On the Threat of Metastability in an Asynchronous Fault-Tolerant Clock Generation Scheme

Gottfried Fuchs, Matthias Függer and Andreas Steininger
Vienna University of Technology — Embedded Computing Systems Group
{fuchs, fuegger, steininger}@ecs.tuwien.ac.at

***Abstract*—Due to their handshake-based flow control, asynchronous circuits generally do not suffer from metastability issues as much as synchronous circuits do. We will show, however, that fault effects like single-event transients can force (sequential) asynchronous building blocks such as Muller C-Elements into a metastable state. At the example of a fault-tolerant clock generation scheme, we will illustrate that metastability could overcome conventional error containment boundaries, and that, ultimately, a single metastable upset could cause even a multiple Byzantine fault-tolerant system to fail. In order to quantify this threat, we performed analytic modeling and simulation of the elastic pipelines, which are at the heart of our physical implementation of the fault-tolerant clocks. Our analysis results reveal that only transient pulses of some very specific width can trigger metastable behavior. So even without consideration of other masking effects the probability of a metastable upset to propagate through a pipeline is fairly small. Still, however, a thorough metastability analysis is mandatory for circuits employed in high-dependability applications.**

I. Introduction

Modern CMOS technologies allow for ultra high speed processing and enormous circuit complexity comprising several millions of transistors per die [1]. The main driving force for the tremendous technological progress is the proceeding downscaling of feature size. Recent publications reveal that this gain in computational power, however, comes at a price [2]. In the first place, process variations during manufacturing make it difficult to comply with the demanding requirements for clocking high-speed synchronous designs. Secondly, the reduced voltage swing as well as the smaller critical charges make circuits more susceptible to the adverse effects of particle hits, crosstalk and electromagnetic interference (EMI) and may cause single event upsets [2], [3]. This trend also makes multiple faults (correlated or uncorrelated) more likely, and the traditional single fault assumption that has become a de facto standard in many fault-tolerant systems needs to be questioned.

Asynchronous, quasi delay insensitive (QDI) circuits [4] are suitable for solving many of these problems quite naturally. They exhibit inherent robustness, do not rely on sophisticated clock distribution, and naturally tolerate parameter variations. Therefore they can be considered for future fault-tolerant systems. QDI circuits solve a notorious problem of the synchronous world very elegantly, namely metastability. While in the Globally Asynchronous Locally Synchronous (GALS [5]) paradigm that is increasingly applied in systems-on-chip the crossing of clock boundaries inevitably introduces metastability issues (albeit these can be mitigated by sacrificing performance), the handshake principle of purely asynchronous designs effectively eliminates the potential for metastability (at least confines it to specific residual cases like arbiters).

As a matter of fact, however, metastability is inherent to any bistable element [6], [7], and it is the key purpose of a design style — be it synchronous or asynchronous — to avoid marginal triggering conditions for bistable elements. The advantage of QDI circuits over GALS is that they generally need not consider metastability issues as part of their normal operation, as they solve synchronization issues on the conceptual level. The situation, however, changes when fault effects need to be considered as well. Here the handshaking principle can no longer protect the bistable elements from becoming metastable, as they are not operated in a closed environment anymore, and the adverse power of faults cannot a priori be restricted. In particular, it appears that short pulses, namely single event transients, caused by particle hits or EMI, have the potential of causing metastable behavior of bistable elements by violating their specified timing constraints. Unfortunately, avoiding such timing constraints is as impossible as building truly delay insensitive bistable elements. Therefore this issue needs to be further investigated.

Due to its known unavoidability in synchronous designs, metastability has received a lot of attention over the years. In this context its impact and its sources are quite well researched. An analytic model has been derived to allow an estimation of the Mean Time Between Upsets (MTBU) that result from metastability in synchronous circuits [8]–[11]. This model has been confirmed by countless simulations and experiments [11]–[19] and continuously refined [12], [15], [16]. By contrast, an asynchronous design does not suffer from the same metastability problems during normal operation, therefore the arbiter problem has become the main issue of interest, while relatively little knowledge exists on metastability propagation in asynchronous elements (e.g. [20]).

This lack of investigations in the asynchronous domain is our main motivation for taking a closer look at fault-induced metastability effects in QDI circuits in this paper. We will use the example of a fault-tolerant hardware clock that we developed in our DARTS-project (Distributed Algorithms for Robust Tick-Synchronization)[1] and identify the metastability critical elements of our asynchronous design. Based both on

[1] The project DARTS received funding from the Austrian bm:vit (FIT-IT, contract no. 809456-SCK/SAI). *DARTS webpage*: *http://ti.tuwien.ac.at/darts/*

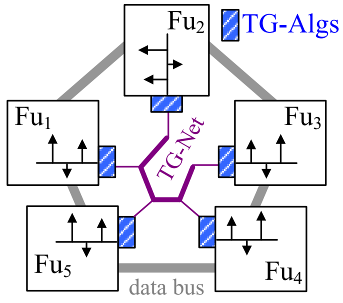

Fig. 1. The DARTS clocking scheme.

analytic modeling and on simulations we will derive a quantitative understanding for the conditions that cause metastability in and for its propagation through Muller C-Elements.

The paper is structured as follows: After a brief introduction of our DARTS architecture in the following section, we will then investigate this circuit's potential for metastability generation and propagation in Section III. Next, Section IV will be concerned with the derivation of an analytic model for the circuit and its metastable decay. The propagation of metastability will be studied by means of simulation in Section V. Finally, Section VI concludes the paper and presents future prospects.

II. The DARTS Clock Generation Scheme

Starting from the insight that future technology generations are likely to be more susceptible to faults, our DARTS-project focused on a very prominent single point of failure of many synchronous systems, namely the clock [21]. In order to make the clock fault tolerant we proposed the clocking scheme depicted in Fig. 1. In this approach each *functional unit* Fu_i of a synchronous system-on-chip design is augmented with an instance of a *tick-generation algorithm TG-Alg*. The interconnection via a dedicated *tick-generation network TG-Net* allows the TG-Alg instances to generate a globally synchronized local clock that each TG-Alg provides to its attached functional unit. Given the fact that our approach directly aims at the *generation* of a fault-tolerant clock (without the need for quartz oscillators) the implementation of the TG-Alg clearly has to follow an asynchronous design style.

The TG-Alg implementation presents an adaptation of a well known clock synchronization algorithm from the distributed systems community to the requirements of VLSI design. In particular, we implemented a slightly modified version of Srikanth & Toueg's consistent broadcast primitive [22] shown in Algorithm 1.

