

Analog Micropipeline Rings for High Precision Timing

Scott Fairbanks
Cambridge University
Scott.Fairbanks@cl.cam.ac.uk

Simon Moore
Cambridge University
Simon.Moore@cl.cam.ac.uk

Abstract

I use asynchronous FIFO stages that are connected in rings to generate and deliver highly precise timing signals. I introduce a Micropipeline FIFO control stage that oscillates at frequencies greater to that found in a ring of three unloaded inverters. Tokens spread evenly through FIFO rings built from this control under certain conditions. The tokens are ‘locked’ into an equally separated pattern by a classical feedback control where the actuator is the FIFO control stage. The actuating variable is the stage latency which varies according to the temporal separation of its inputs. When the tokens in the FIFO ring are equally spaced, the relative phases of the nodes in the system assume predictable values.

This technique allows the division of a better than three gate delay cycle time into an arbitrarily large number of phases whose precision is limited only by the limits of the fabrication process and noise. Applications that need a precise time reference can benefit from this technique. A/D conversion, clock recovery, and multi-phase clocking solutions are briefly sketched.

1. Introduction

This paper advances work from two papers presented at previous meetings of this conference.

Ebergen et al [4] explored how data tokens distribute themselves throughout a free running FIFO.

An accurate characterization of the FIFO was performed using a delay characterization for an asynchronous control stage that took into account the temporal proximity of its inputs. With this characterization and the response times of the source and sink to the FIFO, they predicted how tokens could be expected to distribute themselves over a section of free running FIFO. They confirmed the predictions using HSPICE simulations.

Winstanley et al [10] continues this investigation by closing the section of FIFO into a ring and explaining why the tokens in the ring sometimes gather into a bunch and some-

times do not. They were able to force the tokens to bunch in the ring and were also able to force them to spread apart evenly throughout the ring by use of a novel state holder or ‘keeper’ circuit at the output of the C-element.

This current paper again looks at rings of asynchronous FIFO control. However the focus is to amplify and exploit the effect whereby tokens evenly distribute themselves around the ring. I refer to this state as *locked*.

2. Theory

2.1. Tokens and Phases

A token in a FIFO usually represents the presence of data. In a Micropipeline [9] its presence is signaled when the binary output of adjacent stages are complements.

Data movements in Micropipeline FIFO stages are often viewed as atomic. The data is there or it is not. In this paper, a token can be smeared across a number of stages. When the tokens in the ring are locked, the outputs of adjacent FIFO stages have a fixed phase relationship with respect to each other.

A token experiences 180° of phase shift when it moves between two stages within a ring built from asynchronous FIFO stages that communicate using Micropipeline signaling. If tokens are confined to a ring and spread evenly through the ring, then the phase shift at the output of each stage must be:

$$\phi = \frac{\text{tokens}}{\text{stages}} * 180^\circ \quad (1)$$

‘Fractional number of tokens per stage’ and ‘phase offset per stage’ are then equivalent descriptions.

The total number of phases available, assuming differential signaling, is twice the number of stages.

2.2. The FIFO Stage

The FIFO stage I use is shown in Figure 1. This stage implements the C-element function.

Complementary values are held on the output by the cross coupled inverters that I refer to as a *keeper*. When the

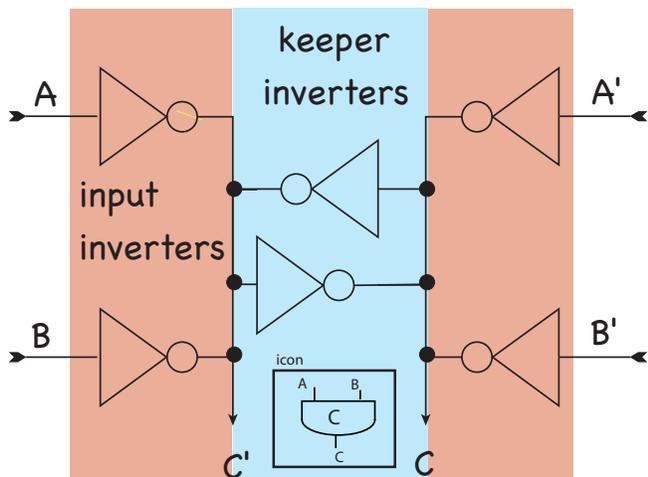


Figure 1. Analog C-element

inputs agree, the state of the output is set to the same value as the inputs. When the inputs disagree the binary state of the output is the value of the inputs when they last agreed. The actual potential at the output of this gate is removed from the supply or ground by an amount determined by the voltage divider formed by the inverter connected to the asserted input against the keeper and the inverter connected to the unasserted input. I call this an Analog C-element because its output voltage is expected to take on a range of values and its inputs sense this same range of values.

