

A 155-MHz Clock Recovery Delay- and Phase-Locked Loop

Thomas H. Lee, *Member, IEEE*, and John F. Bulzacchelli, *Student Member, IEEE*

Abstract—This paper describes a completely monolithic delay-locked loop (DLL) that may be used either by itself as a deskewing element, or in conjunction with an external voltage-controlled crystal oscillator (VCXO) to form a delay- and phase-locked loop (D/PLL). By phase shifting the input data rather than the clock, the DLL and D/PLL provide jitter-peaking-free clock recovery. Additionally, the jitter transfer function of the D/PLL has a low bandwidth for good jitter filtering without compromising acquisition speed. The D/PLL described here exhibits less than 1° rms jitter on the recovered clock, independent of the input data density. No jitter peaking is observed over the 40-kHz jitter bandwidth.

I. INTRODUCTION

THIS paper describes the performance of a completely monolithic delay-locked loop (DLL) that may be used either by itself as a deskewing element, or in conjunction with an external voltage-controlled crystal oscillator (VCXO) to form a delay- and phase-locked loop (D/PLL [1]) that enables jitter-peaking-free clock recovery at the SONET OC-3 frequency of 155.52 MHz. In addition, it will be shown that the D/PLL also may possess a low jitter bandwidth to provide jitter filtering without compromising acquisition performance.

Section II provides a general background on the problem of clock recovery, while Section III examines the properties of DLL's and D/PLL's. Section IV presents specific implementation details for the DLL, Section V presents experimental results, and Section VI concludes with a summary.

II. BACKGROUND

The ability to regenerate binary data is an inherent advantage of digital transmission of information. To perform this regeneration with the fewest bit errors, the received data must be sampled at the optimum instants of time. Since it is generally impractical to transmit the requisite sampling clock signal separately from the data, the timing information is usually derived from the incoming data itself. The extraction of this implicit signal is called

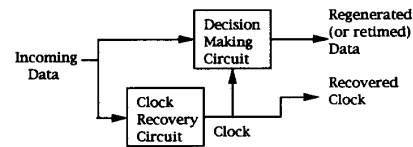


Fig. 1. Simplified block diagram of a digital receiver.

clock recovery, and its general role in digital receivers is illustrated in Fig. 1.

When the incoming data signal has spectral energy at the clock frequency, a synchronous clock can be obtained simply by passing the incoming data through a bandpass filter, often realized either as an LC tank or surface-acoustic wave (SAW) device, tuned to the nominal clock frequency. Because of bandwidth restrictions, however, in most signaling formats the incoming data signal has no spectral energy at the clock frequency, complicating clock recovery. Such signals must first undergo appropriate nonlinear preprocessing ahead of the resonator.

While these clock recovery circuits generally offer rapid acquisition (typically in the hundreds of clock cycles), they operate essentially at a fixed clock frequency, and provide no output in the absence of an input. Furthermore, the phase of the resonator's output may not necessarily be correct to provide optimally timed sampling for the decision circuit and thus generally requires compensation. Proper phase alignment of data and clock must be provided on an open-loop basis and is therefore difficult to maintain over temperature, supply, and process variations. Finally, neither LC - nor SAW-based clock recovery circuits are practically realizable in monolithic form.

Another classical approach employs phase-locked loops (PLL's), as illustrated in Fig. 2. Because of its amenability to monolithic construction, a PLL is an attractive alternative to tuned circuit clock recovery. Furthermore, conventional PLL's offer a comparatively wide tuning range, and provide an output of some kind at all times, in contrast with resonant approaches. However, because the desired loop bandwidths are often much smaller than the tuning range, acquisition is generally comparatively slow, even with frequency-acquisition aids.

To appreciate a more subtle difficulty with conventional PLL realizations, consider the linearized block diagram of a second-order PLL shown in Fig. 3, and the associated root locus diagram of Fig. 4. The input-output phase

Manuscript received April 27, 1992; revised July 15, 1992.

T. H. Lee was with Analog Devices, Wilmington, MA 01887. He is now with Rambus Inc., Mountain View, CA 94040.

J. F. Bulzacchelli is with the Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9203623.

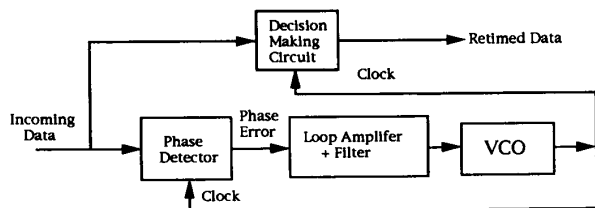


Fig. 2. PLL-based clock recovery system.