Essentially, the operation of Algorithm 1 is based on two concurrent rules, the *relay*- and the *increment*- rule. A node emits *tick*(k) if it has received these *tick*(k) messages from sufficiently many ($f+1$) other nodes, following the relay rule. If a node has received $2f+1$ *tick*(k) messages, the increment rule triggers and produces a new *tick*($k+1$). It can be proven that in a setup of $n \geq 3f+2$ nodes Algorithm 1 tolerates up to f arbitrary (Byzantine) faults while still maintaining a

Algorithm 1 Byzantine tolerant tick generation [23]

```
variables
    $k$ : integer := 0
initially send tick($0$) to all [once]
// Relay Rule
if received tick($l$) from at least $f+1$ remote processes with $l \geq k$
then
    send tick($k$), ..., tick($l$) to all [once]; $k := l$
// Increment Rule
if received tick($k$) from at least $2f+1$ remote processes then
    send tick($k+1$) to all [once]; $k := k+1$
```

computable worst case precision among all non-faulty components.

The original, software based algorithm had to be modified in several ways to allow for an efficient VLSI implementation. A block diagram of the resulting ASIC implementation is shown in Fig. 2. For a suitable implementation the unbounded k numbers of *tick*(k) messages had to be substituted by simple up/down transitions in order to keep the TG-Net and the hardware in general as small respectively high-speed as possible. Sending up/down-transitions (i.e. events) only instead of integer numbers (states) implies that every TG-Alg located at, say node p, locally has to keep track of ticks sent and received so far. Obviously this requires some kind of counter. Fortunately a closer look at Algorithm 1 shows that not the absolute value of k is relevant, but rather the difference between number of received tick messages l (per remote node) and number of messages k, p transmitted. This facilitates the employment of an up/down counter (per remote node), which counts up whenever p receives a message from the respective remote node and down when p transmits a message. The up/down counter's width is bounded by the worst-case precision. One such asynchronous up/down counter has been implemented via pairs of elastic pipelines [24] for each remotely connected node. Computation of the difference is achieved by the removal of "matching" clock ticks from the pipelines. This is achieved via a suitable interconnection of the elastic pipelines in conjunction with an additional Muller C-Element that removes ticks. The *Pipe Compare Signal*

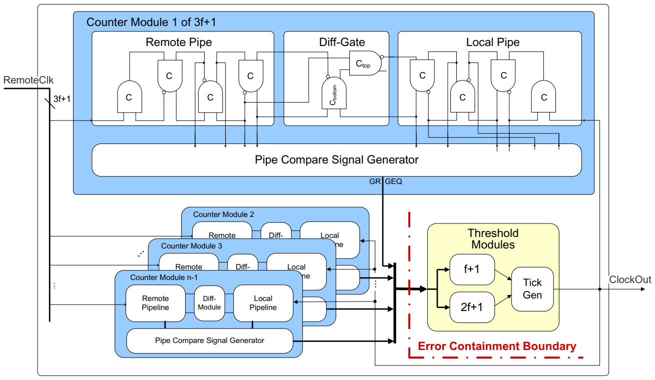

Fig. 2. The DARTS implementation.

Generator PCSG evaluates the fill levels of the remote and local elastic pipeline pairs and supplies the *Threshold Modules (THMs)* with status inputs (both blocks are pure combinational logic without any storage elements). The THMs basically represent the relay- and increment-rules of Algorithm 1 and trigger the generation of a new clock tick if sufficiently many ($f + 1$ respectively $2f + 1$) previous ticks have been received so far.

We have formally analyzed the modified algorithm as well as the resulting design and derived performance measures and correctness proofs for our implementation. Our ongoing experimental evaluations of a prototype ASIC seem to confirm these predictions in practice [25], [26].

III. Potential for Metastability in DARTS

A. Error containment in DARTS

Since Algorithm 1 originally was written for software, it is based on a system model that implies error containment at node level: The $f + 1$ respectively $2f + 1$ thresholds ensure that messages from faulty nodes will be safely ignored (unless supported by other, correct ones), thus avoiding any fault propagation from a faulty to a non-faulty node.

In our DARTS implementation the $f+1$ respectively $2f+1$ thresholds are realized by the THMs. In this sense it is natural to consider the THMs as the error containment boundaries in our system. Given that a few timing constraints are met in the chip layout, it has been proven, that the DARTS implementation tolerates f unrestricted (i.e. Byzantine) faults [27]. Although we have tried to give our proof on a very low level of abstraction, carefully incorporating many implementation related details that have not been considered in Srikanth/Toueg's original proof for the algorithm, we had to postulate that every function module performs correctly (if not affected by a fault). This assumption is valid of course only if the module's environment (particularly its inputs) is within specification, which is also supported by the proofs. For the closed system we consider, every element creates a proper environment for the subsequent one(s). Obviously, however, it is not possible to postulate any restrictions for external faults affecting the system, such as transients caused by particle hits or by electromagnetic interference. In such an event a module may be operated off-spec and hence behave essentially arbitrary. While in many cases such a behavior will still be "benign" in the sense that it will still be successfully masked, it is known that off-spec operation may also trigger metastability. Metastable effects are a notorious threat in synchronous circuits, but, as already mentioned, little is known about their generation and propagation in asynchronous designs like our DARTS chip. In particular, it is hard to judge how well our error containment boundaries (THMs) will perform for metastable effects. In the following we will therefore focus on these issues.

B. The metastability phenomenon

Metastability is an undesired property of bistable elements whose input space is continuous-valued [28]. In terms of hardware the usual showcase is a latch cell whose function is to properly output a HI or LO, while its input voltage and/or time between certain edges are ultimately continuous. The actual problem is that for a borderline case at the input the bistable element may need an unbounded time to decide which of the discrete output states to assume. This is a fundamental problem that has been proven to be unsolvable within bounded time [6]. In a properly designed digital system, however, the input voltage is either clearly HI or clearly LO (with steep transitions in between), thus essentially avoiding the borderline case with respect to voltage. Similarly the design style (either synchronous or handshake based/asynchronous) rules out the occurrence of borderline cases with respect to the time between transitions (in the synchronous case this simply means observing the setup- and hold time of a flip-flop, for example). Therefore metastability is usually encountered only in the very restricted context of synchronizers and arbiters, where it is well researched [15], [17], [29]. Although, as already mentioned, metastability cannot be avoided in principle, there are means to make its occurrence arbitrarily improbable, and models exist to (statistically) estimate the mean time between metastable upsets [10], [13].

There are three different ways in which metastability can manifest itself:

1) Excessive delay of a well-shaped transition. This will normally cause timing violations in the subsequent (synchronous) circuit.
2) The output assumes an undefined value in between HI and LO for an unbounded time. This can lead to a propagation of the undefined logic level or to ambiguous interpretation by different subsequent circuit elements and hence to malfunction.
3) Oscillation of the output. The problem here is that the edges generated by this self-oscillation are not in causal relation with the input.

