



Unfaithful Glitch Propagation in Existing Binary Circuit Models

Matthias Függer, Thomas Nowak, Ulrich Schmid

► To cite this version:

Matthias Függer, Thomas Nowak, Ulrich Schmid. Unfaithful Glitch Propagation in Existing Binary Circuit Models. IEEE Transactions on Computers, 2016, 65 (3), pp.964-978. 10.1109/TC.2015.2435791 . hal-01231501

HAL Id: hal-01231501

<https://hal.science/hal-01231501>

Submitted on 20 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unfaithful Glitch Propagation in Existing Binary Circuit Models

Matthias Függer, Thomas Nowak, and Ulrich Schmid

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 channels (P), inertial delay channels (I), or the elaborate Delay Degradation Model (DDM) channels proposed by Bellido-Díaz et al. We accomplish our goal by considering the border between solvability and non-solvability 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 DDM. This is in opposition to the impossibility of solving bounded SPF in real (physical) circuit models. On the other hand, for binary circuit models with constant-delay channels, we prove that SPF cannot be solved even in unbounded time; again in opposition to physical circuit models. Consequently, indeed none of the binary value-domain models proposed so far (and that we are aware of) faithfully captures glitch propagation of real circuits. We finally show that these modeling mismatches do not hold for the weaker eventual SPF problem.

Index Terms—Circuit models, glitch propagation, binary models, modeling issues

1 INTRODUCTION

BINARY value-domain models that allow to model glitch propagation have always been of interest, especially in asynchronous design [22]: Pure delay channels 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 [3]. 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 [16].

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 Delay Degradation Model (DDM) proposed by Bellido-Díaz et al. [3], which is 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 recently been stimulated by the increasing importance of incorporating fault-tolerance in circuit design [7]: Reduced voltage swings and smaller critical charges make circuits more susceptible to particle hits, crosstalk, and electromagnetic interference [13], [17]. 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.

For example, the Byzantine fault-tolerant distributed clock generation approach DARTS [12] makes use of standard asynchronous circuit components, for example micropipelines [21], which store clock ticks received from other nodes; a new clock tick is generated when sufficiently many micropipelines are non-empty. Clearly, since any “wait-for-all” mechanism may deadlock in the presence of faulty components, handshaking had to be replaced by threshold logic in conjunction with some bounded delay assumptions. This way, DARTS can tolerate arbitrary behavior of Byzantine faulty nodes, except for the generation of pulses with a duration that drive the Muller C-elements of a pipeline into metastability. Analyzing the propagation of such pulses along a pipeline is thus important in order to assess the achievable resilience against such threats [11]. The situation is even worse in case of self-stabilizing algorithms [9], which must be able to recover from an arbitrary initial/error state: Neither handshaking nor any bounded delay condition can be resorted to during stabilization in an algorithm like the one presented by Dolev et al. [8]. Consequently, glitches and the possibility of metastability cannot be avoided.

This research was partially funded by the Austrian Science Fund (FWF) projects SIC (P26436) and RiSE (S11405).

- M. Függer is with the Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany. When this research was conducted, he was with the Embedded Computing Systems Group, Vienna University of Technology, 1040 Vienna, Austria. E-mail: mfuegger@mpi-inf.mpg.de
- T. Nowak is with the Département d’Informatique, École normale supérieure, 75230 Paris, France. When this research was conducted, he was with the Laboratoire d’Informatique, École polytechnique, 91128 Palaiseau, France. E-mail: thomas.nowak@ens.fr.
- U. Schmid is with the Embedded Computing Systems Group, Vienna University of Technology, 1040 Vienna, Austria. E-mail: s@ecs.tuwien.ac.at.

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 none of the existing binary-value candidate models proposed in the past captures glitch propagation adequately. Searching for alternative models is hence an important challenge for future research on asynchronous circuits.

Detailed contributions. In Section 2, we define the Short-Pulse Filtration (SPF) problem. It is essentially the problem of building a one-shot inertial channel, i.e., a channel that does not produce arbitrarily short pulses at its output, and is hence closely related to glitch propagation. We show that (unbounded) SPF is solvable in the physical circuit model of Marino [16] while bounded SPF is not. Our solvability result a fortiori carries over to the solvability of the weaker eventual SPF problem, which allows to produce arbitrarily short pulses during bounded time. These (im)possibility results are depicted in Fig. 1, in column “physical”.

In Section 3, we present a generic binary value-domain model for digital clocked and clockless circuits, and introduce the SPF problem. 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 be expressed by appending channels with delay at the gate’s inputs and outputs. The simplest channel is a pure delay channel, which propagates its input signal with a fixed delay and without any decay, i.e., a pulse has the same duration at the channel’s input and output.

In Section 4, we prove that even unbounded SPF is unsolvable when only pure, i.e., constant-delay channels are available (cf. Fig. 1, column “constant”). This contradicts the solvability of unbounded SPF with physical circuits established in Section 2.

In Section 5, we turn our attention to a generalization of constant-delay channels, termed *bounded 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. [3] (DDM), which additionally has a decay component, as a non-forgetful single-history channel.

In Section 6, we prove that bounded SPF is solvable if just a single forgetful or non-forgetful single-history channel with non-constant delay is available (cf. Fig. 1, columns “(non-)forgetful”). However, this is again in contradiction with the impossibility of implementing bounded SPF with physical circuits established in Section 2.

In Section 7, we prove that weakening SPF to eventual SPF fails to witness the above modeling mismatch: Eventual SPF can be solved both with single-history and physical

bounded SPF	X	✓	✓	X
SPF	X	✓	✓	✓
eventual SPF	✓	✓	✓	✓
	constant	forgetful	non-forgetful	physical

Fig. 1. Possibility (✓) and impossibility (X) results for constant, non-constant forgetful, non-const. non-forgetful, and physical circuit models. Arrows mark implications.

channels (cf. last row in Fig. 1).

Related Work. Unger [22] 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. [3] proposed the DDM model, and justified 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 [15] 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 [2] extended this work, by showing the equivalence of arbiter, synchronizer, latch, and inertial delay channels.

Marino [16] developed a general comprehensive theory of metastable operation, and provided impossibility proofs for metastability-free synchronizers and arbiter circuits for several continuous valued circuit models. Branicky [5] proved the impossibility to construct a time unbounded, deterministic, and time-invariant arbiter in an ordinary differential equations model. In a model based on continuous automata, this was studied by Mendler and Stroup [18].

Brzozowski and Ebergen [6] 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.

2 SPF IN PHYSICAL CIRCUITS

In this section, we will introduce the SPF problem in the model of Marino [16] and use the classic results obtained for bistable elements to determine the solvability/impossibility border of the SPF problem for real (physical) circuits. It turns out that SPF is solvable with a physical circuit if it is allowed to have an unboundedly large reaction time, but that SPF is not solvable if bounded reaction time is demanded.

The model of Marino 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 a single input and a single output solves *Short-Pulse Filtration (SPF)*, if it fulfills the following requirements:

- (i) If the input signal is constantly logical 0, then so is the output signal.
- (ii) There exists an input signal such that the output signal attains logical 1 at some point in time.
- (iii) There exists some fixed $\varepsilon > 0$ such 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' . Informally, this condition prohibits output signals that may be interpreted as pulses (as defined in Section 3) with a duration less than ε .

A physical circuit solves *bounded SPF* if additionally:

- (iv) There exists a time T such that, if the input signal switches to logical 1 by time t , then the output signal value is either logical 0 or logical 1 at time $t + T$ and remains logical 0 respectively logical 1 thereafter.

We will next argue why there is no physical circuit that solves bounded SPF, but that there are physical circuits solving unbounded SPF.

2.1 Impossibility of Bounded SPF

The proof is by reduction to the non-existence of a physical bistable storage element that stabilizes within bounded time in the model of Marino [16]. A *single-input bistable element* is a physical circuit with a single input and a single output that fulfills properties (i) and (ii) of SPF as well as:

- (iii') 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 (iv) has to hold.

The following Corollary 1, which proves the non-existence of a single-input bistable element that stabilizes within bounded time, follows immediately from Theorem 3 in [16].

Corollary 1. *There is no single-input bistable element stabilizing within bounded time.*

Now assume, for the sake of a contradiction, that there existed a physical circuit solving bounded SPF and consider the circuit shown in Fig. 2, with the NOR's initial output equal to 1 and the inverter's initial output equal to 0 at time $t = 0$.

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 logical 0, the SPF's output signal will always be logical 0 due to property (i) of the SPF. Thus the circuit shown in Fig. 2 (left) will always drive a logical 0 at its output, which confirms property (i) for the bistable element.

Now let u be an input pulse that makes the SPF circuit produce a logical 1 at its output. Letting t' be the first time

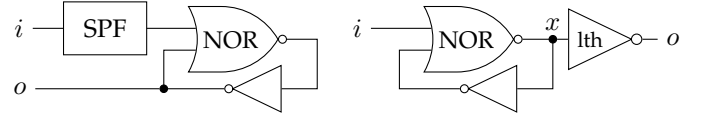


Fig. 2. Left: Building a bistable storage element from a circuit solving SPF. Right: A circuit solving SPF with a low-threshold inverter.

