

# A Distributed Synchronized Clocking Method

David R. Rolston, *Member, IEEE*, David M. Gross, Gordon W. Roberts, *Fellow, IEEE*, and David V. Plant, *Senior Member, IEEE*

**Abstract**—This paper will present a novel method to generate and distribute a synchronous clock to multiple nodes in a distributed system. Total system synchronization is established by adjusting the internal delays of each node so that the delay between all adjacent pairs of nodes becomes identical. The system is based on the principles of phase-locked and delay-locked loops but does not discuss the methods and details of phase acquisition, jitter or lock-in time. The system is composed of a master node used to generate clock pulses and multiple slave nodes used to align the pulses. A Matlab Monte Carlo simulation of the linear behavior of the system is presented which not only validates the theoretical description, but also can be used as a good tool to gauge the performance of any particular system scenario. Selected HSpice simulations are then presented which show the operating characteristics of certain scenarios involving differing interconnect lengths between nodes that correspond to specific Matlab simulations.

**Index Terms**—Clock, distributed, phase-locked loop (PLL), synchronization.

## I. INTRODUCTION

A PERIODIC pulse train of voltage or current is the typical method used to synchronize a digital system. The common H-tree architecture has been exploited in many different designs and has proved to be extremely effective for intra-chip operation even at operating frequencies greater than a gigahertz [1]. Other techniques, such as the rotary traveling-wave oscillator [2], or cooperative ring oscillators [3] offer substantially the same intra-chip clock distribution but use means other than the H-tree. An example of the requirements for well-synchronized intra-chip clock distribution is the microprocessor pipeline. A pipeline typically contains multiple stages of registers separated by combinational logic and uses a synchronous clock distributed to each register to step data through the pipeline [4]. The arrival of the clock pulses at each register must be very well synchronized and they are typically derived from a common

clock source [5]. However, there are some situations where synchronous operation has been replaced by a combination of protocols and memory buffering. This has been done because maintaining perfect clock synchronization among many sub-systems has been extremely difficult to maintain, especially when the subsystems are substantial distances apart (further than a few centimeters). A method of measuring the send and return arrivals of a pulse along a long transmission line to create multiple synchronous clocks was proposed by Grover [6] and highlights the utility of highly synchronous systems. The distribution of a central clock signal to multiple distant points creates a delay and skew problem because the distances cannot be precisely determined or modeled. To adapt differing data rates and eliminate skew and transfer problems from one region to another many computer architectures, such as the IA-64 microprocessor [7], have used intricate phase-locked loop (PLL) techniques, FIFO memories and control signals to regulate the flow of data between two locally synchronous but globally asynchronous sub-systems such as external memory and internal CPU cache.

This paper describes a method of generating and synchronizing clock pulses that are distributed to multiple end points independent of their separation. The distributed synchronous clock (DSC) generates, distributes, and maintains a globally synchronized periodic pulse-train to multiple distant nodes in a system such that every node receives a pulse at precisely the same moment regardless of its position in the system. The DSC system is a fully integrated technology based on PLL techniques and is essentially independent of the transmission medium among the nodes. It can be applied to long distance communications, within electrical or optical backplane or bus structures, and eventually to inter-chip and intra-chip applications as circuit densities increase. The DSC also provides a means for distributed control using each node in the system to adjust its own internal characteristics to maintain synchronization.

This paper will begin with a description of the basic concept of the DSC. A block diagram will be provided and the method of generating and maintaining a balanced system will be outlined. In Section III, the block diagram in Section II will be formulated as an HSPICE CMOS circuit and will be briefly described and characterized. Since the emphasis of this paper is to present a working concept, detailed analysis of the PLL structures, such as lock-in time and jitter, will not be addressed. Section IV will provide details of the circuit simulation results and Section V will present an analytic model for the steady-state solution of the circuit. The model will provide insight into the operating range of the circuit. Finally, the last sections will discuss potential applications and proposed advancements in the circuit structure that may be implemented.

Manuscript received September 8, 2003; revised August 27, 2004. This work was supported by a Natural Sciences and Engineering Council (NSERC) Collaborative Research and Development (CRD) grant. This paper was recommended by Associate Editor C.-W. Jen.

D. R. Rolston was with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada. He is now with the Reflex Photonics Inc., Montreal, QC H3A 1B9, Canada (e-mail: drolston@reflexphotonics.com).

D. M. Gross was with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada. He is now with the Teradyne Systems Inc., Boston, MA 02118-2238 USA.

G. W. Roberts is with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada, and also with DFT Microsystems Inc., Montréal, QC H3A 2E6, Canada.

D. V. Plant is with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada.

Digital Object Identifier 10.1109/TCSI.2005.851683

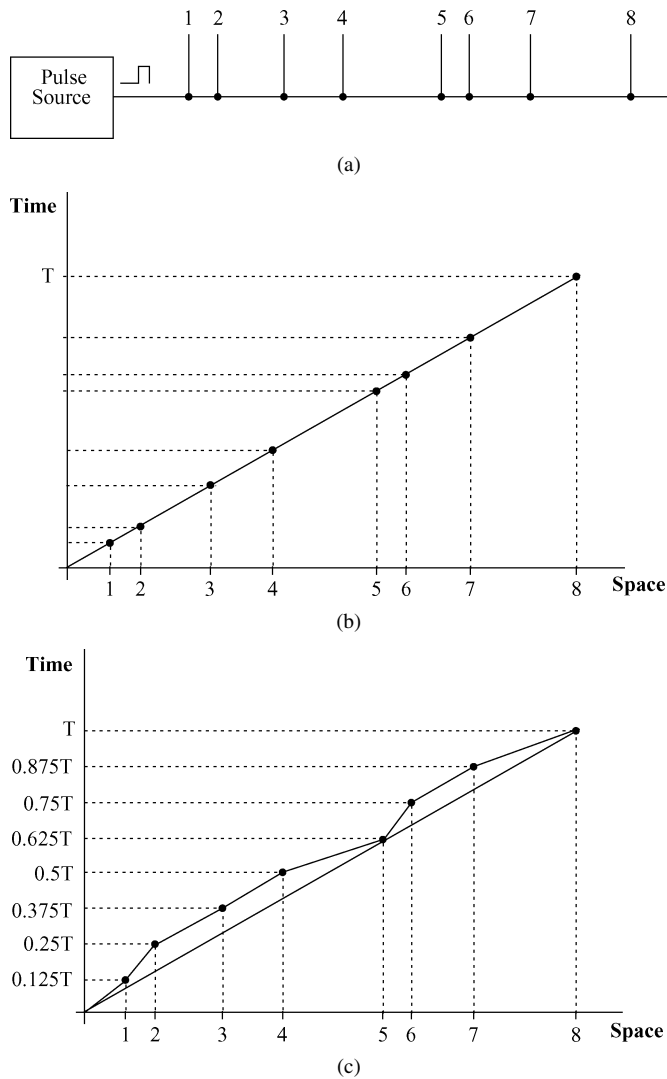


Fig. 1. (a) Tapped transmission line. (b) Space-time diagram for a constant velocity pulse. (c) Space-time diagram for a "balanced" system.

## II. THEORY

The fundamental theory behind the work presented in this paper is predicated on the ability of the system to change the average velocity of a pulse between pairs of nodes in the system. Fig. 1(a) shows a signal line with eight (8) randomly spaced tap points and Fig. 1(b) shows the corresponding space-time diagram assuming a *constant average velocity* along the entire length. Fig. 1(b) shows the "random arrival times" of the pulses along the vertical axis corresponding to the "random positions" of the nodes. Fig. 1(c) then shows the space-time diagram *AFTER* the signal line has been *balanced*; where the average velocity between pairs of nodes is now *DIFFERENT* with respect to the other pairs of nodes. This method essentially forces the arrival times of the pulses to become equal to each other affecting the spacing along the vertical axis but not the total time. This in turn affects the average velocity of the pulses between pairs of nodes since the nodes' physical positions are fixed. In this case, with eight nodes and a total delay of  $T$  s, each node must receive a pulse every  $0.125 T$  s.

The inherent tradeoff with this type of circuit is between the precise phase alignment of the signals and the system's

overall frequency. To achieve precise phase alignment, the system's oscillation frequency must adapt to the total delay of the system and is therefore determined by the average delay between nodes. Although telecommunication applications such as the SONET/SDH time base standards require very stringent clocking and locking requirements so that the sampling rates and the quality of service (QoS) can be maintained, for computing applications these requirements may be somewhat more relaxed and allow for less precise clock oscillation frequencies. The proposed system can usually be made such that the operating frequency is always within a certain range (for example, always between 450 and 550 MHz). Therefore, it seems reasonable that as long as the logic, buffering and memory can still operate, it is possible to take advantage of the precisely phase aligned clocking signals to accomplish such architectures such as pipelining but on a larger scale. An example of an arbitrarily set clock frequency computing environment is the common practice of overclocking (or underclocking) a CPU within a PC to push the performance of the CPU. In such a case, the clock frequency is typically arbitrarily set by varying external resistors and/or capacitors. The PC functions up to a certain point after which errors start to occur. The point here is that a processor chip set, unlike a digital signal processing (DSP) sampling chip set, may not have to perform at a fixed and exact frequency.