Case (1) either leads to a delayed recognition of the input change, or in the worst case, to a propagation of the metastability to the next bistable element. In context with usual technologies case (2) has most often been encountered and researched especially for synchronous systems. Case (3) has quite rarely been reported from experiments and observations in practical applications, therefore we will concentrate on case (2) in this paper.

C. Metastability generation and propagation in DARTS

Independent of the actual fault origin, we can distinguish two types of node faults with respect to their effect:

- Faults whose effects are not visible at the affected node's output are ineffective from the system's point of view. Obviously they have been masked at some point within the node (the THM, e.g.), and will hence not be considered further.
- All remaining, effective faults can therefore be projected to an erroneous behavior of the affected node's output.

Consequently our concern must be how this faulty behavior is perceived at the other nodes' inputs. Notice that this approach automatically embraces faults of the TG-Net as well. We will therefore study propagation of faults, experienced at the node input, into the node itself. Our aim is to quantify the threat of propagating the fault over the error containment boundary, i.e., the THMs.

Recall from Section II that in order to reduce the implementation overheads for the TG-Net we decided to convey events only (instead of tick numbers in the original algorithm), and derive the tick numbers by virtue of up/down counters at the receiving inputs. As a result substantial parts of this counter — namely the remote pipeline and the PCSG — are located between the node input and the THMs. In our specific context this means that we have to investigate how prone to metastability these modules are. It will turn out later, that the remote pipeline can be viewed as a kind of metastability filter protecting the THMs.

Considering this background, a closer analysis of the circuit modules shown in Fig. 2 with respect to metastability yields the following status:

- The purely combinational blocks, namely PCSG and THMs (with the exception of the Tick Generation Module, see Fig. 2), are not bistable (no internal state, no positive feedback) and are hence not prone to metastability. They are, however, capable of propagating a metastable state. Consider the k-of-n threshold gate as an example: With $k-1$ HI inputs the element is just below its threshold, such that a metastable state on one of the remaining inputs may be propagated to the output. Notice, however, that in all other cases a metastable input will not be propagated.
- State-holding elements, in particular Muller C-Elements, are contained in the Elastic Pipelines, Difference Module and Tick Generation Module. As already mentioned we have formally derived timing constraints under which these Muller C-Elements are guaranteed to operate properly and we have considered all of these constraints in our implementation. So there is no threat of metastability in the fault-free case. When operated off-spec in case of faults, however, these Muller C-Elements do have the potential of becoming metastable. At the same time it is well known from the synchronous domain that bistable elements have a quite good "synchronizing property" when confronted with a metastable input. So it will be interesting to study the Muller C-Element's respective performance in our context.

In summary, in our DARTS circuit, we do find a potential for the *generation* of metastability — namely in case of a fault — and for its *propagation*. In one worst case scenario a single particle hit may cause an f-resilient system to fail: Assume (i) the particle creates a transient at the affected node's output. In at least $f+1$ of the receiving nodes this transient then (ii) might trigger metastability that further propagates through (iii) their elastic pipelines and (iv) their threshold modules such that it reaches their outputs. This creates more than f faulty outputs, which violates the failure assumption. The scenario appears to be extremely unlikely, as each of the steps (i)–(iv) exhibits only a very low probability to occur. Still it seems worthwhile to get some *quantitative* estimation for the probability of such an incidence. The subsequent chapters will be concerned with deriving such an estimation. We will be conservative in assuming that the above mentioned masking by the threshold module is not effective, and consider the "synchronizing" effect of the Muller C-Elements constituting the elastic pipeline as the only barrier for a metastable event triggered by a marginal receiver input (corresponding to steps (ii) and (iii) above). By assuming this pipeline to be empty we further disable the logical masking of the Muller C-Element as reported, e.g., in [30].

IV. Basic Models

In the synchronous domain, a widely accepted circuit model for a latch has been derived that allows an analytic treatment of metastability [8]–[11]. Its basic structure is essentially always the one shown in Fig. 3: The latch circuit is composed of two cross-coupled inverters, each of which is modeled as an inverting amplifier with gain G, followed by an RC low-pass filter with time constant τ.

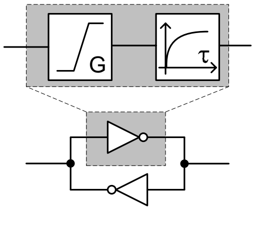

Fig. 3. Traditional model for metastability prediction.

The circuit is considered for the case the latch is opaque, which, in terms of mathematics, relates to the homogeneous case of the differential circuit equation. The (approximated) solution indicates that, starting from the metastable state ("balance point") any small asymmetry in the initial capacitor charges will cause an exponential rise (e^{t/T_c}) of the output voltage towards the upper or the lower limit. The inverter's gain bandwidth product G/τ determines the exponential time constant T_c and thus the resolution time t_{res} required to reach the threshold of the subsequent input.

Under the assumption of periodic alternation between transparent and opaque (period T_{clk}) and for an input signal with exponentially distributed distance between transitions (rate R_{dat}) this model finally yields the widely used equation for the prediction of the mean time between metastable upsets (MTBU):

$$MTBU = e^{t_{res}/T_c} \cdot \frac{T_{clk}}{T_0} \cdot \frac{1}{R_{dat}} \qquad (1)$$

where T_0 is a technology parameter. Although we ultimately want to come up with a similar dependability prediction, at the first glance it seems that we cannot gain much from this existing knowledge for our problem:

- Our target circuit is the Muller C-Element, not a latch; therefore the circuit model must be adapted.
- The Muller C-Element has a different function; in particular it does not “capture”, but rather performs a continuous mapping between inputs and output. Therefore the homogeneous treatment seems inappropriate.
- In our asynchronous circuit there is no strictly synchronous activity; as a result we cannot define a resolution time.
- The probability of input events that trigger metastability is not a priori known.

Consequently, as our next step, we need to derive a circuit model for the Muller C-Element which we can then use as a building block for the model of the elastic pipeline we are actually interested in. Based on the circuit diagram of our Muller C-Element library cell (which is a van Berkel type of circuit [31]) we propose the model shown in Fig. 4 (which we believe is suitable for other circuit types as well).

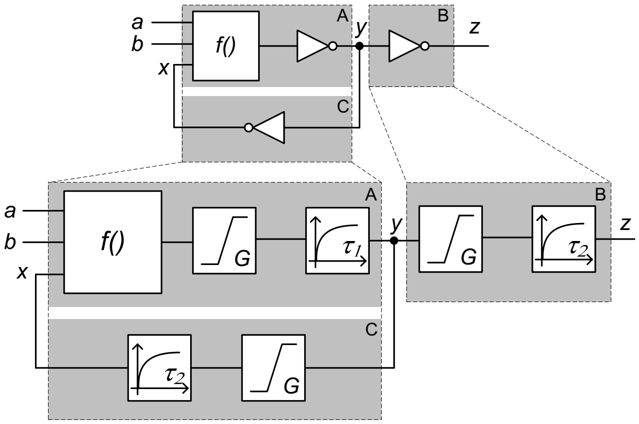

Fig. 4. Proposed model for the Muller C-Element.

