Towards Binary Circuit Models That Faithfully Capture Physical Solvability

Matthias Függer, Robert Najvirt, Thomas Nowak, Ulrich Schmid

Towards Binary Circuit Models That Faithfully Capture Physical Solvability

Matthias Függer*, Robert Najvirt*, Thomas Nowak†, Ulrich Schmid*

* Vienna University of Technology, Vienna, Austria
Email: {fuegger,rnajvirt,s}@ecs.tuwien.ac.at
† École normale supérieure, Paris, France
Email: thomas.nowak@ens.fr

Abstract—In contrast to analog models, binary circuit models are high-level abstractions that play an important role in assessing the correctness and performance characteristics of digital circuit designs: (i) modern circuit design relies on fast digital timing simulation tools and, hence, on binary-valued circuit models that faithfully model signal propagation, even throughout a complex design, and (ii) binary circuit models provide a level of abstraction that is amenable to formal correctness proofs. A mandatory feature of any such model is the ability to trace glitches and other short pulses precisely as they occur in physical circuits, as their presence may affect a circuit's correctness and its performance characteristics.

Unfortunately, it was recently proved [Függer et al., ASYNC'13] that none of the existing binary-valued circuit models proposed so far, including the two most commonly used pure and inertial delay channels and any other bounded single-history channel, is realistic in the following sense: For the simple Short-Pulse Filtration (SPF) problem, which is related to a circuit's ability to suppress a single glitch, they showed that every bounded single-history channel either contradicts the unsolvability of SPF in bounded time or the solvability of SPF in unbounded time in physical circuits, i.e., no existing model correctly captures *physical solvability* with respect to glitch propagation.

We propose a binary circuit model, based on so-called involution channels, which do not suffer from this deficiency. In sharp contrast to what is possible with all the existing models, they allow to solve the SPF problem precisely when this is possible in physical circuits. To the best of our knowledge, our involution channel model is hence the very first binary circuit model that realistically models glitch propagation, which makes it a promising candidate for developing more accurate tools for simulation and formal verification of digital circuits.

I. INTRODUCTION

The steadily increasing complexity of digital circuit designs in conjunction with the large simulation times of accurate analog simulations fuel the need for circuit models that meet two core requirements: (i) Allow for fast and sufficiently accurate simulations, and (ii) facilitate formal correctness proofs that reflect reality, i.e., physical solvability, meaning that a problem is solvable in the model if and only if it is solvable in reality. We call a model that meets both (i) and (ii) a *faithful* model.

With respect to (i), there is a considerable body of work on timing analysis of circuits based on approximating the involved differential equations [1], [2], [3], [4], [5]. However,

these approaches suffer from large simulation times and high memory consumption. Popular VHDL or Verilog simulators hence employ *digital* timing simulations, based on continuous-time, discrete-value (essentially binary), rather than analog-value, circuit models. Their modeling accuracy crucially depends on the ability to accurately predict the propagation of signal transitions throughout a circuit.

Quite obviously, proper timing models are equally important with respect to (ii): Bi-stable elements like latches, flip-flops, and arbiters fail to work correctly when glitches or signal transitions occur at improper times, and may cause metastability (including high-frequency pulse trains due to oscillatory metastability) [6] on that occasion. Since such phenomenons cannot simply be assumed to have vanished at the occurrence of the next clock transition or the next handshake signal in today's high-speed circuits, the precise prediction of the presence/absence of glitches and similar short pulses is crucial for any correctness analysis.

Finding a binary circuit model that indeed faithfully models physical circuits is a difficult task, however. One feasible way to guide such a search is to select some mandatory feature that any such model must satisfy, and to check the ability of a candidate model to do so. Like proposed by Függer et al. [7], we rely on the ability of a model to *realistically* solve a simple but representative glitch propagation problem called *short-pulse filtration* (SPF), with the precise meaning that SPF is solvable in a candidate model if and only if there is a physical circuit that solves it. We are happy to provide the very first continuous-time binary-value circuit model, based on so-called *involution channels*, that realistically solves SPF, along with some physical justification of its success. Nonwithstanding the fact that being realistic is only necessary but not sufficient for being faithful, we are convinced that our model constitutes a big step forward towards a faithful model, which will eventually form the basis of improved tools for simulation and formal verification of digital circuits.