### A. Circuit Implementation

There are two main aspects of the DSC system: 1) the method of clock generation and distribution and 2) the method of phase alignment within each node of the system to balance the entire system. The clock-pulse generation and distribution are based on a ring topology where all nodes in the system are sequentially connected and the last node is connected back to the first. The pulses originate from the master node and propagate through each node (the slave nodes) in the system. The clock-pulse generator must produce as many pulses as there are nodes in the ring (or at least an integer multiple). The phase alignment within each node is accomplished by adjusting pre and post internal delay lines to phase align a clockwise and counterclockwise propagating pulse. Although an individual slave node cannot balance the entire system by itself, it can adjust the phase error between two incoming pulses so that they eventually both arrive at the same time at the same point. It is only when all slave nodes are performing this phase alignment that the system can become synchronized assuming the number of pulses equals the number of nodes (or at least an integer multiple). The method each slave node uses to align the system is analogous to equally spaced cars on a highway. If each car measures the distance to the next and previous car, then all cars can eventually become equidistant from one another by adjusting their velocities; this concept is alluded to in [8]. A critical criterion of the clock-pulse alignment is that the mechanisms within each slave node must be independent from each other and act only on the pulses local to the node since there is no global control system.

### B. Clock-Pulse Generation

The clock pulse generator is based on a PLL design where the output of the oscillator is split so that one signal is directly

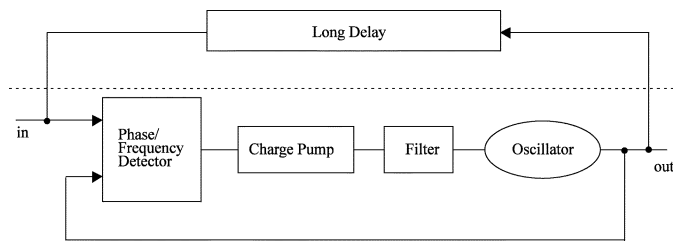


Fig. 2. Fundamental concept for clock generation and distribution using PLL concepts.

fed back into the phase-frequency detector (PFD) and the other is passed through a long delay as shown in Fig. 2. Although this configuration appears to be somewhat of a “free-running oscillator,” the oscillation frequency is bounded by the length of the long delay line. This “bounding effect” will become more evident in the next section once the circuit has been augmented to include the other two (2) delay lines and an  $n$ -bit counter. Assuming that Fig. 2 is in fact composed of seven individual nodes, the block diagram can now be redrawn as that in Fig. 3(a). In Fig. 3(a), node 8 is the master node and generates the clock pulses and the seven other nodes in the ring are the slave nodes. Note that it is important to include the delay of the eighth node as shown by the elements in Fig. 3(a) called “half delays;” otherwise the timing of the system becomes more difficult to calibrate. The master node contains a voltage-controlled oscillator (VCO), a phase-frequency error detector (PFD), and a loop filter. Once the long-delay line of slave nodes has been sufficiently primed with pulses, the PFD can lock the VCO to a particular frequency so that an integer number of pulses exist in the long-delay line of nodes and a steady-state behavior can be achieved. For example, if each slave node had a delay of 10 ns (and both “half delays” had a delay of 5 ns) for a total delay of 80 ns around the ring, then if the VCO was to operate at 100 MHz, the ring would carry exactly eight pulses. However, this circuit is not stable and can act as a free-running oscillator since other steady-state solutions exist. If the same slave node delay of 10 ns is assumed, an 87.5-MHz oscillation would produce exactly seven pulses in the ring separated by 11.42 ns or a 112.5-MHz oscillation would produce nine pulses separated by 8.88 ns.

By including a second ring [called the slow ring (SR)] identical to the first ring [now called the fast ring (FR)], and by including a synchronous counter (for example a 3-bit counter for eight nodes), this circuit can be improved so that the same number of pulses exists in the ring as there are nodes (or at least an integer multiple). As shown in Fig. 3(b), it must be assumed that the FR and SR rings follow the exact same path through the slave nodes and have the same delays between pairs of nodes. However, the SR is connected to the most significant bit of the synchronous counter, and it is this path that allows the correlation between nodes and clock pulses and eliminates the “free-running oscillator” nature of the circuit. The 3-bit counter increases the period of the 100-MHz oscillation from 10 to 80 ns in the most significant bit. Unlike the circuit in Fig. 3(a), the VCO is now constrained to work around only one operating point. If the VCO was to operate at 87.5 MHz (as above), seven pulses would exist in the FR, but only 0.875 of a pulse would

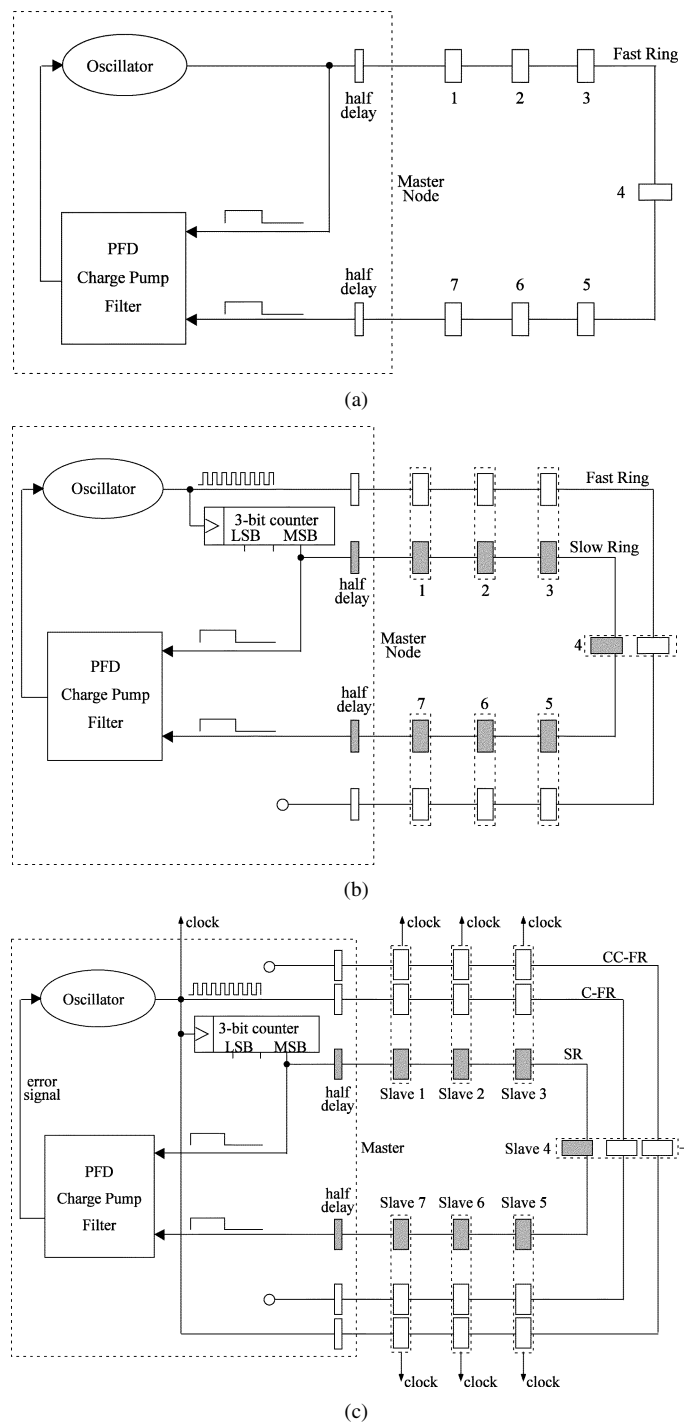


Fig. 3. (a) Block diagram for the first iteration of a clock generation and distribution circuit in the master-node. (b) Block diagram for the second iteration of a clock generation and distribution circuit in the master-node that correlates the number of pulses to the number of nodes. (c) Block diagram of final clock generation and distribution circuit in the master-node that includes an additional counter-propagating line for the slave-node pulse balancing.

exist in the SR. The next valid operating frequency for the VCO would be 200 MHz where 16 pulses separated by 5 ns would propagate in the FR and 2 pulses separated by 40 ns would exist in the SR.