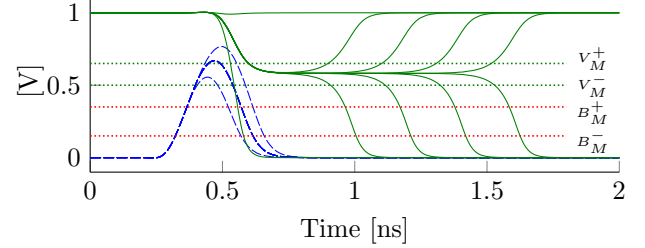


Fig. 3. Input signals at i (dashed blue) and corresponding signals at x (solid green) for the SPF implementation shown in Fig. 2 on the right.

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 (iii) of the SPF. 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, irrespective 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 (ii) and (iii') of the bistable element.

Due to the use of a circuit solving bounded SPF 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. 2 (left) produces a logical 1 by time $t + T$, a contradiction to the non-existence of a single-input bistable element stabilizing in bounded time. We hence obtain:

Theorem 1. *No physical circuit solves bounded SPF.*

2.2 Possibility of Unbounded SPF

To show the existence of a circuit solving unbounded SPF, we make use of a circuit known as a metastability filter (see, e.g., [14, p. 40]). We will show that the circuit in Fig. 2 (right), comprising of a storage loop and a subsequent low-threshold inverter as a metastability filter, solves SPF. According to Marino [16], for every storage loop there exist pulses that drive the internal state of the storage loop (x in Fig. 2 on the right) into a metastable region for an unbounded time. However, it is possible to determine safe bounds V_M^- and V_M^+ around this metastable region such that whenever the storage loop output x leaves $[V_M^-, V_M^+]$, it settles to stable values 0 or 1 within bounded (and short) time. Fig. 3 shows input signals i and corresponding signals x of the SPF implementation in Fig. 2 (right). Note that, although there exist input signals that drive x into a metastable state arbitrarily long, whenever x leaves $[V_M^-, V_M^+]$, it converges quickly to a stable state. Specifically, signal x crosses the marked region $[B_M^-, B_M^+]$ below $[V_M^-, V_M^+]$ at most once and

remains within only for bounded time. We may use this fact and place a low-threshold inverter after x driving output o as in Fig. 3 (right). The low-threshold inverter is designed in a way such that input voltages within $[0, B_M^-]$ are mapped to logical 1 at its output, and input voltages within $[B_M^+, 1]$ to logical 0. We thus obtain a physical circuit that solves (unbounded) SPF. Hence:

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

3 BINARY SYSTEM MODEL

We consider a binary-valued signal model with continuous time, i.e., signal values are from $\mathbb{B} = \{0, 1\}$ that evolve over time $\mathcal{T} = [0, \infty)$.

A *signal* is a function $\mathcal{T} \rightarrow \mathbb{B}$ 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 $e = (t, x)$ is a pair 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 allow for the simple but rigorous handling of initial values of channels in our framework. An *event list* is a (finite or infinite) sequence of events. To every signal, there corresponds an event list $(e_n)_{n \geq 0} = ((t_n, x_n))_{n \geq 0}$ with the following properties:

- S1) There is always an initial event at time $-\infty$.
- S2) The sequence $(t_n)_{n \geq 0}$ of event times is strictly increasing and discrete.
- S3) Values are alternating: $x_n \neq x_{n+1}$

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

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 (positive) constant-delay channels. A *constant-delay* channel c with delay parameter $\delta > 0$ and initial value $x \in \mathbb{B}$ produces at its output the input signal delayed by δ , i.e.,

$$c(s)(t) = \begin{cases} x & \text{if } t < \delta \\ s(t - \delta) & \text{if } t \geq \delta \end{cases} \quad (1)$$

Note that a physical realization of a constant-delay channel with initial value x requires a multiplexer, which supplies the constant-delay channel's input with the initial value x during $(-\infty, 0)$ and switches to the actual input s at the system initialization (reset) time 0.

Circuits are obtained by interconnecting a set of input ports and a set of output ports, forming the external interface of a circuit, and an arbitrary set of zero-time combinational gates via single-input single-output channels that model circuit delays. 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 $C = (G, I, O, c, n)$, where

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 fundamental to our model and, thus, our results.

- 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 $\mathbb{B}^{d_b} \rightarrow \mathbb{B}$, where d_b is the in-degree, i.e., the number of incoming neighbors, of b . By a slight abuse of notation, b also denotes the Boolean 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 $v \in I$ (*input ports*) has in-degree $d_v = 0$.
- C5) Every $v \in O$ (*output ports*) has in-degree $d_v = 1$.
- C6) n is a function that maps every vertex v in G to a linearly ordered subset $n_v = \{v_1, \dots, v_{d_v}\}$ 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 .

Note that there are also zero-input Boolean gates 0 and 1 that represent constant signal values 0 and 1.

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 hold: 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 and $c_{v,o}$ the channel representing edge (v, o) . Let now b be a Boolean gate with d incoming neighbors v_1, v_2, \dots, 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,

$$s_b(t) = b(c_{v_1,b}(s_{v_1})(t), \dots, c_{v_d,b}(s_{v_d})(t))$$

for all $t \in \mathcal{T}$.

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 the "mirror channel" c with $c(s) = s$ for all signals s 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 constant-delay channels, this is always the case. (We will prove this in Lemma 2.)

Short-Pulse Filtration. A *pulse* p of length $\Delta > 0$ at time T is a signal of the form

$$p(t) = \begin{cases} 0 & \text{if } t < T \text{ or } t \geq T + \Delta \\ 1 & \text{if } T \leq t < T + \Delta \end{cases} \quad (2)$$

A signal *contains a pulse* of length $\Delta > 0$ at time T if its event list contains the two consecutive 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 constant 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 constant 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 the following condition holds:

- F6) There exists a $K > 0$ such that, in all executions with a pulse as the input signal whose last event is at time T , the output signal does not change anymore after time $T + K$. (*Bounded stabilization time*)

A circuit *solves eventual SPF* if conditions (F1)–(F4) and the following condition hold:

- F5e) There exists an $\varepsilon > 0$ and a $K > 0$ such that, in all executions with a pulse at time T as the input signal, the output signal does not contain a pulse of length less than ε after time $T + K$. (*Eventually no short pulses*)

Note that we do not require that the signal is eventually 1 forever to solve SPF, but only for longer than the minimal pulse duration ε . However, starting from a circuit solving SPF according to our definition, one can easily construct a circuit solving the stronger version with strict stabilization by adding a storage loop at the end.

4 UNSOLVABILITY OF SPF WITH CONSTANT DELAY CHANNELS

In this section, we show 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 that 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 SPF.

4.1 Dependency Graphs

For each constant-delay circuit with a single input port and a single output port, we introduce its *dependency graph*, which describes the way the output signals may depend on the input signals. Its nodes are either the constant Boolean values 0 and 1 or of the form $\langle v, \tau \rangle$, signifying that the signal value at gate v at time $t - \tau$ may influence the output signal at time t .

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 *dependency graph* $DG(t)$ of C at time t is a directed graph with vertices $\langle v, \tau \rangle$, where v is a vertex in G and τ a time, or from \mathbb{B} . It is defined as follows:

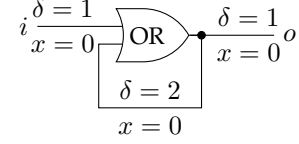


Fig. 4. Example circuit

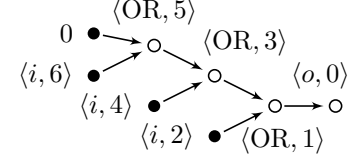


Fig. 5. Example dependency graph $DG(6)$

- The pair $\langle o, 0 \rangle$ is a vertex of $DG(t)$.
- If $\langle v, \tau \rangle$ is a vertex of $DG(t)$ and (u, v) is an edge in G such that $\tau + \delta(u, v) \leq t$, then the pair $\langle u, \tau + \delta(u, v) \rangle$ is also a vertex of $DG(t)$ and there is an edge in $DG(t)$ from $\langle u, \tau + \delta(u, v) \rangle$ to $\langle v, \tau \rangle$.
- If $\langle v, \tau \rangle$ is a vertex of $DG(t)$ and (u, v) is an edge in G such that $\tau + \delta(u, v) > t$, then $c_{u,v}$'s initial value $x(u, v)$ is a vertex of $DG(t)$ and there is an edge in $DG(t)$ from $x(u, v)$ to $\langle v, \tau \rangle$.

Because all $\delta(u, v)$ are strictly positive, the dependency 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 $\langle i, \tau \rangle$, 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).

As an example, consider the circuit shown in Fig. 4. The dependency graph $DG(6)$ is shown in Fig. 5. 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 (input) signal values $s_i(4)$, $s_i(2)$, and $s_i(0)$. Thus, in particular, $s_o(6)$ is the same for signals $s_i = \text{in}_1$ and $s_i = \text{in}_2$ depicted in Fig. 6: Indeed, the signal values agree for all three measure points $\langle i, 6 \rangle$, $\langle i, 4 \rangle$, and $\langle i, 2 \rangle$, i.e., $\text{in}_1(6 - 6) = \text{in}_1(0) = \text{in}_2(0) = 0$, $\text{in}_1(6 - 4) = \text{in}_1(2) = \text{in}_2(2) = 1$, and $\text{in}_1(6 - 2) = \text{in}_1(4) = \text{in}_2(4) = 0$.

Generalizing the observations from the example, by following paths in the dependency graph, we thus observe:

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 ,*

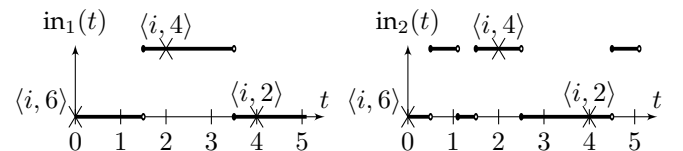


Fig. 6. Two input signals with measure points (\times), labeled with corresponding input leaf names, leading to the same output value at time 6

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 $s_o(t) = \tilde{s}_o(\tilde{t})$.