The forward path comprises two inverter stages, a first one (shaded block A in Fig. 4) that implements the logic combination $f()$ of the inputs a,b and the feedback x from block A’s output y, and a second one (block B) that is just an ordinary inverter. The output of the latter forms the output z of the whole Muller C-Element. In van Berkel’s basic circuit this output is directly fed back to the input of the first stage. In practice, however, an extra inverter (block C) is often used for this feedback to prevent loading effects from influencing the storage loop formed by the feedback. We have chosen to model the latter implementation, as it is the one we have used in our chip, too. Like in the latch model from Fig. 3 we model the inverter as an amplifier followed by an RC low-pass filter. As already mentioned the first stage additionally comprises a (time free) function block $f()$ for the signal combination. The important property here is to be able to consider analog values of the inputs and determine an analog output voltage, but in a much simpler manner than by using the exact transistor characteristics. This simplicity shall allow us to reduce the analysis to the main causes and effects, and makes an extensive iterative simulation of a multi-stage pipeline under varying input and circuit parameters tractable. As for the modeling we have chosen to use an appropriate combination of “minimum” and “maximum” functions over input (and intermediate) voltages to properly reflect the signal combination performed by series and parallel connection of transistors.

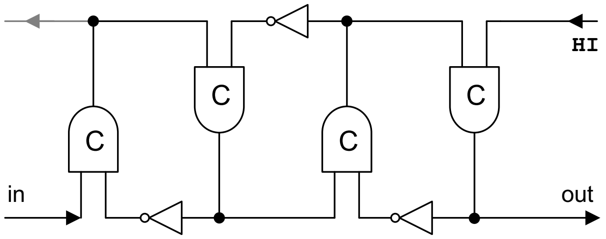

Fig. 5. Elastic pipeline with input $in(t)$ and output $o(t)$

By tuning the short input pulse response of our model to that of the SPICE model of our ASIC standard cell Muller C-Element we obtained the following parameter values $\tau_1 = 0.11ns$, $\tau_2 = 0.1ns$ and $G = 10$.

It should be noted that while our model shows striking similarities with the traditional latch model there are two differences beyond the existence of the function block $f()$:

1) As block A includes the function $f()$, it has different complexity, than the ordinary inverters B and C, hence it is unrealistic to assume equal RC constants (as is usually done for the inverters in the latch model for simplicity).
2) The amplifier in our simulation model cuts off the output at the positive/negative supply limits.

By appropriately combining the Muller C-Element models as in the Counter Module in Fig. 2 we can finally build a model for our elastic pipeline, which is depicted in Fig. 5. It has a single input in and output[2] out. The following analysis/simulations will investigate under which input signal conditions, the output out becomes metastable. For this purpose we further assume an empty pipe, waiting for the next input transition. In order to account for the interconnect delays between the C-Elements we have introduced pure delays in the connections. In the next section we will proceed by using this model to study the effects and the propagation of metastability in the elastic pipeline through both, analytic modeling and simulation. While analytic modeling shall help us get a feeling for the decay of a metastable state over time, we will study its propagation by means of simulations. To this end we will apply a short pulse at the elastic pipeline’s input and vary its length to create adverse behavior. Although in practice a marginal pulse amplitude may trigger metastability as well, we will only consider pulses with full scale amplitude and marginal pulse width. A variation of the amplitude would prohibitively increase the parameter space, and moreover we believe that our approach, while being more tractable, still reflects all relevant effects inside the elastic pipeline.

[2] In our case this corresponds to the output port connected to the leftmost PCSG input.

V. Quantitative Assessment

We will start our analysis with deriving a closed form equation for the metastability decay of the last pipeline stage.

A. Metastability decay

Let us first reconsider the function of a Muller C-Element: Even for an input pulse that is much longer than the Muller C-Element's settling time, the output only changes under certain "arming conditions", while the output retains its value otherwise. Metastability can only be triggered by marginal pulses on "armed" inputs. Let us consider an example: A positive input pulse on input a will only cause an output transition, if the input b is already at logic HI, while the output z is at logic LO. Under these conditions b has no other effect than opening all internal paths for a and/or x to move the output to logic HI. In the beginning it is solely up to the pulse at a to cause z to rise. When this has occurred, the identical feedback x will finally keep the output at logic HI independent of the level of a. However, before this feedback arrives, a is still responsible for keeping the output at HI. Metastability will occur, if a transits to logic LO just when the feedback is about to arrive. At the very point when a changes to LO the closed loop from y over the feedback into x is left on itself and determines the voltage level at the output. Notice the strong analogy to the case of the D-latch here: The trailing edge of the pulse at a triggers a "sampling" of the feedback signal x, just as in the case when a D-latch is switched from transparent to opaque.

Assuming that all pipeline stages are near the metastable state, we are interested in how the parameters G, τ_1 and τ_2 influence the decay time. For simplicity of mathematical treatment, we assume without loss of generality here that the input value domain is symmetrical $[-0.5, 0.5]$ around 0, i.e. -0.5 corresponds to a logic LO and 0.5 to a logic HI value. Similar assumptions have been made for the model of the D-latch in [10], e.g.

Under the assumption that the C-Element is near the metastable balance point, its amplifiers behave linearly yielding the following set of differential equations for the two stages' output voltages $z = x$ and y. According to the arguments above we can concentrate on the homogeneous solution, where, $f(a, b, x) = x$,

$$y = Gx - \tau_1 y'$$
$$z = x = Gy - \tau_2 x'.$$

A Laplace transform leads to

$$Y = GX - \tau_1(sY - y_0)$$
$$X = GY - \tau_2(sX - x_0)$$

where y_0, respectively, x_0 are the initial values of y, respectively, x at time 0. After algebraic transformations this yields

$$X = \frac{sx_0 + (x_0/\tau_1 + Gy_0/\tau_2)}{s^2 + s(1/\tau_1 + 1/\tau_2) + (1 - G^2)/(\tau_1\tau_2)}.$$