Although the circuit of Fig. 3(b) can produce the same number of pulses as there are nodes, the pulses may not all arrive at their respective nodes simultaneously simply because

TABLE I  
ACTIONS REQUIRED ON VARIABLE DELAY LINES WITHIN A SLAVE-NODE GIVEN THREE POSSIBLE PULSE ARRIVAL SCENARIOS AT REFERENCE LINE

Comparing Fast “cw” and “ccw” pulses	Variable Delay of C-FR		Variable Delay of CC-FR	
	Pre-delay	Post-delay	Pre-delay	Post-delay
C-FR arrives before CC-FR	increase	decrease	decrease	increase
C-FR arrives after CC-FR	decrease	increase	increase	decrease
C-FR same time as CC-FR	no change	no change	no change	no change

the interconnect links between nodes would not be identical to each other. The total delay may be 80 ns but the pulses may not be coincident with all the nodes all simultaneously. The master node PLL and the 3-bit counter can only determine the *average total delay* around of the system; an additional control mechanism is required in each node to align the pulses.

### C. Clock-Pulse Alignment

Since there is no centralized control to aid in synchronization, each slave node must use only local information to balance the system. The adjustment of a slave node's internal delay is also based on PLL/delay-locked loop (DLL) techniques to measure the phase error between two (2) signals. Therefore, the circuit proposed in Fig. 3(b) must be adapted to provide two (2) signals with which to form the error signals. The FR, which carries multiple clock pulses, is duplicated to include a second, counter-propagating fast ring. The fast rings are now called: the clockwise fast ring (C-FR) and the counter-clockwise fast ring (CC-FR). The pulses traverse the nodes in a sequential but opposite order. Fig. 3(c) shows the modified circuit that now includes three paths: the SR, the C-FR, and the CC-FR.

Using the C-FR and CC-FRs, each slave node can now detect three (3) different scenarios for the two incoming pulses: 1) a pulse on the C-FR arrives *before* a pulse on the CC-FR; 2) both pulses arrive simultaneously; or 3) a pulse on the C-FR arrives *after* a pulse on the CC-FR. The block diagram in Fig. 4 shows the internal structure of a slave node and where the reference point is defined. The signals called *error* and *error\_bar* are generated by the skew between the two (2) pulses at the reference point. These signals cause the variable delays called “Pre” and “Post” on each of the rings to change. Table I shows the scenarios and the possible actions taken by the slave nodes. For a constant frequency pulse train, it is then possible to lead one pulse and lag the other pulse so that they both eventually arrive at the reference point at the same time. Ideally, the “Pre” delay is equal in magnitude to the “Post” within the same slave node. This is required so that the total delay around the system remains constant implying a constant oscillation frequency from the master VCO. However, in real systems, such as the one presented in this paper, the “Pre” and “Post” delays may vary nonlinearly—the magnitude of the “Pre” delay may be more (or less) than the magnitude of the “Post” delay. In such a case, the total delay around the system could change implying that the oscillation frequency of the VCO must either increase or decrease. This nonideality is overcome by mimicking the delay changes of the C-FR and CC-FRs in the SR ring as shown by the extra arrows and darkened delay boxes in Fig. 4.

The complete DSC system, shown in Fig. 3(c), contains the master and slave node circuitry as well as the passive interconnect links between the nodes. The three signal paths: SR, C-FR,

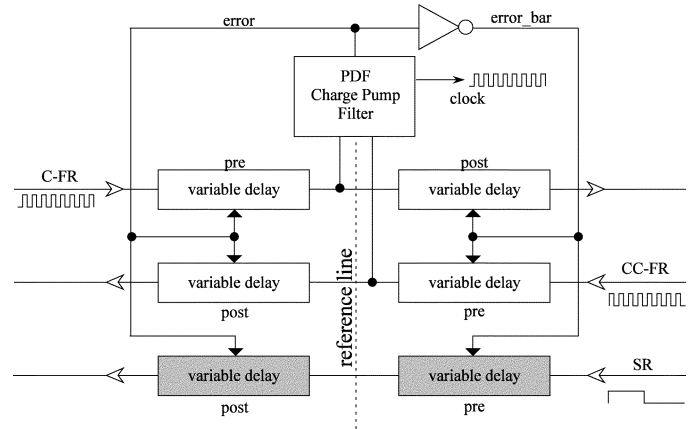


Fig. 4. Block diagram for the slave-node internal self-balancing circuit.

and CC-FR are also explicitly shown. The interconnect links could be coaxial cables, optical fibers, twisted pairs or any other transmission medium where either end of the interconnect is properly terminated to avoid signal reflections. Once the system is balanced, each node would output in-phase square-waves with the same frequency as the master node VCO.

The analytical model presented in the Section V assumes that all interconnect links (SR, C-FR, and CC-FR) between a pair of nodes are all equal. However, it is possible that these links could be somewhat mismatched and could be a source of phase error. Although the circuit presented in the next section explicitly uses three independent links between nodes, future designs will integrate the three signal paths into one physical medium. In [9], a fully bi-directional electronic circuit is described and may be used to pass both the C-FR signal and the CC-FR signal on the same physical medium (this is even more plausible if the medium is optical fiber based). Furthermore, the SR signal can be combined with the C-FR signal using a two level signaling approach (similar to PAM signalling). Strictly speaking, the only real function that the SR line serves is to create the same number of pulses as there are nodes in the system; if there are eight nodes, there must be eight pulses.

### III. CIRCUIT DESCRIPTION

The DSC system was simulated using level-3 parametric HSpice transistor models from a specific Mitel 1.2-micron CMOS wafer run. The circuits were almost entirely designed using CMOS structures and virtually all signals were single-ended 5-V rail-to-rail logic levels. There were a total of 2,131 transistors in the complete 8-node DSC system model, 225 transistors per slave node and 556 transistors in the master node. The model used some minor ideal components such as

pure resistor-capacitor networks for each loop filter, and an ideal voltage amplifier in the charge pumps.

The circuits described in this paper are used only as a proof-of-concept for the theory. Many circuit design improvements can be made to these circuits that will enhance the performance especially if the nonlinear behavior of the slave nodes is eliminated and the loop delay of the master PLL is reduced. Changing the circuit technology to a 3.3-V 0.25- $\mu\text{m}$  BiCMOS process or an heterojunction bipolar transistor (HBT) process would also greatly improve the performance but at the expense of more challenging circuit implementations.

#### A. Master Node Circuit

The master node circuit consists of a PFD, a charge-pump, a loop filter, a VCO, a 3-bit synchronous counter and six constant delay blocks (the half delay elements). These elements are standard subcomponents used in a digital PLL (DPLL) designs [10]. Most of these circuit structures are standard but some subtleties were borrowed from the literature [11]. The only notable subcircuit in the master node is the fixed-delay element. These delays are used to mimic the first and second halves of the nominal delay of a slave node. These half delays are required so that the master node also appears to have the same total nominal delay as the other slave nodes.

#### B. Slave-Node Circuit

The slave node consisted of a PFD, a charge-pump, a loop filter, and six variable delay lines. The charge-pump and loop filter were identical to that of the master node, but the PFD circuit and variable delay line required a more specialized design. The PFD circuit was designed to minimize the variable delay line bias voltage, and the delay line had to be carefully analyzed to determine its operating range. The slave-node PFD was designed to produce a pulse that was only as wide as the *difference* between the closest pair of rising edges. This new circuit could limit the pulsewidth to, at maximum, half the original pulsewidth of the master PFD. Details of this and the other circuits used in both the master and slave nodes can be found in [12]–[14].

The variable-delay lines in the slave nodes consist of chains of tunable current-starved inverters using source-drain MOSFET resistances. To maintain a 50/50 duty cycle for the pulses passing through the variable-delay line, pairs of well-matched inverters were placed before each delay element. As shown in Fig. 5, there were a total of eight delay elements in each variable delay line that were all controlled by the same bias voltage. In addition, there was a capacitance of 100 fF attached to the output of each delay element to help increase the effective RC constant of the delay element. In Fig. 6, the curve “variable delay line” shows the plot of the delay versus bias voltage. The performance was slightly nonlinear within the desired operating range. The nominal delay of the delay line at 2.5 V was 12 ns. In Fig. 6, the curve “two complementary biased variable delay lines” shows the plot of two cascaded variable delay line with opposite voltage bias. The total nominal delay is 23.5 ns and it increases with either an increase or decrease in bias voltage—ideally, this curve should have been flat, i.e., no change with bias voltage. The useful operating range of this

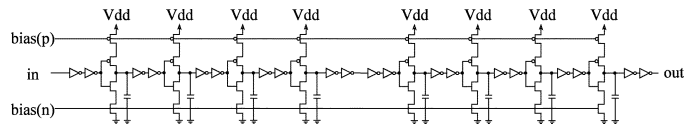


Fig. 5. Transistor-level model for the variable delay line within each slave-node.

