An Event Spacing Experiment

Anthony J. Winstanley, Aurelien Garivier, and Mark R. Greenstreet

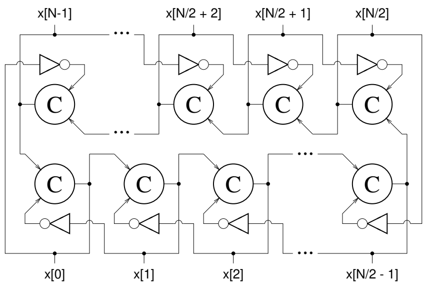
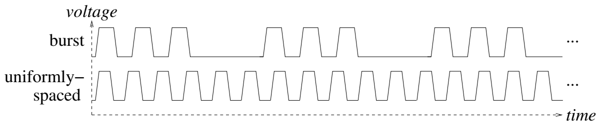


Figure 1. A Self-Timed Ring



(Sampling the output of one stage in a ring.)

Figure 2. Burst and Evenly Spaced Events

Abstract

Keywords: attractors, Charlie Diagrams, hysteresis, phase transitions, self-timed rings, timing analysis.

1 Introduction

Consider a simple self-timed ring such as the one depicted in figure 1. Such rings are ubiquitous in self-timed designs (e.g. [14, 11]), and their performance has been studied in many contexts. For example, [13] analyzed throughput assuming each stage has a fixed time for each operation. Self-timed rings with exponentially distributed processing times were analyzed in [6]. Xie and Beerel have developed tools that analyze general networks of self-timed processors for general probabilistic models [17, 18]. 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 2. This paper reports on our efforts to understand the causes and implications of this bursting behaviour: why do bursts occurs? and how can bursting behaviour be controlled?

While we believe that questions such as those asked above are legitimate grounds for scientific inquiry in their own right, we recognize that some readers are eager to see a practical motivation for exploring such issues. If a self-timed design uses bundled control, then the data-path must be capable of operating at the *minimum* cycle times of the burst. Long-term throughput, however, is typically determined by the *average* rate. Therefore, bursting behaviour leads directly to lost performance. In another direction, we are interested in the possibility of using self-timed pipelines to implement delay lines. Unlike chains of simple buffers, a self-timed pipeline will not drop pulses due to timing asymmetries. On the other hand, if the pipeline clusters events into bursts, then it fails to provide a predictable delay. As yet another application, various researchers (e.g. [12]) have proposed using self-timed pipelines to distribute clock signal – using the handshaking protocol to ensure that local timing constraints are satisfied. The bursts of typical self-timed rings correspond to a clock jitter far greater than can be tolerated in high-performance synchronous design. Finally, we believe that a better understanding of the fundamental properties of self-timed circuits is a prerequisite to discovering new applications for these designs.

In the remainder of this paper, we focus on understanding the temporal behaviour of self-timed rings. In particular:

- We describe why existing models are inadequate for understanding the bursting behaviours of self-timed rings, and present a new model that qualitatively explains this phenomenon.
- We use our model to design a self-timed ring in which bursting can be controlled. We present a CMOS implementation of this design.
- We characterize the transition between bursting and evenly spaced behaviours. Namely, it is a critical phase

transition with hysteresis [10].

- We describe why our model is inadequate to capture the quantitative details of the phase transition described above. From this, we infer properties that a more accurate model must possess.

- In order to test our model, we have designed, fabricated, and tested a chip that implements our new design. Our chip switches between burst and evenly spaced behaviours according to the value of an externally applied current reference.

Our chip provides real, physical measurements for testing our theory. These experiments confirm the nature of the phase transition predicted by the model. They also revealed unanticipated phenomena that motivate future research.

2 Models

The key to understanding the timing properties of self-timed rings lies in finding an appropriate model. Circuit simulators such as SPICE [9] 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, these models are complicated. In particular, 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 timed automata techniques such as [8] and [19]. Close to our current problem, Amon and Hulgaard [7, 1] 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 [2, 4] and intersymbol interference [3]. As described in section 2.1, 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 [2], Chandramouli and Sakallah model closely spaced input events with a function that applies a correction term to a delay model for a single enabling event. However, their model lacks the continuity of the models that we present below. In particular, for large separations, Chandramouli and Sakallah's model assumes that the effect of the earlier signal on the output time is negligible. Our experiments show that even very small dependencies can be critical in determining whether a ring has evenly spaced or clustered events.

When we started this research, we conjectured that bursting behaviour is caused by the intersymbol interference that occurs due to the time that it takes to charge the output capacitance of the C-elements. Recognizing that Charlie Diagrams as described in [5] provided the continuity that would be needed in any model that would explain bursting behaviour, we decided to extend these diagrams to model the dependence of delay on the time since the last output event. In the remainder of this section, we sketch the original Charlie Diagram model and then describe how we extended it.