Some things to note about the Analog C-element are:

Symmetry It is completely symmetric. I don't know of any other electrically symmetric C-elements. This C-element alone is used as the FIFO stage in this paper. The shortest path back to any node through adjacent stages is three gate delays, ensuring robust oscillations if all inverters are of the same size.

Economy It uses 12 transistors. None of the transistors are in series. A layout made for this stage in a 180nm process using $2\mu\text{m}$ wide PMOS transistors and $1\mu\text{m}$ wide NMOS transistors was $6 \times 6\mu^2$. The drains of the inverters connected to the inputs should be shared to reduce the drain capacitance for a stage by a third. The square layout lends itself to simple placement and abutment of cells. The short distance between cells results in insignificant wire loads.

Static Current If the inputs disagree then static current is drawn. This is a problem if this FIFO stage were used in an asynchronous environment where it might wait an unbounded period of time for its inputs to agree.

When used in a ring FIFO with locked tokens, the inputs are always transitioning.

Performance When the inputs are in phase, the input inverters work together and are hardly impeded by the keeper inverter, which is mostly disabled because the inputs to the keepers inverters are switching as they try to resist the voltage change.

When the inputs are out of phase, the output settles at a voltage removed from the rail. This results in shorter gate delays when the final input arrives because the output potential must then be charged over a smaller voltage range.

Predictability There are no internal nodes between the rails and the output. Internal nodes store charge that could be dumped onto the output and cause unexpected phase shifts.

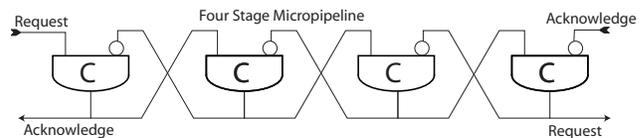


Figure 2. Four Stage Micropipeline

Figure 3 shows the closed loop FIFO stage delay vs temporal separation of the inputs. I refer to this plot as a *Separation curve*. Both papers [4, 10] characterize a FIFO stage with an open loop delay versus temporal input separation characterizations to predict their FIFO's behavior. The open loop characteristics of the analog Micropipeline FIFO stage differs significantly from its closed loop behavior. The Separation plot shows that the delay of the stage is at a maximum when its inputs arrive simultaneously. The delay tapers to a nadir near 50ps. There is a small hump from 55ps to 100ps. The delay then tapers out to a constant value for large input separation times. The open loop characterization of the analog Micropipeline did not have the hump that starts at 55ps. Instead the delay monotonically tapered to a constant delay value of 18ps at around a 200ps separation.

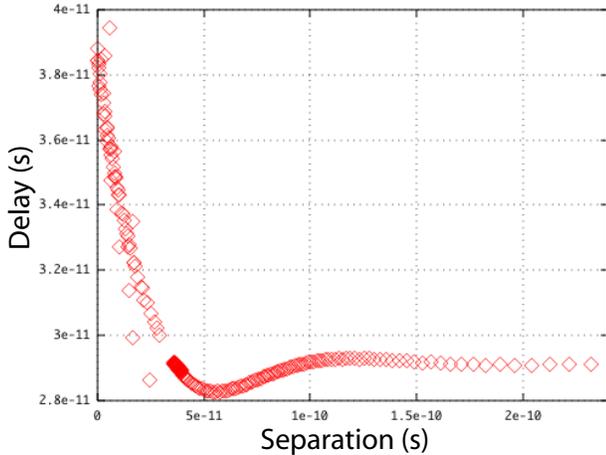


Figure 3. Delay vs Separation plot

2.3. The Locking Mechanism

Assume that tokens start in a bunch, each in adjacent stages of a ring or closed FIFO. The first token starts to move around the ring. During its first lap it experiences the delay of a stage when the inputs arrive separated by infinite time. The second token follows the first as soon as the first token leaves a hole in the stage it occupied. Initially it will follow the first token closely. The short separation time between it and the first token causes it to experience greater delay than the first token as it is moved between stages, just as the Separation curve predicts. The greater delay of the second token causes it to be pushed back from the first, causing their separation to increase. The effect where a gate's delay increases as the temporal separation of inputs shrinks was named the Charlie¹ effect in [4, 2]. Each of the tokens pushes away from its front neighbor in this way until their separation reaches the point on the Separation curve that presents the same delay as the first token sees with its very long Separation, which is about 30ps and results in a stage delay of 29ps. If the second token were to continue Separating from the first token, then its move delay would become less than the move delay of the first token and it would catch back up. I call this the *holding distance*.

The separation between the last token and the first token shrinks because the Charlie effect increases the time required for the token movement, effectively pushing the tail of the last token towards the nose of the first token. If the last

¹ Named after Charles Molnar, a pioneer in early studies of the fine grained behavior of token movement in asynchronous pipelines.

token is pushed back far enough so that the first token begins to operate on the hump of the Separation curve found between separations of 55ps to 100ps, then the first token will be accelerated towards the last as it begins to be delayed less.