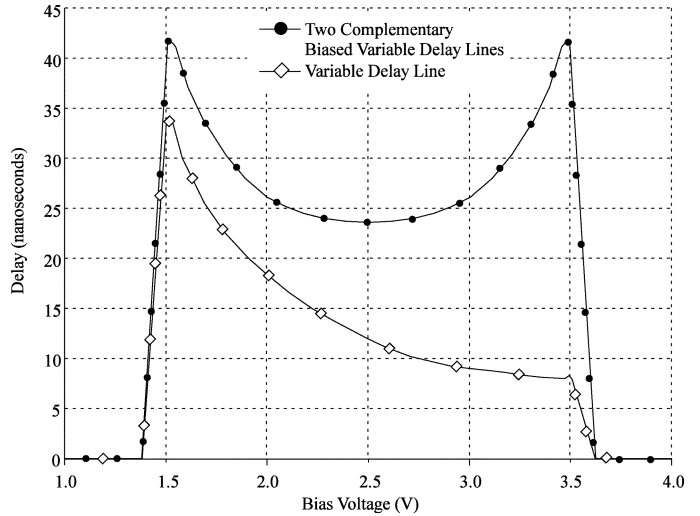


Fig. 6. Variable delay line response w.r.t. applied voltage bias for a) the circuit of Fig. 5, and b) two Fig. 5 cascaded circuits with complementary bias conditions.

circuit was assumed to be between 1.8 and 3.2 V. These limits were obtained empirically through HSpice simulations where it was determined that the variable delay line did not function properly beyond these limits. Other delay lines can also be considered, such as differential delay elements, or other single ended delay elements such as that found in [15] and [16].

#### C. Transmission-Line Interconnect Links

The interconnection link between slave nodes was modeled in HSpice using the ideal (or lossless) transmission line model that was based on a coaxial cable with zero loss-tangent; the impedance and total delay could be specified. The HSpice simulation incorporated a 50-Ohm impedance matched transmission line between each pair of nodes. The total delay of each transmission line was set at the beginning of each simulation but the set of eight delays modeled a typical random variation in lengths (and therefore total delay) for the eight interconnects.

### IV. HSPICE SIMULATION

The system's lock-in sequence was partitioned into four (4) regions: 1) the reset region, 2) the pulse priming region, 3) the master node lock-in region, and 4) slave node lock-in region. Fig. 7(a) shows a typical plot of the VCO bias voltage in the master node. During the first 50 ns, the 3-bit counter was reset and all PLL/DLL action was disabled. In the next 700 ns the 3-bit counter was enabled and the VCO was allowed to operate at a nominal voltage bias of 2.5 V where a square-wave pulse train with a frequency of 30 MHz was produced and allowed to propagate along each path. Note that the nominal voltage bias of 2.5 V was designed (by way of appropriately designing the VCO) to produce *approximately* eight pulses within the C-FR

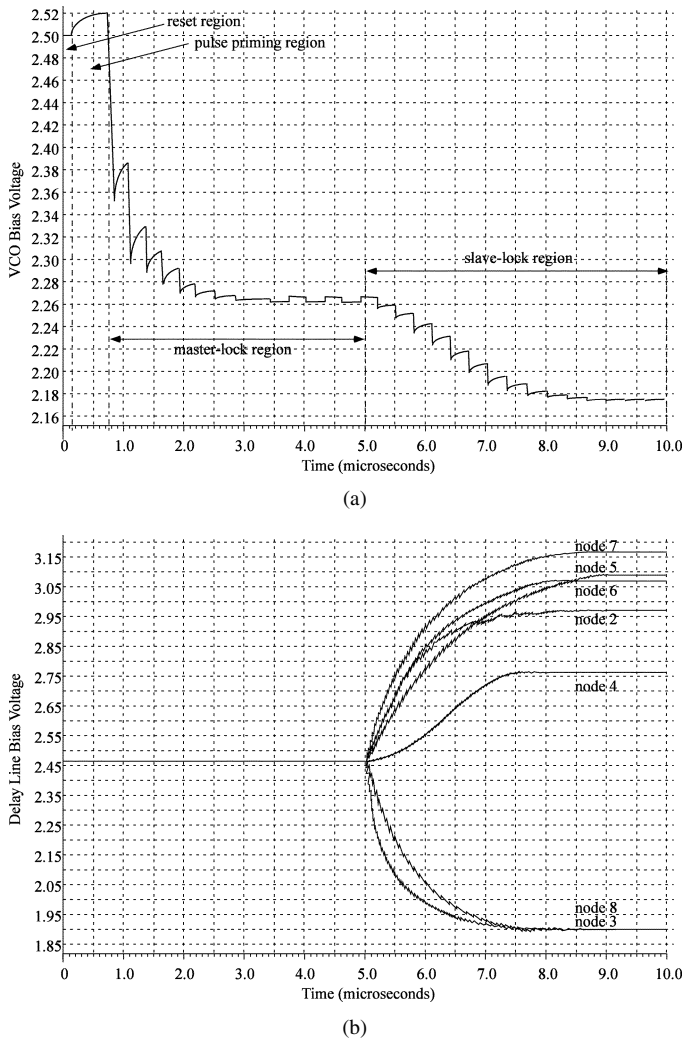


Fig. 7. (a) Typical response of the "error signal" to the VCO of Fig. 3(c) given a particular operating scenario. (b) Typical responses of the "error signals" to the variable delay lines of Fig. 4 in each of the seven slave-nodes in the system.

and CC-FR paths and *approximately* 1 pulse in the SR path. During this region, the slave nodes had their variable delay line bias voltage set to 2.5 V so that each slave node had a total delay of 23.5 ns. After a sufficient amount of time, the master node PLL was enabled and the bias voltage on the master node VCO began to re-adjust based on the phase errors between the SR path and the internal loop-back path through the 3-bit synchronous counter. The bias voltage on the master node's VCO monotonically moved toward another bias voltage on which it finally settled. Once the master node steady state was reached, exactly eight pulses existed in both the C-FR and CC-FR paths and 1 pulse existed in the SR path.

The final region of the DSC system balancing was to enable the DLL action within each slave node. A signal was sent to all the slave nodes from the master node (using a daisy-chained or any other reasonably efficient method) to enable each slave node DLL. The typical settling time of the slave-lock region was roughly 3500 ns, but again depended greatly on the RC loop-filter characteristics in each slave node and the interconnect link scenario. As shown by the slave-lock region in Fig. 7(a), the bias voltage on the master node VCO also had to re-adjust

TABLE II  
TYPICAL CIRCUIT DATA—OPERATING FREQUENCY, SKEW, BIAS  
VOLTAGE—FOR THE OPERATING REGIONS DEPICTED IN FIG. 7(a)

Values of interconnect link lengths (nominal 10-nsec):	
L1	10.212 nsec
L2	10.332 nsec
L3	11.141 nsec
L4	10.544 nsec
L5	9.780 nsec
L6	9.556 nsec
L7	11.002 nsec
L8	10.864 nsec
Initial Period of Oscillation of the VCO:	
	30.0 nsec (33.24-MHz)
Time Period between 750-nsec and 5000-nsec:	
Duration of Transient for Master-PLL:	2.518 sec
Steady-State value of analog voltage on VCO:	2.68 Volts
Period of oscillation of VCO in steady-state:	34.2 nsec (29.24-MHz)
Total skew of rising-edge of node clock:	1.80 nsec
Total skew of falling-edge of node clock:	1.74 nsec
Time Period between 5000-nsec and 10000-nsec:	
Duration of Transient for Master-PLL:	3.7 sec
Steady-State value of analog voltage on VCO:	2.18 Volts
Period of oscillation of VCO in steady-state:	33.4 nsec (29.94-MHz)
Total skew of rising-edge of node clock:	0.90 nsec
Total skew of falling-edge of node clock:	0.66 nsec

due to the nonlinear behavior of the slave node's variable delay lines. Fig. 7(b) shows how each slave node's variable delay line voltage bias changed until steady-state behavior was reached.

HSpcice simulations were run for several different interconnect link length scenarios. An interconnect link length is the physical medium connecting two nodes—this could be a coaxial cable or an optical fiber. One simulation used link lengths that were all close to the nominal value of 10 ns. These were  $L1 = 10.212$  ns,  $L2 = 10.332$  ns,  $L3 = 11.141$  ns,  $L4 = 10.544$  ns,  $L5 = 9.780$  ns,  $L6 = 9.556$  ns,  $L7 = 11.002$  ns, and  $L8 = 10.864$  ns. The simulation was run for 10 000 ns with 0.1-ns step size and Table II summarizes certain characteristics of the system. Another simulation used more widely varying interconnect link lengths:  $L1 = 22.905$  ns,  $L2 = 10.154$  ns,  $L3 = 11.356$  ns,  $L4 = 20.349$  ns,  $L5 = 8.480$  ns,  $L6 = 15.898$  ns,  $L7 = 9.849$  ns, and  $L8 = 15.669$  ns and the results, before and after system balancing are shown in Fig. 8(a) and (b), respectively. This simulation had a spread of 7.95 ns before slave-lock and a spread of 0.8 ns after slave-lock; this is a 163.4% relative improvement—this is an example where HSpice was able to converge to steady-state value, any residual spread in the pulses can be attributed to the nonlinearities within the slave node variable delay lines. An example of an extreme case for interconnects link length before and after system balancing are shown in Fig. 9(a) and (b), respectively. In this case, the variable delay lines within each slave node did not have sufficient delay to properly align the pulses.