Overview of our contributions. Binary value, continuous time circuit models based on pure and inertial delay channels [8] have been introduced several decades ago, and are still heavily used in existing digital design tools. However, those simple models cannot express such subtle phenomenons as decaying glitches: While pure delay channels propagate even very short glitches as is, unlike real circuits, inertial

delay channels make unrealistically strong assumptions [9] by requiring a glitch to propagate unchanged when it exceeds some minimal length, and to completely vanish otherwise. Unfortunately, in aggressively timed circuit designs, even very short glitches cannot be considered as second-order effects and thus neglected, as they may affect the correctness of the chip.

More elaborate digital channel models, in particular, the DDM model proposed by Bellido-Díaz et al. [10] (originally termed PID model), have hence been introduced, which made their way into accurate digital timing analysis tools already [11]. Although the experimental validation of the DDM model in [10] showed good accuracy for the evaluated examples, the question of the general ability of such a model to actually capture the behavior of physical circuits remained open: Is any of the proposed models realistic, in the sense that it allows to solve (resp. not to solve) a given problem precisely when this is (resp. is not) the case for a physical circuit?

And indeed, Függer et al. [7] showed that any model with *bounded single-history channels*, including pure delay, inertial delay, and DDM channels, is *not* realistic in the case of the simple—but representative—*Short-Pulse Filtration* (SPF) problem: The SPF problem is the problem of building a one-shot variant of an inertial delay channel. As for inertial delay channels, no short pulse may appear at the SPF output; in case of long input pulses, they may be passed unaltered or augmented in duration (including ∞). The stronger variant of *bounded* SPF requires the output to settle in bounded time.

Since Barros and Johnson [12] proved that the problems of building an inertial delay, a latch, a synchronizer and an arbiter are all equivalent, the (un)solvability of (bounded) SPF is a suitable test for a model's ability to realistically model glitch propagation with respect to physical circuits: On the one hand, Marino [9] formally proved that problems like SPF cannot be solved in a physical model when the output is required to stabilize in bounded time [7]. On the other hand, a simple storage loop with a high-threshold filter at its output (see Fig. 7) solves SPF in unbounded time: As shown in the SPICE simulation traces in Fig. 1, sufficiently large input pulses (largest blue dashed one) just cause the storage loop to change its state (to 1) instantaneously (left-most green solid one), very small input pulses (smallest blue dashed one) do not affect the storage loop (bottom green solid one). Critical input pulses (middle blue dashed ones, overlapping, therefore appearing as if they were one pulse) cause the storage loop to become metastable for an unbounded time, eventually resolving to either state 0 or 1. Therefore, appending a high threshold filter with threshold (marked by the red dotted line) clearly above the metastability region results in a clean (= non-metastable) output signal, which either remains at 0, or makes a single transition to 1. Hence, with real circuits, SPF is solvable, while its stronger bounded variant is not.

A single-history channel, as introduced in [7], is characterized by a delay function $\delta(T)$ that may depend on the difference T between the time of the input transition and that of the previous output transition. Fig. 2 illustrates this relation and the involved delays. Pure delay, inertial delay, and DDM
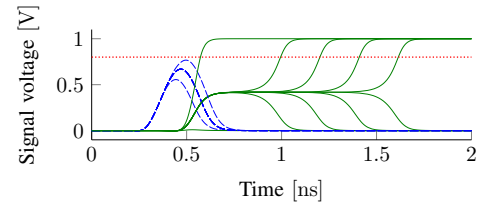

Fig. 1. Analog simulation traces of a CMOS SPF, implemented as a storage loop followed by a high-threshold filter. The dashed (blue) curves represent the input signal, the solid (green) ones give the output of the storage loop. The horizontal line at 0.8 marks the filter threshold level.
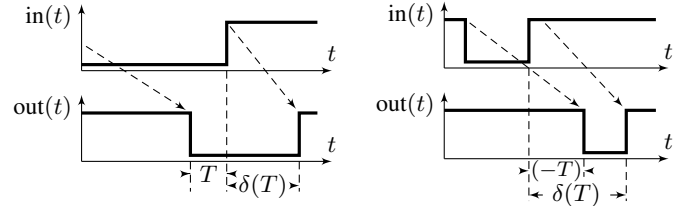

Fig. 2. (i) left: Input/output signal of a single-history channel, involving the input-to-previous-output delay T and the resulting output-to-input delay $\delta(T)$. (ii) right: Input transition with negative input-to-last-output delay T.

channels are all single-history channels with an upper and lower *bounded* delay function. Interestingly, as shown in [7], binary circuit models based on channels with pure (= constant) delays do not even allow to solve unbounded SPF. On the other hand, bounded single-history channels with non-constant delays, including inertial delay and DDM channels, allow to design circuits that solve bounded SPF. Since this contradicts reality, as argued above, none of the existing binary circuit models is realistic.