2.1 The 2D Charlie Diagram

Recognizing that output delay depends on the relative arrival times of the input events to a multiple input gate, Charlie Diagrams measure the output delay of a gate from the *average* of the arrival times of the input events [5]. For a two-input gate, this delay is parameterized by the half-difference of the arrival times:

$$\begin{array}{rcl} t_c & = & m + \mathsf{Charlie}(s) \\ \text{where:} & & \\ m & = & (t_a + t_b)/2 \\ s & = & (t_a - t_b)/2 \end{array} \quad (1)$$

Figure 3 shows a typical Charlie Diagram. The curve of $\mathsf{Charlie}(s)$ 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.

For stage i of a ring, signal $\mathsf{x[i\text{-}1]}$ is the a input to the stage; signal $\mathsf{x[i\text{+}1]}$ is the b input of the stage; and signal

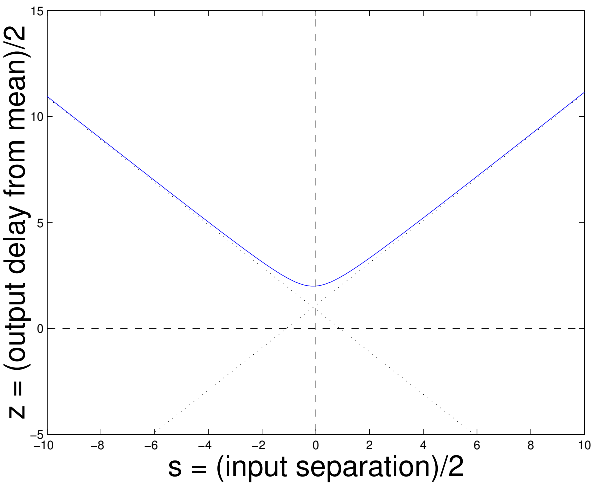


Figure 3. A Charlie Diagram

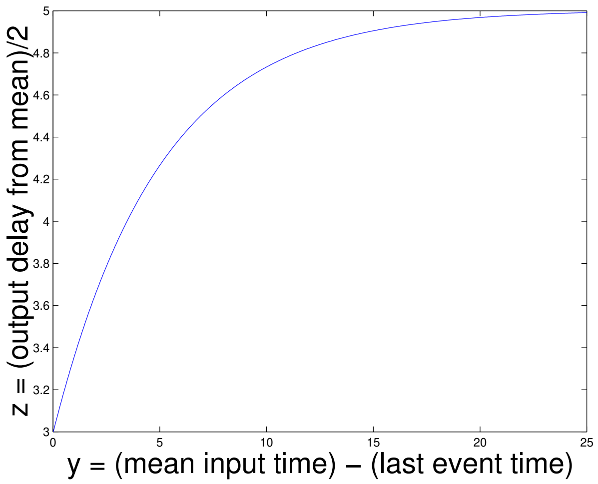


Figure 4. "Drafting"

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:

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

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. These are the simultaneous switching effects described in [4, 2].

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. Returning to the scenario where input a changes after b, we note that the delay from an event at input a to an event at output c is the forward delay, $\mathsf{Charlie}(s) - s$. If input a changes a long time after b, then s is large and positive, and the forward delay is small. Conversely, if a changes only slightly after b, then s is smaller, and the forward delay increases. Because the dependence of gate delay on the relative arrival time of the input events is naturally modeled by Charlie Diagrams, we call this dependence the "Charlie Effect".

In section 3.3, we show that the monotonicity with which the curve of a Charlie Diagram approaches its asymptotes implies that events in the ring are evenly spaced. In practice, most self-timed rings produce bursts of events. Therefore, we extend the model to include another important phenomenon: the dependence of output delay on the time since the previous output event.

2.2 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 4 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.

We model drafting by extending the Charlie Diagram to

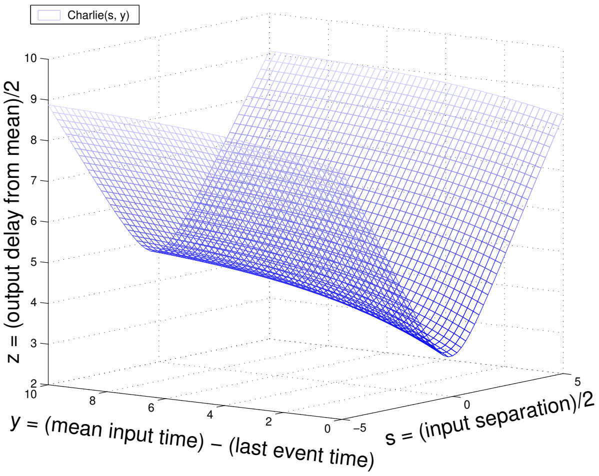


Figure 5. A Three-Dimensional Charlie Diagram

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 5 shows such a Charlie Diagram.