This is not the drafting effect described by the previous research. The drafting effect is where a second token experiences less delay than a first token that it follows because the voltage signaling the first move has not transitioned completely to the rails when the second token move is initiated. The move signal is then charged over a smaller potential to complete the second move, which takes less time and causes the tokens to bunch.

The peak to peak voltage swing is fairly constant for all separation distances of the analog Micropipeline FIFO control that is used in this paper. It is at a maximum when the inputs arrive at the same time, which doesn't describe the drafting effect.

To understand the cause of the hump in the Separation curve we need to closely look at the dynamics of a token movement. When the inputs to a FIFO stage agree, the output is driven towards its new binary value. The new value signals a Request to the subsequent stage in the FIFO. If the subsequent stage is empty, it consumes the token and sends an Acknowledgment back to the initial stage. The Acknowledgment means that the inputs to the first stage will now disagree. The output voltage of the stage is divided because the inverter connected to the Request input to the stage and a keeper inverter are maintaining the previous binary output value while the inverter connected to the Acknowledge input is attempting to drive the output in the opposite direction.

The output voltage of the next stage dips to the voltage divided value one stage latency later than the first for this same reason. The inverter connected to the Acknowledge input of the first stage is then weakened slightly because it is controlled by the output voltage of that second stage. The resistance between the rail and the output of the inverter controlled by this Acknowledge input then increases and changes the potential of the voltage divider, moving it closer towards the full voltage of the binary value the C-element is maintaining. This new voltage divided value, in turn, has a further effect on the voltage divided value of the stage's two neighbors, but this effect is barely discernible.

The delay of the FIFO stage increases for a small range of separations beyond the nadir of the Separation curve because of this feedback of the voltage divided output from the next stage. This is the source of the hump.

Figure 4 shows eight plots. Each plot shows the separation times for one of eight token in a 24 stage closed FIFO for its first 250 moves. Notice that the curves have the same character as any simple feedback control system has towards a step response. The curves look very similar to a

plot of the temperature near the heat source of a house during winter after the thermostat is turned on. Initially the target is overshoot by a large amount and then there is a large amount of ringing about the target while the controlled variable converges.

The effect of the electronic force between two like-charged particles is similar to the interaction between tokens in a ring. Intuition about how charged particles interact can be applied with tokens in a ring. Equally spaced charges confined to the same closed path, such as electrons orbiting at the same energy level around a nucleus, arrange themselves into the lowest energy state which is equidistant. The main difference in the charged particle system and the tokens in a ring is that the tokens have a limited separation over which they affect each other. This illustration might be helpful.

In the simulations used to draw Figures 4 and 5, one hundred moves required about 7.5ns to 10ns.

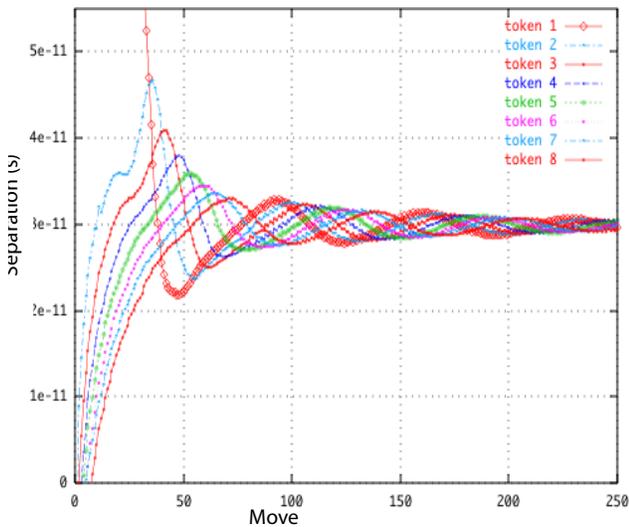


Figure 4. Separation between a token and its predecessor versus number of Moves for the eight tokens in a 24 stage closed FIFO

Figure 5 shows the separation time for the first of eight tokens for its first 250 moves in rings of different sizes. The shorter rings converge faster for two reasons. First, the tokens have less distance to spread. But also they operate on a steeper portion of the Separation curve. The slope of the Separation curve is the loop gain of our control system. A

token moved out of place by a noise source is corrected quicker if the tokens operate on a steeper part of their Separation curve. Notice that the half loaded FIFO, 16 stages, locks within the first 50 moves which takes about 4ns.