In this paper, we propose a class of single-history channel models with *unbounded* delay functions that are realistic: Like their bounded counterparts, their delay is upper bounded; however, it is not bounded from below. As shown in Section IV, these negative delays are crucial for accurately modeling glitch suppression. We coined the term *involution channel* for our channels, as we require their negative delay functions to be involutions, i.e., $-\delta(T)$ must form its own inverse (which implies that $\delta(T)$ is strictly increasing and concave). To increase the size of our class of involution channels, we actually allow the delay functions $\delta_\uparrow$ and $\delta_\downarrow$ for rising and falling transitions to be different, and require both $-\delta_\downarrow(-\delta_\uparrow(T)) = T$ and $-\delta_\uparrow(-\delta_\downarrow(T)) = T$. We will prove that the solvability/unsolvability border of SPF in a binary-valued circuit model based on our involution channels is exactly the same as in physical reality.

Major contributions and paper organization. (1) In Section II, we use a simple analog channel model to demonstrate that assuming delay functions which are involutions is not artificial: It reveals that the standard first-order model used, e.g., in [13] actually gives a simple instance of general involution channels, which are introduced formally in Section IV. Our binary circuit model, as well as the SPF problem, are defined in Section III. In Section V, we explain how to use our model to explicitly construct output and intermediate signals of a circuit, given the input signals. (2) In Section VI, we prove that the

simple circuit consisting of a storage loop and a high-threshold filter solves unbounded SPF in the involution channel model. (3) In Section VII, we show that bounded SPF is impossible to solve with involution channels. In a nutshell, our proof inductively constructs an execution that can determine the final output only after some unbounded time. It exploits a surprising continuity property of the output of an involution channel with respect to the presence/absence of glitches at the channel input, which is due to the involution property (unboundedness) of the delay functions. Together, our results reveal that a binary circuit model based on involution channels indeed allows to solve SPF precisely when this is possible in physical circuits, i.e., is realistic.

Related Work. We are not aware of much existing work that directly relates to the particular problem studied in our paper: Unger [8] proposed a general technique for modeling asynchronous sequential switching circuits, based on combinational circuit elements interconnected by pure and inertial delay channels. Brzozowski and Ebergen [14] formally proved that it is impossible to implement Muller C-Elements and other state-holding components using only zero-time logical gates interconnected by wires without timing restrictions. Bellido-Díaz et al. [10] proposed the PID model, and justified its appropriateness both analytically and by comparing the model predictions against SPICE simulations. In [15], the PID model (now called *Delay Degradation Model* DDN) was generalized from inverters to (N)AND and (N)OR gates. In the meantime, thanks to considerable efforts like [16], [15] spent on the question of how to extract the DDN model parameters from technology parameters, the DDN model has already made its way into digital timing analysis tools [11]. Nevertheless, since the results of Függer et al. [7] revealed that none of the above binary circuit models can be realistic (and hence faithful), there is still room for improvement.

II. THE EXPRESSIVE POWER OF INVOLUTION CHANNELS

In this section we will argue that involution channels are indeed a reasonable basis for a binary circuit model, in the sense that they naturally match a (generalized) standard analog model: For any pair of $\delta_\uparrow$, $\delta_\downarrow$, there is a generalized standard analog channel model consisting of a pure delay component, a slew-rate limiter with generalized switching waveforms, and a comparator, as shown in Fig. 3, which has $\delta_\uparrow$, $\delta_\downarrow$ as its corresponding delay functions. Note carefully, though, that we do not claim that Fig. 3 is the only analog model that leads to involution delay functions; there may of course be many others as well. Vice versa, the fact that some well-known analog model leads to involutions does not at all make our results incremental: Besides the fact that, to the best of our knowledge, no analog modeling paper [1], [2], [3], [4], [5] addressed the properties of corresponding delay functions, it is of course not possible to generalize results obtained for some *particular* involution to involutions in general.