Proof: For a path π in G , denote by $\delta(\pi)$ the sum of delays $\delta(u, v)$ over all edges (u, v) of π . For every vertex v of G and every time $t \in \mathcal{T}$, let $\mathcal{P}(\rightarrow v, t)$ be the set of maximum length paths π ending in v such that $\delta(\pi) \leq t$.

It is clear, by iterating Eq. (1), that the value of $s_v(t)$ is uniquely determined by the collection of values $s_u(t - \delta(\pi))$ where u is the start vertex of $\pi \in \mathcal{P}(\rightarrow v, t)$. Moreover, by maximality of π , if $u \neq i$, then $s_u(t - \delta(\pi))$ only depends on the initial values of channels of incoming edges to u . Hence $s_v(t)$ is uniquely determined by the collection of values $s_i(t - \delta(\pi))$ where $\pi \in \mathcal{P}(\rightarrow v, t)$ starts at i . This holds in particular for $v = o$. \square

This lemma has as an immediate consequence:

Lemma 2. *If C is a circuit with only constant-delay channels, then for all assignments of input signals $(s_i)_{i \in I}$ there exists a unique execution of C extending this assignment.*

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

Proof: Let $\varepsilon > 0$ be strictly smaller than all positive values of the form $\delta(u, v) + \tau - t$ where (v, τ) is an intermediate vertex of $DG(t)$ and (u, v) is an edge in G . If no such intermediate vertex or edge exists, choose $\varepsilon > 0$ arbitrarily.

Let (v, τ) be an intermediate vertex of $DG(t)$ and (u, v) be an edge in G . If $t + \varepsilon - \tau < \delta(u, v)$, then clearly $t - \tau < \delta(u, v)$, because $\varepsilon > 0$. On the other hand, if $t - \tau < \delta(u, v)$, then $\delta(u, v) + \tau - t$ is positive and hence $\delta(u, v) > t + \varepsilon - \tau$ by choice of ε . Thus, the conditions $t - \tau < \delta(u, v)$ and $t + \varepsilon - \tau < \delta(u, v)$ are equivalent. This shows that the two dependency graphs $DG(t)$ and $DG(t + \varepsilon)$ and hence all dependency graphs in between are equal. \square

4.2 Unsolvability Proof

Assume by contradiction that 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 p of some length and a time t such that the corresponding output signal's value at time t is 1.

By Lemma 3, 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 .

Clearly, $DG(\tilde{t}) = DG(t)$ for all times \tilde{t} between t and $t + \varepsilon$, in particular, for $\tilde{t} = t + \varepsilon/2$. Denote by Δ the infimum of input pulse lengths (where all pulses start at the same time) 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} . By definition of the infimum Δ , there exists an input pulse p with the above property of length at most $\Delta + \varepsilon/4$. We show that its corresponding output signal s_p contains a pulse of length strictly less than ε , in contradiction to the no short pulses property (F5).

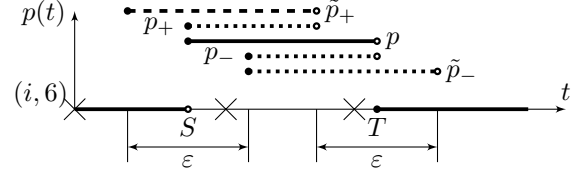


Fig. 7. Input pulse p , together with its derived pulses and measure points for time \tilde{t}

Denote by S the time of p 's upwards and by T the time of p 's downwards transition. Now let p_+ be the pulse whose upwards transition is at time S and whose downwards transition is at time $T - \varepsilon/2$. If $S \geq T - \varepsilon/2$, then let p_+ be the zero signal instead. The length of p_+ is either strictly less than Δ or it is the zero signal. Hence, by the definition of Δ and the no-generation property (F3), its corresponding output signal's value at time \tilde{t} is 0. This implies that there exists a measure point for time \tilde{t} within $[T - \varepsilon/2, T)$, because p and p_+ coincide everywhere else (see marked measure point on the right in Fig. 7).

Because we chose ε to be smaller than all differences of distinct measure points for time t (and hence also for time \tilde{t}), we see that there is no measure point for \tilde{t} in the interval $[T, T + \varepsilon/2)$.

Likewise, by defining p_- as the pulse with upwards transition at time $S + \varepsilon/2$ and downwards transition at time T , we infer that there is one measure point for time \tilde{t} in the interval $[S, S + \varepsilon/2)$ and there is no measure point for \tilde{t} in the interval $[S - \varepsilon/2, S)$ (see Fig. 7).

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

Likewise, by considering p shifted into the future by $\varepsilon/2$, we see that also $s^{\tilde{p}_+}(\tilde{t} - \varepsilon/2) = 0$. 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, this concludes the proof.

5 BOUNDED SINGLE-HISTORY CHANNELS

This section formally introduces the notion of bounded single-history channels 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 bounded single-history channel propagates each event, occurring at time t , of the input signal to an event at the output happening after some bounded *output-to-input* delay $\delta(T)$, which depends on the *input-to-previous-output* delay $T = t - t'$ where t' is the most recent output event. Note that T is positive if the channel delay is short compared to the input signal transition times, and negative otherwise. In case FIFO order would be invalidated, i.e., $t + \delta(T) \leq t'$, such that the next output event would not occur after the previous one, both events annihilate.

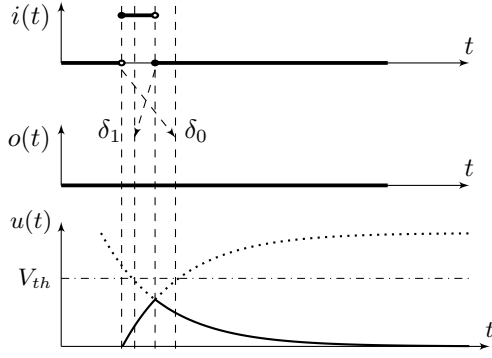


Fig. 8. Annihilating events and analog interpretation: the binary input signal triggers a switching curve. When the switching curve crosses a threshold level, a binary output event is generated.

Viewed from the perspective of an analog circuit model, annihilating events represent a signal change that ultimately does not reach the switching threshold, and hence do not result in a new (digital) event, see the example shown in Fig. 8: Let us assume that the event list of a channel input signal i is $((-\infty, 0), (1, 1), (2, 0))$, i.e., represents a pulse of length 1 starting at time 1. For the rising transition at time 1, the most recent output transition occurred at $-\infty$. Thus $T = 1 - (-\infty) = \infty$, and the transition is mapped to the (potential) output transition at time $1 + \delta_0 = 1 + \delta(\infty)$. In the analog model, this corresponds to the channel reaching the threshold voltage at time $1 + \delta_0$. The next (falling) transition at time 2 is mapped to a (potential) output transition at time $2 + \delta_1 = 2 + \delta(2 - (1 + \delta_0)) = 2 + \delta(1 - \delta_0)$. In our example, $2 + \delta_1 < 1 + \delta_0$, so both potential output transitions annihilate. In the analog model, this is reflected by the fact that the output voltage did not reach the threshold voltage.

There are two natural variants of bounded single-history channels that differ in the definition of the time of the previous output when determining the input-to-previous-output delay: one may either take the time of the last non-annihilated output event, or of the last output event even if it was annihilated. We dub these two variants *forgetful* and *non-forgetful* bounded single-history channels. In terms of the example in Fig. 8, a positive input transition occurring at time $t > 2$, i.e., after the shown input pulse, would have parameter $T = t - (-\infty)$ in the forgetful and parameter $T = t - (2 + \delta_1)$ in the non-forgetful case. We give formal definitions of both variants below.

At the end of this section, we give a list of channel models that are special cases of our definition of bounded single-history channels. It reveals that all the binary-value channel models proposed in literature are instances of either forgetful or non-forgetful bounded single-history channels.

Formally, a *bounded single-history channel* c is characterized by an initial value $x \in \mathbb{B}$, a nondecreasing delay function $\delta : \mathbb{R} \rightarrow \mathbb{R}$ such that $\delta(\infty) = \lim_{T \rightarrow \infty} \delta(T)$ is finite and positive, and the fact whether it is forgetful or not. In the rest of the paper, we will drop the qualifier “bounded” when referring to bounded single-history channels. We detail the channel behavior in the next two subsections.

5.1 Forgetful Single-History Channels

This class of channels includes the classical inertial delay channels as used, for example, in VHDL simulators [1].

Their behavior is defined by the following algorithm: Let s be a signal. In case the channel’s initial value x_{init} is equal to the initial value of s , or if there is an event at time 0 in the event list of s , let the channel’s *input list* $((t_n, x_n))_{n \geq 0} = ((t_0, x_0), (t_1, x_1), \dots)$ be the event list of s . Otherwise, let the channel’s input list be the event list of s with an additional event at time 0 and value equal to the initial value of s . Algorithm 1 iterates the input list and updates the *output list*, which will define the channel’s output signal $c(s)$. Note that $T = \infty$ is possible. In this case, $\delta(T) = \delta(\infty) = \lim_{T \rightarrow \infty} \delta(T)$, which is finite by assumption.

