Temporal Properties of Self-Timed Rings

Anthony Winstanley and Mark Greenstreet

Department of Computer Science, University of British Columbia, 2366 Main Mall, Vancouver, BC V6T 1Z4, CANADA
fwinstan, mrgg@cs.ubc.ca

Abstract. Various researchers have proposed using self-timed networks to generate and distribute clocks and other timing signals. We consider one of the simplest self-timed networks, a ring, and note that for timing applications, self-timed rings should maintain uniform spacing of events. In practice, all previous designs of which we are aware cluster events into bursts. In this paper, we describe a dynamical systems approach to verify the temporal properties of self-timed rings. With these methods, we can verify that a new design has the desired uniform spacing of events. The key to our methods is developing an appropriate model of the timing behaviour of our circuits. Our models are more accurate than the simplistic interval bounds of timed-automata techniques, while providing a higher level of abstraction than non-linear differential equation models such as SPICE. Evenly spaced and clustered event behaviours are distinguished by simple geometric features of our model.

1 Introduction

Like many problems in hybrid systems, the problem of analyzing temporal clustering of events in self-timed rings involves a modeling problem. Clustering shows up in detailed circuit models (e.g. SPICE), but such models are too detailed to provide insight into the causes of the clustering or possible solutions. At the other extreme, timed automata tools and logic simulators use bounded delay models that are too coarse to distinguish the timing modes of self-timed circuits. We require a model that captures the non-linear timing dependencies of real circuits while abstracting away most of the details of a low-level circuit model.

We propose a new model of timing behaviour. The time of the output event of a logic element is expressed as a function of the times of the input events and the time of the previous output. This model recognizes the fact that gate delay times depend not only on the time of the last input event, but also on the time separation of these events. For a two input gate, this model can be represented as a surface in a three dimension space. In a regular structure such as a self-timed ring, the times of events are determined by an iterated map on this surface. We identify the fixpoints of this map and the stability properties of these equilibria. In particular, we show that self-timed rings exhibit critical behaviours that lead to phase transitions with hysteresis between evenly spaced and clustered equilibria. These phase transitions are in the *timing* of events; the *logical* behaviour of the circuit is unaffected. In other words, the timed, discrete event model gives rise to a continuous map with Hopf bifurcations. We present simple, geometric criteria for identifying these phenomena.

Self-timed rings are introduced in section 2. Section 3 describes our timing model, an extension of the "Charlie Diagrams" introduced in [EFS98]. Section 4 presents the main results of this paper; in particular, we derive geometrical criteria for classifying the behaviour of a ring as clustered or evenly spaced. In section 5, we apply our approach to two examples. In the first example, we use a simple, synthetic model that admits easy analysis and provides the motivation for our circuit design. In the second example, we describe a circuit that we have designed and is currently in fabrication in a 0.35μ CMOS process. This latter example shows the value of our model as a design guide while revealing numerical limitations of our current methods for generating Charlie Diagrams from SPICE models.

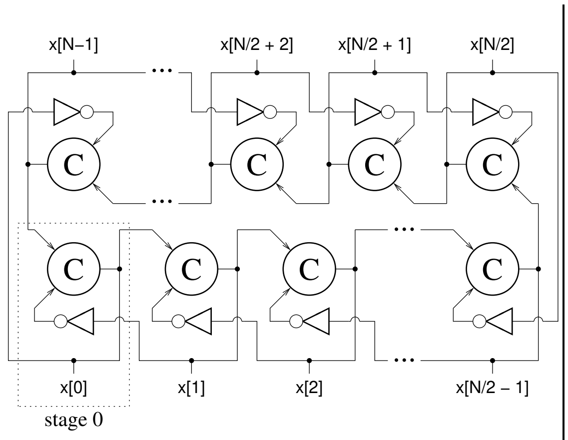

Fig. 1. A Self-Timed Ring

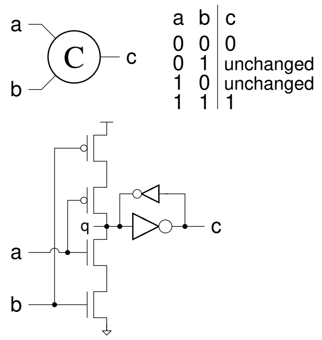

Fig. 2. A Muller-C Element

2 Self-Timed Rings

Self-timed circuits use handshake signals to control the sequencing of operations. These handshake signals take the role of clocks in synchronous designs. Figure 1 shows a ring composed of self-timed, handshaking stages (see [Sut89]). Each stage consists of a Muller C-element and an inverter. As illustrated by the transition table in figure 2, a C-element drives its output to the value of its inputs when its inputs agree; when the inputs of the C-element have different values, then the output of the C-element retains the value from the last time the inputs agreed.

The schematic in the lower half of figure 2 shows the CMOS implementation of the C-element used in this paper.