Fig. 3 shows a block diagram of an idealized analog circuit corresponding to so constructed involution channels, and a sample waveform. The pure delay shifts the binary-valued
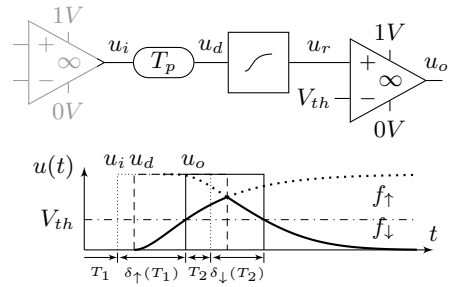

Fig. 3. Simple analog channel model.

input u_i in time by some T_p; e.g., accounting for underswing. The slew rate limiter exchanges the step functions of the resulting u_d with instances of $f_\uparrow$ and $f_\downarrow$, shifting them in time such that the output u_r is continuous and switches between strictly increasing and decreasing exactly at u_d switching times. The comparator generates u_o by again discretizing the value of this waveform comparing it to the threshold voltage V_{th}. One can show that

$$\delta_\uparrow(T) = -f_\uparrow^{-1}(f_\downarrow(T + \delta_\infty^\downarrow)) + \delta_\infty^\uparrow \quad \text{and}$$
$$\delta_\downarrow(T) = -f_\downarrow^{-1}(f_\uparrow(T + \delta_\infty^\uparrow)) + \delta_\infty^\downarrow, \tag{1}$$

where $\delta_\infty^\uparrow = T_p + f_\uparrow^{-1}(V_{th})$ and $\delta_\infty^\downarrow = T_p + f_\downarrow^{-1}(V_{th})$.

As a special case, consider a slew rate limiter implemented as a first-order RC low pass filter; the switching waveforms are $f_\downarrow(t) = 1 - f_\uparrow(t) = e^{-t/\tau}$ here, with τ being the RC time constant. Inserting these functions into (1), we obtain

$$\delta_\uparrow(T) = \tau \ln(1 - e^{-(T + T_p - \tau \ln(V_{th}))/\tau}) + T_p - \tau \ln(1 - V_{th})$$
$$\delta_\downarrow(T) = \tau \ln(1 - e^{-(T + T_p - \tau \ln(1 - V_{th}))/\tau}) + T_p - \tau \ln(V_{th}).$$

In the remainder of this paper, these specific channels will be called *exp-channels*. Note that, in general, switching waveforms are not restricted to exponential RC-charging curves.

To round-off this section, we also provide a glimpse of our on-going model validation experiments. The purpose of these experiments is to further contribute to answering the question of whether our involution channel model is not only realistic but also quantitatively matches the behavior of physical circuits well. Fig. 4 shows the delay function of a single inverter from the UMC 65 nm standard cell library obtained in SPICE simulations (dots). Like in the work by Bellido-Díaz et al. [10], we used a pulse train consisting of equally spaced pulses with decaying duration as the input waveform; a symmetric 50% threshold has been used for determining the resulting $\delta(T)$. The dashed fitting of an involution channel shows a good match. The (solid) waveform in Fig. 5 was finally obtained by simulating a 5 inverter chain (out1, out3 and out5 shown). The dashed binary signals depict the nicely matching predictions obtained by our model, using $\delta(T)$ from Fig. 4.

III. BINARY CIRCUIT MODEL

Since the purpose of our work is to replace analog models like the one in the previous section by a purely digital model, we will now formally define the binary-value continuous-time circuit model used in the remainder of this paper.
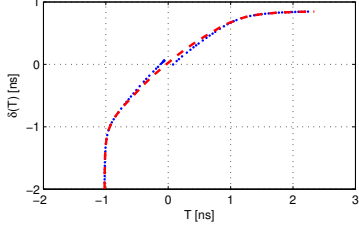
Fig. 4. Delay $\delta(T)$ (blue dots) of a UMC 65 nm inverter plotted over the input-to-previous-output delay T, computed from a SPICE simulation using a decaying pulse train as the input waveform. The fitting curve (dashed red) compensates the inevitable simulation errors for very short pulses $T \approx 0$.
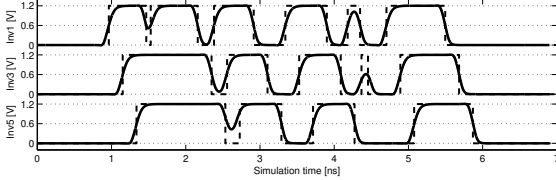

Fig. 5. A sample waveform (solid) of an inverter chain simulated in SPICE, along with the corresponding predictions (dashed) of our model.