Algorithm 1 Forgetful channel algorithm

```

1: output  $\leftarrow ((-\infty, x_{\text{init}}))$ 
2: for  $n = 1$  to  $\text{length}(\text{input})$  do
3:    $(t, x) \leftarrow (t_n, x_n)$ 
4:    $(t', x') \leftarrow$  last event of output
5:   if  $x \neq x'$  then
6:      $T \leftarrow t - t'$ 
7:     if  $t + \delta(T) > t'$  then
8:       add  $(t + \delta(T), x)$  to output
9:     else
10:      remove  $(t', x')$  from output
11:    end if
12:  end if
13: end for

```

Note that the output sequence’s first event is always the initial event $(-\infty, x)$, all other events have positive times (since $\delta(\infty) > 0$), 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 $n \geq N$ make no changes to the output sequence at times $\leq t$. The next lemma (Lemma 4) proves this property and makes the limit output list as n tends to infinity well-defined. So, even if the input list is infinite, there exists a well-defined (infinite) output list S that is the result of the described algorithm. The channel’s output signal $c(s)$ is then defined by the event list S :

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 the list S as defined by the Algorithm 1.

The following lemma shows that the output list eventually stabilizes, by proving that events that are too far back in time cannot interfere with the currently considered input event.

Lemma 4. Denote by S_n the output list after the n th iteration of the forgetful channel algorithm, and by $S_n|t$ its restriction to the events at times at most t . For all t there exists an N such that $S_n|t$ is constant for all $n \geq N$.

Proof: The lemma is trivial if the input list is finite, so we assume it to be infinite.

Because the sequence of input event times $(t_n)_{n \geq 0}$ tends to infinity, there exists an N such that

$$t_N \geq \max(t, t - \delta(-\delta(\infty))) . \quad (3)$$

The choice of the constant $-\delta(-\delta(\infty))$ can be explained by the fact that the current output event is removed if and only

if $\delta(T) \leq -T$ and that we always have $T > -\delta(\infty)$. We show by induction that $S_n|t = S_N|t$ for all $n \geq N$. This is trivial for $n = N$, so let $n > N$, which implies $t_n > t_N$.

Let (t', x') be the last element in S_{n-1} , and $T = t_n - t'$. The case $x_n = x'$ is trivial, so let $x_n \neq x'$. We distinguish two cases, depending on whether $\delta(T) > -T$ or not:

Case 1: $\delta(T) > -T$. Because δ is nondecreasing, we have $\delta(T) \leq \delta(\infty)$, and hence $T > -\delta(\infty)$ and also $\delta(T) \geq \delta(-\delta(\infty))$. This implies $t_n + \delta(T) > t_N + \delta(-\delta(\infty)) \geq t$ by using (3). Hence $S_n|t = S_{n-1}|t = S_N|t$ by the induction hypothesis.

Case 2: $\delta(T) \leq -T$. We show that $t' > t$ by contradiction: Let $t' \leq t$. Then $T = t_n - t' > t_N - t \geq 0$, by (3). From $\delta(\infty) > 0$, we thus obtain $T > -\delta(\infty)$. Hence $\delta(T) \geq \delta(-\delta(\infty))$ by monotonicity of δ . By assumption, $\delta(-\delta(\infty)) \leq \delta(T) \leq -T = t' - t_n$, which implies $t_n \leq t' - \delta(-\delta(\infty))$, i.e., $t_N < t - \delta(-\delta(\infty))$. This is a contradiction to (3), which shows that $t' > t$. Hence $S_n|t = S_{n-1}|t = S_N|t$ by the induction hypothesis. \square

5.2 Non-Forgetful Single-History Channels

The DDM channels introduced by Bellido-Díaz et al. [3] are not covered by the above forgetful single-history channels, since they have been designed to reasonably match analog RC waveforms: Analog signals like exponential functions do not “forget” sub-threshold pulses: If an input pulse (train) takes the output voltage very close to the switching threshold (but not above), then even a very short subsequent pulse can cause a threshold crossing. Moreover, the output voltage does not drop to zero immediately (cf. the falling waveform in the bottom part of Fig. 8) in the absence of a subsequent pulse. Hence, DDM channels cannot be modeled via delay functions $\delta(T)$ that depend on the input-to-previous *non-annihilated* output delay T . To also cover such models, we hence introduce non-forgetful single-history channels, the delay function of which may depend on the last *annihilated* output event.

The output event list generation 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. Intuitively, the variable r retains some memory of the analog signal voltage, translated into the time domain; note that this approach was already employed in the DDM event generation algorithm [3, Fig. 13]. Similar to the forgetful case, our Algorithm 2 determines the output signal $c(s)$ of a non-forgetful single-history channel c , given input signal s with input event list $((t_n, x_n))_{n \geq 0}$.

Lemma 5. *If an event (t', x') is deleted in line 11 of Algorithm 2, then $r = t'$.*

Proof: Assume by contradiction that this is not the case, and let n be the first iteration where the statement is violated. Denote by t'_k, T_k and r_k the values of t', T and r at the end of the k^{th} iteration, respectively. Our assumption is $t_n + \delta(T_n) < r_{n-1}$.

Then it must hold that $n \geq 2$, as in iteration $n - 2$ some event (τ, x_{n-2}) must have been added to the output list that was deleted in iteration $n - 1$, due to $\tau' = t_{n-1} + \delta(T_{n-1}) \leq r_{n-2} = \tau$. Furthermore, in iteration n , our assumption of

Algorithm 2 Non-forgetful channel algorithm

```

1: output  $\leftarrow ((-\infty, x_{\text{init}}))$ 
2:  $r \leftarrow -\infty$ 
3: for  $n = 1$  to length(input) do
4:    $(t, x) \leftarrow (t_n, x_n)$ 
5:    $(t', x') \leftarrow$  last event of output
6:   if  $x \neq x'$  then
7:      $T \leftarrow t - r$ 
8:     if  $t + \delta(T) > r$  then
9:       add  $(t + \delta(T), x)$  to output
10:    else
11:      remove  $(t', x')$  from output
12:    end if
13:     $r \leftarrow t + \delta(T)$ 
14:  end if
15: end for

```

deleting some event with a time different from $r_{n-1} = \tau'$ implies $\tau'' = t_n + \delta(T_n) \leq \tau'$. However, from $t_{n-1} < t_n$, $\tau \geq \tau'$ and monotonicity of δ , $t_{n-1} + \delta(t_{n-1} - \tau) < t_n + \delta(t_n - \tau')$, i.e., $\tau' < \tau''$, which provides the required contradiction. \square

Thus, an event is either deleted in the next iteration, or never deleted. The output sequence's first event $(-\infty, x)$ is obviously never deleted.

By analogous arguments, one can show that the sequence of event times is strictly increasing, with an alternating sequence of values. Unlike in the case of forgetful channels, however, the event list generation algorithm may produce events that occur at finite *negative* times, which will be removed from the final output. If the input list is finite, the algorithm clearly halts. If not, we again have the same stabilization property as for forgetful single-history channels, which we will provide in Lemma 6 below. Thus the algorithm's final output list S is again well-defined and we can define:

Definition 2. For input signal s , the output signal $c(s)$ of the non-forgetful single-history channel c is the signal whose event list is the list S as defined by Algorithm 2 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 .

The next lemma shows that the output list eventually stabilizes for non-forgetful channels. Surprisingly, its proof is more direct than of the analogous statement for forgetful channels.

Lemma 6. *Denote by S_n the output list after the n -th iteration of the non-forgetful channel algorithm, and by $S_n|t$ its restriction to the events at times at most t . For all t , there exists an N such that $S_n|t$ is constant for all $n \geq N$.*

Proof: The lemma follows from the fact that an event can only be deleted one iteration after it was added to the output list, and the fact that in each iteration n , we have $T > -\delta(\infty)$ and thus $t_n + \delta(T)$ is lower bounded by $t_n + \lim_{\tau \rightarrow 0^+} \delta(-\delta(\infty) + \tau)$. \square

5.3 Examples of Single-History Channels

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

- 1) A classic *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 whether it is forgetful or not.
- 2) An inertial channel is a forgetful single-history channel whose delay function δ is of the form

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

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 .

- 3) The DDM-channels of Bellido-Díaz et al. [3] are non-forgetful with delay function

$$\delta(T) = t_{p0} \cdot \left(1 - e^{-(T-T_0)/\tau}\right) \quad (4)$$

for certain (empirically determined) positive parameters t_{p0} , τ , and T_0 . Note that we have $\delta(T_0) = 0$, $\lim_{t \rightarrow \infty} \delta(T) = t_{p0}$, and $\frac{d\delta(T)}{dT}|_{T=0} = t_{p0}/\tau$ here.

6 BOUNDED SPF WITH ONE NON-CONSTANT DELAY CHANNEL

In this section we prove that bounded SPF is solvable as soon as there is a single non-constant-delay bounded single-history channel available. More specifically, we show 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 that solves bounded SPF. Different circuits, and hence proofs, are used for different types of channels.

For a single-history channel with delay function δ , let $\delta_\infty = \delta(\infty) = \lim_{t \rightarrow \infty} \delta(t)$ with $0 < \delta_\infty < \infty$. The right limit of δ at $-\delta_\infty$ is denoted by $\delta_{\text{inf}} = \lim_{t \rightarrow 0^+} \delta(-\delta_\infty + t)$; note that $\delta_{\text{inf}} = -\infty$ is allowed here.