Figure 3 illustrates the operation of a self-timed ring. Let the output of n stages be represented by the array x$[0..n-1]$. By the operation of inverters and C-elements, stage i is enabled to change the value of its output, x$[i]$, when x$[i]$ differs from x$[i-1]$, and x$[i]$ is the same as x$[i+1]$. We say that a stage holds a "token" if the value of its output is different than that of its successor, and a "bubble" if their output values are the same. In figure 3, stages that

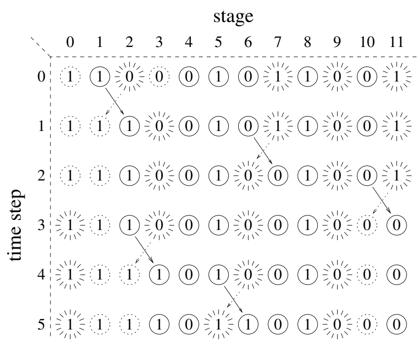

Fig. 3. Operation of Self-Timed Ring

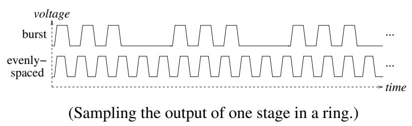

Fig. 4. Burst and Evenly Spaced Events

hold tokens are represented with a solid circle, ○; stages that hold bubbles and are not enabled are represented with a dotted circle, ◌; and stages that hold bubbles and are enabled are represented with a "star burst", ☼. It is convenient to rephrase the rules for when stages are enabled in terms of tokens and bubbles: stage i is enabled if it holds a bubble and stage $i-1$ holds a token. After stage i performs its action, stage i will hold a token, and stage $i-1$ will hold a bubble. In other words, performing an action exchanges a token and a bubble; tokens move forward around the ring, and bubbles move backward. The number of tokens in the ring is invariant, as is the number of bubbles. For example, if there are 6 tokens and 6 bubbles at time step 1, there will always be 6 tokens and 6 bubbles in the ring. The sum of the number of tokens and the number of bubbles is the number of stages.

In many applications of self-timed rings tokens and/or bubbles are used to convey data values. The invariants on the number of tokens and the number of bubbles guarantee that data values are neither lost nor duplicated. In our current analysis, we focus on the timing of operations in the ring and do not give further consideration to any data values that may be conveyed with the tokens or bubbles.

Self-timed rings are ubiquitous in self-timed designs (e.g. [Wil91,SS93]), and their performance has been studied in many contexts. For example, [Thi91] analyzed throughput assuming each stage has a fixed time for each operation. Self-timed rings with exponentially distributed processing times were analyzed in [GS90]. Xie and Beerel have developed tools that analyze general networks of self-timed processors for general probabilistic models [XB97,XKB99]. All of these analyses have focused on the long-term throughput of the self-timed network.

Although long-term throughput is an important measure, the details of time separation between consecutive events is also important. In particular, in most self-timed rings, events occur in "bursts" as depicted in figure 4. Noting that stages in the ring are idle between bursts, even spacing can produce higher performance. Furthermore, evenly spaced events are desirable if the self-timed network is used to generate or distribute clocks and other timing signals in an otherwise synchronous design. Our work was motivated by these applications where self-timed circuits are used within a synchronous system. Although even spacing is desirable, all designs that we had seen prior to the ones described in section 5 produced bursts of events. In the remainder of this paper, we present methods for analyzing the timing modes of self-timed rings.

3 Models

As with many hybrid systems, the key to understanding the timing properties of self-timed rings lies in finding an appropriate model for the ring. Circuit simulators such as SPICE [Nag75] use non-linear ordinary differential equations (ODEs) to model the circuit, and numerically integrate these equations to predict the circuit's behaviour. These ODE models can be quite accurate, and they correctly predict the burst behaviour that is observed by laboratory measurements. However, the device models are complicated, and even the models for small rings have dozens to hundreds of variables. Thus, whatever their virtues for accuracy, ODE models are too detailed to provide the insight into the causes of burst behaviour and how it can be controlled.

Another approach is to model the system as having discrete values that change at instants in continuous time. This is the approach taken by discrete event simulators (e.g. SHIFT [DGV96]) and timed automata techniques [LPY97,Yov97]. Close to our current problem, Amon and Hulgaard [HBAB95,AH99] have developed algorithms for computing bounds on the separation of events given bounds on operation times. All of these models specify the range of possible event times for each operation with an interval. Such models admit a wider range of behaviours than occur in practice. In particular, they show that bursts and evenly spaced events are both admitted by the models, but they don't predict which behaviour actually occurs.