Partial fraction decomposition (assuming different singularities), leads to

$$X = \frac{\frac{\gamma+\delta(\alpha+\beta)}{2\beta}}{s - (\alpha + \beta)} - \frac{\frac{\gamma+\delta(\alpha-\beta)}{2\beta}}{s - (\alpha - \beta)}$$

with

$$\alpha = -\frac{1}{2}\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)$$
$$\beta = \frac{1}{2\tau_1\tau_2}\sqrt{\tau_1{}^2 + \tau_2{}^2 + (4G^2 - 2)\tau_1\tau_2)}$$
$$\gamma = \frac{y_0 G}{\tau_2} + \frac{x_0}{\tau_1}$$
$$\delta = x_0.$$

The inverse Laplace transform finally leads to

$$x(t) = z(t) = e^{\alpha t}\left(\frac{\gamma + \delta\alpha}{\beta}\sinh(\beta t) + \delta\cosh(\beta t)\right) \qquad (2)$$

In contrast to the latch analysis, we do not assume matched y_0, x_0 for the two stages. However, for simplicity, we assume, that $x_0 = 0$ at time 0 (i.e. the feedback path is in the middle of a transition), while y_0 is arbitrary (depending on the stage's input history). This leads to an algebraic solution of how fast the output z will decay from metastability. From (2) we thus obtain

$$z(t) = \frac{\gamma}{2\beta}\left(e^{(\alpha+\beta)t} - e^{(\alpha-\beta)t}\right).$$

With the additional assumptions $\alpha < 0$, $\beta > 0$ and $G \geq 1$ (which are fulfilled for meaningful circuit parameters) we further can approximate

$$z(t) \approx \frac{\gamma}{2\beta}e^{(\alpha+\beta)t}. \qquad (3)$$

Interestingly, although at first sight our setup is substantially different from the latch analysis in the synchronous design paradigm, (3) is analogous to (1) obtained for latch metastability decay. This is a strong indication that the synchronizing property of a single stage of the elastic pipeline compares quite well to that of a D-latch in the synchronous case.

B. Simulation

Because of the non-linearities in our elastic pipeline model (recall that our inverters have linear gain G but cutoff at positive and negative supply voltage), the above considerations on the output behavior are limited to the region around the metastability balance point. In the following we will generalize these by presenting numerical results that we obtained from simulating the complete non-linear elastic pipeline model. To quantitatively characterize the metastability decay properties of elastic pipelines, we feed input in of the elastic pipeline from Fig. 5 (comprising 4 stages) with perfectly shaped pulses of varying width. For simplicity (and in contrast to Section V-A), we assume a normalized voltage domain of

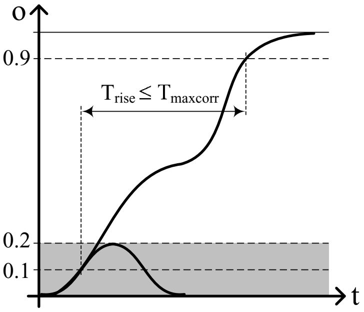

Fig. 6. Correctness criteria for the output $o(t)$. The output must either (i) remain in the gray region or (ii) have a bounded transition time T_{rise}.

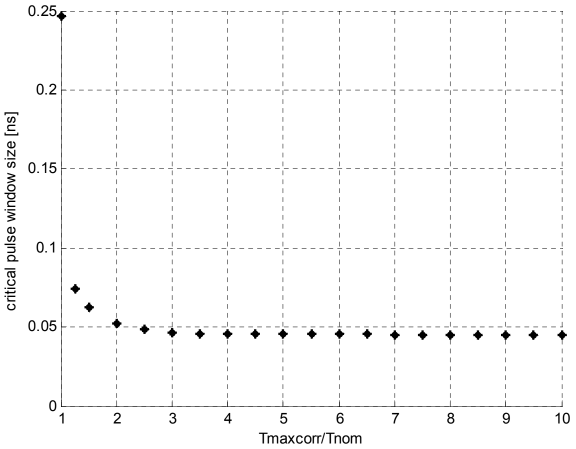

Fig. 7. Critical pulse window size vs. $T_{maxcorr}/T_{nom}$

$[0, 1]$, i.e., our input function space is the set of all functions $i \in I : \mathbb{R} \rightarrow [0, 1]$, s.t.,

$$i(t) = \begin{cases} 0 & \text{for } 0 \leq t < t_b \\ 1 & \text{for } t_b \leq t < t_b + \Delta \\ 0 & \text{for } t_b + \Delta \leq t \end{cases}$$

for a constant $t_b > 0$, and varying pulse width Δ. Note, that t_b must be chosen large enough to ensure that initialization of the elastic pipeline has stabilized. Clearly, for large Δ, the output produces a valid LO to HI transition. We say that a pulse $i(t)$ is a *critical pulse*, iff feeding the elastic pipe with $i(t)$ does not produce a valid output at the last stage's output out. We further define a *valid output* $o(t)$, as one for which either

$$\forall t : o(t) \leq 0.2, \tag{4}$$

i.e., it is a valid LO output, or

$$\forall t_1 \leq t_2 : o(t_1) \in (0.1, 0.9) \wedge o(t_2) \in (0.1, 0.9) \Rightarrow \\ t_2 - t_1 \leq T_{\mathit{maxcorr}} = 3T_{\mathit{nom}}, \tag{5}$$

i.e., it has rise time less than the maximum correct rise time T_{maxcorr}, defined as three (the value of three will be motivated later in this section) times the nominal rise time T_{nom}. T_{nom} is obtained by measuring the rise time of the elastic pipeline when fed with a long pulse that clearly does not generate metastability. The motivation for both criteria, is that (i) pulses which remain beneath 20% of voltage are consistently recognized as LO and (ii) those which traverse the $(0.1, 0.9)$ not too slowly are recognized as correct LO to HI transitions. The correctness criteria (4) and (5) are graphically depicted in Fig. 6. The value of $T_{\mathit{maxcorr}}/T_{\mathit{nom}} = 3$ for the quotient of the maximum correct rise time T_{maxcorr} and the nominal rise time T_{nom} in (5) was chosen in accordance with simulations for different $T_{\mathit{maxcorr}}/T_{\mathit{nom}}$. The simulation, like all subsequent simulations, were carried out in MATLAB with a stiff/trapezoidal ODE and maximum step size of $1ps$. Since, for the parameter setting of $\tau_1 = 0.1, \tau_2 = 0.11, G = 10$ and inter-stage connect $d = 0.5ns$, results in the order of the attainable quantization error $(1fs)$ were obtained, we set $G = 1.66$ during the experiments, unless otherwise stated. In the sequel let the *critical pulse window* be the smallest interval $[\Delta_L, \Delta_U]$, such that all critical pulses have pulse width $\Delta \in [\Delta_L, \Delta_U]$. We will further call $\Delta_U - \Delta_L$ the *critical pulse window size*. The simulation results for varying $T_{\mathit{maxcorr}}/T_{\mathit{nom}}$ are presented in Fig. 7. It can be seen, that, while for low values of $T_{\mathit{maxcorr}}/T_{\mathit{nom}}$ (i.e. when requiring shorter rise time) the critical pulse window size decreases dramatically, for values ≥ 3 it remains nearly constant. Thus analogous simulation results are obtained for all $T_{\mathit{maxcorr}}/T_{\mathit{nom}} \geq 3$, making (5) a plausible correctness criterion for the rise time.