Signals. A *falling transition* at time t is the pair $(t, 0)$, a *rising transition* at time t is the pair $(t, 1)$. A *signal* is a (finite or infinite) list of alternating transitions such that

S1) the initial transition is at time $-\infty$; all other transitions are at times $t \geq 0$,

S2) the sequence of transition times is strictly increasing,

S3) if there are infinitely many transitions in the list, then the set of transition times is unbounded.

To every signal s corresponds a function $\mathbb{R}_+ \to \{0, 1\}$ whose value at time t is that of the most recent transition. We follow the convention that the function already has the new value at the time of a transition, i.e., the function is constant in the half-open interval $[t_n, t_{n+1})$ if t_n and t_{n+1} are two consecutive transition times. A signal is uniquely determined by such a function and its value at $-\infty$.

Circuits. 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 any gate input, channel input and output port is attached to only one input port, gate output or channel output. Moreover, gates and channels must alternate on every path in the circuit.

Formally, a *circuit* is described by a directed graph whose vertices are partitioned into *input ports*, *output ports*, *channels*, and *gates*; with edges constrained as described above. Every channel is assigned a channel function, which maps the input to the output. Section IV specifies the properties of this function for our involution channels. Every gate is assigned a Boolean function that maps its inputs to the output.

Executions. An execution of circuit C is an assignment of signals to vertices (formalized as a collection of signals s_v for all vertices v of C) that respects the (timed) channel functions and (untimed) Boolean gate functions.

Short-Pulse Filtration. A *pulse* of length Δ at time T has initial value 0, one rising transition at time T, and one falling transition at time $T+\Delta$. A signal *contains a pulse* of length Δ at time T if it contains a rising transition at time T, a falling transition at time $T + \Delta$ and no transition in between.

A circuit *solves Short-Pulse Filtration (SPF)* if it fulfills the following conditions. Note that we allow the circuit to behave arbitrarily if the input signal is not a (single) pulse.

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

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

F3) There exist an input pulse such that the output signal is not the zero signal. *(Nontriviality)*

F4) There exists an $\varepsilon > 0$ such that for every input pulse the output signal never contains a pulse of length less than ε. *(No short pulses)*

A circuit *solves bounded SPF* if additionally the following condition holds:

F5) There is a $K > 0$ such that for every input pulse the last output transition is before time $T + K$ if T is the time of the last input transition. *(Bounded stabilization time)*

IV. INVOLUTION CHANNELS

Intuitively, a channel propagates each transition at time t of the input signal to a transition at the output happening after some *output-to-input* delay $\delta(T)$, which depends on the *input-to-previous-output* delay T. Note that T can be negative if two input transitions are close together, as is the case in Fig. 2 (ii).

Formally, an *involution channel* is characterized by an *initial value* $I \in \{0, 1\}$ and two strictly increasing concave *delay functions* $\delta_\uparrow : (-\delta_\infty^\downarrow, \infty) \to (-\infty, \delta_\infty^\uparrow)$ and $\delta_\downarrow : (-\delta_\infty^\uparrow, \infty) \to (-\infty, \delta_\infty^\downarrow)$ such that both $\delta_\infty^\uparrow = \lim_{T \to \infty} \delta_\uparrow(T)$ and $\delta_\infty^\downarrow = \lim_{T \to \infty} \delta_\downarrow(T)$ are finite and

$$-\delta_\uparrow\big(-\delta_\downarrow(T)\big) = T \text{ and } -\delta_\downarrow\big(-\delta_\uparrow(T)\big) = T \quad (2)$$

for all applicable T. All such functions are necessarily continuous and strictly increasing. For simplicity, we will also assume them to be differentiable; δ being concave thus implies that its derivative δ' is monotonically decreasing. If multiple channels in a circuit share a common input signal, as depicted in Fig. 6, we require that they all have the same initial value I. This is without loss of generality, as one can always replicate the input signal.

The behavior of involution channels is defined as follows: *Initialization:* If the channel's initial value I is different from the initial value X of the channel input signal s and s has no transition at time 0, add transition $(0, X)$ to s ("reset"). *Output transition generation algorithm:* Let $t_1, t_2, \ldots$ be the times of the transitions of s, and set $t_0 = -\infty$ and $\delta_0 = 0$.