Typical hardware delay models specify a delay after the *last* input event that enables the change. Using such a model, the time at which the output of a C-element changes, t_c is given by $\max(t_a, t_b) + \delta$, where t_a is the time of the change of the a input; t_b is the time of the change of the b input; and δ is some value with $\delta_{\min} \leq \delta \leq \delta_{\max}$. Let $t_{c,\max}$ be the latest time at which an output may change. For $t_a < t_b$, $\frac{\partial}{\partial t_a} t_{c,\max} = 0$, and for $t_a > t_b$, $\frac{\partial}{\partial t_a} t_{c,\max} = 1$. An equivalent observation holds for $\frac{\partial}{\partial t_a} t_{c,\min}$, and for derivatives with respect to t_b. The ODE models for circuits don't exhibit such discontinuities. To remain consistent with the ODE models, the delay intervals of these traditional hardware models must be fairly large. This is what makes them too imprecise for our purposes.

More accurate delay models account for the effects of closely spaced input events [DM+94,CS96] and intersymbol interference [DP98]. As described in section 3.2, when enabling input events are closely spaced, the delay from the last input event to the resulting output event is greater than when the input events are more widely separated. In [CS96], this phenomenon is modeled with function that applies a correction term to a delay model for a single enabling event. This is similar to the Charlie Diagram model described in section 3.1. However, the model in [CS96] lacks the continuity of the Charlie Diagram. In particular, for large separations, it assumes that the effect of the earlier signal on the output time is negligible. As described in section 5, even very small dependencies can be critical in determining whether a ring has evenly spaced or clustered events.

3.1 The 2D Charlie Diagram

The starting point for our models are the "Charlie Diagrams" first described in [EFS98]. Based on observations like those above, Charles Molnar (after whom the diagrams were named) proposed measuring the output delay from the *average* of the two input arrivals. This delay is parameterized by the half-difference of the arrival times:

$$\begin{array}{l} t_c = m + \mathrm{Charlie}(s) \\ \text{where:} \\ m = (t_a + t_b)/2 \\ s = (t_a - t_b)/2 \end{array} \tag{1}$$

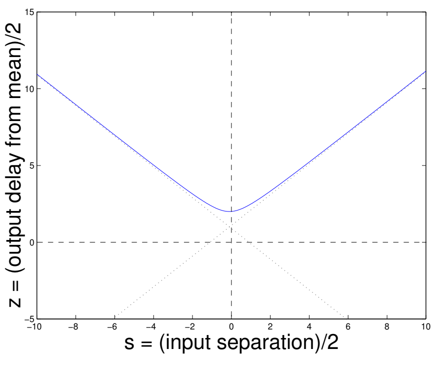

Fig. 5. A Charlie Diagram

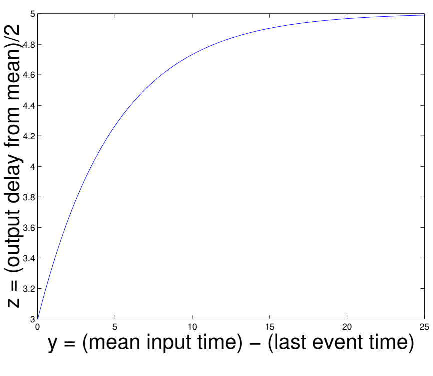

Fig. 6. "Drafting"

Figure 5 shows a typical Charlie Diagram. The curve of Charlie(s) versus s resembles a hyperbola. For large separations of the input events, the output time approaches the time of the last input plus some constant. Thus, the Charlie Diagram has asymptotes with slopes of ± 1. We assume that all stages in the ring are identical and that their behaviour is causal; these assumptions imply that the asymptotes of the Charlie Diagram intersect the vertical axis at positive values.

Charlie Diagrams can be used to model any two-input gate, and the obvious higher dimensional diagrams can be used to model gates with more inputs. We use Charlie Diagrams to model stages of the ring, where each stage consists of a C-element and an inverter. For stage i, signal x[i-1] is the a input to the stage; signal x[i+1] is the b input of the stage; and signal x[i] is the output of the stage. The forward delay of stage i is the time from receiving an event on signal x[i-1] until producing an event on x[i]. Likewise, the reverse delay is the time to propagate an event on x[i+1] to x[i]. Using the Charlie Diagram notation, we obtain:

$$\begin{array}{ll} \delta_F = t_c - t_a = \text{Charlie}(s) - s, & \text{forward delay} \\ \delta_R = t_c - t_b = \text{Charlie}(s) + s, & \text{reverse delay} \end{array} \tag{2}$$

3.2 The Charlie Effect

We now examine how the curve of a Charlie Diagram approaches the asymptotes. Consider a scenario where both a and b make low-to-high transitions, and a changes after b. If a changes a long time after b, then the p-channel transistor controlled by b will be in its cut-off region, and the n-channel transistor controlled by b will be fully conducting as a changes. Furthermore, the node between the two n-channel transistors will be close to ground potential. This allows a relatively fast transition on signal q and therefore on the output c. On the other hand, if a changes only slightly after b, then the transistors controlled by b will both be partially conducting as a changes. This results in a greater delay from the transition of a to the transition of c. Similar effects occur if a changes before b.