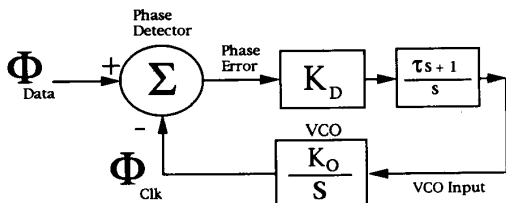


Fig. 3. Linearized block diagram of second-order PLL.

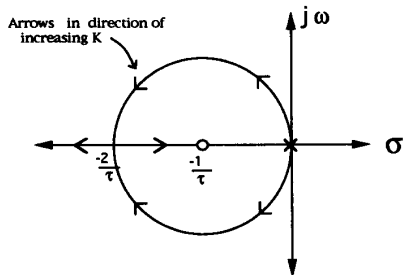


Fig. 4. Root locus for second-order PLL.

transfer function (also known as the jitter transfer function) for this system is as follows:

$$H(s) = \frac{\phi_{clk}(s)}{\phi_{data}(s)} = \frac{K(1 + \tau s)}{s^2 + K\tau s + K}$$

where $K = K_D K_O$.

As seen in the Bode diagram of Fig. 5, the magnitude of the jitter transfer function of such a system exceeds unity over some range of frequencies because of the presence of the closed-loop zero at a frequency lower than that of the poles. This gain in excess of unity is known as jitter peaking and is particularly objectionable in systems that employ many clock recovery units in cascade (such as in repeaters) because of the resulting exponential growth of jitter [3].

The traditional solution is to increase the damping ratio to reduce the spacing between the zero and the lowest frequency pole. As can be inferred from the root locus, however, the amount of jitter peaking can be reduced but never completely eliminated since the zero always occurs at a lower frequency than that of the poles. Additionally, frequency acquisition speed often suffers as damping increases in PLL's that employ acquisition aids, such as that described in [2]. Hence, simply increasing the damping ratio is frequently unsatisfactory.

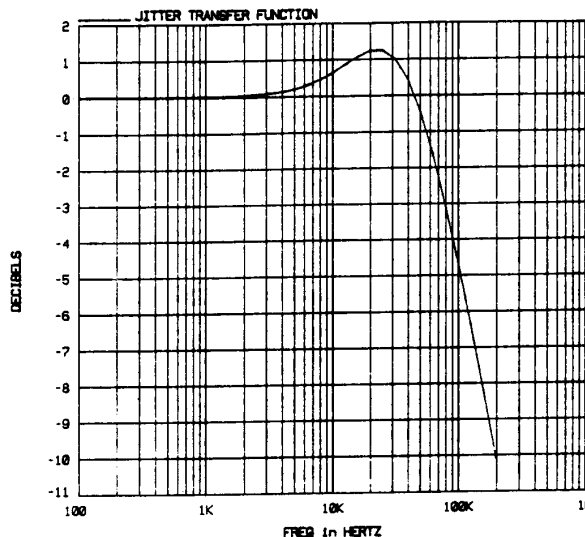


Fig. 5. Jitter transfer function of critically damped second-order PLL.

Note that the problem of jitter peaking disappears if a way can be found to move the necessary loop-stabilizing zero to a frequency higher than that of the lowest frequency pole. One possibility is to employ a loop of different order. As shown in the root locus of Fig. 6, a third-order loop, for example, permits placement of the zero as desired. Unfortunately, the conditional stability of such loops makes them difficult to use.

Another option is to retain a second-order loop, but to move the necessary loop-stabilizing zero out of the forward path so that there is then no closed-loop zero. The next section explores methods for accomplishing this goal.

III. THE DELAY- AND PHASE-LOCKED LOOPS

Traditional PLL's adjust the phase of the clock to obtain phase alignment with the incoming data. However, significant advantages accrue if one instead shifts the input data to align with the clock, as in the two clock recovery schemes shown in Fig. 7.

A. The Delay-Locked Loop (DLL)

If a clock signal of precisely the correct frequency is available, the DLL connection shown in solid lines may be used. As seen, the DLL employs a voltage-controlled phase shifter (VCPS) driven under loop control to align data with the incoming clock. The loop is first-order, as shown in the linearized block diagram of Fig. 8, with the VCPS characterized by a gain constant K_ϕ , with units of radians/volt. A single loop integrator suffices to drive the steady-state phase error to zero.

Note that the jitter transfer function of the DLL simply equals zero; the DLL cannot transfer jitter to the clock since the same (external) clock both generates and retimes the data. Although this fact does not imply that the DLL's decision circuit then functions with improved bit-error rate

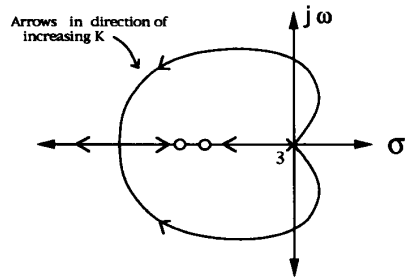


Fig. 6. One possible root locus for a third-order PLL.