Synchronizer-like behavior. From Section V-A we know that a stage of the elastic pipeline, namely a Muller C-Element, behaves like a classical D-latch with respect to metastability decay. This is promising insofar, as cascading elastic pipeline stages then should have a similar positive effect on metastability "filtering" as cascading latches in a synchronizer in the synchronous case. The first set of simulations thus is intended to determine the critical pulse window and the critical pulse window size for elastic pipelines of size $1 \leq n \leq 4$. The results are depicted in Fig. 8 (where the critical pulse window is plotted as a vertical line) and Fig. 9. As can be seen from Fig. 9, the critical pulse window size decreases exponentially with the number of stages (about ten times per stage). From Fig. 8 we can deduce, that the window size is mainly reduced by attenuation of short pulses to valid LO outputs (also termed electrical masking). Furthermore we can observe that the critical pulse width is below 125ps in all cases, which means that, considering our $0.18\mu m$ technology, only extremely short pulses are capable of creating metastability. This allows the conclusion that it is extremely unlikely to encounter such failures.

Variation of parameters. A further interesting question is, how the parameters τ_1, τ_2 and G influence the metastability decay property of an elastic pipeline. For this purpose we used a 4 stage pipeline and varied each of the three parameters, while holding the others constant. The resulting critical pulse windows for varying G are plotted in Fig. 10, the region

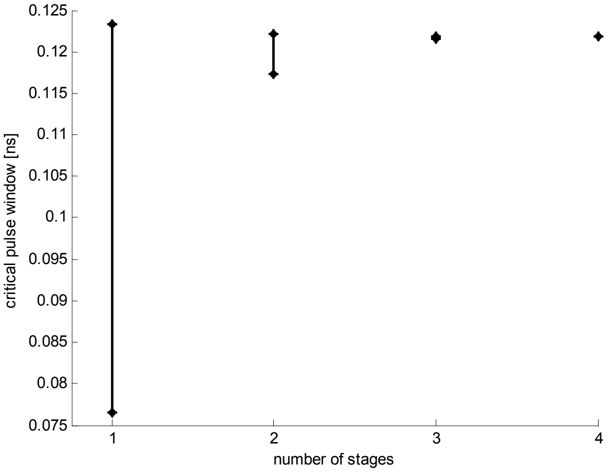

Fig. 8. Critical pulse window vs. number of pipeline stages

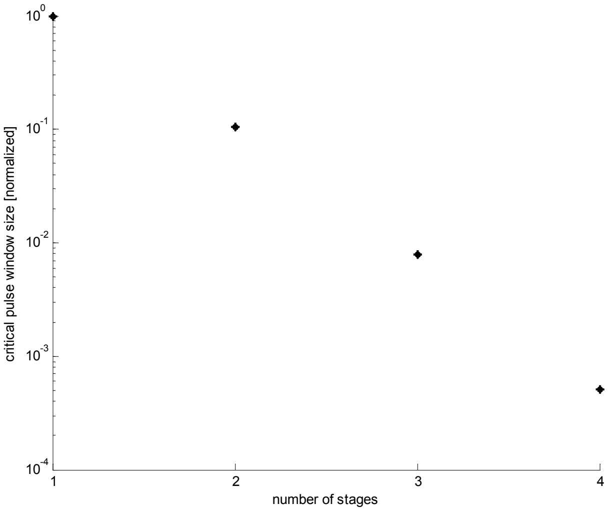

Fig. 9. Critical pulse window size vs. number of pipeline stages

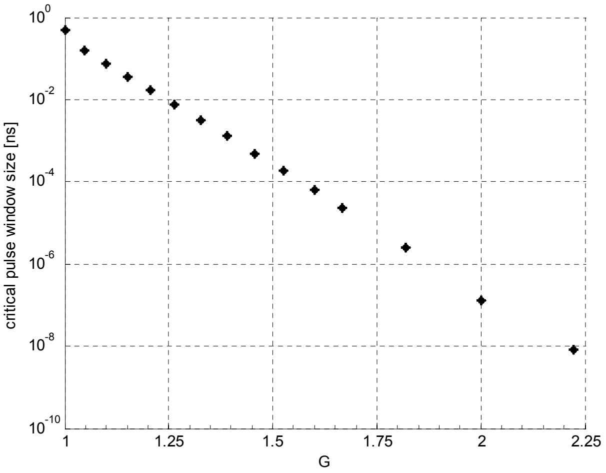

Fig. 12. Critical pulse window size vs. inverter gain G

around $G = 1.66$ in detail in Fig. 11 and the critical pulse window sizes for $G \leq 2.5$ in Fig. 12. Note, that the simulations were performed for larger G, too. However, the obtained window sizes were in the order of the quantization error of $1fs$. It can be observed that while variations of G have a negligible effect on the critical pulse window position (changing G from 5 to 10 yields less than 1% change in the critical pulse window), they do have a tremendous (exponential) effect in its size: Changing G from 1 to 2 reduces the critical pulse widow size by about 7 orders of magnitude.

Fig. 13 and Fig. 14 depict the critical window and its size for varying τ_1 and τ_2. Both abscissas have been normalized to $\tau_1 = 0.1, \tau_2 = 0.11$. It can be observed that both time constants only have an approximately linear dependence on the critical window size. A doubling of τ_1, e.g. roughly doubles the critical pulse window size.

As a result of our parameter analysis we can conclude that both, critical pulse window as well as its size are only moderately sensitive to variations in the RC constants. In contrast, variations in the inverter gain G have a tremendous effect, particularly on the critical pulse window size. Like in the case of the traditional D-latch, high inverter gain (or more generally high gain bandwidth product) tends to harden the design against metastability.

Quantitative projections. Based on our results it seems safe to conclude that the probability of metastability generation and propagation in an elastic pipeline is extremely low, and many similarities to the traditional synchronizers can be identified. Still, however, one should be careful when trying to derive quantitative MTBU predictions from our results: One reason is that the probability of encountering malicious short transients solely depends on the actual environmental conditions and the properties of the target technology under consideration. Knowledge of these parameters is indispensable for quantifying the probability of metastability generation.

Another reason is the validity of our model: In our trend studies we not only had to vary the considered parameters but also needed to "finetune" the input pulse size against both borders Δ_L and Δ_U (iteratively) to determine the respective critical window each time. The resulting huge number of simulation runs explains why we have deliberately restricted our model to the essential effects. Punctual correspondence checks that we performed with the full SPICE model confirmed all qualitative trends that we discussed above.