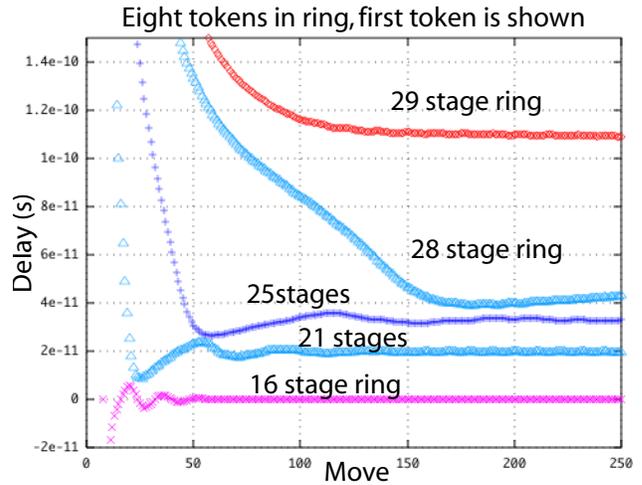


Figure 5. Separation between the first token and the last (8th) token versus number of Moves for first token in rings of various size.

Notice that the first token in the 29 stage FIFO ring fails to lock. The separation distance between the first and last token is on the constant delay portion of the Separation curve just to the right of the hump and so it will not change. The seven tokens behind it closely follow each other at the holding distance, about 30ps apart. When the tokens spread out from each other because of the Charlie effect, they are not able to push the last token close enough to the first token to force it over the hump.

The trajectory of the first token in the 28 stage FIFO ring looks like it is about to flatten out just beyond the lock range at around 80 moves. Then the last token is delayed just enough so that the distance between it and the first token places the last token over the hump of the Separation curve. The first token is then delayed even less on the next move, causing the separation between the first and last token to close even further. A similar action trickles back through the pack of tokens and the tokens eventually lock.

Table 1 shows the numbers of tokens in a stage that lock for different sizes of rings. The column labeled phases shows the number of evenly spaced phases on the nodes of

<i>N</i>	<i>Tokens</i>	<i>Phases</i>	<i>N</i>	<i>Tokens</i>	<i>Phases</i>
3	2	6	12	4,6,8	6x4,4x6
4	2	4x2	13	4,6,8,10	26
5	2	10	14	4,6,8,10	14x2
6	2,4	6x2	15	4,6,8,10,12	30,10x3,6x5
7	2,4	14	16	6,8,10	8x4,4x8,8x4
8	2,4,6	8x2,4x4	17	6,8,10,12	34
9	4,6	18	18	6,8,10,12	6x6,18x2
10	4,6	10x2	19	6,8,10,12	38
11	4,6,8	22	20	8,10,12	10x4,4x10,8x5

Table 1. Numbers of tokens causing locks in rings with up to twenty stages.

the rings. When the column has two numbers separated by an ‘x’, as in 4x2, it means that each of four phases has a copy at a distinct location in the ring. A ring loaded at half occupancy has a 90° phase shift per stage according to Eq (1). Because each stage is differential, a copy of a signal is found in every other stage. Similar reasoning puts a copy of each phase in every third stage for a ring FIFO that is a third occupied.

Notice that rings of Micropipeline control can only be loaded with even numbers of tokens.

2.4. Variations

The analog C-element functions with any number of buffers on the output that are in the timing loop. The cycle time with one stage buffering at the output is 279ps, with two stages of buffering the period is 411ps. The hump on the Separation curve is filtered out in both these cases. A smaller range of token occupancies lock with these configurations partly because this hump is gone and also because the increased latency makes it more difficult to push the separation times towards the dynamic part of the Separation curve. Yet both configurations force tokens to lock and both are robust to transistor mismatch, and power supply and other sources of noise.

Other alterations that might find use in some applications is to break the symmetry of the FIFO stage and place an unequal number of buffers on the Request and Acknowledge signals.

2.5. Performance and Power

Hspice simulations for this paper were done with models extracted from a 1.8V 180nm process. The FO4 delay for an inverter with PMOS/NMOS transistor ratio of two in this process is 88ps. A ring of three unloaded inverters oscillates with a period of 158ps.

The period for the signals produced by a ring built from analog Micropipeline control is:

$$Period = 4 * D + 2 * S \quad (2)$$

Where D is the stage delay from the last arriving input and S is the separation time when locked. Because the stage delay increases as the Separation time decreases, the frequency of all locked rings is nearly the same. The cycle times for all rings that I simulated, when locked, fell between 153ps, when a ring is half full, and 208ps when 8 tokens occupied an eleven stage ring.

The peak to peak voltages for all nodes in the rings that locked fell between 1.49 and 1.58V.

The average current draw per stage when the ring is locked is within a couple percent of 1.4mA regardless of the number of stages or tokens. The peak to peak current draw per stage varies dramatically depending on the number of distinct phase in a ring. An eight stage ring loaded with four tokens has sixteen nodes that represent four copies of four distinct phases. If the same ring is loaded with two tokens, it would have two copies of eight distinct phases. A ring with a high number of distinct phases draws nearly constant current because the current needs of the gates in the various stages are evenly spaced through time. Figure 6 shows the relationship between the peak to peak current, or the magnitude of the AC current, per stage with respect to the number of distinct phases found in the ring.