## V. ANALYTICAL MODEL

To verify that the DSC system is capable of balancing a wide variety of interconnect link length scenarios, a simple analytical model of the system was developed that could be simulated using a Monte Carlo analysis in MATLAB. By writing out the seven unique expressions (for an 8-node system), (1a)–(1g),

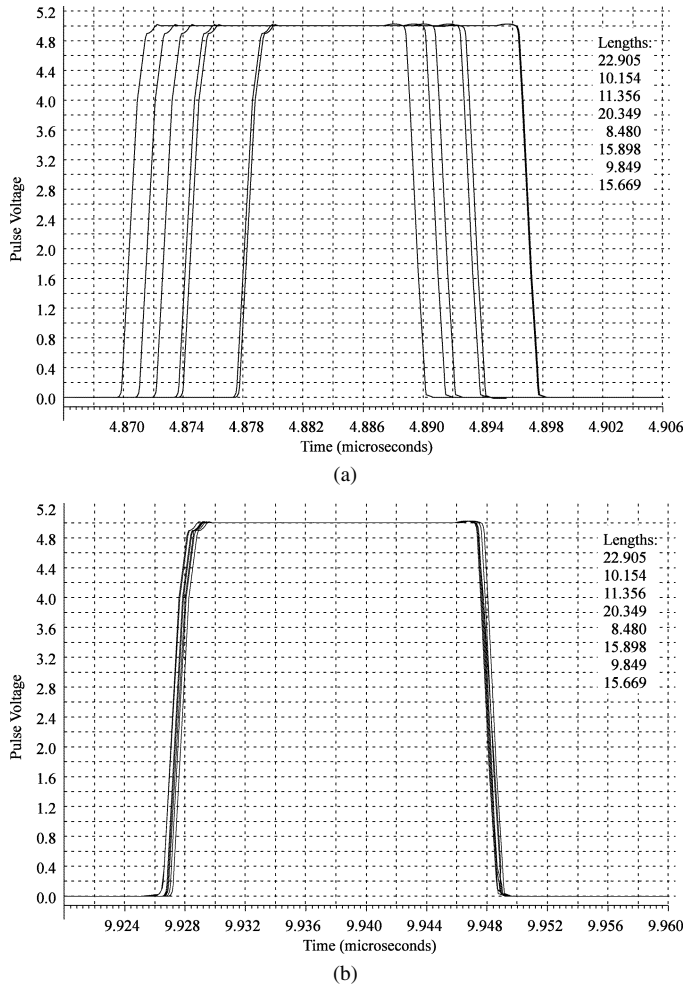


Fig. 8. (a) HSpice simulation of the clock outputs of the seven slave-nodes BEFORE pulse alignment. (b) HSpice simulation of the clock outputs of the seven slave-nodes AFTER a SUCCESSFUL pulse alignment.

that equate the clockwise and counter-clockwise delays from a reference to each slave node, the slave node differential delays could be calculated and provide a unique solution for each scenario of interconnect link length delay. Equations (1a)–(1g), (2) and (3a)–(3e) are derived from Fig. 10 and involve the interconnect link delays:  $L1, L2, L3, L4, L5, L6, L7,$  and  $L8$ , and the nominal internal slave node delays:  $Ai, Ao, Bi, Bo, Ci, Co, Di, Do, Ei, Eo, Fi, Fo, Gi, Go, Xi,$  and  $Xo$ . These in turn generate the required variable internal slave node delays:  $\Delta A, \Delta B, \Delta C, \Delta D, \Delta E, \Delta F, \Delta G, \Delta X$  for a balanced system (typically  $\Delta X$  was set to zero). For simplicity, the reference points for the master and slave nodes in the analytical model are labeled:  $\xi, \alpha, \beta, \chi, \delta, \varepsilon, \phi,$  and  $\gamma$ . The model assumes steady-state (i.e.: after the system has balanced itself) and the resulting differential delays within each slave node are calculated

$$\begin{aligned} \frac{(\xi \text{ to } \alpha \text{ clockwise})}{1} &= \frac{(\xi \text{ to } \alpha \text{ counterclockwise})}{7} : \\ &- 7\Delta X + 7Xo + 7L1 + 7Ai + 7\Delta A \\ &= Xi + \Delta X + L8 + Go + Gi + L7 + Fo + Fi + L6 \\ &+ Eo + Ei + L5 + Do + Di + L4 + Co + Ci + L3 \\ &+ Bo + Bi + L2 - \Delta A + Ao \end{aligned} \quad (1a)$$

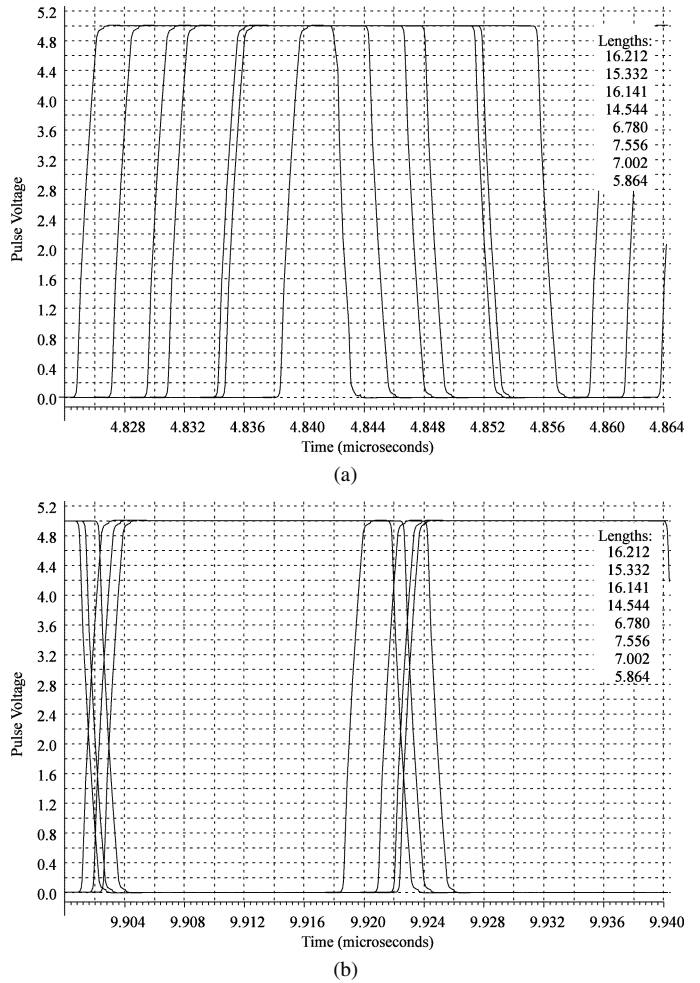


Fig. 9. (a) HSpice simulation of the clock outputs of the seven slave-nodes BEFORE pulse alignment given a very disruptive link interconnect scenario. (b) HSpice simulation of the clock outputs of the seven slave-nodes AFTER a FAILED pulse alignment due to the very disruptive interconnect link scenario (slave-nodes failed to balance).

$$\begin{aligned} \frac{(\xi \text{ to } \beta \text{ clockwise})}{2} &= \frac{(\xi \text{ to } \beta \text{ counterclockwise})}{6} : \\ &- 6\Delta X + 6Xo + 6L1 + 6Ai + 6Ao + 6L2 + 6\Delta B + 6Bi \\ &= 2Xi + 2\Delta X + 2L8 + 2Go + 2Gi + 2L7 + 2Fo \\ &+ 2Fi + 2L6 + 2Eo + 2Ei + 2L5 + 2Do + 2Di \\ &+ 2L4 + 2Co + 2Ci + 2L3 - 2\Delta B + 2Bo \end{aligned} \quad (1b)$$

$$\begin{aligned} \frac{(\xi \text{ to } \chi \text{ clockwise})}{3} &= \frac{(\xi \text{ to } \chi \text{ counterclockwise})}{5} : \\ &- 5\Delta X + 5Xo + 5L1 + 5Ai + 5Ao + 5L2 + 5Bi \\ &+ 5Bo + 5L3 + 5\Delta C + 5Ci \\ &= 3Xi + 3\Delta X + 3L8 + 3Go + 3Gi + 3L7 + 3Fo \\ &+ 3Fi + 3L6 + 3Eo + 3Ei + 3L5 + 3Do + 3Di \\ &+ 3L4 - 3\Delta C + 3Co \end{aligned} \quad (1c)$$

$$\begin{aligned} \frac{(\xi \text{ to } \delta \text{ clockwise})}{4} &= \frac{(\xi \text{ to } \delta \text{ counterclockwise})}{4} : \\ &- 4\Delta X + 4Xo + 4L1 + 4Ai + 4Ao + 4L2 + 4Bi \\ &+ 4Bo + 4L3 + 4Ci + 4Co + 4L4 + 4\Delta D + 4Di \\ &= 4Xi + 4\Delta X + 4L8 + 4Go + 4Gi + 4L7 + 4Fo \\ &+ 4Fi + 4L6 + 4Eo + 4Ei + 4L5 - 4\Delta D + 4Do \end{aligned} \quad (1d)$$