Charlie Diagrams provide a simple way to model these effects. If input a changes a long time after b, then s is large and positive. Conversely, if a changes only slightly before b, then s is small and positive. The delay from an event at input a to an event at output c is $t_c - t_a = \text{Charlie}(s) - s$. The dependence of output delay on relative arrival times described above is reflected in the curve of the Charlie Diagram approaching the asymptotes monotonically from above. In general, when enabling events for a multiple input gate arrive at roughly the same time, the delay from the last input event to the output is greater than when the arrival times of the events are widely separated. Because this effect is naturally modeled by Charlie Diagrams, we call this effect the "Charlie Effect".

The curves of the Charlie Diagrams described in [EFS98] monotonically approach their asymptotes from above. This is due to the Charlie Effect described above. As we show in section 4, this monotonicity implies that events are evenly spaced. In practice, most self-timed rings produce bursts of events. The authors of [EFS98] noted this discrepancy – it shows that their Charlie Diagram model neglects phenomena that are crucial to understanding the spacing of events.

3.3 The Drafting Effect

We extend the Charlie Diagram to model the effects of the output capacitance of the gate. Due to this capacitance, output transitions are not instantaneous. Instead, the voltage of the output asymptotically approaches the level of the power supply or ground. If input events are closely spaced, then the output of the gate will still be a significant distance from the power or ground rail when a new transition occurs. This allows subsequent transitions to occur faster than in the case where the output has reached a value closer to the rail. We call this phenomenon "drafting," after the practice of bicyclists to ride in closely spaced lines to reduce wind drag. Just as the lead cyclist reduces the work required of those behind her, the lead token in a burst allows subsequent tokens to propagate with reduced delay. The handshake protocol prevents trailing tokens from overtaking earlier ones (fear serves an equivalent purpose for bicyclists). As an example, figure 6 shows the time from an input event to the corresponding output event for our FIFO stage as a function of the input period. In this example, both inputs of the FIFO change simultaneously.

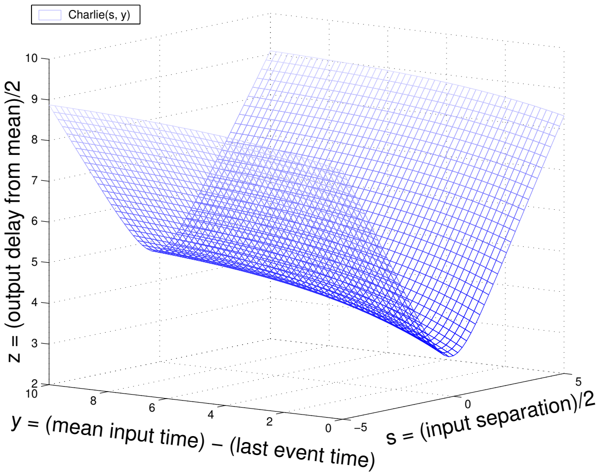

Fig. 7. A Three-Dimensional Charlie Diagram

3.4 The 3D Charlie Diagram

We model drafting by extending the Charlie Diagram to three dimensions. As with the original Charlie Diagram, the time separation of the input events is drawn along one axis of the domain. The other domain axis is the time from the last output event to the mean of the input event times. Figure 7 shows such a Charlie Diagram.

4 Analysis

Consider a burst of events propagating around a self-timed ring. We'll call the leading event in the burst event 1, and number the subsequent events consecutively. In the following analysis, we assume that the spacing of events in the burst remains invariant as the burst travels around the ring. This is intuitively plausible (otherwise, the burst would alternately expand and contract, etc. as it propagated). While we don't have a proof for this conjecture, we observed it in all cases for a wide variety of synthetic models and models from real circuits.

Now consider event 1. By the invariance assumption above, each stage that propagates event 1 has the same separation of input events, call this separation s_1. Likewise, each stage that propagates event 1 has the same time since its last output change, call this y_1. The values s_1 and y_1 specify a point on the 3D Charlie Diagram. These observations about event 1 apply to any event. If there are n_T tokens in the ring, and these tokens form a burst, then the operation of the ring is completely characterized by the values of s for each of the n_T events in the burst, and the n_T values for y. This cycle of n_T points on the 3D Charlie Diagram define an *operating point* for the ring.

In the remainder of this section, we will show how to identify the steady state operating points of a self-timed ring. This identification is based on geometric properties of the 3D Charlie Diagram. In particular, we show that our invariance assumption above implies that the forward latency, δ_F is the same for all event propagations. If the events form a burst, then there are two values of δ_R: one for the leading event in the burst, and the other for all other events. This means that the leading event of the burst corresponds to one point on the Charlie Diagram, and all other events correspond to one other point. If the events are evenly spaced, then all events occur at the same point on the Charlie Diagram.