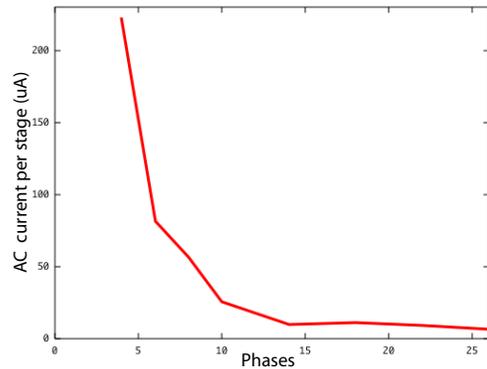


Figure 6. Peak to peak current draw per stage in rings of FIFO with different numbers of distinct phases

The rings operate under a wide range of supply voltages. Figure 7 shows the relationship between the frequency at

which the ring oscillates and the supply voltage. The nearly linear relationship suggests that an analog FIFO ring is well suited for use as a frequency locked loop.

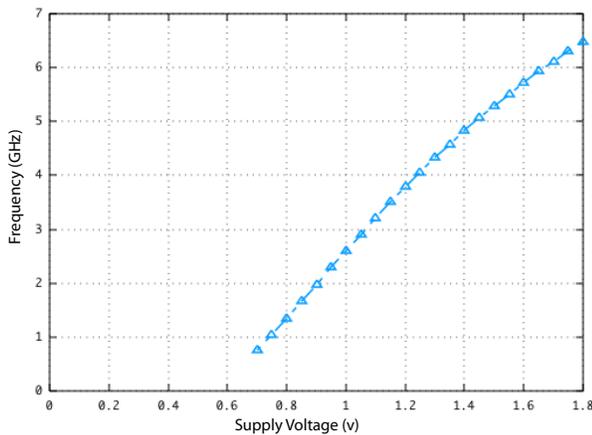


Figure 7. Frequency against supply voltage

Table 5 shows how transistor mismatch affects the performance of the FIFO rings. I use a mismatch model that chooses a threshold voltage for each transistor from a 1 sigma Absolute Gaussian distribution of 10mV and a 1 sigma Absolute Gaussian current matching distribution of 2%. Hspice performed a thirty sweep Monte Carlo simulation.

Table 3 shows how transistor mismatch accumulates through a string of unloaded inverters using this process and with the applied mismatch modeling. These numbers are presented as a reference point since there does not appear to be a standard for reporting transistor mismatch data. The last three columns report the average deviation from the ideal phase with inverters of fanout 1, 2, and 3 respectively.

Of the 30 Monte Carlo sweeps for the fifteen stage FIFO ring, the ring failed to lock during one sweep, and the ring failed to sustain oscillations for one sweep. The results from those two runs are not included in the average deviation calculation for the data in Table 5. In the case where the ring failed to sustain oscillations, transistor mismatch caused one of the adjacent stages to respond sooner to a voltage transition from a stage than the other. This simulation exposes the limits of dependable operation for this FIFO control in the presence of transistor mismatch. Increasing the size of the keeper inverters to 1.5 times the size of the input inverters yielded robust and locked oscillations for all thirty sweeps. The average period of the signals slows to

203ps with this alteration. Increasing the keeper strength stabilizes the oscillation and keeps the output from switching until the inputs have decisively changed state.

I define the instant of an event in the FIFO rings as when the differential output values cross. An artifact of this choice is that there is no skew between a signal and its complement. The phases in the FIFO rings are listed next to their complement in Table 5. The maximum distance between any two gates in the fifteen stage inverter ring is fourteen gates. Because FIFO communication signaling is bi-directional, the furthest number of stages between any two nodes is half the number of FIFO stages. This accounts for part of the higher timing precision.

Another explanation for the higher precision is found by plotting the open loop separation curve for the analog C-element under different extremes of my chosen transistor mismatch model. I plotted five cases. In each case I assigned a different extreme threshold voltage and current factor to the NMOS and PMOS transistors. Figure 8 shows the results. The legend lists the extremes chosen for each plot. Each plot is represented by four couplets of letters and a negative or plus sign (+/-). ‘I-n=+’ means that the NMOS transistors for this simulation received current factor parameters on the more conductive extreme of the mismatch model. ‘T-p--’ means that the PMOS transistors in this simulation received threshold voltage offsets on the later turn-on threshold voltage extreme of the mismatch model.