In the rest of this section, let c^* be a single-history channel that is not a constant-delay channel. This is equivalent to saying that its delay function δ is non-constant for $T > -\delta_\infty$, because $T_n > -\delta_\infty$ in every step of the channel algorithm. This argument proves the following lemma:

Lemma 7. *A single-history channel with delay function δ is a constant-delay channel if and only if δ is constant in the open interval $(-\delta_\infty, \infty)$.*

Note that $\delta_{\text{inf}} < \delta_\infty$ in case of a non-constant delay channel. From the fact that $-\delta_\infty < T_n \leq \infty$ in every step of the channel algorithm, we also obtain:

Lemma 8. *All events in the event list of a single-history channel's input signal are delayed by times within $[\delta_{\text{inf}}, \delta_\infty]$.*

6.1 Forgetful Channels

In this subsection, assume that c^* is forgetful. Consider circuit C_{ff} depicted in Fig. 9, which contains channel c^* as well as two constant-delay channels. For the moment assume that the initial value of c^* is 0. We will show at the end of this subsection that bounded SPF is also solvable with c^* if its initial value is 1.

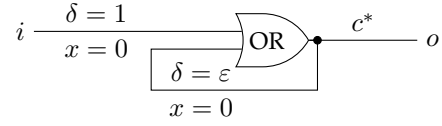


Fig. 9. Circuit C_{ff}

It remains to describe how to choose delay parameter $\varepsilon > 0$. We will show in the following that for each non-constant-delay forgetful single-history channel c there exists a $\gamma(c) > 0$ such that $c(s)$ is the zero signal whenever s is a pulse of length less than $\gamma(c)$. More generally we will show that, if signal s does not contain pulses of length greater or equal to $\gamma(c)$, then $c(s)$ is the zero signal. We then choose $0 < \varepsilon < \gamma(c^*)$ for the delay parameter ε in circuit C_{ff} .

If the input signal of circuit C_{ff} is a pulse of length at least ε , then the signal s_{OR} at the OR gate is eventually stable 1 because of the ε -delay feedback loop, and hence the circuit's output signal is eventually stable 1. 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 that the circuit's output signal is zero.

Let δ be the delay function of a single-history channel c . We define:

$$\gamma(c) = \inf \{ \Delta > 0 \mid \Delta - \delta_\infty + \delta(\Delta - \delta_\infty) > 0 \} \quad (5)$$

We will prove $\gamma(c^*) > 0$ in Lemma 10. Before characterizing the non-constant-delay channels as those c with $\gamma(c) > 0$, we need a preliminary lemma on pulse-filtration properties of non-constant-delay channels. It shows that very short pulses are suppressed by explicitly running the channel algorithm for one pulse.

Lemma 9. *Let c be a non-constant-delay bounded single-history channel with initial value 0. If s is a pulse of length less than $\gamma(c)$, then $c(s)$ is zero.*

Proof: The event list of signal s consists of two events $(R, 1)$ and $(S, 0)$, possibly preceded by an additional event $(0, 0)$, depending on whether $R = 0$ or $R > 0$. Because the initial value of c is 0, we may assume without loss of generality that the sequence consists of only these two events.

After the first iteration of Algorithm 1, the output list is equal to $((-\infty, 0), (R + \delta_\infty, 1))$. Hence, in the next iteration,

$$T = S - R - \delta_\infty < \gamma(c) - \delta_\infty,$$

i.e., $T + \delta_\infty < \gamma(c)$. By definition of $\gamma(c)$, this implies

$$(T + \delta_\infty) - \delta_\infty + \delta((T + \delta_\infty) - \delta_\infty) \leq 0,$$

and thus $T + \delta(T) \leq 0$. Thus, the event $(R + \delta_\infty, 1)$ gets removed from the output list and the output signal is the constant-zero signal. \square

We are now ready to prove the characterization of constant single-history channels in terms of its ability to suppress some pulse and the value of $\gamma(c)$. The starting point in the proof is to note that the pulse's rising transition is delayed by δ_∞ and that $t' = \delta_\infty$ and $T = \Delta - \delta_\infty$ in the following iteration of the algorithm if Δ denotes the pulse length. The falling transition annihilates the rising

transition if and only if $\delta(T) \leq t'$. In other words, if $\Delta + \delta(\Delta - \delta_\infty) \leq \delta_\infty$. It then suffices to compare this with the definition of $\gamma(c)$.

Lemma 10. *Let c be a single-history channel with initial value 0. The following statements are equivalent:*

- 1) c is not a constant-delay channel.
- 2) There exist a pulse s such that $c(s)$ is the zero signal.
- 3) $\gamma(c) > 0$

Proof: Let δ be the delay function of c . If s is a pulse of length Δ , then $c(s)$ is zero if and only if

$$\Delta - \delta_\infty + \delta(\Delta - \delta_\infty) \leq 0.$$

This implies $\gamma(c) \geq \Delta$ and hence establishes the equivalence of (2) and (3). If we can show that c is not a constant-delay channel if and only if

$$\exists \varepsilon > 0 : \delta(-\delta_\infty + \varepsilon) \leq \delta_\infty - \varepsilon, \quad (6)$$

then we can choose $\Delta = \varepsilon$, concluding the proof.

The sufficiency of Eq. (6) for c not being a constant-delay channel is immediate. To prove the necessity of Eq. (6), assume that c is not a constant-delay channel. Then there exist $\beta, \beta' > 0$ such that $\delta(\beta - \delta_\infty) < \delta(\beta' - \delta_\infty)$ and since δ is nondecreasing, $\delta(\beta - \delta_\infty) < \delta_\infty$. Thus, there exists a $z > 0$, such that,

$$\delta(\beta - \delta_\infty) \leq \delta_\infty - z. \quad (7)$$

There are two cases for z : If $\beta \leq z$, we obtain from Eq. (7) that $\delta(\beta - \delta_\infty) \leq \delta_\infty - \beta$. Choosing $\varepsilon = \beta$ shows that Eq. (6) holds. Otherwise, i.e., if $\beta > z$, we obtain from Eq. (7) and the fact that δ is nondecreasing

$$\delta(z - \delta_\infty) \leq \delta(\beta - \delta_\infty) \leq \delta_\infty - z.$$

Choosing $\varepsilon = z$ shows that Eq. (6) holds. \square

Note that, while Lemmas 9 and 10 hold for both forgetful and non-forgetful single-history channels, the following lemma does fundamentally not hold for arbitrary non-forgetful channels. It is the generalization of Lemma 9 to pulse trains, which is obvious due to the forgetting nature of forgetful channels.

Lemma 11. *Let c be a non-constant-delay bounded forgetful single-history channel with initial value 0. Let s be a signal that does not contain pulses of length greater or equal to $\gamma(c)$ and that is not eventually equal to 1. Then $c(s)$ is the zero signal.*

Proof: The lemma is proved by inductively repeating the proof of Lemma 9 for all pulses contained in s . \square

Using Lemma 11, we can now prove that circuit C_{ff} solves bounded SPF by checking off the requirements.

Lemma 12. *Circuit C_{ff} solves bounded SPF.*

Proof: We first note that, given an input signal, there is a unique execution for circuit C_{ff} according to Lemma 2, because the sole non-constant bounded channel c^* is not part of a feedback loop. The well-formedness properties (F1) and (F2) of SPF are hence fulfilled. The non-generation property (F3) is also obvious.

If the input signal is a pulse of length at least ε , then $s_{\text{OR}}(t) = 1$ for all $t \geq S + 1$, and hence $s_o(t) = 1$

for all $t \geq S + 1 + \delta^*(\infty)$. In particular, this shows the nontriviality property (F4).

If the input signal is a pulse of length less than ε , then $s_{\text{OR}}(t)$ only contains pulses of lengths less than ε , hence less than $\gamma(c^*)$ by the choice of ε . By Lemma 11, the output signal is zero in this case. This, together with the above, shows (F5) and (F6). \square

It remains to show that assuming c^* to have initial value 0 is not restricting: If its initial value is 1 we modify circuit C_{ff} by adding an inverter before and after channel c^* . A proof analogous to Lemma 12's yields:

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

6.2 Non-Forgetful Channels

Theorem 4 reveals that a single non-constant-delay non-forgetful single-history channel c^* (with initial value 0) also allows to solve bounded SPF:

Theorem 4. *Let c^* be a non-constant-delay non-forgetful bounded single history channel with initial value 0. Then there exists a circuit solving SPF whose channels are all either constant-delay channels or c^* .*

6.2.1 Proof of Theorem 4: Case Distinction

Let δ be the delay function of c^* . Recall from Lemma 7 that $\delta_{\text{inf}} < \delta_\infty$, since δ is non-decreasing and not constant. We distinguish three cases for function δ with respect to its behavior at $-\delta_{\text{inf}}$.

1. There exists a $t > -\delta_{\text{inf}}$ such that $\delta(t) < \delta_\infty$.
2. $\delta(t) = \delta_\infty$ for all $t > -\delta_{\text{inf}}$, and
 - 2.1 δ is continuous at $-\delta_{\text{inf}}$, i.e., at $-\delta_{\text{inf}}$ its left limit $\lim_{t \rightarrow 0^-} \delta(-\delta_{\text{inf}} + t)$ equals its right limit δ_∞ .
 - 2.2 δ is not continuous at $-\delta_{\text{inf}}$, that is, we have $\delta^- = \lim_{t \rightarrow 0^-} \delta(-\delta_{\text{inf}} + t) < \delta_\infty$.

Note that this covers all possible cases since $\delta(t) \leq \delta_\infty$ for all $t \in \mathbb{R}$ by monotonicity.