3 Analysis

In this section, we sketch our analytical results for characterizing whether events are evenly spaced or bursting. A more detailed analysis can be found in [15, 16].

3.1 The unique forward delay assumption

To improve our intuition, we constructed a family of Charlie Diagrams assuming hyperbolic dependence of delay on separation and a negative-exponential dependence on the time since the last output event.

$$\begin{aligned} u(s) &= 1 + \sqrt{1 + (s + 0.1)^2} \\ \mathsf{Charlie}(s, y) &= u(s) + \beta(1 - e^{-\alpha(y + u(s))}) \end{aligned} \quad (3)$$

Where y is the time from the last input event to the average of the arrival times of the input events. In this model, the parameter α gives the time constant for the negative exponential drafting term, and parameter β gives the strength of the drafting effect. By varying the values of these parameters, we could obtain clustering or evenly-spaced behaviours. Figure 5 corresponds to this model with $\alpha = 0.2$ and $\beta = 3.0$.

For both clustering and evenly spaced events, we observed that δ_F was the same for all events in rings that were token limited, and δ_R was the same for all events in rings that were bubble limited. We strongly suspect that any "reasonable" Charlie Diagram will exclude steady state behaviours that don't have either a unique forward delay for all actions or a unique reverse delay, but we do not yet have a proof for this conjecture. In the following, we will assume a token limited ring, and write δ_F^* for the unique forward delay of all events. When events clumped in bursts, the first event in the burst had a large reverse delay, δ_{R0}, corresponding to the interval between the last event of one burst and the first event of the next. The other events of the burst had *exactly* the same reverse delay, δ_{R1}, with $\delta_{R1} < \delta_{R0}$. When events were evenly spaced, $\delta_{R0} = \delta_{R1}$.

3.2 Classifying the ring's behaviour

Let t_{last} and t_{next} be the time of any two consecutive events at stage i. We exploit two ways to compute $t_{next} - t_{last}$. Writing m to denote the average of the input arrival times, the definitions of y and $\mathsf{Charlie}(s, y)$ yield: $y = m - t_{\text{last}}$, and $\mathsf{Charlie}(s, y) = t_{\text{next}} - m$. We conclude: $t_{next} - t_{last} = y + \mathsf{Charlie}(s, y)$. Now we note that stage $i + 1$ transitions δ_F^* time units after t_{last}, and that t_{next} occurs δ_R later; thus: $t_{next} - t_{last} = \delta_F^* + \delta_R$. Adding the parts of equation 2 yields $\delta_F^* + \delta_R = 2\mathsf{Charlie}(s, y)$. We conclude: $\mathsf{Charlie}(s, y) = y$. Graphically, we intersect the surface of a three-dimensional Charlie Diagram with the $\mathsf{Charlie}(s, y) = y$ plane to obtain a curve that includes the operating points of any burst or evenly spaced equilibrium. We write $\mathsf{Charlie}_y(s)$ to denote this curve. In the remainder of this section, we restrict our attention to the $\mathsf{Charlie}_y(s)$ curve.

Given that all events have the same forward delay, the operating point(s) for any steady state solution must lie on the line $\mathsf{Charlie}_y(s) = s + \delta_F^*$. If $\partial\mathsf{Charlie}_y(s)/\partial s < 1$ at the point of intersection, then that intersection is a stable attractor. On the other hand, if $\partial\mathsf{Charlie}_y(s)/\partial s > 1$, then the point of intersection is unstable.

Let n_T denote the number of tokens in a ring and n_B the number of bubbles. Balancing token and bubble flow yields:

$$n_B\delta_F^* - \delta_{R0} - (n_T - 1)\delta_{R1} = 0 \quad (4)$$

Noting that δ_{R0} and δ_{R1} are determined by δ_F^*, the Charlie Diagram, and the $y = \mathsf{Charlie}(s, y)$ constraint, we solve for δ_F^* in equation 4 using standard root finding techniques. In the case of evenly spaced events, $\delta_{R1} = \delta_{R0}$, from which we derive:

$$\mathsf{Charlie}_y(y) = \frac{n_T + n_B}{n_T - n_B} s \quad (5)$$

The above observations yield the following procedure for classifying the timing behaviour of a self timed ring:

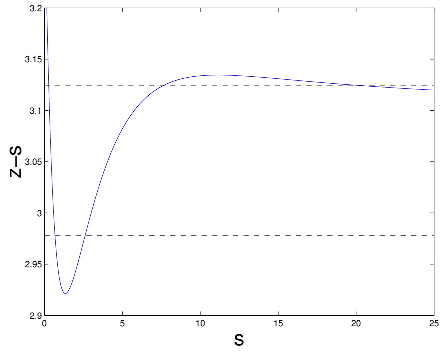


Figure 6. $z = \mathsf{Charlie}_y(s)$**, high drafting**

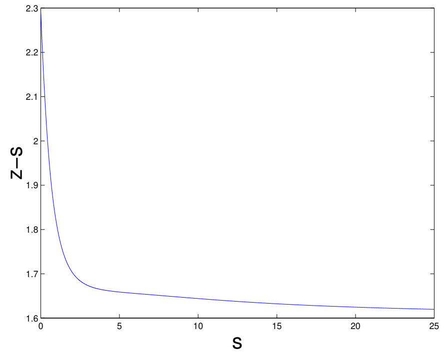


Figure 7. $z = \mathsf{Charlie}_y(s)$**, low drafting**

1. Find the curve $\mathsf{Charlie}_y(s)$ by intersecting the surface of the Charlie Diagram with the $\mathsf{Charlie}(s, y) = y$ plane.

2. Compute the intersection of $\mathsf{Charlie}_y(s)$ with the line $\mathsf{Charlie}_y(s) = ((n_T+n_B)/(n_T-n_B))s$. If such intersections exist, then the one with $\partial\mathsf{Charlie}_y(s)/\partial s < 1$ is stable, and evenly spaced. All solutions for figure 7, and the left intersection for the lower dashed line in figure 6 depict stable, evenly spaced solutions.

3. Determine $\max(\partial\mathsf{Charlie}_y(s)/\partial(s))$. If this value is greater than one (see figure 6), then burst behaviours are possible. If it is less than one (see figure 7), then burst behaviours are excluded.

4. If burst behaviours are possible, find the feasible values of δ_F^* by solving equation 4. If there is a solution with $\delta_{R0} \neq \delta_{R1}$ where $\partial\mathsf{Charlie}_y(s)/\partial(s) < 1$ at both points, then the ring has stable, burst behaviours.

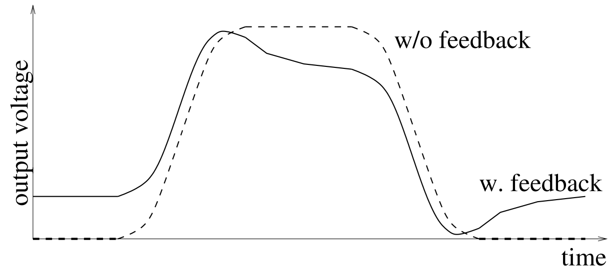


Figure 8. Cancelling Drafting with Feedback

3.3 2D Charlie Diagrams predict evenly spaced events

Any 2D Charlie Diagram can be represented by a 3D Charlie Diagram where $\mathsf{Charlie}(s, y)$ is independent of y, the time since the last event. In this case, the curve for $\mathsf{Charlie}(s, y) = y$ is just the original 2D Charlie Diagram. The slope of the right asymptote of the Charlie Diagram is one. If the curve of the Charlie Diagram approaches this asymptote monotonically from above, then the slope of the curve is always less than one. By the arguments above, this implies evenly spaced behaviour.

4 A Ring with Controlled Event Spacing

With the explanations and analysis of clustering behaviour from the previous sections, we set out to design a ring stage that produces evenly spaced events. We started with the family of simple Charlie Diagrams given by equation 3 studying the behaviour as α and β vary. For example, with $\alpha = 0.2$, evenly spaced events are the only stable behaviour for $\beta < 0.63$; for $0.63 < \beta < 1.18$, both evenly spaced and burst are stable; and with $\beta > 1.18$, only the burst behaviour is stable. We conclude that the system exhibits critical phase transitions between evenly spaced and bursts: the behaviour snaps from one to the other when the critical value is crossed. Furthermore, the system has hysteresis: for $0.63 < \beta < 1.18$, both behaviours are stable. If the system enters this region from the evenly spaced side ($\beta < 0.63$) then it will continue to exhibit evenly spaced behaviour in this region. On the other hand, if the system enters this region from the burst side ($\beta > 1.18$), it will continue to exhibit burst behaviour. The attractors in the model are structurally stable: they hold in an open set of parameter values. This means that a circuit with the same qualitative dynamics will exhibit the same structural stability. In particular, achieving evenly spaced behaviour does not require exact matching of the circuit parameters or device sizes across the ring stages.

Having identified drafting as the cause of event clustering, we sought to design an "anti-drafting" mechanism. Drafting occurs because the output of the C-element asymptotically approaches the power supply rails, thus monotonically increasing the distance for the next transition. Our innovation is to make the output "bounce" instead. Using a small amount of negative feedback, the output settles to values that are slightly inside the rails. If this feedback is suitably delayed, then the output will overshoot this target and then asymptotically approach it. Figure 8 shows this approach where the dashed line is the C-element output without feedback, and the solid line is the output with negative feedback. The feedback causes the delay for the next event to *decrease* as the time since the last event increases. With a sufficient bounce, this cancels the drafting effect, and produces evenly spaced events.

We now have our hypotheses:

H1: Bursting behaviour is due to drafting. Negative feedback can cancel drafting and produce evenly spaced events.