Notice that as the Separation distance between inputs shrinks, all five plots converge. When a FIFO ring is locked, it will be operating on the dynamic part of the Separation curve near the zero Separation point where these plots converge. This plot says that the locked rings are very insensitive to transistor mismatch when operating in this region. This helps to explain the small amount of skew between the different phases of the FIFO ring.

My best explanation for this phenomenon is that when the Separation time is short, the totem or crossover currents are the greatest. A transistor that is poor at conducting charge is also poor at siphoning charge from the output current when transitioning. When the separation time is long, the inverter connected to the input asserted first is fully conducting when the second inputs asserts and the output finally transitions. If the transistor is weak it won’t help the output transition much, if it is strong it will help the output transition much. The difference between the two cases is great. If the separation time is short, then if the transistor is weak it won’t help charge the output much, but it also won’t steal much charge from the output in crossover current. If the transistor is strong, it will contribute much charge to the output but it will also siphon much crossover current from the output. The differences between the strengths of transistors are mitigated. I’m not completely satisfied with this answer, but it is the one I’ve chosen to sell to the public.

3 Stages 2 Tokens		4 Stages 2 Tokens		5 Stages 2 Tokens		7 Stages 4 Tokens		13 Stages 8 Tokens		15 Stages 8 Tokens	
179ps avg period		154ps avg period		162ps avg period		156ps avg period		165ps avg period		158ps avg period	
phases	avg. dev	phases	avg. dev	phases	avg. dev						
0,3	0	0,2	0	0,5	0	0,7	0	0,13	0	0,15	0ps
1,4	0.8ps	1,3	1.5ps	1,6	0.6ps	1,8	0.8ps	1,14	0.8ps	1,16	0.8ps
2,5	0.6ps	0,2	1.5ps	2,7	1.9ps	2,9	0.9ps	2,15	0.8ps	2,17	1.7ps
		1,3	1.5ps	3,8	2.0ps	3,10	1.4ps	3,16	0.8ps	3,18	2.3ps
				4,11	1.5ps	4,9	1.7ps	4,17	1.0ps	4,19	2.5ps
						5,12	1.6ps	5,18	1.2ps	15,20	2.5ps
						6,13	1.9ps	6,19	1.1ps	6,21	2.7ps
								7,20	2.6ps	7,22	2.9ps
								8,21	1.6ps	8,23	2.3ps
								9,22	1.9ps	9,24	3.2ps
								10,23	0.9ps	10,25	3.5ps
								11,24	0.9ps	11,26	3.4ps
								12,25	1.1ps	12,27	3.0ps
										13,28	2.2ps
										14,29	1.5ps

Table 5. Rings with the average deviation from the ideal phase from 30 Monte Carlo sweeps

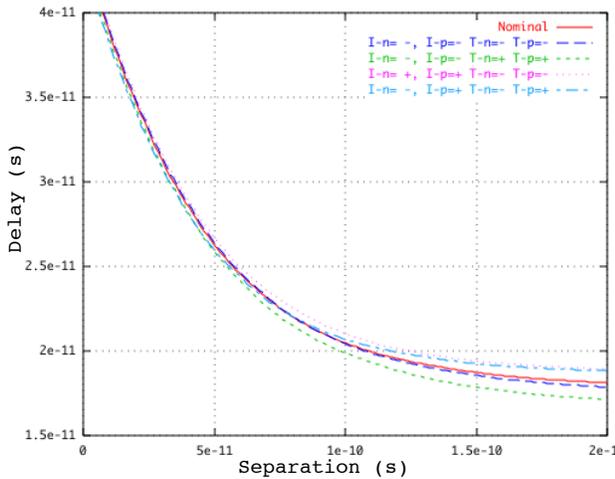


Figure 8. Open loop separation curve for Analog C-element with different extremes of transistor mismatch

15 Stage Inverter Ring			
period →	826ps	1270ps	1730ps
Stage	FO1	FO2	FO3
0	0ps	0ps	0ps
1	0.8ps	1.7ps	2.5ps
2	1.2ps	2.3ps	3.5ps
3	1.5ps	2.7ps	4.1ps
4	2.1ps	3.2ps	4.8ps
5	2.3ps	3.3ps	5.2ps
6	2.4ps	3.5ps	5.2ps
7	2.5ps	3.9ps	5.9ps
8	2.6ps	4.0ps	6.3ps
9	3.1ps	4.7ps	7.3ps
10	3.2ps	5.0ps	8.0ps
11	3.4ps	5.2ps	8.7ps
12	3.6ps	5.4ps	9.1ps
13	3.5ps	5.4ps	9.2ps
14	3.8ps	5.6ps	10.2ps

Table 3. Transistor mismatch skew with inverters of different fanout

3. Applications

This section discusses potential applications for the technology introduced in the first part of this paper.