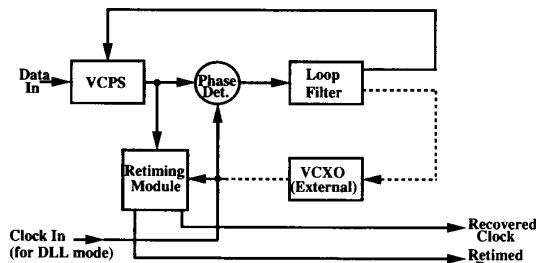


Fig. 7. Block diagram of DLL and D/PLL.

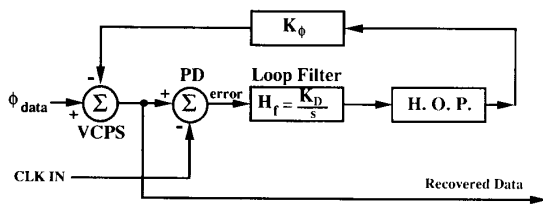


Fig. 8. Linearized block diagram of DLL. (The block labeled H.O.P. represents higher order poles and is discussed in Section IV-A.)

(BER) as a consequence, the BER of *subsequent* clock recovery stages would improve.

Phase errors are nulled out with a speed commensurate with the loop bandwidth $K_D K_\phi$ so that increasing the loop bandwidth makes the DLL acquire more quickly. Furthermore, because only phase acquisition needs to take place, acquisition can be faster than that of PLL's or even of resonant filter approaches. And, unlike the case of a PLL, jitter filtering is independent of the loop bandwidth, so that only acquisition speed and loop stability considerations bound the loop bandwidth.

It should be noted that, in general, any real VCPS will have a finite range. As a consequence, it is possible for the VCPS to be driven to the end of its control range under certain circumstances. Once this occurs, acquisition is prematurely halted, and data can be lost.

This predicament can be avoided by resetting the DLL. Before acquisition begins, the integrator can be initialized to the midpoint of the phase shifter's range. Since the initial phase error must lie between $-\pi$ and π radians ($= \pm 0.5$ unit intervals, or clock periods) from the middle of its range. Hence, in principle, acquisition will always be

successful as long as the total range of the VCPS exceeds 2π radians.

In practice, a range considerably larger than 2π radians is desirable, for jitter on the incoming data can temporarily drive the phase shifter more than $\pm \pi$ radians from the middle of its range. Additional range is also needed if the integrator is not initialized to the exact midpoint of the control range. For most applications, a total range of approximately 3π radians is sufficient.

While this type of DLL exhibits no jitter peaking (because the jitter transfer function is zero), it suffers from the need for an exact external frequency reference. In most cases of practical interest, no such reference is available local to the clock recovery module.

B. The Delay- and Phase-Locked Loop (D/PLL)

A loop that does not require an external frequency reference is the combined D/PLL comprising all of the elements shown in Fig. 7. As shown, the D/PLL contains two parallel control loops. The phase detector, loop amplifier, and VCXO form the core of a PLL, while the phase detector, loop amplifier, and VCPS form the core of a DLL, and is effectively summed with the first loop.

The two loops in the D/PLL act in concert to null out phase errors as follows: if the clock is behind the data, the phase detector drives the VCXO to a higher frequency and simultaneously increases the delay through the VCPS. Both of these actions serve to reduce the initial phase error since the faster clock picks up phase, while the delayed data loses phase. Eventually, the initial phase error is reduced to zero.

To investigate the loop dynamics, consider the linearized block diagram of Fig. 9 and the associated loop transmission magnitude plot of Fig. 10. As can be seen, the low-frequency loop transmission is controlled by the second-order PLL path, while the high-frequency loop behavior is controlled by the first-order DLL path.

The advantages of this connection derive from the manner in which loop stabilization is accomplished. Rather than realizing the zero in the loop filter as is customary (e.g., with a resistor in series with an integrating capacitor), the *phase shifter* provides the necessary zero, as is readily verified by considering the loop transmission behavior. The first-order DLL path sums with the second-order PLL path to yield a loop transmission that inflects at the intersection (at $\omega = K_O/K_\phi$) of the two components. Hence the zero location (the inflection point) is set by a ratio of gain constants, in contrast with prior art.

To establish that the arrangement of Fig. 9 also evades jitter peaking, let us derive explicitly the jitter transfer function for the D/PLL.

From the block diagram of Fig. 9, the control voltage v_1 may be expressed as follows:

$$v_1 = \frac{K_D}{s} (\phi_{\text{data}} - K_\phi v_1 - \phi_{\text{clk}}).$$

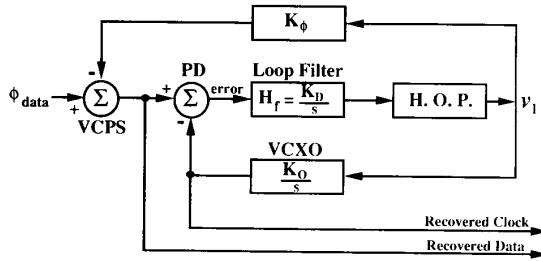


Fig. 9. D/PLL linearized block diagram.