H2: If the amount of drafting can be controlled (e.g. by controlling the strength of the negative feedback above), then the ring will exhibit critical phase transitions with hysteresis between evenly spaced and bursting modes of operation.

H3: The evenly spaced and bursting modes are structurally stable. For either mode, there exist a region of operation where the mode is robust against small perturbations such as variations in the strength of the feedback, the circuit parameters, or the operating conditions. In particular, bursting is not a consequence of electrical mismatches between stages.

H4: 3D Charlie Diagrams provide an accurate model for classifying the temporal properties of events in self-timed rings.

To test these hypotheses, we designed, fabricated, and tested a chip.

5 A Test Chip

5.1 Negative Feedback

The first task of our design was to design a ring stage that used negative feedback as an "anti-drafting" mechanism. Our first approach was to attach a three inverter ring to the output of a FIFO stage as shown in figure 9. In this design, transistors n1, n2, p1, and p2 form a dynamic, inverting C-element. Inverter i2 buffers the output of the C-element, and inverter i3 is a keeper. Inverter i1 provides an inverted input for the acknowledgement from the successor stage.

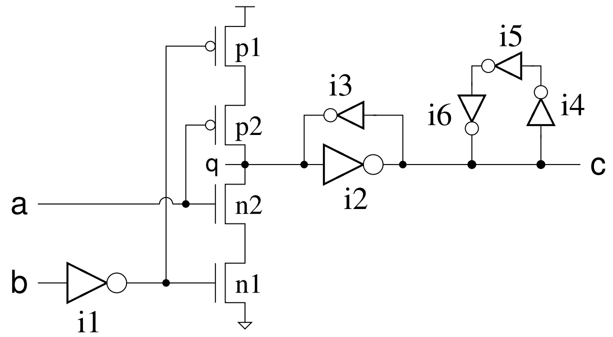


Figure 9. A Ring Stage With Feedback

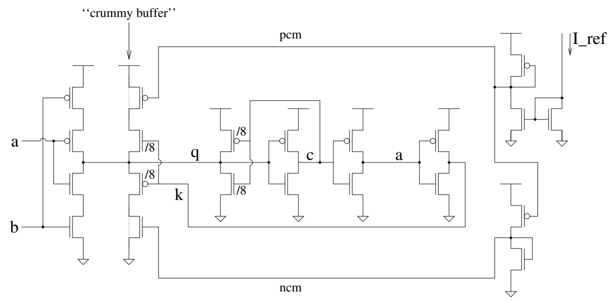


Figure 10. A Ring Stage With "Crummy Buffer" Feedback

Inverters i4, i5, and i6 provide the negative feedback with a delay. Inverter i2 is designed to be much stronger than i6. When inverter i2 switches, i6 is initially pulling the same direction and accelerates the transition. After this transition propagates through inverters i4 and i5, inverter i6 pulls against i6, moving the level on node c slightly away from the power supply rail. This creates the bounce described in the previous section.

HSPICE simulations showed that this design failed to prevent bursting behaviour regardless of the relative strengths of inverters i2 and i6. We tried to understand this by creating 3D Charlie Diagrams from HSPICE simulation data. However, the surface was flat to within the numerical resolution of HSPICE in the regions of interest. We were forced to think instead of simulate. We realized that while the bounce decreases the drafting effect, it also reduces the swing of the signal that is supplied to the next stage causing the transistors in the next stage to be less than fully conducting. This negated the Charlie-effect, and events remained clustered. Here, the intuition from the Charlie-Diagram helped us to diagnose the problem and realize that it could not be solved by methods such as adjusting transistor sizes.

Our solution, as shown in figure 10, is to apply the negative feedback to node q of the C-element. This required some care to avoid spurious transitions. Between transitions, node q can be driven by the keeper only. We must ensure that our feedback circuit does not overpower the keeper. Our solution was to use a "crummy buffer" with an n-channel pull-up and a p-channel pull-down. With this arrangement, the n-channel pull-up of the crummy buffer fights the n-channel pull-down of the keeper. As both transistors are n-devices, the matching of their relative strengths is fairly robust against parameter variations, variations in operating voltage or temperature, etc. Likewise, the p-channel pull-down in the crummy buffer fights the p-channel pull-up of the keeper. We found that matching the sizes of the transistors in the crummy buffer to those in the keeper provided excellent matching over the entire range of process parameters. In figure 10, all P-channel devices have a shape factor twice that of the N-channel devices. Devices marked "/8" indicate transistors with 1/8 the width of the others.

Finally, we placed current limiting transistors in series with the crummy buffer's pull-up and pull down. We bias these transistors from a current mirror that is regulated by an external current reference. This allows us to experiment with the strength of the feedback.