- *Iteration:* Determine the tentative list of *pending* output transitions: Recursively determine the output-to-input delay δ_n for the input transition at time t_n by setting
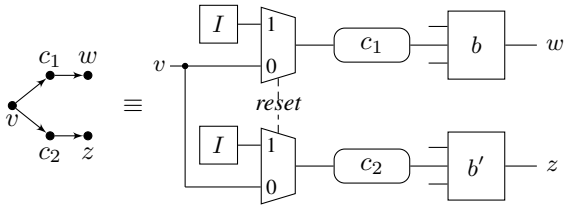
Fig. 6. A circuit (graph) with vertex v (being an input or a gate), gates w, z, and channels c_1 and c_2 (on the left) and the physical equivalent (on the right). Both channels must have the same initial value I; b and b' are the Boolean functions assigned to gates w and z, respectively.

$\delta_n = \delta_\uparrow(t_n - t_{n-1} - \delta_{n-1})$ if t_n is a rising transition and $\delta_n = \delta_\downarrow(t_n - t_{n-1} - \delta_{n-1})$ if it is falling. The nth and mth pending output transitions *cancel* if $n < m$ but $t_n + \delta_n \geq t_m + \delta_m$. In this case, we mark both as canceled.

- *Return:* The channel output signal $c(s)$ has initial value I and contains every pending transition at time $t_n + \delta_n$ that has not been marked as canceled.

We say an involution channel is *strictly causal* if $\delta_\uparrow(0) > 0$, which is equivalent to $\delta_\downarrow(0) > 0$ due to (2). One can show a minimum delay property for these channels:

Lemma 1: A strictly causal involution channel has a unique $\delta_{\min}$ defined by $\delta_\uparrow(-\delta_{\min}) = \delta_{\min} = \delta_\downarrow(-\delta_{\min})$, which is positive. The channel delay for any non-canceled transition of an involution channel is at least $\delta_{\min}$.

In the rest of the paper, we assume all channels to be strictly causal involution channels.

V. CONSTRUCTING EXECUTIONS OF CIRCUITS

The definition of an execution of a circuit as given in Section III is "existential", in the sense that it only allows to check for a given collection of signals whether it is an execution or not. And indeed, in general, circuits may have no execution or may have several different executions. By contrast, in case of circuits involving strictly causal involution channels only, executions are unique and can be constructed iteratively: We give a deterministic algorithm below.

Given a circuit C with strictly causal involution channels, let $(s_i)_{i \in \mathcal{I}}$ be any collection of signals for all the input ports $\mathcal{I}$. Since all output ports are driven by gates we can identify the output port with the output of its driving gate. The channel with predecessor x (an input port or a gate output) and successor y (a gate input) is denoted by the tuple (x, y). The algorithm iteratively generates the list of transitions of s_σ of (the output of) every vertex σ in the circuit, and hence the corresponding function $s_\sigma(t)$. In the course of the execution of this algorithm, a subset of the generated transitions will be marked *fixed*: Non-fixed transitions could still be canceled by other transitions later on, fixed transitions will actually occur in the constructed execution. The detailed algorithm is as follows:
Initialization: For all channels (v, w) in C, $s_{(v,w)} = ((-\infty, I))$ initially, with I being the initial value of channel (v, w). According to the implicit reset of our channels introduced in Section IV, the transition $(0, X)$ is also added to $s_{(v,w)}$ if the initial transition $(-\infty, X)$ of s_v satisfies $X \neq I$. Note that this is well-defined also in case of channels (v, w)
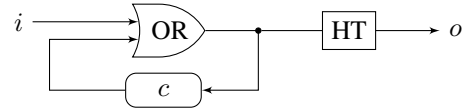
and (v, w') attached to the same v, as we require $s_{(v,w)} = s_{(v,w')}$ initially in this case; see Section IV. Further, for a gate v, $s_v = ((-\infty, X))$ initially, where X is the value of the Boolean function corresponding to v applied to the values of the initial transitions in s_σ for all of v's predecessors σ. The zero-input gates 0 and 1 used for generating constant-0 and constant-1 signals have $s_0 = ((-\infty, 0))$ and $s_1 = ((-\infty, 1))$, respectively. Initially, all transitions at $-\infty$ are fixed and all others are not.
Iteration: If there is no non-fixed transition left, terminate with the execution made up by all fixed transitions. Otherwise, let $t \geq 0$ be the smallest time of a non-fixed transition.