More abstractly, the operation of the ring defines a mapping between points on the Charlie Diagram. We are interested in limit cycles of this mapping. In the rest of this section, we show how to locate and characterize these limit cycles.

4.1 The Constant ﬃF assumption

The conjecture stated above that the spacing of events in a burst remains invariant as the burst travels around the ring is equivalent to assuming that δ_F is the same for all events. We would like to find sufficient conditions on the Charlie Diagram than guarantee this conjecture. For the time being, we note that we have never observed a counter-example and will assume a fixed value for δ_F in the remainder of our analysis. We write δ_F^* to denote this common forward latency.

4.2 The $z = y$ Constraint

An enabling of stage i requires two input events: one from stage $i-1$ and the other from stage $i+1$. The $z = y$ constraint says that for any limit cycle of the Charlie Diagram, the midpoint of the input arrival times coincides exactly with the midpoint of the time of the last output event and the time of the output event triggered by these inputs. In this section, we show that this condition is implied by the definition of the Charlie Diagram and the handshake rules for the ring.

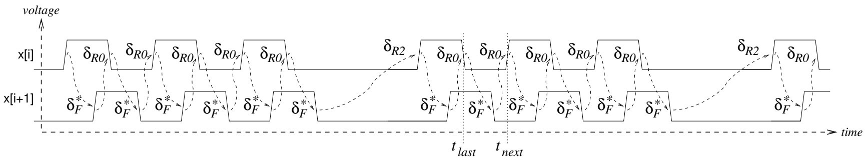

Fig. 8. Forward and Reverse Delays

Let t_{last} and t_{next} be the time of any two consecutive events at some stage i. Let $z = \text{Charlie}(s, y)$ where

$$y = m - t_{last} \tag{3}$$

and m and s are defined as in equation 1. Let $\delta_R \in \{\delta_{R0}, \delta_{R1}\}$ be reverse latency of stage i leading to the event at time t_{next}. We derive:

$$\begin{array}{rrcll} & t_{next} &=& m + z, & \text{eq. 1} \\ \wedge & m &=& t_{last} + y, & \text{eq. 3} \\ \wedge & t_{next} &=& t_{last} + \delta_F^* + \delta_R, & \text{see fig. 8} \\ \wedge & \delta_F^* + \delta_R &=& 2z, & \text{eq. 2} \\ \Rightarrow & y &=& z & \end{array} \tag{4}$$

Graphically, we can intersect the surface of a three-dimensional Charlie Diagram with the $y = z$ plane to obtain a curve that includes the operating points of any burst or evenly spaced equilibrium. We write $\text{Charlie}_{y=z}(s)$ to denote this curve. In the remainder of this section, we restrict our attention to the $\text{Charlie}_{y=z}(s)$ curve.

4.3 The δ_F^* Constraint

The assumption that δ_F^* is the forward latency for all events means that all points on a limit cycle lie on the intersecton of the $\text{Charlie}_{y=z}(s)$ curve with a line corresponding to δ_F^*. We call this the δ_F^* constraint.

More specifically, the points of a limit cycle lie at the intersection of

$$\begin{aligned} z &= \mathrm{Charlie}(s, y), \\ z &= y, \\ \text{and } z &= s + \delta_F^* \end{aligned} \tag{5}$$

If $\partial \mathrm{Charlie}_{y=z}(s)/\partial s < 1$ at the point of intersection, then that intersection is stable – in other words, if the ring is operating at a point near that intersection, it will converge to that stable point. On the other hand, if $\partial \mathrm{Charlie}_{y=z}(s)/\partial s > 1$, then the point of intersection is an unstable equilibrium.

Noting that $\partial \mathrm{Charlie}_{y=z}(s)/\partial s$ approaches -1 as s approaches $-\infty$ and $+1$ as s approaches $+\infty$ we classify five scenarios as exemplified in figures 9 and 10. In the "low drafting" case, $\partial \mathrm{Charlie}_{y=z}(s)/\partial s$ approaches $+1$ monotonically as s goes to infinity. For any value of δ_F^* greater than the asymptotic forward delay, the $z = s + \delta_F^*$ line intersects the $\mathrm{Charlie}_{y=z}(s)$ curve at a single point. As there is only one solution to equation 5, all events have the same operating point, and therefore the same reverse latency. Thus, the first scenario has evenly spaced events.

