses at i can only show up at s_{OR} when they occur in phase C of some round.

The main arguments of the proof that C_{NF} solves bounded SPF are as follows: Properties (F1) and (F2) trivially hold. As for (F3), if the circuit's input signal is 0, then the channel's input signal s_{OR} is 0 during phase C of all rounds $k \geq 0$ as well. It can be proved (see below) that if this is the case, then the channel's output signal $s_{c^*(OR)}$ during phase F is 0 for all rounds $k \geq 0$. Since phase F is the only phase where o could possibly produce a non-0 output due to the AND gate, both (F3) and (F5) follow. Property (F4) is implied by the fact that there exists an input signal i such that s_{OR} contains a pulse during phase C of some round $k \geq 0$: If this is the case, one can show that c^*'s output signal is 1 during the entire phase F of some round. Finally, Property (F6) follows from the fact that all delays are bounded.

Figures 9 and 10 depict the input s_{OR} and the output

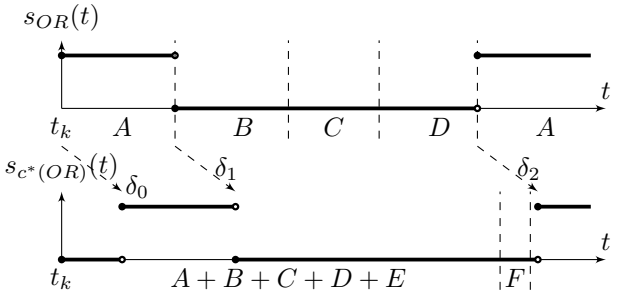

Fig. 9. Case 2: In- and Output of c^* in circuit C_{NF} without pulse in phase C

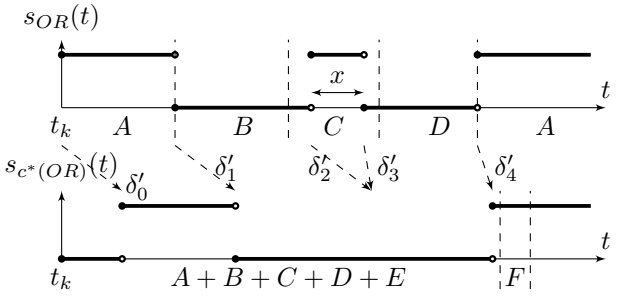

Fig. 10. Case 2: In- and Output of c^* in circuit C_{NF} with pulse in phase C

$s_{c^*(OR)}$ of the channel in the absence and in the presence of a pulse in phase C. By suitably choosing the phase durations A–F, it is guaranteed that if a pulse of length $x \leq C$ occurs during phase C (of round k), it is canceled, i.e., $x + \delta'_3 \leq \delta'_2$. However, the output event delayed by δ'_3 affects the delay δ'_4 of the next output event, no matter how short x was: Instead of being delayed by δ_2, it is delayed by $\delta'_4 < \delta_2$, thus being scheduled *before* the beginning of output phase F (of round $k+1$).

VIII. Impossibility of Bounded Short-Pulse Filtration in Physical Systems

Similar to Section V, we will now contrast the possibility of solving bounded SPF in our binary model with non-constant delay channels established in Theorems 3 and 4 by the impossibility of building such a circuit in reality, as proved in Theorem 6 below.

The proof is by reduction to the non-existence of a physical bistable storage element that stabilizes within bounded time in the model of [3]. A *(single-input) bistable element* is a physical circuit with input port i and output port o that, under the assumption that its input signal is either always 0 or a single positive pulse, fulfills (the physical variants of) Properties (F1) – (F4) of SPF as well as (F5'): If the output is logical 1 at some time t, it also remains logical 1 at all times larger than t. For a *(single-input) bistable element stabilizing within bounded time*, additionally (F6) has to hold. For the non-existence of such a circuit, we can utilize a classical result:

Theorem 5 (Marino [3, Theorem 3])**.** *For all single-input bistable elements, all times $t \geq 0$ and all times $T > 0$, there is a time $t_2 > t_1$ such that an input signal that is continuously 0 during time $[0, t_1)$, continuously 1 during time $[t_1, t_2)$, and again continuously 0 from time t_2 on (that is, a pulse), makes the bistable element's output o switch from logical 0 to logical 1 after time $t + T$.*

Now assume, for the sake of a contradiction, the existence a physical circuit solving bounded SPF and consider the circuit shown in Fig. 11, with the NOR's initial output equal to 1 and the inverter's initial output equal to 0 at time $t = 0$.

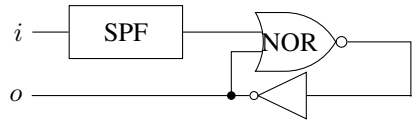

Fig. 11. Building a bistable storage element from a circuit solving SPF

It is not difficult to prove that this circuit implements a single-input bistable element stabilizing within bounded time: In case the input signal i is always 0, the SPF's output signal will always be logical 0 due to property (F3) of the SPF. Thus the circuit shown in Fig. 11 will always drive a logical 0 at its output, which confirms property (F3) for the bistable element.