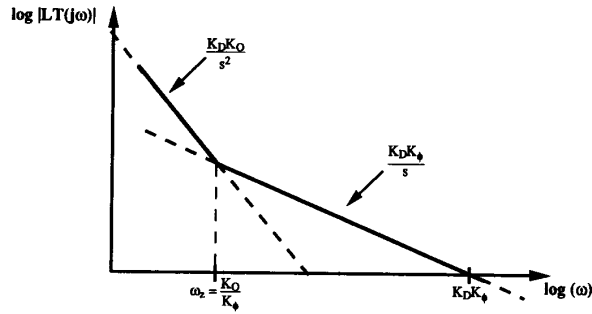


Fig. 10. D/PLL loop transmission.

Now, we may also write:

$$\phi_{\text{clk}} = v_1 \frac{K_O}{s}$$

Eliminating v_1 from these two expressions leads to the desired jitter transfer function:

$$H(s) = \frac{\phi_{\text{clk}}(s)}{\phi_{\text{data}}(s)} = \frac{K}{s^2 + K\tau s + K}$$

where $K = K_D K_O$ and $\tau = K_\phi / K_O$. Note that this jitter transfer function is all-pole, with the same closed-loop poles as a standard second-order PLL (and hence the same stability), but there is no closed-loop zero. Thus, as long as one picks K and τ so that the loop possesses a damping ratio ζ greater than or equal to 0.707, this D/PLL exhibits no jitter peaking. This condition is readily satisfied by locating the zero well below loop crossover.

Elimination of the closed-loop zero also lowers the bandwidth of the jitter transfer function, perhaps dramatically. In a conventional overdamped PLL, the jitter transfer function has a bandwidth nearly equal to the location of the higher frequency pole because the closed-loop zero nearly cancels the low-frequency pole. In this D/PLL, however, it is the lower frequency pole that sets the closed-loop bandwidth because the loop transmission zero does not appear in the closed-loop transfer function. If the D/PLL is overdamped, the low-frequency pole resides near the loop transmission zero, as easily seen from the root locus. Thus, if a PLL and D/PLL possess identical pole locations, the D/PLL can have a much smaller jitter bandwidth. This difference becomes most prominent at high damping ratios, where the two poles are widely

separated in frequency. The implication is that for the same loop crossover frequencies (and hence the same acquisition speed), the D/PLL can provide much more jitter filtering.

C. Acquisition Behavior of the D/PLL

The acquisition behavior of the D/PLL can be studied most conveniently by considering separately the response of the loop to an initial phase error and the response to an initial frequency error. Assuming that the system is linear, the total response can be obtained by superposition. We will later articulate the conditions under which the assumption of linearity holds.

Let us first assume that the frequency of the D/PLL's VCXO equals the bit rate of the incoming data, but that an initial phase error ϕ_0 exists. The D/PLL's response to this initial phase error has two distinct stages. During the first stage, the control voltage is adjusted to a new value with a time constant ($= 1/K_D K_\phi$) of the DLL portion of the loop (recall that the DLL controls the loop behavior at high frequencies). Simultaneously, the phase-shifter delay and the VCXO frequency are adjusted. Since the frequency of the VCXO cannot be pulled very far from the bit rate of the incoming data, the phase of the VCXO does not change very rapidly. Consequently, the VCXO's phase may be considered stationary during the first stage of acquisition. As a result, the phase error is reduced toward zero as fast as the DLL bandwidth $K_D K_\phi$ during this first stage.

During the second stage of acquisition, the control voltage decays slowly (with the time constant of the slow pole) from its new value back to its original value, for at the start of the second stage, the VCXO's frequency does not match the bit rate of the incoming data. (The VCXO's frequency only matches the bit rate of the incoming data when the control voltage is at its original value.) Because of this frequency error, the phase of the VCXO with respect to the incoming data changes (albeit slowly). As the phase of the VCXO changes, the delay of the phase shifter is adjusted so that the phase of the delayed data tracks that of the VCXO. This process continues until the control voltage reaches its original value, at which point the VCXO's frequency equals the bit rate of the data, and acquisition is complete.

It must be stressed that the phase error between the delayed data and the clock may be insignificant during the second stage of acquisition. Examination of the block diagram of Fig. 9 readily shows that the phase error of the D/PLL is proportional to the time derivative of the control voltage. Since the control voltage is changing during the second stage of acquisition, a nonzero phase error can persist for a long time. However, if the control voltage changes slowly, the magnitude of this slow tail can be negligible. Overdamping the loop is advantageous in this context since the time constant of this slow mode is that of the low-frequency pole, which in turn is nearly the lo-

cation of the loop transmission zero. Overdamping moves the zero, and hence the frequency of the slow mode, to lower frequencies and thereby reduces the magnitude of the error tail. Under such conditions, the second stage of acquisition can be ignored, for the data can be successfully regenerated as soon as the first stage of acquisition is completed.