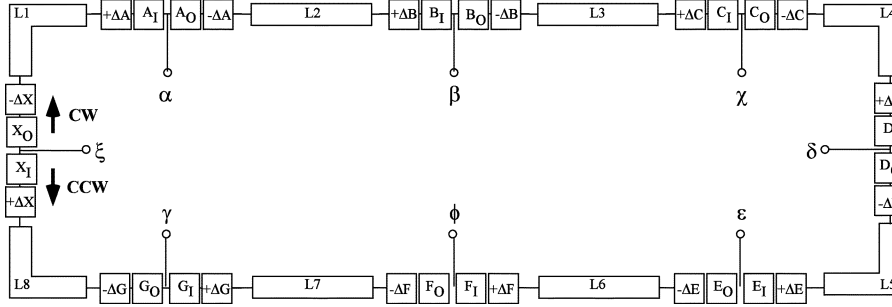


Fig. 10. Model used to generate steady-state analytical solutions for any set of interconnect link scenario.

$$\begin{aligned} \frac{(\xi \text{ to } \varepsilon \text{ clockwise})}{5} &= \frac{(\xi \text{ to } \varepsilon \text{ counterclockwise})}{3} : \\ -3\Delta X + 3X_o + 3L1 + 3A_i + 3A_o + 3L2 + 3B_i + 3B_o + 3L3 \\ + 3C_i + 3C_o + 3L4 + 3D_i + 3D_o + 3L5 + 3\Delta E + 3E_i \\ &= 5X_i + 5\Delta X + 5L8 + 5G_o + 5G_i + 5L7 + 5F_o \\ &\quad + 5F_i + 5L6 - 5\Delta E + 5E_o \end{aligned} \quad (1e)$$

$$\begin{aligned} \frac{(\xi \text{ to } \phi \text{ clockwise})}{6} &= \frac{(\xi \text{ to } \phi \text{ counterclockwise})}{2} : \\ -2\Delta X + 2X_o + 2L1 + 2A_i + 2A_o + 2L2 + 2B_i \\ + 2B_o + 2L3 + 2C_i + 2C_o + 2L4 + 2D_i + 2D_o \\ + 2L5 + 2E_o + 2E_i + 2L6 + 2\Delta F + 2F_i \\ &= 6X_i + 6\Delta X + 6L8 + 6G_o + 6G_i + 6L7 \\ &\quad + 6\Delta F + 6F_o \end{aligned} \quad (1f)$$

$$\begin{aligned} \frac{(\xi \text{ to } \gamma \text{ clockwise})}{7} &= \frac{(\xi \text{ to } \gamma \text{ counterclockwise})}{1} : \\ -\Delta X + X_o + L1 + A_i + A_o + L2 + B_i \\ + B_o + L3 + C_i + C_o + L4 + D_i + D_o \\ + L5 + E_o + E_i + L6 + F_o + F_i + L7 - \Delta G + G_i \\ &= 7X_i + 7\Delta X + 7L8 + 7\Delta G + 7G_o \end{aligned} \quad (1g)$$

$$\begin{aligned} [\text{Variable Delay Matrix}] * \text{Variable Delay Vector} \\ &= [\text{Link Node Matrix}] * \text{Link Node Vector} \\ &\quad + [\text{Constant Delay}] * \Delta X \end{aligned} \quad (2)$$

$$\text{Variable Delay Matrix} = \begin{bmatrix} 8, & 0, & 0, & 0, & 0, & 0, & 0; \\ 0, & 8, & 0, & 0, & 0, & 0, & 0; \\ 0, & 0, & 8, & 0, & 0, & 0, & 0; \\ 0, & 0, & 0, & 8, & 0, & 0, & 0; \\ 0, & 0, & 0, & 0, & 8, & 0, & 0; \\ 0, & 0, & 0, & 0, & 0, & 8, & 0; \\ 0, & 0, & 0, & 0, & 0, & 0, & 8 \end{bmatrix} \quad (3a)$$

[see (3b) and (3c) shown at the bottom of the page], and

$$\text{Variable Delay Vector} = [\Delta A, \Delta B, \Delta C, \Delta D, \Delta E, \Delta F, \Delta G] \quad (3d)$$

$$\text{Constant Delay} = [8, 8, 8, 8, 8, 8, 8]. \quad (3e)$$

$$\text{Link Node Matrix} = \begin{bmatrix} -7, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & -7, & 1, & 1, \\ -6, & -6, & 2, & 2, & 2, & 2, & 2, & 2, & -6, & -6, & 2, & -6, \\ -5, & -5, & -5, & 3, & 3, & 3, & 3, & 3, & -5, & -5, & -5, & -5, \\ -4, & -4, & -4, & -4, & 4, & 4, & 4, & 4, & -4, & -4, & -4, & -4, \\ -3, & -3, & -3, & -3, & -3, & 5, & 5, & 5, & -3, & -3, & -3, & -3, \\ -2, & -2, & -2, & -2, & -2, & -2, & 6, & 6, & -2, & -2, & -2, & -2, \\ -1, & -1, & -1, & -1, & -1, & -1, & -1, & 7, & -1, & -1, & -1, & -1, \\ 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & -7, & 1; \\ 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & -6, & 2; \\ 3, & -5, & 3, & 3, & 3, & 3, & 3, & 3, & 3, & 3, & -5, & 3; \\ -4, & -4, & 4, & -4, & 4, & 4, & 4, & 4, & 4, & 4, & -4, & 4; \\ -3, & -3, & -3, & -3, & 3, & -3, & 5, & 5, & 5, & 5, & -3, & 5; \\ -2, & -2, & -2, & -2, & -2, & -2, & 6, & -2, & 6, & 6, & -2, & 6; \\ -1, & -1, & -1, & -1, & -1, & -1, & -1, & -1, & 7, & -1, & -1, & 7 \end{bmatrix} \quad (3b)$$

$$\begin{aligned} \text{Link Node Vector} &= [L1, L2, L3, L4, L5, L6, L7, L8, A_o, A_i, B_o, B_i, \\ &\quad C_o, C_i, D_o, D_i, E_o, E_i, F_o, F_i, G_o, G_i, X_o, X_i] \end{aligned} \quad (3c)$$



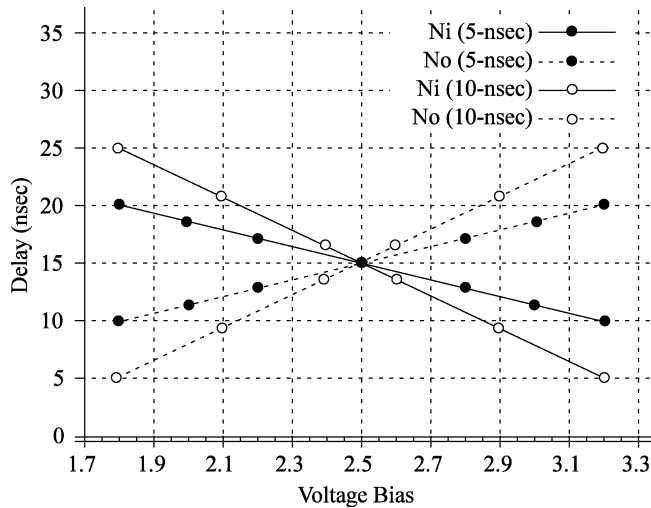


Fig. 11. Linearized variable delay line version of Fig. 6—used in the Monte Carlo simulations to determine “failures” due to a theoretical out-of-range voltage bias in a slave-node.

Thousands of groups of eight interconnect link length delays were randomly generated and corresponding sets of eight variable internal slave node delays were obtained by reducing the matrix of (3). For each percentage tolerance level of interconnect link length, the group of eight randomly generated interconnect link length delays would be within the interval (nominal\_link\_length  $\pm$  [(percent\_tolerance/100) \* nominal\_link\_length]). In general, the Monte Carlo simulations were conservative estimates of the viability of the system because the linear nature of the model was unable to incorporate feedback mechanisms of PLLs. These simulations at least provide a lower bound on the possible successful balancing of a system for a given set of interconnect link length scenarios.