Figure 11 shows the time intervals between successive transitions of the first C-element for a ring with 15 stages initialized to hold 6 tokens (i.e. 3 pulses). The data was obtained from HSPICE simulations. At the beginning of the simulation, the reference current was 0μA, and the ring operated with bursts of events. With 6 tokens, these bursts have 2 short output cycles, and one long output cycle. where the current attached to I_ref is 0μA. Figure 11 shows what happens when the current is increased to 30μA: all the measured distances become equal, and the ring switches into evenly spaced mode.

When we decrease the current back to 0μA, the ring remained evenly spaced in HSPICE simulation. It appeared that *positive* feedback was required to switch the circuit back to burst mode, something outside the range of the present design. As we describe shortly in section 5.3, the fabricated chip exhibits both critical transitions. This discrepancy between HSPICE simulation and the real chip remains to be explained.

5.2 The Chip

We implemented a chip with a thirty stage ring with a stage design based on the one presented in the previous section. In addition to the feedback mechanism described above, we added several other features to each stage to facilitate testing our hypotheses:

- We introduced a programmable asymmetry into the ring stage. In parallel with the inverter that drives node c, we placed an inverting tri-state buffer. We added a

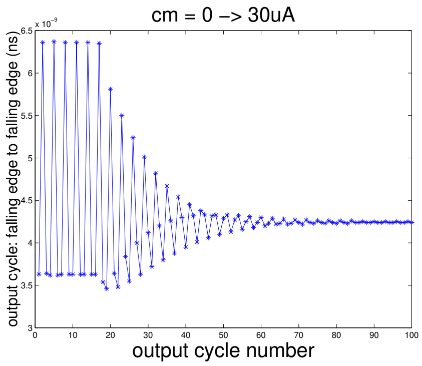


Figure 11. Burst → Evenly Spaced

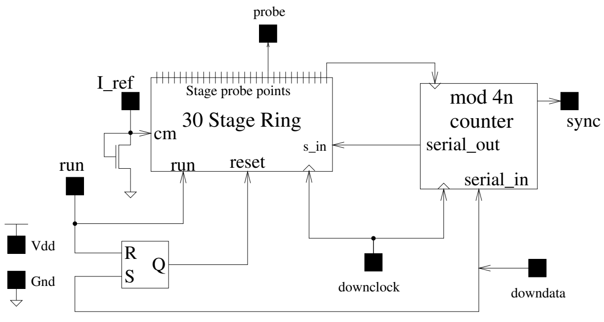


Figure 12. The Test Chip

serial register with one bit per ring stage. When this bit is true, the buffer is enabled, when false, it is disabled. This allows us to selectively alter the output delay of the ring stages.

- We added a loading mechanism to set the number of tokens in the ring. A reset signal forces all stage outputs to false. The first stage of the ring includes a multiplexor so that it can receive its forward input from the last stage of the ring or from an external input. To maintain matching of the stages, the other stages include an equivalent multiplexor with the control input set to always accept the output of the previous stage.

  After resetting the ring, we load the desired number of tokens into the ring. The ring starts running when the multiplexor is set to close the ring. A side-effect of this mechanism is that the ring always starts with its events clumped in a burst.

- We added probe points to each stage. These were chains of three buffers from node c of each stage, culminating with a 50Ω driver. One of these drivers was connected to a bonding pad. The driver for each stage can be probed using a micro-manipulator probe.

Figure 12 shows the block diagram for the chip. To work with an existing probe card for small designs, the chip has eight pads that we multiplexed to provide the functionality that we wanted. When the chip is reset, tokens are loaded into the ring, and the serial configuration registers are loaded. The ring starts running when the run signal goes high. In addition to the ring, we included a programmable divider to output a lower frequency sync pulse for oscilloscopes and other test instruments. By setting the modulus of this counter to a multiple of the number of tokens in the ring, we obtain a strobe that works effectively with either bursting or evenly spaced events.

5.3 Test Results

We fabricated our design using the Canadian Microelectronics Corporation 0.35μ process. We made measurements on the fabricated chip to test the hypotheses advanced in section 4.

Our first hypothesis states that evenly spaced events can be achieved through the use of negative feedback. We loaded the ring with 12 tokens (i.e. pulses, one event for each rising edge and one for each falling). This ensured token limited operation. As shown in figure 13 when the chip was operated with I_ref $= 0\mu$A, the ring operated with bursting events, as predicted by the model. Figure 14 shows operation with I_ref $= 200\mu$A, we observed evenly spaced events. Thus, sufficient negative feedback produced evenly spaced events as predicted by our model.