Let us now consider the response of the loop to an initial frequency error of $\Delta\omega$. We may simplify the situation considerably by recalling that the second stage of phase acquisition just studied is essentially one of frequency acquisition. The frequency error at the beginning of the second stage is equal to ϕ_0/τ , while the phase error is negligibly small (if the loop is sufficiently well damped). Hence, by analogy, we conclude that the phase error due to an initial frequency error decays away with the time constant of the low-frequency pole, which is nearly that of the zero. A VCXO with its excellent center frequency accuracy and narrow tuning range is generally required to ensure that this phase error tail is negligible.

Unlike the second-order PLL, the D/PLL realizes rapid acquisition without compromising jitter filtering. While phase errors are nulled out as quickly as the DLL bandwidth $K_D K_\phi$, the jitter transfer function's bandwidth is predominantly controlled by the low-frequency pole at (very nearly) K_O/K_ϕ . Increasing K_D makes the D/PLL acquire more rapidly, but does not diminish the D/PLL's ability to filter jitter.

In practice, the D/PLL's ability to filter jitter is limited by the finite range of the phase shifter, the variability of the incoming data's bit rate, and the instability or uncertainty of the VCXO's center frequency. Together, the variability of the bit rate and the instability of the VCXO's center frequency determine the VCXO's required tuning range, as the tuning range must be large enough to ensure that the VCXO's frequency can always be pulled to the bit rate of the incoming data.

An important consideration is that the entire tuning range of the VCXO is not necessarily usable, for one must guarantee that the phase shifter is never driven out of range. (Recall that the VCXO and the phase shifter share the same control voltage.) As explained earlier, the phase shifter stabilizes the D/PLL by implementing the necessary loop transmission zero. If the phase shifter is ever driven to the end of its range, the loop transmission zero disappears, and the D/PLL becomes unstable. Hence, signal swings must be arranged to guarantee that the VCXO's control range is a *subset* of the phase shifter's control range.

If the VCXO's control range is a subset of that of the phase shifter, it follows that

$$\frac{\Delta\omega}{K_O} \leq \frac{\Delta\phi}{K_\phi}$$

where $\Delta\omega$ equals the VCXO's required tuning range (in radians per second), and $\Delta\phi$ equals the phase shifter's to-

tal range (in radians). Rearranging terms, one obtains

$$\frac{K_O}{K_\phi} \geq \frac{\Delta\omega}{\Delta\phi}.$$

This last equation expresses a lower limit on the bandwidth of the jitter transfer function, as the bandwidth of the jitter transfer function is approximately K_O/K_ϕ .

As a numerical example, consider a VCXO with a tuning range of 200 ppm of the bit rate, and a phase shift range of 3π radians. The lower limit on the bandwidth is then only 21 ppm of the bit rate, which accommodates exceptionally good jitter filtering.

On the other hand, use of a relaxation-type oscillator with its poorer center frequency stability greatly diminishes the D/PLL's ability to filter jitter. Suppose $\Delta\omega$ were 50% of the bit rate, and that the phase shifter range were still 3π radians. In this instance, the lower limit on the jitter bandwidth would be 2500 times larger than in the VCXO example. Because its narrow tuning range improves jitter filtering, a VCXO is highly desirable in a D/PLL.

A final consideration is that the D/PLL may require resetting if rapid acquisition is required because of the finite range of the voltage-controlled phase shifter. Normally, initial phase errors are rapidly nulled out during the first stage of phase acquisition. However, the first stage of acquisition, which is basically the acquisition process of the DLL, can halt prematurely if the VCPS is driven to the end of its range.

As in the DLL, initializing the D/PLL's integrator to the midpoint of the phase shifter's range eliminates this problem, provided that the total range of the phase shifter is sufficient. As before, a total range of approximately 3π radians is sufficient.

If rapid acquisition is unimportant, then resetting is unnecessary. As long as the bit rate of the incoming data lies within the tuning range of the VCXO, the D/PLL eventually acquires. If the phase shifter is driven to the end of its range, the first stage of acquisition terminates prematurely and a significant phase error may remain. Now, if the phase shifter is at the end of its range, then the VCXO must also be as well, since the VCXO's control range is a subset of that of the phase shifter. The resulting frequency error slews the phase error toward zero in an open-loop manner. Once the phase error is zero, closed-loop behavior resumes and the loop ultimately settles to a condition of zero phase and frequency error.