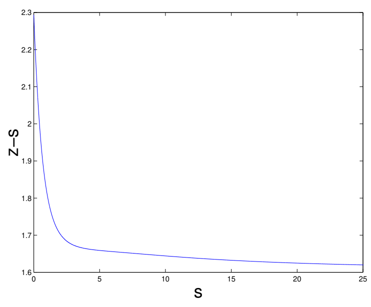

Fig. 9. z = $\mathrm{Charlie}_{y=z}$(s), low drafting

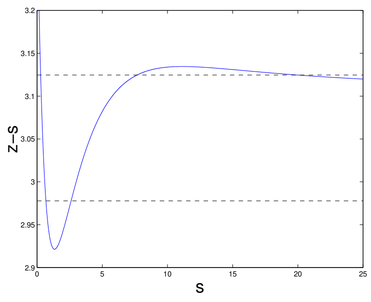

Fig. 10. z = $\mathrm{Charlie}_{y=z}$(s), high drafting

In the second scenario, $\partial \mathrm{Charlie}_{y=z}(s)/\partial s$ takes on values greater than one for some range of s. In this range, the drafting effect dominates the Charlie Effect, and forward delay increases with increases of the arrival time of the input token. If δ_F^* is large enough, then equation 5 is satisfied at three points: the leftmost and rightmost of these points are stable, and the inner one is unstable. The upper, dashed line in figure 10 illustrates this scenario. This solution gives rise to the burst of events: δ_{R0} corresponds to the right intersection, and δ_{R1} corresponds to the left intersection.

The third, fourth, and fifth scenarios arise for smaller values of δ_F^* as shown by the lower, dashed line in figure 10. This can either be a stable, evenly spaced mode where all events take place at the left intersection; an unstable, evenly spaced mode where all events take place at the right intersection; or an unstable, burst mode that exits to the stable burst mode or the evenly spaced mode.

4.4 Computing ffi$_F^\Lambda$

Not all values of δ_F^* are feasible for any particular number of tokens and bubbles. Every time a stage performs an action, it exchanges a token and a bubble. This imposes a balance in the rates of token and bubble propagation which in turn determines the feasible values for δ_F^*.

Let the number of tokens be n_T, and the number of bubbles be n_B. We derive another constraint by considering a token that makes one complete orbit of the ring. This token is processed by n stages, with a delay of δ_F^* at each stage. The total time for the orbit is $n\delta_F^*$. For any i, stage i processes each of the n_T tokens during the same interval. After processing one token, the stage waits δ_F^* for the bubble to come back from stage $i+1$, and then δ_R additional time to process the bubble. For the first token in a burst, $\delta_R = \delta_{R0}$. For the remaining $n_T - 1$ tokens, $\delta_R = \delta_{R1}$. Setting the time for one token to go around the ring equal to the time for one stage to process n_T tokens yields:

$$
\begin{aligned}
n\delta_F^* &= (n_T)\delta_F^* + \delta_{R0} + (n_T - 1)\delta_{R1} \\
\Rightarrow \quad 0 &= n_B\delta_F^* - \delta_{R0} - (n_T - 1)\delta_{R1}
\end{aligned}
\tag{6}
$$

For any value of δ_F^* we compute the intersection of $z = s + \delta_F^*$ and $z = \mathrm{Charlie}_{y=z}(s)$ to find the possible values of s and thereby the values of δ_{R0} and δ_{R1}. Thus, the right side of equation 6 is a function of δ_F^*; the roots of this function are the feasible values of δ_F.

In evenly spaced mode, $\delta_{R0} = \delta_{R1}$. In this case, equation 6 can be rewritten as $n_B\delta_F = n_T\delta_R$. Substitutions for equation 2 yield:

$$
z = \frac{n_T + n_B}{n_T - n_B} s \tag{7}
$$

4.5 Classification

Given a three-dimensional Charlie Diagram, and values for n_T and n_B, we classify by the following procedure:

1. Find the curve $z = \mathrm{Charlie}_{y=z}(s)$ by intersecting the surface of the Charlie Diagram with the $z = y$ plane.
2. Compute the intersection of $z = \mathrm{Charlie}_{y=z}(s)$ with $z = ((n_T + n_B)/(n_T - n_B))s$. This is the evenly spaced solution. If $\partial\mathrm{Charlie}_{y=z}(s)/\partial s < 1$ at the intersection, then the evenly spaced solution is stable (i.e. an attractor). If $\partial\mathrm{Charlie}_{y=z}(s)/\partial s > 1$, then the evenly spaced solution is unstable.
3. Determine $\max(\partial\mathrm{Charlie}_{y=z}(s)/\partial(s))$. If this value is greater than one, then burst behaviours are possible. If it is less than one, then burst behaviours are excluded.
4. If burst behaviours are possible, find the feasible values of δ_F^* by solving equation 6. If there is a solution with $\delta_{R0} \neq \delta_{R1}$ where $\partial\mathrm{Charlie}_{y=z}(s)/\partial(s) < 1$ at both points, then the ring has stable, burst behaviours.