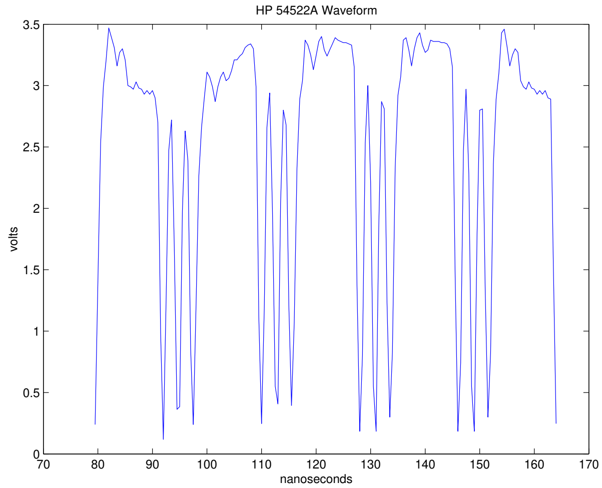


Figure 13. I_ref $= 0\mu$A**: Bursting Events**

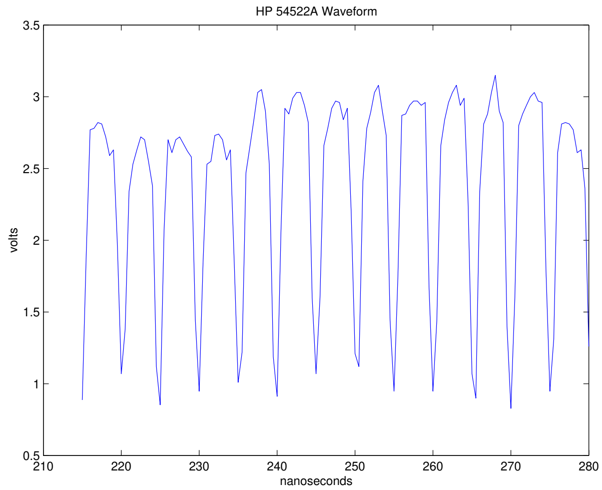


Figure 14. I_ref $= 200\mu$A**: Evenly Spaced Events**

Our second hypothesis is that the ring should exhibit critical phase transitions between bursting and evenly spaced events, and that these transitions should exhibit hysteresis. Our HSPICE simulations indicated that we should only be able to observe the bursting to evenly spaced transition. We again loaded the ring with 12 tokens, started the ring with I_ref $= 0\mu$A, and gradually increased I_ref. The ring operated with bursting events, gradually increasing in frequency. When I_ref reached 132μA[1], operation snapped to evenly spaced events. In other words, the events did not gradually become more evenly spaced with increasing I_ref. This indicates a critical phase transition as predicted by the model.

When I_ref was decreased again, we observed a transition back to bursting events at I_ref $= 107\mu$A. While this disagrees with our HSPICE simulations, it is in accordance with the prediction of our 3D Charlie Diagram models that indicates that the phase transition should exhibit hysteresis.

Another observation that we made was that when the ring was in bursting mode with 107μA $<$ I_ref $<$ 132μA, the burst became less stable. In particular, the trailing pulse of the burst would occasionally fall back around the ring and become the leading pulse. We don't know why this occurs. We are particularly intrigued that this phenomenon occurs at the same reference current, to within the resolution of our measurements, as the current for the critical transition from evenly spaced to bursting. In other words, this bursting appears to coincide exactly with the feasibility of evenly spaced behaviour. The evenly spaced mode appears to be completely stable without anomalies for all values of I_ref above 107μA. We suspect that the instability of the bursting mode may be related to noise in the circuit that was not included in our HSPICE models, but we have no data for testing this conjecture.

Our third hypothesis is that bursting and evenly spaced behaviours are structurally stable. We repeated our experiments with one stage set to operate with a slow output buffer and the other 29 stages set to operate with fast buffers. The results measured were equivalent to those with all buffers fast. This indicates that bursting and evenly spaced behaviours are structurally stable as predicted by our model: they are not artifacts of asymmetries in the ring.

6 Limitations of Charlie Diagrams

We return to Charlie Diagrams to evaluate our fourth hypothesis: 3D Charlie Diagrams provide an accurate model for classifying the temporal properties of events in self-timed rings. We clearly found 3D Charlie Diagrams useful for designing our test chip, and the test results agree qualitatively with the model. As mentioned in section 5, numerical limitations of HSPICE prevented us from using 3D Charlie Diagrams as a quantitative tool while designing the chip.

During the fabrication of the chip, we sought to remedy this problem. Our idea was to integrate the Jacobian matrix (i.e. the partial derivatives) for the ODE model along with the circuit state. From this integrated Jacobian, we could infer the tangent surface to the 3D Charlie Diagram at each data point. These tangent surfaces would allow us to perform the classification described in section 3.2.

As HSPICE does not provide the numerical hooks for this calculation, we chose Matlab for this experiment and started with a highly simplified, ODE model of a C-element that exhibited drafting when used in a ring. We constructed the 3D Charlie Diagram for this simplified model, and much to our shock, it predicted evenly spaced events. On closer examination, the culprit turned out to be the shape of the input waveforms that we were applying to the C-elements. When we performed a fix point iteration to ensure that the input waveforms had the same shape as the output waveforms, then the Charlie Diagrams made the correct prediction. Unfortunately, this is roughly equivalent to simulating each possible ring configuration and offers little predictive value.