6.2.2 Cases 1 and 2.1

For Cases 1 and 2.1, we show that circuit C_{NF} depicted in Fig. 12 solves bounded SPF. All its clocks $CLK_{A/C/F}$ produce a signal with period $A + B + C + D$, where parameters A to D are chosen later on in accordance with δ . Let $\tau_k = k(A + B + C + D)$ denote the beginning of the k -th round, for $k \geq 0$. Clock 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 $[\tau_k + A + B, \tau_k + A + B + C)$. Such a clock can easily be built from constant-delay channels and inverters only. Clock CLK_A 's output signal is 1 during $[\tau_k, \tau_k + A)$ and 0 during $[\tau_k + A, \tau_{k+1})$. The output signal of CLK_F is 0 during $[\tau_k, \tau_k + E) \cup [\tau_k + E + F, \tau_{k+1})$ and 1 during $[\tau_k + E, \tau_k + E + F)$. Again, E and F are chosen later on in accordance with δ .

Abbreviating $t_k = \tau_k + 2$, we observe that circuit C_{NF} generates a signal s_{OR} at the input of channel c^* , which is the OR of two sub-signals that consist of four phases

within time $[t_k, t_{k+1})$, $k \geq 0$ (i.e., per round): Phase A (of round k) denotes the interval of times $[t_k, t_k + A)$, phase B the interval $[t_k + A, t_k + A + B)$, phase C the interval $[t_k + A + B, t_k + A + B + C)$ and phase D the interval $[t_k + A + B + C, t_k + A + B + C + D)$. The value of s_{OR} is 1 during phase A , and 0 during phases B and D . During phase C it is either 0 or contains a pulse, depending on signal i . Analogously, we define output phase F (of round k) as the interval of times $[t_k + E, t_k + E + F)$. Note that phase E and F of round k follow phase D of round k , and overlap with phase A of round $k + 1$.

Informally, for Cases 1 and 2.1, circuit C_{NF} solves bounded SPF according to the following reasoning: Properties (F1) and (F2) trivially hold for circuit C_{NF} . Clearly, if the circuit's input signal is 0, then the channel's input signal s_{OR} is 0 during phase C of all rounds $k \geq 0$. Subsequently, we will prove that if this is the case, then the channel's output signal $c^*(s_{OR})$ 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 $k \geq 0$. We will prove below that if this is the case, then the channel's output signal is 1 during phase F of round $k + 1$. Essentially, this follows from a reduced delay of the rising transition at the end of phase D , caused by not forgetting the (canceled) pulse in phase C . From this and the fact that all delays are bounded, (F6) follows.

Case 1. In this case, we will show that it is sufficient to choose

- (i) $C > 0$, $D > 0$ and $0 < \Delta < \delta_\infty$ such that $\delta(C + D - \delta_{\inf}) \leq \delta_\infty - \Delta$. Such values for C , D and Δ exist, because of the assumption of Case 1.
- (ii) $\varepsilon > 0$, $\varepsilon' > 0$ and $C > 0$ small enough such that $\delta_\infty - \varepsilon' \geq \delta_{\inf} + \varepsilon + C$ and $\varepsilon' < \Delta/4$.
- (iii) $C > 0$ and $\varepsilon' > 0$ small enough such that $\delta(C + \varepsilon' - \delta_\infty) \leq \delta_{\inf} + \varepsilon$.
- (iv) $A = B > \max(\varepsilon', \Delta, \delta_\infty - \delta_{\inf})$ and large enough such that $\delta(A - \delta_\infty) \geq \delta_\infty - \varepsilon'$.
- (v) $E = \delta_\infty - \Delta$ and $F = \Delta/2$.

It is not too hard to check that Assumptions (i)–(v) are compatible with each other.

Figures 10 and 11 depict signal s_{OR} in absence and presence of a pulse. The next three lemmas prove formally the behavior of channel c^* shown in these two figures. Lemmas 13 and 14 verify Figure 10, i.e., that there is no output pulse in phase F if there is no input pulse in phase C . Lemma 15 then shows the accuracy of Figure 11, i.e., that the output is 1 during the whole duration of phase F if there is an input pulse in phase C . The proofs essentially are an explicit execution of the channel algorithm and making sure that the parameters were chosen correctly.

Lemma 13. *The channel's output signal $c^*(s_{OR})$ has value 0 during output phase F of round 0.*

Proof. The signal is depicted in Fig. 10: Signal s_{OR} 's transition to value 1 at time t_0 is delayed by c^* by $\delta_0 = \delta_\infty > 0$. Its next transition back to value 0 at time $t_0 + A$

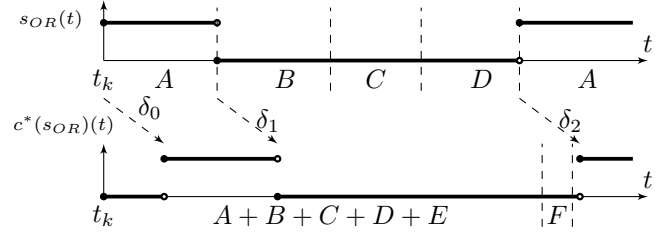


Fig. 10. Case 1: Input and output of channel c^* in circuit C_{NF} if phase C does not contain a pulse

is delayed by, say, δ_1 . Because of Lemma 8, $\delta_1 \geq \delta_{\inf}$. From this and Assumption (iv) on A ,

$$A + \delta_1 > (\delta_\infty - \delta_{\inf}) + \delta_{\inf} = \delta_0 .$$

It follows that output $c^*(s_{OR})$'s transition to 0 does not cancel $c^*(s_{OR})$'s transition to 1 from before. All of s_{OR} 's following transitions occur at times at least $t_0 + A + B$, and by (iv), at times greater than $t_0 + \delta_\infty - \delta_{\inf}$. Since all these transitions are delayed by at least δ_{\inf} time, none of them can cancel $c^*(s_{OR})$'s transition to 1 at time $t_0 + \delta_\infty$ either. Since channel c^* has initial value 0, it follows that its output has value 0 during $[0, t_0 + \delta_\infty)$. Since

$$t_0 + \delta_\infty > t_0 + \delta_\infty - \Delta/2 = t_0 + E + F ,$$

the channel's output indeed has value 0 during output phase F of round 0. \square

Lemma 14. *If signal s_{OR} does not contain a pulse within phase C of round k , signal $c^*(s_{OR})$ has value 0 during output phase F of round $k + 1$.*

Proof. Assume the input signal s_{OR} of channel c^* does not contain a pulse within phase C of round k . The signal is depicted in Fig. 10.

Signal s_{OR} 's transition to value 1 at time t_k is delayed by c^* by $\delta_0 \leq \delta_\infty$.

There is no transition of s_{OR} before s_{OR} 's transition back to value 0 at time $t_k + A$. Let δ_1 be its delay. Because of (iv), and δ being non-decreasing, $A + \delta_1 > (\delta_\infty - \delta_{\inf}) + \delta_{\inf}$. Thus, and because transitions are delayed by at least δ_{\inf} , none of the transitions from time $t_k + A$ on may cancel $c^*(s_{OR})$'s transition to 1 at time $t_k + \delta_0$.

The transition of s_{OR} to value 1 at time $t_{k+1} = t_k + A + B + C + D$ is delayed by δ_2 , where

$$\delta_2 = \delta(B + C + D - \delta_1) \geq \delta(B - \delta_\infty) \geq \delta_\infty - \varepsilon' , \quad (8)$$

because of Assumption (iv). Together with (ii) this yields

$$\delta_2 > \delta_\infty - \Delta/4 . \quad (9)$$

It will thus not occur at output $c^*(s_{OR})$ before time $t_{k+1} + \delta_\infty - \Delta/4$, and thus, by (v), not before the end of output phase F of round $k + 1$ at time $t_{k+1} + \delta_\infty - \Delta/2$.

Furthermore, from (8) and (iv),

$$B + C + D + \delta_2 > \delta_\infty \geq \delta_1 ,$$

because (iv) in particular implies $B > \varepsilon'$. It follows that output $c^*(s_{OR})$'s transition to 1 does not cancel $c^*(s_{OR})$'s transition to 0 at time $t_k + A + \delta_1$. All s_{OR} 's subsequent transitions occur at earliest at time $t_{k+1} + A > t_{k+1} + \delta_\infty -$

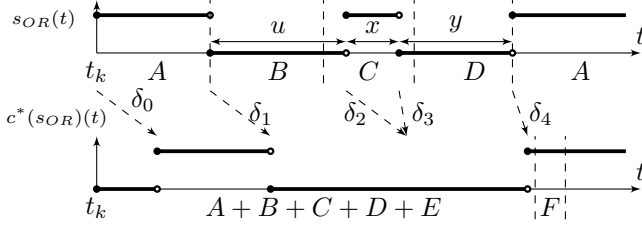


Fig. 11. Case 1: Input and output of channel c^* in circuit C_{NF} if phase C contains a pulse

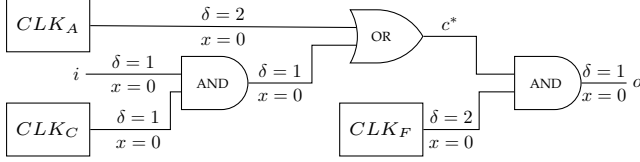


Fig. 12. Circuit C_{NF} used in Cases 1 and 2.1

δ_{\inf} , by (iv) and the fact that they are delayed by at least δ_{\inf} , hence cannot cancel $c^*(s_{OR})$'s transition to 1 at time $t_{k+1} + \delta_2$. Thus, $c^*(s_{OR})$ has value 0 during $[t_k + A + \delta_1, t_{k+1} + \delta_2]$. Together with (9), this implies that $c^*(s_{OR})$'s value is 0 during phase F of round $k + 1$. \square

By analogous means we derive the following lemma, whose proof can be found in the supplementary material.