However, when calculating quantitative probabilities below 10^{-10} one cannot simply disregard second order effects, nonlinearities, parasitics and numerical inaccuracies. It is therefore not surprising that we have identified considerable mismatches between our simple model predictions and SPICE simulations with respect to quantitative results, particularly for extremely short pulse widths.

VI. Conclusions

At the example of our DARTS clock generation circuit we have illustrated that asynchronous circuits — even those

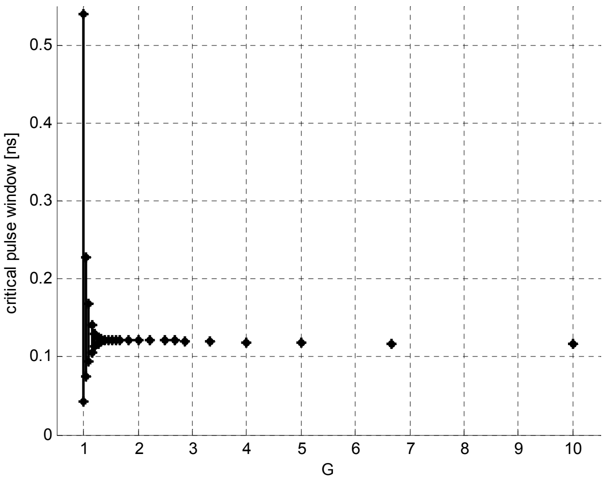

Fig. 10. Critical pulse window vs. inverter gain G

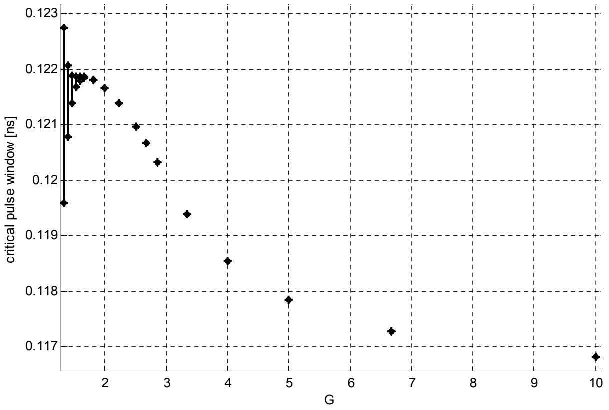

Fig. 11. Critical pulse window vs. inverter gain G (close-up)

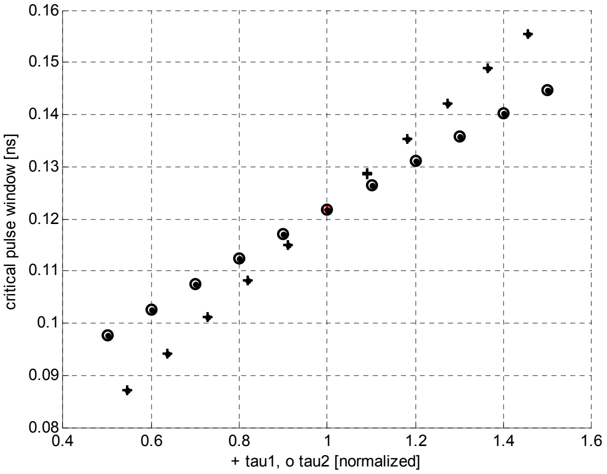

Fig. 13. Critical pulse window vs. RC constants τ_1 and τ_1

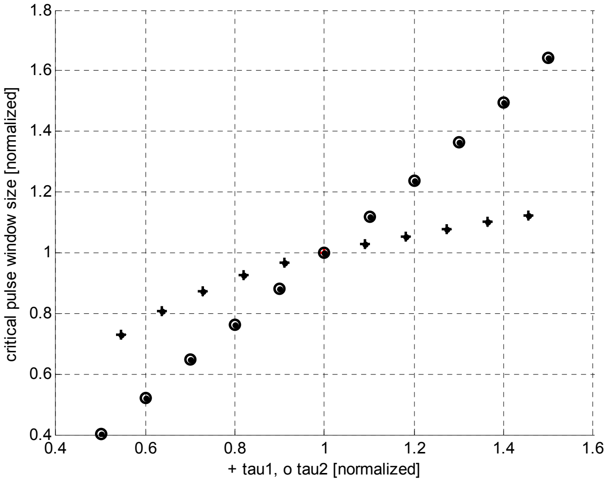

Fig. 14. Critical pulse window size vs. RC constants τ_1 and τ_2

that are completely free of metastability during normal operation — have a non-zero potential of becoming metastable under the impact of faults. We have argued that in principle metastable effects are capable of overcoming traditional error containment boundaries like masking and thus are particularly dangerous.

To facilitate a quantitative assessment of this threat we have proposed a model for the Muller C-Element, based on which we performed both, an analytic examination as well as a simulation study of metastability propagation in an elastic pipeline. Although the existing metastability models from the synchronous domain could not directly be applied to our problem, our solution showed many similarities to them. Our results consistently confirm that a Muller C-Element performs like a D-latch in resolving metastable events. In the specific case of the elastic pipeline, there is a small window of critical input pulse widths capable of producing a metastable event that propagates to the pipeline output. For the circuit parameters we identified for our $0.18\mu m$ chip implementation the critical window turned out smaller than 1fs, which allows us to faithfully judge the occurrence of metastable upsets at the pipeline output as highly improbable. A variation analysis showed that the size of the critical window linearly depends on the actual values of the RC time constants, while it is exponentially reduced for higher gain of the CMOS inverters and for increasing number of pipeline stages.

It would be tempting to apply our results for computing concrete failure probabilities and relate them to those due to other fault sources. While the presented approach is definitely useful in this context, several further steps need to be taken: Other masking effects need to be considered, and complementary information, in particular the probability of fault pulses with critical length must be known. Moreover, several aspects of the model need to be refined in order to provide trustworthy quantitative predictions in the range of extremely low failure probabilities. An extension of the model to further circuit elements, such as SR-latches, would also be interesting. These issues form the outline of our future work.

Acknowledgements

The authors want to thank the participants of the Dagstuhl seminar on *Fault-Tolerant Distributed Algorithms on VLSI Chips*[3] for the stimulating discussions on our early ideas. Special thanks go to Ulrich Schmid for ogranizing this seminar and for pointing us to this topic as well as for his valuable inputs that contributed to this work.