3.1. Analog Digital Conversion

As the frequency at which we modulate data increases, so will the sampling frequencies needed to convert that signal to the digital domain. High frequency signals turn simple interconnect into transmission lines and excite reactive parasitics that would otherwise remain unobservable. Because the rings of asynchronous control that I have presented are able to precisely demarcate time in quantities much smaller than the cycle time, they can be used to sample analog signals at relatively low frequencies while still fulfilling the Nyquist rate criteria.

The Nyquist criteria says that to reconstruct an analog signal from a digital data, the analog signal must be sampled at a frequency that is twice that of the highest frequency component of the analog signal. Rather than sample with a high frequency signal equal to or greater than the Nyquist rate, a relatively low frequency ring that generates multiple evenly spaced phases can be used. For example, an 18Ghz analog signal would need to be sampled at 36GHz to satisfy the Nyquist criteria. Alternatively, one could use the precisely demarcated phases generated by a three stage FIFO ring with six phases that operates at 6Ghz. Each phase could be used to sample the analog signal. This scheme achieves the necessary 36Ghz sampling rate.

3.2. Clock Recovery

Clock Recovery circuits are used between communicating synchronous modules that don't share the same clock. Based on the frequency and phase of the sent data, a local clock is 'recovered' by the receiver.

A clock recovery solution is sketched in Figure 9. Seven samples of the data line are used to resolve the phase difference between the sending data and a clock signal generated by a FIFO ring. Each phase and its complement operate a passgate latch. Each latch samples the input data line. Each of the latches are pulsed by an adjacent phase of the control ring. XOR-ing the outputs of adjacent latches reveals the relative phase of the sending and receiving clock. The latches controlled by signals advanced in phase relative to the sending clock will all successfully latch the data, and the XORs will not assert. The latches controlled by signals retarded in phase relative to the sending clock will all fail to latch the data and the XOR connected to these latches will not assert. The latch controlled by a signal in phase with the sending clock will just barely latch the data. The XOR with inputs from the output of that latch and its predecessor latch will assert because only it's two inputs differ.

Depending on the relative phase of the received data, the supply is momentarily throttled or reduced to a different level, temporarily changing the frequency and adding or subtracting phase from the loop. The more phases that the

receiver uses to locate the relative phase of the received data, the more likely a latch will go metastable and fail to detect the transition. But then the next latch will certainly detect the edge and trigger the later XOR. The more phases that are used to detect the clock edge, the better the accuracy and the more likely that the metastability will occur. The only adverse effect is that the receiver's clock will be advanced or retarded perhaps a bit too much. This is the quantization noise of the system.

Oversampling clock recovery circuits have been proposed[11]. Their effectiveness is limited because of the tradeoff in the number of sampling phases against the frequency of the local clock. More phases requires more stages in the local oscillator. More gate delays mean slower frequencies. Phase interpolators [8] can be used to multiply the number of phases without compromising frequency but multiple oscillators are needed and the jitter on the derived edges limits the effectiveness of this solution.

3.3. Clock Generation and Distribution

The number of gate delays per pipeline stage is steadily reducing with each new generation of high performance processor. Minor sources of skew on the order of fractions of a gate delay were once negligible, but are now significant.

- Generating complementary signals by running a signal through short uneven chains of inverters results in uncertainties that must be modeled as skew. Half a gate delay of skew is significant when logic paths are dipping below ten gate delays [1]. The ideas presented above yield low-skew *fully complementary signals*. This frees valuable pico-seconds to perform computations and allows clock frequencies to continue increasing.
- Harris [5] derives the following equation to relate the maximum tolerable clock skew for Domino logic as a function of the number of clock phases (N) used.

$$t_{max-skew} = \frac{\frac{N-1}{N} * T_c - t_{hold} - t_{precharge}}{2} \quad (3)$$

Using this equation he presents the following table of values for maximum clock-skew and necessary pre-charge time for different values of N assuming a sixteen gate delay clock period.

Notice that as the number of clock phases increases, the maximum skew tolerated also increases. Unfortunately the extra clock skew gained by using more phases is not available at higher frequencies where it is most needed using the current solutions detailed in Harris' book. FIFO rings remove this limit.