(i) Mark all transitions at t fixed.
(ii) For each newly fixed transition from step (i), occurring in s_σ where σ is a predecessor of a gate v: If signal s_v's current value $s_v(t) = X$ differs from the value of v's Boolean function applied to the values $s_{\sigma'}(t)$ for all of v's predecessors σ' (which also include σ), add the transition $(t, 1 - X)$ to s_v and mark it fixed.
(iii) For each newly fixed transition $(t, x) \in s_v$ from steps (i) or (ii), occurring in s_v of a gate output or an input port: For each successor channel (v, w) of v, apply the iteration step of (v, w)'s transition generation algorithm with input signal s_v, output signal $s_{(v,w)}$, and current input transition (t, X). If this leads to a cancellation in $s_{(v,w)}$, remove both canceling and canceled transition from the list. No fixed transition will ever be removed this way.

One can show that this algorithm indeed constructs an execution of C.

VI. POSSIBILITY OF UNBOUNDED SPF

We next show that unbounded SPF is solvable in a circuit model with strictly causal involution channels. We do this by verifying that the circuit in Fig. 7 indeed solves SPF. The circuit was inspired by the physical solution of Fig. 1, and consists of a fed back OR-gate forming the storage loop and a subsequent high-threshold filter (implemented by a channel). The channel implementing the high-threshold filter is assumed to be an exp-channel because we have to adjust its parameters appropriately.

Consider a pulse of length Δ at time 0 at the input. One can show that there exists a unique threshold $\tilde{\Delta}$ such that: If Δ is larger, then the output is eventually constant 1. If it is smaller, the output is eventually constant 0. If it is equal, the output is a periodic pulse train with duty cycle 50%.

Finally, one can show that a properly dimensioned high-threshold filter is able to suppress the pulse trains with decreasing or constant 50% duty cycle. We thus obtain:
Theorem 1: There is a circuit that solves unbounded SPF.

Fig. 7. A circuit solving unbounded SPF, consisting of an OR-gate fed back by channel c, and a high-threshold filter HT.

VII. Impossibility of Bounded SPF

We first show that that strictly causal involution channels are continuous in a certain sense that we will define precisely below. We start with a suitable distance of signals.

Definition 1: For a signal s and a time T, denote by $\mu_T(s)$ the total duration in $[0, T]$ where s is 1. That is, $\mu_T(s)$ is the measure of the set $\{t \in [0, T] \mid s(t) = 1\}$.

For any two signals s_1 and s_2 and every T, we define their *distance up to time T* by setting $\|s_1 - s_2\|_T = \mu_T(|s_1 - s_2|)$.

Intuitively, an involution channel is continuous under this measure for two reasons: (i) Due to the continuity of δ, a small change in the time at which an input transition occurs, results in a small change in the time at which the corresponding output transition occurs. This, again, only results in a small change of the input-to-previous-output time for the next input transition, and so on. The technical difficulty is to show that this effect does not result in discontinuities even for an unbounded number of input transitions. (ii) Due to the involution property of δ, one can show that δ is not only continuous in changing the length of input pulses, but also in removing them: An input pulse whose length is arbitrarily small results in a value of δ for the next input transition that is arbitrarily close to the transition's δ value in the case the short pulse was not present at all. Again, the major difficulty lies in showing that this also holds for infinite pulse trains. Note carefully that it is primarily the continuity property (ii) that distinguishes our involution channels from the "unfaithful" single-history channels analyzed by Függer et al. [7], which allow bounded SPF to be solved. We thus establish:

Theorem 2: Let c be a channel and let $T \in [0, \infty)$. Then, the mapping $s \mapsto c(s)$ is continuous with respect to the distance $\|s_1 - s_2\|_T$.

Call a circuit a *forward circuit* if its graph is acyclic. Forward circuits are exactly those that do not contain feed-back loops. Equipped with the continuity of involution channels and the fact that the composition of continuous functions is continuous, one can prove that the inherently discontinuous SPF problem cannot be solved with forward circuits.

Theorem 3: No forward circuit solves bounded SPF.