5 Examples

We have applied the analysis of the previous section to a small handful of examples; both real circuits and synthetic models. We start with our synthetic modeling because it permits succinct description.

5.1 Synthetic Example

Out synthetic model consisted of a ring with 30 stages modeled by the family of 3D Charlie Diagrams described below. We analyzed the ring using the methods described in section 4 and observed the predicted phase transitions between clustered and evenly-spaced events using event driven simulation.

The two-dimensional Charlie Diagram from figure 5 is given by:

$$\mathrm{Charlie}(s) = 1 + \sqrt{1 + (s + 0.1)^2} \tag{8}$$

To simulate drafting, we added a delay that converges exponentially to a limit value:

$$\begin{aligned} u(s) &= 1 + \sqrt{1 + (s + 0.1)^2} \\ \mathrm{Charlie}(s, y) &= u(s) + b(1 - e^{-a(y+u(s))}) \end{aligned} \tag{9}$$

Note that $y + u(s)$ is the time from the previous firing until the gate would fire next without drafting. The exponential convergence to the asymptotic delay corresponds to a simple, first-order, RC delay. Figure 7 corresponds to this model with $a = 0.2$ and $b = 3.0$.

Varying b in equation 9 varies the strength of the drafting effect. Figure 9 corresponds $b = 0.5$, and figure 10 corresponds to $b = 2.0$ (both drawn with $a = 0.2$). For $a = 0.2$, evenly spaced events are the only stable behaviour for $b < 0.63166476$. For $0.63166476 < b < 1.18011777$, both evenly spaced and burst spacings are stable. As b increases in this interval, the intersection of the $z = \mathrm{Charlie}_{y=z}(s)$ curve with the $z = ((n_T + n_B)/(n_T - n_B))s$ line moves to the right. When it reaches the local minima of slope equals 1 point of the curve, the evenly spaced solution becomes unstable. For $b > 1.18011777$, only the burst behaviour is stable.

5.2 Circuit Example

Following the example of [EFS98], we initially considered an entire FIFO stage as a two-input module consisting of a C-element and an inverter. Based on our analysis, we assumed that drafting was the cause of event clustering. A signal that bounced away from the rail should generate a negative drafting effect and produce evenly spaced events. We tried this by adding small, delayed negative feedback circuitry to the output (c) of each stage in our ring.

This did not work as expected: events in the ring clustered into bursts regardless of the feedback that we applied. To understand this outcome, we attempted to construct 3D Charlie Diagrams for the stage using HSPICE, but this effort was hindered by the limited numerical resolution of HSPICE. At the operating point of the ring, the Charlie Diagram is flat to within the resolution of HSPICE.

Elementary continuity arguments imply that the surface must have some residual curvature, but we cannot observe this curvature directly. After further consideration, we realized that the bounce had a side effect of reducing the Charlie Effect. Consider a scenario where the c output of stage i is initially low and the f input from stage $i-1$ goes high before the r input from stage $i+1$ goes low. While waiting for the event from stage $i+1$, the output of stage $i-1$ *decreases* towards its asymptotic value. This *increases* the delay from the subsequent transition of stage $i+1$. Thus, while the negative feedback may have counteracted the drafting as desired, it also counteracted the Charlie Effect, resulting in a ring that continued to generate bursts of events.

To obtain the desired outcome, me moved the feedback point to the internal node, q of the C-element. This required some care to avoid spurious transitions. In the final solution, we use and N-channel pull-up to fight the keeper's pull-down and get good device matching. Likewise we fight the keeper's pull-up with another P-channel device. Our successful solution is shown in figure 11. We included

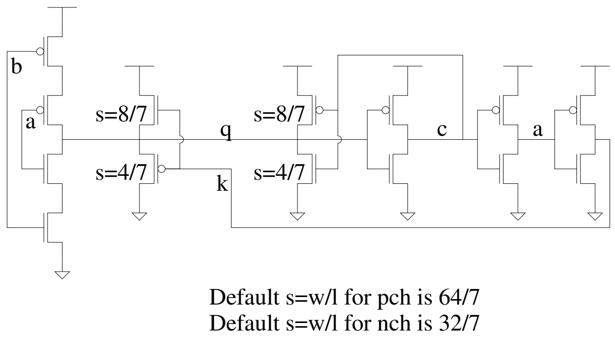

Fig. 11. A Ring Stage With Feedback

current mirrors (not shown in the figure) in series with the feedback devices to allow external control of the strength of the feedback for testing purposes. According to HSPICE simulations, this circuit can be started in burst-mode, and forced to evenly-spaced mode by a control input. Circuit considerations limit the range of our control input, and we can't force the ring back into bursting behaviour. Again, we would like to construct detailed Charlie Diagrams to better understand the behaviour of the circuit, but are hindered by numerical issues.