To evaluate a particular set of interconnect link lengths (and corresponding variable internal slave node delays) a linearized version of the variable delay versus bias voltage originally shown in Fig. 6 was used to conclude a pass/fail result. If the voltage required for a desired variable internal slave node delay was out of range (either too high or too low), then that particular simulation failed. The straight line relationship between bias voltage and differential delay was given by Fig. 11. Two sets of variable delay versus voltage bias approximations were used, one limited to  $\pm 5$  ns for a bias range between 1.8 and 3.2 V and the other limited to  $\pm 10$  ns for the same bias voltage range. Given these linear approximations, the total delay, as a function of bias voltage, remained constant at a delay of 30 ns for both cases.

Table III(a) shows five groups of eight randomly generated interconnect link delays (three of which were simulated in HSpice and presented in the section above). The nominal internal slave node delays have a total value of 30 (i.e.:  $A_i = A_o = B_i = B_o = \dots = X_i = X_o = 15$ ). Table III(b) shows the resulting variable internal slave node delays assuming that each node was linearly approximated using the  $\pm 5$  ns lines of Fig. 11. Table III(c) shows the resulting bias voltages within each slave node, the bold numbers indicate that the slave-node has violated the voltage range between 3.2-V and 1.8-V. Case 5 of Table III(c) shows that *ALL* the bias voltages are in violation of the

TABLE III (a)  
FIVE CASES OF EIGHT RANDOM INTERCONNECT LINK DELAYS. (b)  
CORRESPONDING CALCULATED VALUES FOR DIFFERENTIAL  
DELAYS WITHIN EACH SLAVE NODE FOR EACH OF FIVE  
CASES OF EIGHT INTERCONNECT LINK DELAYS. (c)  
CORRESPONDING CALCULATED VALUES FOR VOLTAGE BIASES GIVEN VALUES  
OF DIFFERENTIAL DELAYS WITHIN EACH SLAVE NODE FOR EACH OF THE FIVE  
CASES OF EIGHT INTERCONNECT LINK DELAYS

Interconnect Delays (time units) Case					
Link	1	2	3	4	5
L1	10.212	10.875	22.905	8.345	16.212
L2	10.332	17.766	10.154	8.639	15.332
L3	11.141	10.906	11.356	9.786	16.141
L4	10.544	11.007	20.349	18.030	14.544
L5	9.780	15.280	8.480	8.393	6.780
L6	9.556	24.549	15.898	8.707	7.556
L7	11.002	11.795	9.849	12.184	7.002
L8	10.864	10.756	15.669	11.554	5.864

(a)

Differential Node Delays “ΔN” and Period “T” (time units) Case					
Node	1	2	3	4	5
ΔA	0.2169	3.2417	-8.5725	2.3597	-5.0331
ΔB	0.3137	-0.4075	-4.3940	4.4255	-9.1863
ΔC	-0.3984	2.8033	-1.4175	5.3442	-14.1484
ΔD	-0.5135	5.9130	-7.4340	-1.9810	-17.5135
ΔE	0.1354	4.7497	-1.5815	0.3307	-13.1146
ΔF	1.0082	-5.6825	-3.1470	2.3285	-9.4918
ΔG	0.4351	-3.3608	1.3365	0.8492	-5.3149
ΔX	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
T	40.4289	44.1167	44.3325	40.7047	41.1789

(b)

Differential Node Voltages (volts) Case					
Node	1	2	3	4	5
Va	2.5304	2.9538	<b>1.2998</b>	2.8304	<b>1.7954</b>
Vb	2.5439	2.4429	1.8848	3.1196	<b>1.2139</b>
Vc	2.4442	2.8925	2.3016	<b>3.2482</b>	<b>0.5192</b>
Vd	2.4281	<b>3.3278</b>	<b>1.4592</b>	2.2227	<b>0.0481</b>
Ve	2.5190	3.1650	2.2786	2.5463	<b>0.6640</b>
Vf	2.6412	<b>1.7044</b>	2.0594	2.8260	<b>1.1712</b>
Vg	2.5609	2.0295	2.6871	2.6189	<b>1.7559</b>
Vx	2.5000	2.5000	2.5000	2.5000	2.5000

**Bold** - indicated over/under voltage

(c)

voltage range and the corresponding HSpice simulation above confirms that this case does not have sufficient locking capability. Note that in each case the period T is different; this is because the total delay around the system is different in each case due to the different interconnect link delays for each scenario.

The Monte Carlo analysis consisted of 101 groups of 10 000 uniformly distributed random sets of eight interconnect link lengths where a nominal link length delay of 10 ns was selected. The tolerance level increased by  $\pm 1\%$  for each group of 10 000 from 0% to 100%. The percentage of total number of failures per 10 000 iterations was calculated for each tolerance level between and plotted in Fig. 12(a) for voltage biases ranges corresponding to maximum differential delays of  $\pm 5$  ns and  $\pm 10$  nss.

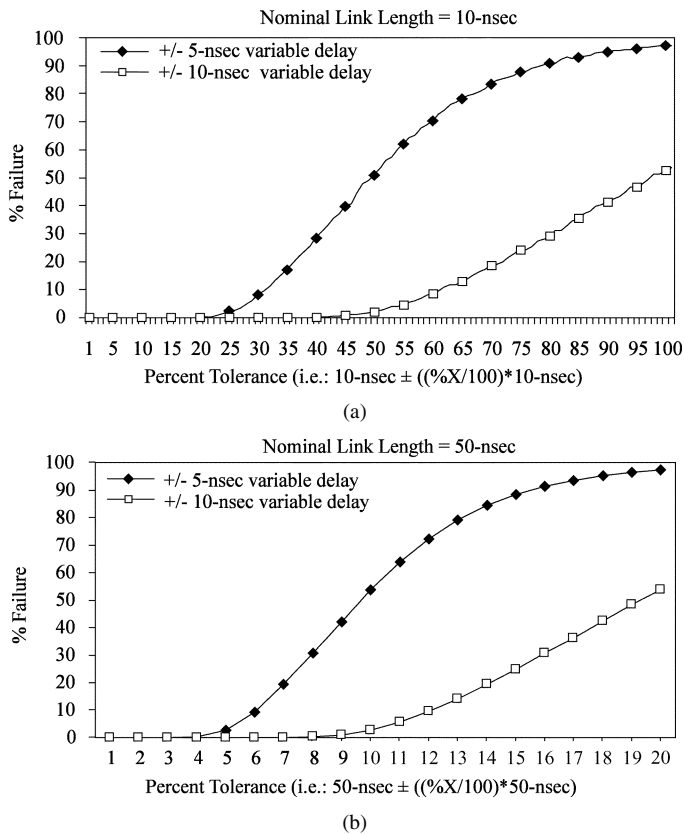


Fig. 12. (a) Monte Carlo simulation to determine theoretical sensitivity of the system to maximum tolerance variations in short (10 ns) nominal interconnect links. (b) Monte Carlo simulation to determine theoretical sensitivity of the system to maximum tolerance variations in long (50 ns) nominal interconnect links.

A second Monte Carlo analysis consisted of 21 groups of 100 000 uniformly distributed random sets of eight interconnect link lengths where a nominal link length delay of 50 ns was selected. The tolerance level increased by  $\pm 1\%$  for each group of 100 000 from 0% to 20%. The percentage of total number of failures per 100 000 iterations was calculated for each tolerance level between and plotted in Fig. 12(b) for voltage biases ranges corresponding to maximum differential delays of  $\pm 5$  ns and  $\pm 10$  ns.

Fig. 12(a) indicates that a nominal interconnect link length of 10 ns could vary up to 20% (between 8 and 12 ns) for  $\pm 5$  ns differential delays and up to 40% (between 6 ns and 14 ns) for  $\pm 10$  ns differential delays and still successfully accommodate any required delay changes. Fig. 12(b) indicates that a nominal interconnect link length of 50 ns could vary up to 4% (between 48 ns and 52 ns) for  $\pm 5$  ns differential delay and up to 8% (between 46 ns and 54 ns) for  $\pm 10$  ns differential delays and still successfully accommodate any required delay changes. Fig. 12(a) and (b) together translates into an absolute value of  $\pm 2$ -ns variation permitted in the interconnect link length tolerance (for  $\pm 5$ -ns differential delay) and  $\pm 4$ -ns variation permitted in the interconnect link length tolerance (for the  $\pm 10$ -ns differential delay).

It is important to note that several examples of interconnect link length scenarios that failed in MATLAB actually passed in HSpice. It is estimated that the link tolerance could increase an additional 5%–10% more and still reliably work in a real

applications, this would be equivalent to sliding the curves in Fig. 12(a) and (b) 5%–10% more to the right along the x-axis.

This analysis of the DSC system demonstrates that there is a useful operating range that can be accommodated using a relatively small differential delay within each slave node. With a range of only  $\pm 5$  ns in each slave node, up to  $\pm 2$ -ns delay variation in the interconnect link length is possible. For example, 10-m of co-axial cable (given a phase-velocity of approximately  $2 \times 10^8$  m/s) has a delay of 50 ns. Given a  $\pm 2$  ns tolerable variation, each co-axial cable link length could vary between 9.6-m and 10.4-m (or  $\pm 40$ -cm).