IV. IMPLEMENTATION OF THE DLL

This section describes the realization of a monolithic DLL in a 3-GHz silicon bipolar process, beginning with some background material on phase detectors.

A. Phase Detectors

Before describing the phase detector actually used in this particular DLL, it is useful first to examine briefly the properties of two related phase detectors. The first of

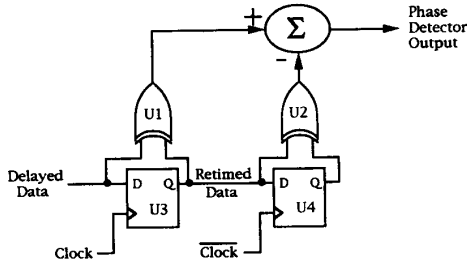


Fig. 11. Hogge phase detector.

these, due to Hogge [4], is shown in Fig. 11. This circuit directly compares the phases of the delayed data and the clock in the following manner. After a change in the state of the delayed data, the *D* input and *Q* output of D-type flip-flop *U3* are no longer equal, causing the output of XOR gate *U1* to go high. The output of *U1* remains high until the next rising edge of the clock, at which time the delayed data's new state is clocked through *U3*, eliminating the inequality between the *D* and *Q* lines of *U3*. At the same time, XOR gate *U2* raises its output high because the *D* and *Q* lines of *U4* are now unequal. The output of *U2* remains high until the next falling edge of the clock, at which time the delayed data's new state is clocked through *U4*.

If we assume that the clock has a 50% duty cycle, *U2*'s output is a positive pulse with a width equal to half the clock period for each data transition. *U1*'s output is also a positive pulse for each data transition, but its width depends on the phase error between the delayed data and the clock; its width equals half a clock period when the delayed data and the clock are optimally aligned. Hence, the phase error can be obtained by comparing the widths of the pulses out of *U1* and *U2*.

Fig. 12 is a timing diagram for this detector with the delayed data and clock optimally aligned (in this case, with the falling edge of the clock). In this case, the output of the phase detector has zero average value, and there is no net change in the loop integrator's output.

If the delayed data are ahead of the clock, the output of the phase detector has a positive average value, as shown in Fig. 13. As a result, the loop integrator's output exhibits a net increase. Conversely, if the delayed data were behind the clock, the phase detector's output would have a negative average value, and the loop integrator's output would exhibit a net decrease.

Plotting the phase detector's average output (assuming maximum data transition density) as a function of phase error yields the sawtooth characteristic shown in Fig. 14. Consistent with Fig. 12, the phase detector's average output equals zero when the phase error between the delayed data and the clock is zero.

While one noteworthy feature of this phase detector is that the decision-making circuit is an integral component of the phase detector (for the output of flip-flop *U3* is the retimed data), this detector does suffer from a sensitivity to the data transition density. Since each triangular pulse

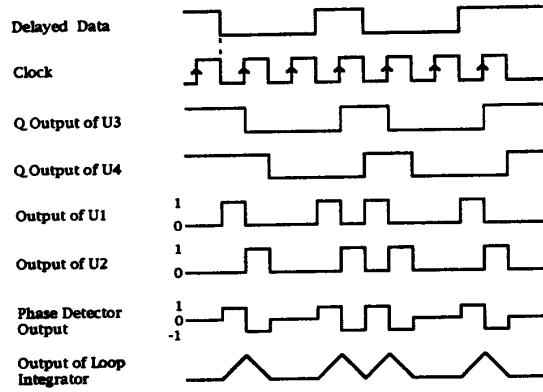


Fig. 12. Waveforms of Hogge's detector with clock and data aligned.

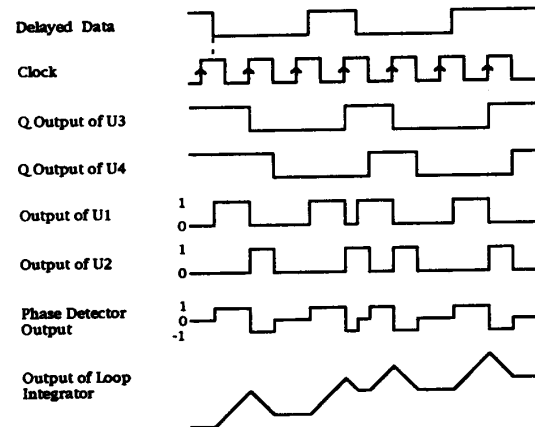


Fig. 13. Waveforms of Hogge's detector with data ahead of clock.