References

[1] D. C. Pham, T. Aipperspach, D. Boerstler, M. Bolliger, R. Chaudhry, D. Cox, P. Harvey, P. M. Harvey, H. P. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Pham, J. Pille, S. Posluszny, M. Riley, D. L. Stasiak, M. Suzuoki, O. Takahashi, J. Warnock, S. Weitzel, D. Wendel, and K. Yazawa, "Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 179–196, 2006.

[2] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," *IEEE Micro*, vol. 23, no. 4, pp. 14–19, Jul. 2003.

[3] M. J. Gadlage, P. H. Eaton, J. M. Benedetto, M. Carts, V. Zhu, and T. L. Turflinger, "Digital device error rate trends in advanced CMOS technologies," *Nuclear Science, IEEE Transactions on*, vol. 53, no. 6, pp. 3466–3471, Dec. 2006.

[4] A. J. Martin, "Limitations to delay-insensitivity in asynchronous circuits," in *Sixth MIT Conference on Advanced Research in VLSI*, 1990, pp. 263–278.

[5] D. M. Chapiro, "Globally-Asynchronous Locally-Synchronous Systems," Ph.D. dissertation, Stanford University, Oct. 1984.

[6] L. Marino, "General theory of metastable operation," *IEEE Transactions on Computers*, vol. C-30, no. 2, pp. 107–115, February 1981.

[7] L. Lamport, "Buridan's principle," SRI Technical Report, Tech. Rep., 1984.

[8] D. Kinniment and D. Edwards, "Circuit technology in a large computer system," in *Proc. Conf. on Computers – Systems and Technology*. Institution of Electronic and Radio Engineers, 1972, pp. 441–450.

[9] H. J. M. Veendrick, "The behavior of flip-flops used as synchronizers and prediction of their failure rate," *IEEE Journal of Solid-State Circuits*, vol. SC-15, no. 2, pp. 169–176, April 1980.

[10] L. Kleeman and A. Cantoni, "Metastable behavior in digital systems," *IEEE Design & Test of Computers*, pp. 4–19, December 1987.

[11] J. U. Horstmann, H. W. Eichel, and R. L. Coates, "Metastability behavior of cmos asic flip-flops in theory and test," *IEEE Journal of Solid-State Circuits*, vol. SC-24, no. 1, pp. 146–157, February 1989.

[12] D. J. Kinniment, A. Bystrov, and A. V. Yakovlev, "Synchronization circuit performance," *IEEE Journal of Solid-State Circuits*, vol. SC-37, no. 2, pp. 202–209, February 2002.

[13] Y. Semiat and R. Ginosar, "Timing measurements of synchronization circuits," in *Proc. IEEE Int. Symp. on Asynchronous Circuits and Systems*. IEEE Computer Society Press, 2003, pp. 1–10.

[14] D. J. Kinniment, K. Heron, and G. Russell, "Measuring deep metastability," in *Proc. IEEE Int. Symp. on Asynchronous Circuits and Systems*. IEEE Computer Society Press, 2006, pp. 1–10.

[15] C. Dike and E. Burton, "Miller and noise effects in a synchronizing flip-flop," *IEEE Journal of Solid-State Circuits*, vol. SC-34, no. 6, pp. 849–855, June 1999.

[16] C. L. Portmann and T. H. Y. Meng, "Supply noise and synchronization errors," *IEEE Journal of Solid-State Circuits*, vol. SC-30, no. 9, pp. 1015–1017, September 1995.

[17] D. J. Kinniment, C. E. Dike, K. Heron, G. Russell, and A. V. Yakovlev, "Measuring deep metastability and its effect on synchronizer performance," *IEEE Transactions on VLSI Systems Circuits*, vol. 15, no. 9, pp. 1028–1039, September 2007.

[18] J. Zhou, D. Kinniment, G. Russell, and A. Yakovlev, "Adapting synchronizers to the effects of on chip variability," in *Proc. IEEE Int. Symp. on Asynchronous Circuits and Systems*. IEEE Computer Society Press, 2008, pp. 39–47.

[19] M. Bhushdan, M. B. Ketchen, and K. K. Das, "Cmos latch metastability characterization at the 65-nm-technology node," in *Proc. IEEE Conference on Microelectronic Test Structures*. IEEE Computer Society Press, 2008, pp. 147–151.

[20] D. Shang, A. Yakovlev, F. Burns, F. Xia, and A. Bystrov, "Low-cost online testing of asynchronous handshakes," in *Proceedings Eleventh IEEE European Test Symposium*, May 2006, pp. 225–232.

[21] N. Seifert, P. Shipley, M. Pant, V. Ambrose, and B. GiII, "Radiation-induced clock jitter and race," in *Proceedings 43rd Annual IEEE International Reliability Physics Symposium*, 17-21, 2005, pp. 215–222.

[22] T. K. Srikanth and S. Toueg, "Optimal clock synchronization," *Journal of the ACM*, vol. 34, no. 3, pp. 626–645, Jul. 1987.

[23] J. Widder, "Distributed computing in the presence of bounded asynchrony," Ph.D. dissertation, Vienna University of Technology, Fakultät für Informatik, 2004.

[24] I. E. Sutherland, "Micropipelines," *Communications of the ACM, Turing Award*, vol. 32, no. 6, pp. 720–738, Jun. 1989, iSSN:0001-0782.

[25] M. Fuegger, U. Schmid, G. Fuchs, and G. Kempf, "Fault-Tolerant Distributed Clock Generation in VLSI Systems-on-Chip," *Sixth European Dependable Computing Conference (EDCC-6)*, Oct. 2006.

[26] M. Ferringer, G. Fuchs, A. Steininger, and G. Kempf, "VLSI Implementation of a Fault-Tolerant Distributed Clock Generation," *IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT2006)*, Oct. 2006.

[27] G. Fuchs, M. Fuegger, A. Steininger, and F. Zangerl, "Analysis of constraints in a fault-tolerant distributed clock generation scheme," *3rd International Workshop on Dependable Embedded Systems (WDES'06)*, Oct. 2006.

[28] L. Lamport, "Using time instead of timeout for fault-tolerant distributed systems," *ACM Transactions on Programming Languages and Systems*, vol. 6, no. 2, pp. 254–280, Apr. 1984.

[29] S. Yang and M. Greenstreet, "Computing synchronizer failure probabilities," in *Proc. Intl. Conference on Design Automation and Test in Europe*. IEEE Computer Society Press, 2007.

[30] Y. Monnet, M. Renaudin, and R. Leveugle, "Asynchronous circuits transient faults sensitivity evaluation," in *Proceedings 42nd Design Automation Conference*, June 2005, pp. 863–868.

[31] K. van Berkel, "Beware the isochronic fork," *Integr. VLSI J.*, vol. 13, no. 2, pp. 103–128, 1992.

[3] Dagstuhl seminar webpage: *http://www.dagstuhl.de*