To show the result for general circuits, we define the k-*unrolled circuit* $C_k(v)$, with $k \geq 0$ and v a gate or input in C, as a forward circuit all whose paths of length at most k and ending in v are equal in $C_k(v)$ and C modulo renaming; we use the unrolling level as a superscript for gates and channels in $C_k(v)$. Care must be taken to place proper initialization gates at the beginning of the paths in the unrolled circuit, in order not to introduce unwanted initial transitions.

To finally deduce the impossibility of bounded SPF, we use the fact that a circuit C that solves SPF with stabilization time bound K can be simulated by an unrolled circuit C_N with depth N larger than the maximum causal depth of any of its transitions, i.e., larger than $K/\delta_{\min}^C$ plus the number of input transitions, where $\delta_{\min}^C$ denotes the smallest $\delta_{\min}$ of all channels in circuit C. This, however, contradicts the fact that no forward circuit, and thus specifically C_N, can solve bounded SPF. We hence obtain the result:

Theorem 4: No circuit solves bounded SPF.

VIII. Conclusions and Future Work

We showed that a binary circuit model based on involution channels is a promising candidate for a faithful model, as it is realistic in the sense that it allows to design circuits solving the SPF problem precisely when this is possible with physical circuits. Our involution channels differ from all existing bounded single-history channels, which do not share this property, in that they are also continuous with respect to dropping small input pulses. Although our results prove that involution channels are superior to all alternative channel models known so far in this respect, there are open challenging questions: We only gave a physical motivation and first simulations to quantitatively assess the modeling accuracy of our model w.r.t. glitch propagation. More detailed experiments and a comparison of simulation times with alternative channels and SPICE is inevitable to assess its practical usability in circuit simulators. Although we believe that it will surpass alternative channel models, we cannot rule out the possibility that a non-faithful model like DDM works better in some situations. Needless to say, addressing both questions requires major efforts and is hence a subject of our current/future research.

References

[1] L. W. Nagel and D. Pederson, "SPICE (Simulation Program with Integrated Circuit Emphasis)," Tech. Rep. UCB/ERL M382, EECS Department, University of California, Berkeley, 1973.

[2] M. A. Horowitz, *Timing Models for MOS Circuits*. PhD thesis, Stanford University, 1984.

[3] T.-M. Lin and C. Mead, "Signal delay in general RC networks," *IEEE TCAD*, vol. 3, no. 4, pp. 331–349, 1984.

[4] L. Pillage and R. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE TCAD*, vol. 9, no. 4, pp. 352–366, 1990.

[5] A.-C. Deng and Y.-C. Shiau, "Generic linear RC delay modeling for digital CMOS circuits," *IEEE TCAD*, vol. 9, no. 4, pp. 367–376, 1990.

[6] L. R. Marino, "General theory of metastable operation," *IEEE ToC*, vol. 30, no. 2, pp. 107–115, 1981.

[7] M. Függer, T. Nowak, and U. Schmid, "Unfaithful glitch propagation in existing binary circuit models," in *Proceedings ASYNC'13*, pp. 191–199, New York City: IEEE Press, 2013.

[8] S. H. Unger, "Asynchronous sequential switching circuits with unrestricted input changes," *IEEE ToC*, vol. 20, no. 12, pp. 1437–1444, 1971.

[9] L. R. Marino, "The effect of asynchronous inputs on sequential network reliability," *IEEE ToC*, vol. 26, no. 11, pp. 1082–1090, 1977.

[10] M. J. Bellido-Díaz, J. Juan-Chico, A. J. Acosta, M. Valencia, and J. L. Huertas, "Logical modelling of delay degradation effect in static CMOS gates," *IEE Proceedings – Circuits, Devices, and Systems*, vol. 147, no. 2, pp. 107–117, 2000.

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

[12] J. C. Barros and B. W. Johnson, "Equivalence of the arbiter, the synchronizer, the latch, and the inertial delay," *IEEE ToC*, vol. 32, no. 7, pp. 603–614, 1983.

[13] F. U. Rosenberger, C. E. Molnar, T. J. Chaney, and T.-P. Fang, "Q-modules: Internally clocked delay-insensitive modules," *IEEE ToC*, vol. 37, no. 9, pp. 1005–1018, 1988.

[14] J. A. Brzozowski and J. C. Ebergen, "On the delay-sensitivity of gate networks," *IEEE ToC*, vol. 41, no. 11, pp. 1349–1360, 1992.

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

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