6 Conclusions and Future Work

We have presented a new model for reasoning about the timing behaviour of digital circuits: the three-dimensional Charlie Diagram. We have shown that the temporal properties of self-timed rings can be understood by examining fixpoints of iterated maps of this model. Simple geometric properties of the model distinguish evenly spaced and bursting behaviours. Three-dimensional Charlie Diagrams model behaviours that cannot be distinguished by traditional, timed-automata style models. At the same time, these Charlie Diagrams provide a much higher level of abstraction than circuit-level differential equation models (such as SPICE). This higher level of abstraction makes verification of timing properties practical.

While traditional timing analysis focuses on estimating delay values for gates and paths, we have presented an analysis that focuses on the *dynamics* of the timing. To this end, our models not only estimate the delays of logic elements but also the derivatives of these delays with respect to event times at the gate's inputs and outputs. We have shown that self-timed circuits have distinct timing modes, with phase transitions between these modes that exhibit hysteresis. Understanding these timing modes allows us to design circuits with timing properties not previously achieved.

We have used these techniques to design a new circuit for a self-timed ring stage. We are currently fabricating a chip based on this circuit to obtain experimental measurements and validate our models.

We currently generate three-dimensional Charlie Diagrams for our circuits through iterated SPICE simulations. In the future, we plan to explore ways of combining analytical and numerical techniques for generating these models to reduce the model generation time and improve model accuracy. In particular, by integrating SPICE's Jacobian matrix along paths between events, we should be able to compute an accurate value for the tangent plane of the Charlie Diagram at each point in spite of errors in the exact value of the point. We are also interested in ways to apply these techniques to other design, verification, and simulation problems.

References

[AH99] Tod Amon and Henrik Hulgaard. Symbolic time separation of events. In *Proceedings of the Fifth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 83–93. IEEE, April 1999.

[CS96] V. Chandromouli and K.A. Sakallah. Modeling the effects of temporal proximity of input transitions on gate propagation delay and transistion time. In *Proceedings of the 33th ACM/IEEE Design Automation Conference*, pages 617–622, June 1996.

[DGV96] A. Deshpande, A. Gollu, and P.P. Varaiya. SHIFT, a formalism and programming language for dynamic networks of hybrid automata. In *Proceedings of the Eigth Conference on Computer Aided Verification*, pages 113–133, 1996.

[DM+94] Florentin Dartu, Noel Menzes, et al. A gate delay model for high-speed CMOS circuits. In *Proceedings of the 31th ACM/IEEE Design Automation Conference*, pages 576–580, June 1994.

[DP98] William J. Dally and John W. Poulton. *Digital Systems Engineering*. Cambridge University Press, 1998.

[EFS98] Jo C. Ebergen, Scott Fairbanks, and Ivan E. Sutherland. Predicting performance of micropipelines using Charlie Diagrams. In *Proceedings of the Fourth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 238–246, April 1998.

[GS90] Mark R. Greenstreet and Ken Steiglitz. Bubbles can make self-timed pipelines fast. *Journal of VLSI and Signal Processing*, 2(3):139–148, November 1990.

[HBAB95] Henrik Hulgaard, Steven M. Burns, Tod Amon, and Gaetano Borriello. An algorithm for exact bounds on the time separation of events in concurrent systems. *IEEE Transactions on Computers*, 44(11):1306–1317, November 1995.

[LPY97] Kim G. Larsen, Paul Petterson, and Wang Yi. UPPAAL: Status and developments. In *Proceedings of the Ninth Conference on Computer Aided Verification*, pages 456–459. Springer, June 1997. LNCS 1254.

[Nag75] L.W. Nagel. SPICE2: a computer program to simulate semiconductor circuits. Technical Report ERL-M520, Electronics Research Laboratory, University of California, Berkeley, CA, May 1975.

[SS93] Jens Sparsø and Jørgen Staunstrup. Delay-insensitive multi-ring structures. *INTEGRATION*, 15(3):313–340, October 1993.

[Sut89] Ivan E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720–738, June 1989. Turing Award lecture.

[Thi91] Lothar Thiele. On the analysis and optimization of self-timed processor arrays. *INTEGRATION*, 12(2):167–187, December 1991.

[Wil91] Ted E. Williams. *Self-Timed Rings and their Application to Division*. PhD thesis, Stanford University, May 1991.

[XB97] Aiguo Xie and Peter A. Beerel. Symbolic techniques for performance analysis of timed systems based on average time separation of events. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 64–75. IEEE Computer Society Press, April 1997.

[XKB99] Aiguo Xie, Sangyun Kim, and Peter A. Beerel. Bounding average time separations of events in stochastic timed Petri nets with choice. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 94–107, April 1999.

[Yov97] Sergio Yovine. Kronos: A verification tool for real-time systems. *Internation Journal of Software Tools for Technology Transfer*, 1(1/2), October 1997.

Comments/Corrections

Comment number(page number): comment