Lemma 15. *If signal s_{OR} contains a pulse within phase C of round k , signal $c^*(s_{OR})$ has value 1 during output phase F of round $k + 1$.*

The combination of Lemmas 13, 14, and 15 now proves Theorem 4 in Case 1.

Case 2.1. In this case, we may choose

- (i) $A = D > \max(0, \delta_{\infty} - \delta_{\inf})$ and large enough such that $\delta(A - \delta_{\infty}) = \delta_{\infty}$. Such an A must exist, because of the assumption of Case 2.1.
- (ii) $B, C, \varepsilon > 0$ small enough such that $B + C + \varepsilon + \delta_{\inf} \leq \delta_{\infty}$.
- (iii) $0 < \varepsilon' < B + C$
- (iv) $\varepsilon > 0$ small enough such that $\delta(-\delta_{\inf} - \varepsilon) \geq \delta_{\infty} - \varepsilon'$. Such a value exists, since δ is continuous at $-\delta_{\inf}$ by the assumption of Case 2.1.
- (v) $B + C > 0$ small enough such that $\delta(B + C - \delta_{\infty}) \leq \delta_{\inf} + \varepsilon$.
- (vi) $E = A + \delta_{\infty}$ and $F = B + C - \varepsilon'$.

Again, it is not hard to verify that Assumptions (i)–(vi) are compatible with each other.

Figures 13 and 14 depict signal s_{OR} in the absence and presence of a pulse. The next lemma does the same job in Case 2.1 as Lemmas 13, 14, and 15 did in Case 1: It proves that the output signal of channel c^* is either all 0 or all 1 during phase F , depending on whether there is an input pulse in phase C . The proof is again an explicit execution of the channel algorithm, and is provided in the supplementary material.

Lemma 16. *Signal s_{OR} 's transition at time t_k is delayed by δ_{∞} , and the channel's output $c^*(s_{OR})$ has value 0 during phase F of round k in the absence of a pulse within phase C of round k , and value 1 in the presence of a pulse.*

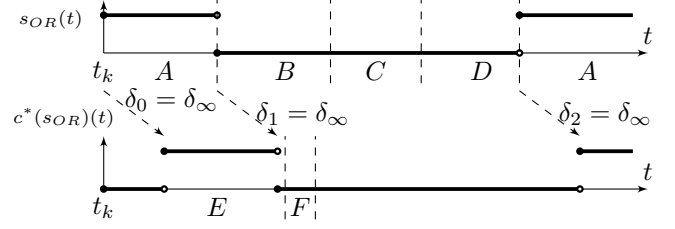


Fig. 13. Case 2.1: Input and output of channel c^* in circuit C_{NF} if phase C does not contain a pulse

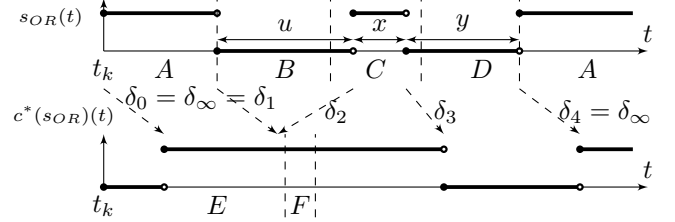


Fig. 14. Case 2.1: Input and output of channel c^* in circuit C_{NF} if phase C contains a pulse

6.2.3 Case 2.2

For this case, circuit C_{NC} depicted in Fig. 15 solves bounded SPF. The algorithm and its proof rest on the following idea: We first show in Lemma 17 that every channel c^* whose δ is in accordance with Case 2.2 does not produce pulses of length within the non-zero interval $[\max(0, \delta^- - \delta_{\inf}), \delta_{\infty} - \delta_{\inf}]$. The remaining part of circuit C_{NC} thus just has to filter out all pulses with duration less than $\max(0, \delta^- - \delta_{\inf})$ (AND gate) and continuously hold all pulses of length $\delta_{\infty} - \delta_{\inf}$ (OR gate). We thus obtain the following key lemma:

Lemma 17. *Let c^* be a non-constant-delay non-forgetful channel chosen in accordance to Case 2.2. If the channel's input signal is a pulse, then its output signal is either 0 or a pulse whose length is not within the non-zero interval $[\max(0, \delta^- - \delta_{\inf}), \delta_{\infty} - \delta_{\inf}]$.*

Its proof can be found in the supplementary material.

If we choose the circuit parameters in Fig. 15 according to $\varepsilon' = \max(0, \delta^- - \delta_{\inf})$ and $0 < \varepsilon < \delta_{\infty} - \delta_{\inf} - \varepsilon'$, it is not difficult to show that the resulting circuit C_{NC} solves bounded SPF in Case 2.2: Properties (F1) to (F3) trivially hold for circuit C_{NC} . To prove (F4), consider that if the input signal i is a pulse of length $2\delta_{\infty}$, the output signal $s_{c^*(i)}$ of c^* is a pulse of length at least δ_{∞} . Thus, the output of the AND gate s_{AND} is a pulse of length at least $\delta_{\infty} - \varepsilon' > \varepsilon$, resulting in the circuit's output o making a transition to 1 and remaining 1 from there on.

Property (F5) directly follows from Lemma 17: If $s_{c^*(i)}$ is a pulse of length smaller than $\max(0, \delta^- - \delta_{\inf}) = \varepsilon'$, then

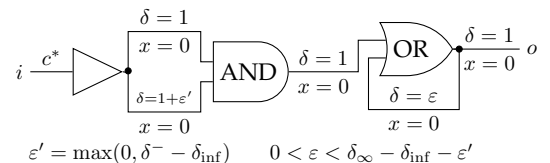


Fig. 15. Circuit C_{NC} used in Case 2.2

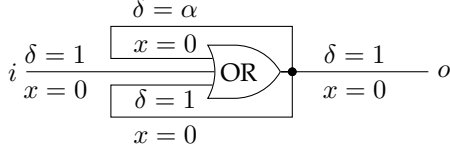


Fig. 16. Circuit C_{ev} solving eventual SPF

it is completely filtered out; s_{AND} and hence o are hence permanently 0. Otherwise, by Lemma 17, $s_{c^*(i)}$ must be a pulse of length at least $\delta_\infty - \delta_{inf}$. Thus, s_{AND} is a pulse of length at least $\delta_\infty - \delta_{inf} - \varepsilon' > \varepsilon$, which is sufficiently long to be permanently captured in the storage looped formed by the OR gate. The circuit's output o hence makes a transition to 1 and remains 1 from there on.

Finally, (F6) is due to bounded channel delays.

7 EVENTUAL SPF WITH CONSTANT DELAYS

We proved that SPF is not solvable with constant-delay channels. In this section, we consider the weaker eventual SPF problem, which drops the “no short pulses” requirement (F5) and replaces it with its eventual analog (F5e). We show that eventual SPF is solvable using only constant-delay channels. More specifically, we prove that circuit C_{ev} in Fig. 16 solves eventual SPF. The circuit contains a delay parameter α , which we will choose to be a positive irrational like $\alpha = \sqrt{2}$.

We will show that the circuit's output is eventually stable at 1 whenever the input is a pulse of positive length. We derive a bound on this stabilization time in terms of the input pulse length Δ . The bound is almost linear in $1/\Delta$: It is in the order of $O(\Delta^{-1-\varepsilon})$ for all $\varepsilon > 0$.

The measure points of circuit C_{ev} for time t are of the form $t - (\alpha k + \ell) - 2$, where k and ℓ are nonnegative integers. This is easy to see by considering the dependency graph for circuit C_{ev} . Indeed, summing the δ 's of the possible paths from the input i to output o of length L gives the values $2 + (\alpha k + \ell)$ where $k + \ell = L - 2$. We can hence characterize the circuit's behavior with the following lemma.

Lemma 18. *In every execution (s_v) of circuit C_{ev} , the following are equivalent: (i) $s_o(t) = 1$, and (ii) there exist nonnegative integers k and ℓ such that $s_i(t - (\alpha k + \ell) - 2) = 1$.*

We may restrict our considerations to input pulses starting at time 0. In the following, let the input signal s_i be a pulse of length $\Delta > 0$. We are looking for the *stabilization time*, which is the minimal time $T = T(\Delta)$ such that, for all $t \geq T$, we have $s_o(t) = 1$.

To prove finiteness and effective bounds on the stabilization time, we relate it to the number-theoretic concept of *discrepancy* of the sequence (αn) modulo 1 (see, e.g., [10]). The discrepancy compares the number of sequence elements in a given interval with their expected number if the elements were uniformly distributed.

For a given nonempty sub-interval $(x, y]$ of $(0, 1]$ and a given positive integer N , let $A(x, y; N)$ denote the number of αn 's with $n \leq N$ that lie in the interval modulo 1: $\alpha n \in (x, y] + \mathbb{Z}$. The expected number of such αn 's is $(y - x)N$. The discrepancy $D_N(\alpha)$ is then defined as the maximum

difference between $A(x, y; N)$ and $(y - x)N$, formed over all nonempty sub-intervals $(x, y]$ of $(0, 1]$.