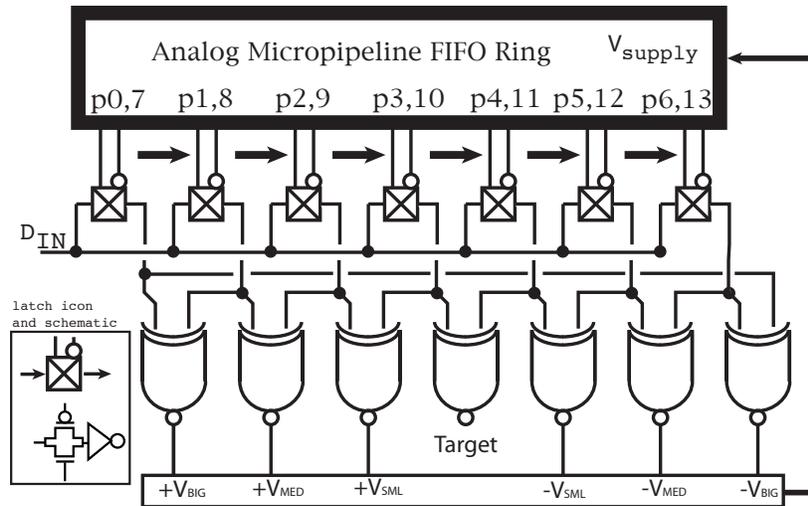


Figure 9. Clock Recovery Circuit using Micropipeline FIFO Ring

N	Precharge	Maximum Skew
2	6	2
3	7.33	3.33
4	8	4
6	8.66	4.66
8	9	5

Table 6. Skew tolerance for various numbers of clock phases, from *Skew-Tolerant Circuit Design* by David Harris [5]. Numbers represent gate delays in a 16 gate delay cycle time.

Output prediction logic[6] is a high performance CMOS design technique. Though promising great speed benefits over even Domino data paths, its adaption has been limited. One study asserts [3]

Generating clock phases accurately with this small of a delay may be infeasible thus limiting the performance gains of OPL gates.

The authors described clocking solution[7] uses a $2.8 \times FO_4$ clock period with six phases. Each phase has a $\pm 15^\circ$ resolution. A three stage FIFO ring with my chosen mismatch models yielded a $1.8 \times FO_4$ clock period with a $\pm 2^\circ$ phase resolution.

Multiple clock phases are typically generated by sending a copy of the clock signal through an open loop chain of inverters. New phases can't be derived at a resolution finer than a gate delay using this method. The phase generation method presented here does not have this limitation. A large

number of precise phases can be derived for even the most aggressive clocking strategies.

The open loop phase generation method has other drawbacks:

- The period through the open loop approximates the clock period but is not constrained to the period of the clock. As soon as the signal leaves the frequency source it is free to drift with respect to the source.
- Assume the chip is clocked too fast and the frequency is reduced to accommodate slower than expected domino logic. The domino style gates receive a phase derived by tapping a signal after a string of inverters. Just because the clock is slowed, the delay between the inverters does not change and the timing violation will remain.

If the speed of a ring made from analog Micropipeline circuitry is reduced by adjusting its supply or through a current limiting method, the tokens stay locked in the ring. When the frequency of the clock is reduced, the delay between phases grows proportionally.

4. Conclusion

I have presented a novel method of achieving high resolution on chip timing. The extent to which time can be demarcated is the limiting factor in many VLSI applications. I discussed three of these applications.

Future work includes:

- Detailed switching noise simulations with extracted parasitics from layout.

- More detailed analysis of the analog C-element with buffering.
- Design of a clock recovery circuit built around an analog Micropipeline ring.

The group here at Cambridge is in the process of building a test chip in collaboration with MIT scheduled for tape out sometime during the summer of 2004. The ideas presented here will be a part of that chip.

5. Acknowledgments

Thanks to Andrew West for squeezing microns from the Analog C-element layouts.

References

- [1] S. I. Association. National technology roadmap for semiconductors. *SEMATECH*, 1997.
- [2] V. Chandramouli, E. Brunvand, and K. F. Smith. Self-timed design in GaAs—case study of a high-speed, parallel multiplier. *IEEE Transactions on VLSI Systems*, 4(1):146–149, Mar. 1996.
- [3] A. Chang and S. Kao. A study and comparison of an output prediction logic adder.
- [4] J. C. Ebergen, S. Fairbanks, and I. E. Sutherland. Predicting performance of micropipelines using Charlie diagrams. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 238–246, 1998.
- [5] D. Harris, editor. *Skew Tolerant Circuit Design*. Morgan Kaufmann Publishers, Inc., 2001.
- [6] C. Sechen. Output prediction logic: A high-performance cmos design technique. In *Proceedings of the 2000 IEEE International Conference on Computer Design*, page 247. IEEE Computer Society, 2000.
- [7] C. Sechen and L. McMurchie. Slides:output prediction logic: A high performance cmos design technique.
- [8] S. Sidiropoulos and M. Horowitz. A semidigital dual delay-locked loop. *IEEE Journal of Solid State Circuits*, 32(12), November 1997.
- [9] I. E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720–738, June 1989.
- [10] A. J. Winstanley, A. Garivier, and M. R. Greenstreet. An event spacing experiment. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 47–56, Apr. 2002.
- [11] C.-K. K. Yang and M. Horowitz. A 0.8 μ m cmos 2.5gb/s oversampling receiver and transmitter for serial links. *IEEE Journal of Solid State Circuits*, 31(12), December 1996.