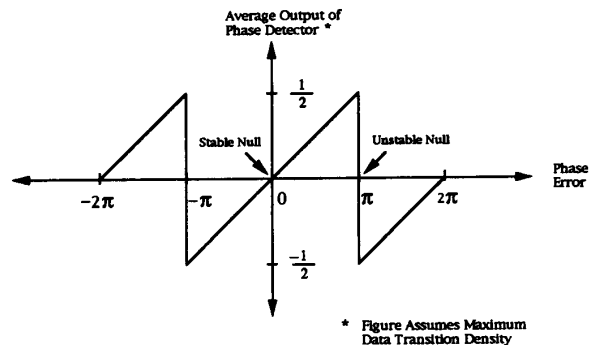


Fig. 14. Transfer characteristic of Hogge's detector.

on the output of the loop integrator has positive net area (see Fig. 12), the presence or absence of such a pulse affects the average output of the loop integrator. The data-dependent jitter thus introduced is often large enough to be objectionable.

The phase detector shown in Fig. 15 greatly reduces this problem by replacing the triangular correction pulses (which have net area, even when the delayed data and clock are properly aligned) with "triwaves" whose net area is zero when clock and data are aligned.

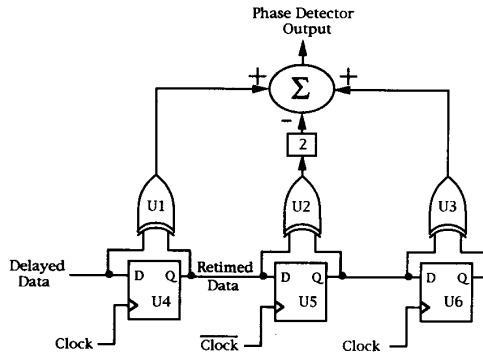


Fig. 15. Triwave phase detector.

As in Hogge's detector, the width of $U1$'s output is dependent on the phase error between the delayed data and the clock, while the outputs of $U2$ and $U3$ are always a half clock cycle wide (assuming that the clock possesses a 50% duty cycle). The phase error can thus be obtained by comparing the variable width pulse from $U1$ with the fixed width pulses from $U2$ and $U3$. Note that the pulses out of $U1$ and $U3$ are weighted by 1, while the pulse out of $U2$ is weighted by -2 .

Fig. 16 is the timing diagram for the triwave detector with the delayed data and the clock optimally aligned. Note that each data transition initiates a three-sectioned transient (the triwave) on the output of the loop integrator, and that this triwave has zero area. Therefore, its presence or absence does not change the average output of the loop integrator. Hence, the triwave detector exhibits a much reduced sensitivity to data transition density.

The triwave detector is somewhat more sensitive to duty cycle distortion in the clock signal, however, than Hogge's implementation, because of the unequal weightings used [5]. This sensitivity to duty cycle can be restored to that of Hogge's implementation with the simple modification shown in Fig. 17. The modified triwave detector uses two distinct down-integration intervals clocked on opposite edges of the clock, rather than a single down-integration of twice the strength clocked on a single clock edge. As a consequence, duty cycle effects are attenuated.

Filtering of the high-frequency ripple present in the output of any of these phase detectors can be provided by the addition of poles placed beyond loop crossover. These "higher order poles" (HOP's), shown in Figs. 8 and 9, can reduce the jitter induced by this ripple to insignificant levels.

The low clock duty cycle sensitivity, small data-dependent jitter, and the integral decision/retiming property of the modified triwave detector are the reasons the present DLL design uses this particular form of phase detector.

In the actual implementation of the phase detector, the necessary summations are performed by steering currents into and out of a capacitor with a differential charge pump,

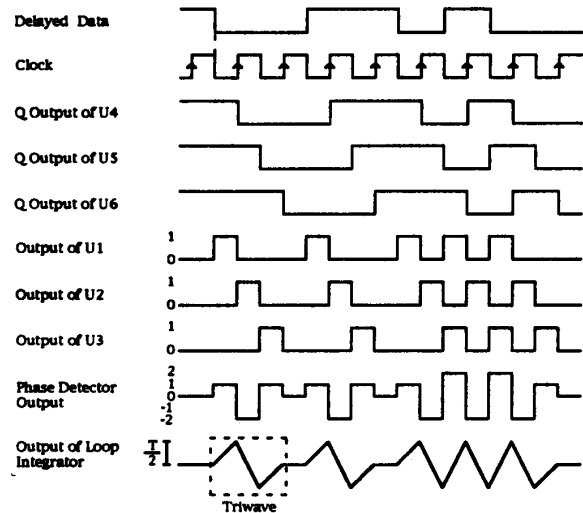


Fig. 16. Waveforms of triwave detector with clock and data aligned.

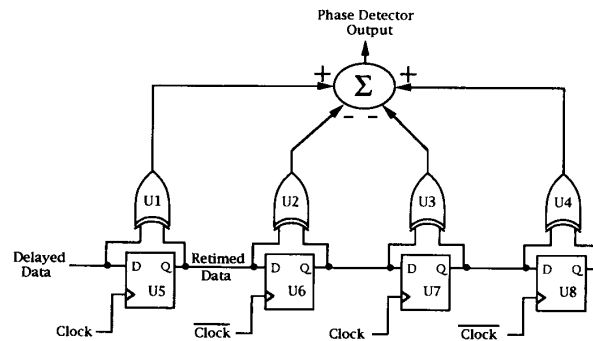


Fig. 17. Modified triwave phase detector.

as shown in Fig. 18. The current switches are connected to the outputs of the XOR gates shown in Fig. 17.