Now let u be an input pulse i that makes the SPF circuit produce a logical 1 at its output, which exists due to property (F4) of the SPF. By definition, u is 0 during $[0, t_1)$ and $[t_2, \infty)$ and 1 during $[t_1, t_2)$ for some $t_1, t_2 \in \mathbb{R}_0^+$. Letting t' be the first time the SPF circuit drives a logical 1 at its output, its output must remain logical 1 within $[t', t' + \varepsilon]$ for some $\varepsilon > 0$ due to property (F5) of the SPF stated in Section V. Assuming that the signal propagation delay of the NOR gate and the inverter is short enough for the inverter's output to reach a logical 1 before time $t' + \varepsilon$, the NOR gate will subsequently drive a logical 0 on its output forever, irrespectively of the output of the SPF circuit. The circuit's output signal o will hence continuously remain logical 1 once it switched to logical 1, which also confirms Properties (F4) and (F6) of the bistable element.

Due to the usage of a circuit solving *bounded* SPF (F6) in the compound circuit, we further obtain that there exists some $T > 0$ such that, for any input pulse u' that switches to logical 1 by time t, the circuit shown in Fig. 11 produces a logical 1 by time $t + T$. This is a contradiction to the non-existence of a single-input bistable element stabilizing in bounded time.

We hence obtain our claimed result:

Theorem 6. *No physical circuit solves bounded SPF.*

IX. Conclusion

We have shown that binary circuit models using single-history channels fail to faithfully model glitch propagation. This includes all binary models known to date. Either channels have constant delay and SPF is not solvable, which is in contradiction to Newtonian reality. Or there is a non-constant delay channel and even bounded SPF is solvable, which is also

in contradiction to Newtonian reality. Future binary models that faithfully model glitch propagation hence cannot have the single-history property. This provides a signpost for future research on circuit models.

ACKNOWLEDGMENTS

We would like to thank an anonymous referee of the ASYNC'12 conference for pointing out the full impact of our results and thereby greatly improving their presentation.

This research was partially supported by the FATAL project (grant P21694) of the Austrian Science Fund (FWF).

REFERENCES

[1] S. H. Unger, "Asynchronous Sequential Switching Circuits with Unrestricted Input Changes," IEEE Trans. Computers, vol. 20, no. 12, pp. 1437–1444, Dec. 1971.

[2] 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 Proc. Circuits, Devices, and Systems, vol. 147, no. 2, pp. 107–117, Apr. 2000.

[3] L. R. Marino, "General Theory of Metastable Operation," IEEE Trans. Computers, vol. 30, no. 2, pp. 107–115, Feb. 1981.

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

[5] C. Constantinescu, "Trends and Challenges in VLSI Circuit Reliability," IEEE Micro, vol. 23, no. 4, pp. 14–19, July 2003.

[6] M. J. Gadlage et al., "Digital Device Error Rate Trends in Advanced CMOS Technologies," IEEE Trans. Nuclear Science, vol. 53, no. 6, pp. 3466–3471, Dec. 2006.

[7] M. S. Maza and M. L. Aranda, "Analysis of Clock Distribution Networks in the Presence of Crosstalk and Groundbounce," Proc. 8th Int'l IEEE Conf. Electronics, Circuits, and Systems (ICECS'01), pp. 773–776, Sep. 2001.

[8] L. R. Marino, "The Effect of Asynchronous Inputs on Sequential Network Reliability," IEEE Trans. Computers, vol. 26, no. 11, pp. 1082–1090, Nov. 1977.

[9] J. C. Barros and B. W. Johnson, "Equivalence of the Arbiter, the Synchronizer, the Latch, and the Inertial Delay," IEEE Trans. Computers, vol. 32, no. 7, pp. 603–614, July 1983.

[10] M. S. Branicky, "Universal Computation and Other Capabilities of Hybrid and Continuous Dynamical Systems," Theoretical Computer Science, vol. 138, no. 1, pp. 67–100, Feb. 1995.

[11] M. Mendler and T. Stroup, "Newtonian Arbiters Cannot Be Proven Correct," Formal Methods in System Design, vol. 3, no. 3, pp. 233–257, Dec. 1993.

[12] J. A. Brzozowski and J. C. Ebergen, "On the Delay-Sensitivity of Gate Networks," IEEE Trans. Computers, vol. 41, no. 11, pp. 1349–1360, Nov. 1992.

[13] M. Függer and U. Schmid, "Reconciling Fault-Tolerant Distributed Computing and Systems-on-Chip," Distributed Computing, vol. 24, no. 6, pp. 323–355, Jan. 2012.

[14] D. Dolev, M. Függer, C. Lenzen, and U. Schmid., "Fault-Tolerant Algorithms for Tick-Generation in Asynchronous Logic: Robust Pulse Generation (Extended Abstract)," Proc. 13th Int'l Symp. Stabilization, Safety, and Security of Distributed Systems (SSS'11), LNCS 6976, Springer, pp. 163–177, Oct. 2011.

[15] D. J. Kinniment, Synchronization and Arbitration in Digital Systems, Wiley, 2007.

[16] P. J. Ashenden, The Designer's Guide to VHDL, 3rd ed., Morgan Kaufmann, 2008.