It is well-known that $D_N(\alpha)/N \rightarrow 0$ if and only if α is irrational. Also, if α has a bounded continued fraction expansion, then $D_N(\alpha) = O(\log N)$ and the constant can be computed [20]. This is, in particular, true for $\alpha = \sqrt{2}$.

The next lemma makes the link between the circuit's stabilization time for an input pulse of length Δ and a number-theoretic parameter $K(\Delta)$. Its proof is an elementary reformulation using Lemma 18.

Lemma 19. *Let $K = K(\Delta)$ be the least integer K such that for all real t there exists an integer k , $0 \leq k \leq K$, with $\alpha k \in (t - \Delta, t] + \mathbb{Z}$. Then, $T(\Delta) \leq \alpha \cdot K(\Delta) + \Delta + 2$.*

Proof: The lemma is trivial if $K = \infty$, so assume the contrary.

Let $t \geq \alpha K + \Delta + 2$. By the definition of K , there exists a k with $0 \leq k \leq K$ and an ℓ such that $t - \Delta - \ell - 2 < \alpha k \leq t - \ell - 2$, which is equivalent to $0 \leq t - (\alpha k + \ell) - 2 < \Delta$.

By Lemma 18, it remains to prove that ℓ is nonnegative. The inequality $t - (\alpha k + \ell) - 2 < \Delta$ is equivalent to $\ell > t - \Delta - \alpha k - 2$. Noting $-\alpha k \geq -\alpha K$ and $t \geq \alpha K + \Delta + 2$ shows $\ell > 0$ and concludes the proof. \square

We now make the connection to the notion of discrepancy with the following lemma, which is proved by just using the definitions in a straightforward way.

Lemma 20. *Let $0 < \Delta \leq 1$. If $D_N(\alpha)/N < \Delta/2$, then $K(\Delta) \leq N$.*

Proof: Suppose the contrary, i.e., that there exists a real t such that, for all $n \leq N$, we have $\alpha n \notin (t - \Delta, t] + \mathbb{Z}$. Let $0 < x < y \leq z < u \leq 1$ such that we can decompose the interval $(t - \Delta, t] + \mathbb{Z} = ((x, y] + \mathbb{Z}) \cup ((z, u] + \mathbb{Z})$ modulo 1. None of the two intervals $(x, y]$ and $(z, u]$ contains an αn modulo 1 with $n \leq N$. Hence $A(x, y; N) = A(u, z; N) = 0$, which implies $2D_N(\alpha) \geq (y - x)N + (u - z)N = \Delta N$, a contradiction. \square

Finally, we prove that eventual SPF is solvable with constant delay channels by using the fact that $D_N(\alpha)/N$ tends to zero as $N \rightarrow \infty$ whenever α is irrational.

Theorem 5. *Circuit C_{ev} solves eventual SPF if α is irrational. If $\alpha = \sqrt{2}$, the stabilization time satisfies $T(\Delta) = O(\Delta^{-1-\varepsilon})$ as $\Delta \rightarrow 0$ for all $\varepsilon > 0$.*

Proof: (F1) and (F2) are obviously fulfilled. Because all initial values of channels are 0, also (F3) holds. Because $D_N(\alpha)/N \rightarrow 0$ whenever α is irrational, for all $\Delta > 0$, there exists some N such that $D_N(\alpha)/N < \Delta/2$. Hence Lemma 20 and Lemma 19 show that $T(\Delta)$ is finite, which shows (F4) and (F5e).

We now prove the bound on the stabilization time. Let $\gamma = -1 - \varepsilon < -1$. There exists a $C_1 > 0$ such that $D_N(\alpha) \leq C_1 \log N$. Because $1 + 1/\gamma > 0$, there exists a $C_2 > 0$ such that $\log N < C_2 N^{1+\gamma}$. Thus if $N \geq \left(\frac{\Delta}{2C_1 C_2}\right)^\gamma$, then $D_N(\alpha)/N \leq C_1 \log N/N < C_1 C_2 N^{1/\gamma} < \Delta/2$, which, by Lemma 20, implies $K(\Delta) \leq (\Delta/2C_1 C_2)^\gamma + 1$ for all $0 < \Delta \leq 1$. That is, $K(\Delta) = O(\Delta^\gamma)$ as $\Delta \rightarrow 0$.

It is easy to see that $K(\Delta) \rightarrow \infty$ as $\Delta \rightarrow 0$. Hence Lemma 19 implies $T(\Delta) = O(K(\Delta))$ as $\Delta \rightarrow 0$. \square

8 CONCLUSION

We showed that binary circuit models using bounded single-history channels, hence all binary models known to date, fail to faithfully model glitch propagation: In the case of constant-delay channels, SPF turned out to be unsolvable, which is in contradiction to physical reality. In case of non-constant-delay channels, even bounded SPF is solvable, again in contradiction to physical reality. Binary models aiming at faithful glitch propagation hence cannot be bounded single-history. The difficulty in the search for appropriate models is that they must not only faithfully capture glitch propagation but should also accurately cover (analog) pulse deformation effects; at the same time, it should have low computational complexity and also facilitate manual or even automated verification of circuits (which rules out non-analytic, e.g., table-based, approaches).

We hope that our results provide a signpost for future research on adequate binary circuit models: As confirmed by the fact that the weaker eventual SPF problem is already solvable with constant-delay channels, SPF is well suited for capturing the peculiarities of glitch propagation while not being overly restrictive. Moreover, in the proofs of our core results, we actually used weaker properties than guaranteed by single-history channels. It may hence be possible to reuse part of those for weaker channel models.

Since any candidate for a faithful delay function must invalidate our impossibility proofs for bounded SPF, part of our current research on this topic is devoted to and in-depth analysis of the properties of the delay functions required by these proofs. We are convinced that suitable delay functions can be identified by invalidating these required properties.

REFERENCES

- [1] P.J. Ashenden, *The Designer's Guide to VHDL*, 3rd ed., Morgan Kaufmann, 2008.
- [2] 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.
- [3] M.J. Bellido-Díaz et al., "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.
- [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] 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.
- [6] 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.
- [7] C. Constantinescu, "Trends and Challenges in VLSI Circuit Reliability," *IEEE Micro*, vol. 23, no. 4, pp. 14–19, July 2003.
- [8] D. Dolev et al., "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.
- [9] S. Dolev, *Self-Stabilization*, MIT Press, 2000.
- [10] M. Drmota and R.F. Tichy, *Sequences, Discrepancies, and Applications*, Springer, 1997.
- [11] G. Fuchs, M. Függer, and A. Steininger, "On the Threat of Metastability in an Asynchronous Fault-Tolerant Clock Generation Scheme," *Proc. 15th IEEE Int'l Symp. Asynchronous Circuits and Systems (ASYNC'09)*, pp. 127–136, May 2009.
- [12] M. Függer and U. Schmid, "Reconciling Fault-Tolerant Distributed Computing and Systems-on-Chip," *Distributed Computing* 24(6), 2012, pp. 323–355.

- [13] 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.
- [14] D.J. Kinniment, *Synchronization and Arbitration in Digital Systems*, Wiley, 2007.
- [15] L.R. Marino, "The Effect of Asynchronous Inputs on Sequential Network Reliability," *IEEE Trans. Computers*, vol. 26, no. 11, pp. 1082–1090, Nov. 1977.
- [16] L.R. Marino, "General Theory of Metastable Operation," *IEEE Trans. Computers*, vol. 30, no. 2, pp. 107–115, Feb. 1981.
- [17] 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.
- [18] 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.
- [19] F.U. Rosenberger et al., "Q-Modules: Internally Clocked Delay-Insensitive Modules," *IEEE Trans. Computers*, vol. 37, no. 9, pp. 1005–1018, Sep. 1988.
- [20] J. Schoissengeier, "The Discrepancy of $(n\alpha)_{n \geq 1}$," *Mathematische Annalen*, vol. 296, no. 1, pp. 529–545, Dec. 1993.
- [21] I.E. Sutherland, "Micropipelines," *Comm. ACM*, vol. 32, no. 6, pp. 720–738, June 1989.
- [22] S.H. Unger, "Asynchronous Sequential Switching Circuits with Unrestricted Input Changes," *IEEE Trans. Computers*, vol. 20, no. 12, pp. 1437–1444, Dec. 1971.



Matthias Függer received his M.Sc. (2006) and PhD (2010) in computer engineering from TU Wien. He worked as an assistant professor at TU Wien, as a post-doctoral researcher at LIX, Ecole polytechnique, and currently as a post-doctoral researcher at the Max-Planck-Institut für Informatik. His main research interest is the formal study of the fundamentals of computationally highly restricted distributed devices; such as (fault-tolerant) distributed algorithms in hardware and biology.



Thomas Nowak received an MSc degree in computer engineering from Vienna UT in 2010 and a PhD degree in computer science from Ecole polytechnique in 2014. He is currently a post-doctoral teaching and research assistant at École normale supérieure in Paris, France. His research focuses on discrete event systems and distributed algorithms in dynamic fault-tolerant networks.



Ulrich Schmid is full professor and head of the Embedded Computing Systems Group at the Institut für Technische Informatik at TU Vienna. He studied computer science and mathematics and also spent several years in industrial electronics and embedded systems design. Ulrich Schmid authored and co-authored numerous papers in the field of theoretical and technical computer science and received several awards and prizes, like the Austrian START-price 1996. His current research interests focus on the mathematical

analysis of fault-tolerant distributed algorithms and real-time systems, with special emphasis on their application in systems-on-chips and networked embedded systems.