Further numerical experiments showed that the classification of bursting versus evenly spaced behaviours is fairly sensitive to the shapes of the waveforms. Simply knowing the delays from 50% transition points is not sufficient. This suggests that any model based on lumped delays is inadequate to make quantitative predictions about steady-state event spacings. On the other hand, both bursting and evenly spaced behaviours appear to be very robust. They persist over wide variations in process parameters, changes in circuit topologies, and large variations between individual stages in the ring. Thus, we strongly suspect that there should be a simple abstraction that captures these robust behaviours and makes accurate, quantitative predictions. Finding such a model is a topic for future research.

7 Conclusions

We have designed and implemented the first known self-timed ring that exhibits uniform spacing of events. Our design uses negative feedback to cancel the effects of drafting, the dependence of switching delay on the time since the previous output event. By varying the strength of this feedback, we can evoke bursting or evenly spaced behaviours. The transition between these two modes is a critical phase transition with hysteresis.

We developed and used 3D Charlie Diagrams for our qualitative analysis. These diagrams correctly identified drafting as the cause of bursting events; they correctly classified the nature of the phase transition; and they provided

[1]For the referees: When writing this section, I noticed that this current is much greater than the 30μA value mentioned in section 5.1. The co-author who made the measurements is unavailable for two months. We will reconcile these numbers for the final paper.

critical insight into how to design real circuits where the event spacing can be controlled. The quantitative details of event spacing depends on not only the times of signal transitions but also the shapes of the waveforms during these transitions. These waveform shapes are neglected by the Charlie Diagram abstraction. Therefore, some other model is needed to capture quantitative details of event spacing.

Measurements on real, physical artifacts are essential to establishing the validity of theoretical models. Accordingly, we implemented a chip to test the hypotheses that we made based on our model. The chip's behaviour transitions between bursting and evenly spaced events according to the value of an externally applied reference current. This transition exhibits a critical phase transition with hysteresis as predicted by the model. In the hysteresis region, the bursting mode shows an instability that is not predicted by the Charlie Diagrams, nor was it predicted by HSPICE simulations. We suspect that it is somehow noise related, but we are intrigued that it occurs only in the region were both bursting and evenly spaced behaviours are allowed.

By performing these experiments, we have obtained a deeper understanding of the operation of self-timed pipelines. We expect that this knowledge will lead to novel applications of self-timed circuits, especially in applications where timing details are critical. We also hope that the questions raised by our experiments will lead to further research and a better understanding of the dynamics of self-timed circuits and systems.

Acknowledgements

This research was funded in part by a grant from the Canadian National Science and Engineering Research Council, a grant from SUN Microsystems, a University of British Columbia Graduate Fellowship, and an internship stipend from the Ecole Normale Supérieure, Cachan. We thank Jo Ebergen, Ian Jones, and Ivan Sutherland for helpful discussions of Charlie Diagrams and the bursting behaviour of self-timed rings. We are indebted to Mike Jackson, Roozbeh Mehrabadi, Roberto Rosales for suggestions and assistance with the tests and measurements of our chip.

References

[1] T. Amon and H. 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, Apr. 1999.

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

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

[4] F. Dartu, N. 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.

[5] J. C. Ebergen, S. Fairbanks, and I. 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, Apr. 1998.

[6] M. R. Greenstreet and K. Steiglitz. Bubbles can make self-timed pipelines fast. *Journal of VLSI and Signal Processing*, 2(3):139–148, Nov. 1990.

[7] H. Hulgaard, S. M. Burns, T. Amon, and G. Borriello. An algorithm for exact bounds on the time separation of events in concurrent systems. *IEEE Transactions on Computers*, 44(11):1306–1317, Nov. 1995.

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

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

[10] T. S. Parker and L. O. Chua. *Practical Numerical Algorithms for Chaotic Systems*. Springer, New York, 1989.

[11] J. Sparsø and J. Staunstrup. Delay-insensitive multi-ring structures. *INTEGRATION*, 15(3):313–340, Oct. 1993.

[12] W.-K. Su. *Reactive-Process Programming and Distributed Discrete Event-Simulation*. PhD thesis, Computer Science Department, California Institute of Technology, 1989.

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

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

[15] A. J. Winstanley. Temporal properties of self-timed rings. Master's thesis, University of British Columbia, Sept. 2001.

[16] A. J. Winstanley and M. R. Greenstreet. Temporal properties of self-timed rings. In *Proceedings of the 11th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, CHARME'01*, Sept. 2001.

[17] A. Xie and P. 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, Apr. 1997.

[18] A. Xie, S. Kim, and P. 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, Apr. 1999.

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