Further analysis of the Monte Carlo simulations showed trends in the failures of certain scenarios of interconnect link lengths. A failure was more likely to happen due to the distribution of the random link lengths. For example, large magnitude interconnect link length changes that were well distributed, such as:  $L_1 = 22.905$  ns,  $L_2 = 10.154$  ns,  $L_3 = 11.356$  ns,  $L_4 = 20.349$  ns,  $L_5 = 8.480$  ns,  $L_6 = 15.898$  ns,  $L_7 = 9.849$  ns, and  $L_8 = 15.669$  ns, were less likely to violate the slave node bias voltage range. Lower magnitude variations, but grouped, interconnect link lengths such as:  $L_1 = 16.212$  ns,  $L_2 = 15.332$  ns,  $L_3 = 16.141$  ns,  $L_4 = 14.544$  ns,  $L_5 = 6.780$  ns,  $L_6 = 7.556$  ns,  $L_7 = 7.002$  ns, and  $L_8 = 5.864$  ns were more likely to violate the voltage ranges.

## VI. CONCLUSION

The main objective of this paper was to demonstrate a novel method to synchronizing multiple distance nodes. A “proof-of-concept” circuit simulation was demonstrated using a comprehensive transistor-level model as well as an analytical approach done in MATLAB.

The analytical simulations indicated that small differential delay adjustment in each slave node could balance a system where the interconnect link lengths varied up to  $\pm 2$  ns between the nodes. This can allow up to a  $\pm 40$ -cm variation in the interconnect’s length.

Although preliminary analysis of this type of system shows promise, there are 3 factors that can adversely affect the performance of the system. These factors are: 1) nonlinearities in the variable delay lines; 2) random circuit variations within the master and slave nodes due of power, temperature, voltage and loading (PTVL) effects; and 3) mismatches in the interconnect link lengths between pairs of nodes. Although the nonlinearities of the circuits are controllable and can be overcome by more intricate designs, more analysis and design is required. The random PTVL variations are more difficult to design for, but the nature of PLL feedback within the master and slave nodes should help regulate these variations. Finally the mismatching between interconnect link lengths between nodes can be reduced or eliminated given techniques suggested in the paper to combine the three signals into one physical medium.

There are still many issues that would need to be addressed, the most significant is the circuit’s response to random noise inducing jitter as well as the lock-in time of the PLLs. Other refinements to the system could be applied such as transitioning to a smaller line-width technology and the use of differential signaling such as CML. The three independent signal lines could also be reduced to just one common transmission medium given

bi-directional circuit such as that proposed in [9]. The system might even be implemented using fundamental carrier frequencies where frequency filters are used and all the PLLs could be implemented using fully analog designs.

## REFERENCES

- [1] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. D. Madland, "A multigigahertz clocking scheme for the pentium 4 microprocessor," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1647–1653, Nov. 2001.
- [2] J. Wood, T. C. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1654–1665, Nov. 2001.
- [3] L. Hall, M. Clements, L. Wentai, and G. Bilbro, "Clock distribution using cooperative ring oscillators," in *Dig.Tech. Papers Symp. VLSI Technology*, 1997, pp. 62–75.
- [4] G. Hinton, M. Upton, D. J. Sager, D. Boggs, D. M. Carmean, P. Roussel, T. I. Chappell, T. D. Fletcher, M. S. Milshtein, M. Sprague, S. Samaan, and R. Murray, "A 0.18- $\mu\text{m}$  CMOS IA-32 processor with a 4-GHz integer execution unit," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1617–1627, Nov. 2001.
- [5] R. A. Omondi, *The Microarchitecture of Pipelined and Superscalar Computers*. Boston, MA: Kluwe, 1999.
- [6] W. D. Grover, "A new method for clock distribution," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 41, no. 2, pp. 149–160, Feb. 1994.
- [7] S. Rusu and G. Singer, "The first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1539–1544, Nov. 2000.
- [8] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. Autom. Contr.*, vol. 47, no. 7, pp. 1091–1107, Jul. 2002.
- [9] K. Ishibashi, T. Goto, T. Hayashi, T. Okada, A. Yamagiwa, M. Shibata, K. Akimoto, N. Hamanaka, T. Takahashi, A. Koyama, and T. Aida, "Simultaneous bidirectional transceiver logic," *IEEE Micro*, vol. 19, no. 1, pp. 14–19, Jan.–Feb. 1999.
- [10] B. Razavi, Ed., *Monolithic Phase-Locked Loops and Phase Recovery Circuits*. New York: IEEE, 1996.
- [11] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications 4th Ed.*. New York: McGraw-Hill, 1999.
- [12] D. M. Gross, "Theory and analysis of a distributed synchronous clocking method," M. Eng. thesis, Dept. of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada, 2004.
- [13] D. R. Rolston, "The design, layout, and characterization of vlsi optoelectronic chips for free-space optical interconnects," Ph.D. dissertations, Dept. of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada, 2000.
- [14] D. R. C. Rolston, D. V. Plant, and G. W. Roberts, "Method and Apparatus for Distributed Synchronous Clocking," U. S. Patent Application #20020031199, Mar. 14, 2002.
- [15] I. A. Young, J. K. Greason, and K. L. Wong, "A PLL clock generator with 5 to 100 MHz of lock range for microprocessors," *IEEE J. Solid-State Circuits*, vol. 27, no. 11, pp. 1599–1606, Nov. 1992.
- [16] G. Kim, M.-K. Kim, B.-S. Chang, and W. Kim, "A low-voltage, low-power CMOS delay element," *IEEE J. Solid-State Circuits*, vol. 31, no. 7, pp. 966–971, Jul. 1996.



**David R. Rolston** (S'90–M'00) received the B.Eng., M.Eng., and Ph.D. degrees in electrical and computer engineering from McGill University, Montreal, QC, Canada, in 1993, 1996, and 2000, respectively.

He worked at FCI-Areva, for two years as a Senior Photonics Scientist where he helped build one of three global R&D research centers in the field of optics and photonics. He is currently Co-founder and Chief Technology Officer of Reflex Photonics Inc., Montreal, QC, Canada. His current activities include the design and testing of optical and optomechanical

design assemblies for optical transceivers and hybrid optical integrated circuit packaging concepts for chip-to-chip optical interconnections. He has published numerous journals and holds several US and international patents and patents pending.



**David M. Gross** received the B.Eng. and M.Eng. degrees in electrical engineering from McGill University, Montreal, QC, Canada, in 2000 and 2005, respectively.

During the B. Eng. course, he interned as an ASIC Designer for Nortel Networks, Ottawa, ON, Canada, working on Nortel's first high-speed DSL modem. Between degrees he returned to Nortel Networks, Ottawa, ON, Canada, as an ASIC Designer where he had been part of 10G-Ethernet chip set development for control and interface units, as well as IEEE standards body liaison. He currently works as an FPGA Designer and Board Clocking Specialist working on issues related to synchronization, signal integrity, power consumption, and reliability for Teradyne Inc., Boston, MA.



**Gordon W. Roberts** (F'04) received the B.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1983 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1986 and 1989, respectively, all in electrical engineering.

He is currently on leave from McGill University, Montreal, QC, Canada, as a Co-founder of DFT Microsystems Inc., where he holds the position of Chief Technical Officer at DFT Microsystems, Montreal, QC, Canada, Canada, Inc. At McGill University, he is a Full Professor and holds the James McGill Chair in Electrical and Computer Engineering. He has co-written five textbooks related to analog integrated circuit design and mixed-signal test. He has published numerous papers in scientific journals and conferences, and he has contributed chapters to various industrially focused textbooks.

Dr. Roberts has held many administration roles within conference organizations; most recently he was the 2003 Program Chair of the IEEE International Test Conference. He has received numerous department, faculty, and university awards for teaching test and electronics to undergraduates, and received several IEEE awards for his work on mixed-signal testing.

**David V. Plant** (S'86–M'89–SM'04) received the Ph.D. degree in electrical engineering from Brown University, Providence, RI, in 1989.

From 1989 to 1993, he was a Research Engineer in the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles (UCLA). In 1993, he joined the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada, as an Assistant Professor, was promoted to Associate Professor 1997 and to Professor in 2004. During the 2000–2001 academic year, he went on leave from McGill to become the Director of Optical Integration at Accellight Networks, Pittsburgh, PA. He is the Principle Investigator (PI) and Scientific Director of the NSERC funded Agile All-Photonic Networks (AAPN) research program, and the PI and Center Director of the Quebec-funded Center for Advanced Systems and Technologies in Communications.

Dr. Plant has received numerous awards, most recently the Carrie M. Derick Award for Graduate Student Supervision, Teaching, and Research in 2004 and was named a James McGill Professor in 2001.