No reset function is provided in this particular implementation. Hence, although acquisition is typically fast (40 clock cycles by design), it is possible for acquisition to terminate prematurely, as described in Section III-C. Acquisition takes considerably longer in such cases.

Because of headroom limitations, resistive loads are used in the charge pump, as seen in Fig. 18. To achieve a sufficiently high dc gain (for low static phase error), a negative resistance is placed in parallel with the resistive loads, as shown in simplified form in Fig. 19. To understand how the negative resistor functions, assume that $Q1$ and $Q2$ act as ideal voltage followers so that $v_e = v_{in}$. Because of the cross-connections of the bases, the current i_{in} is equal to $-v_{in}/2R$, so that $R_{in} = -2R$.

Fig. 20 shows an improved negative resistor that compensates for the nonzero output impedance of the emitter followers by adding diode-connected transistors $Q1$ and $Q2$. If we assume that the transistors possess infinite β and infinite Early voltage, the collector currents of $Q1$

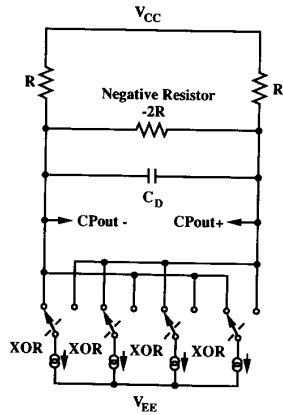


Fig. 18. Phase-detector charge pump.

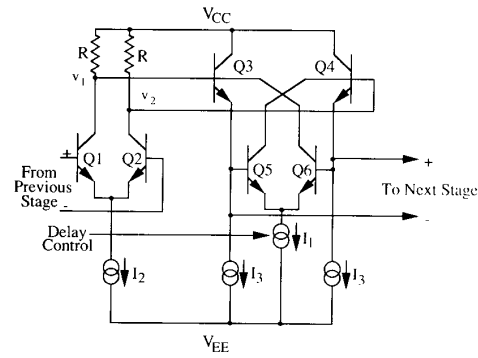


Fig. 21. Delay cell.

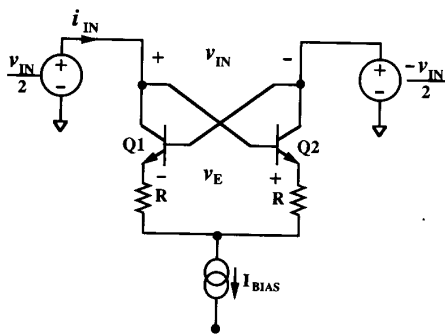


Fig. 19. Simple negative resistance circuit.

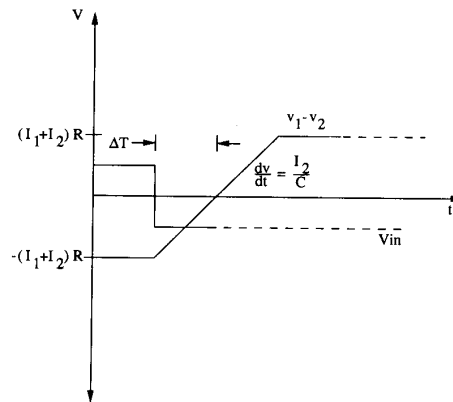


Fig. 22. Idealized delay cell waveforms.

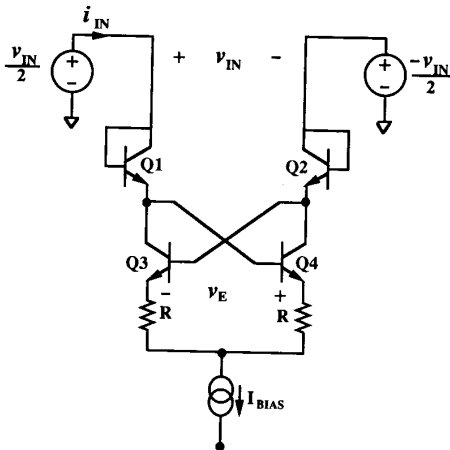


Fig. 20. Improved negative resistance circuit.

and $Q3$ will be equal, as will those of $Q2$ and $Q4$. Hence, $v_{BE1} = v_{BE3}$ and $v_{BE4} = v_{BE2}$. As a result

$$v_{BE1} + v_{BE4} = v_{BE2} + v_{BE3}$$

That is, the voltage drop between the base of $Q1$ and the emitter of $Q4$ equals the voltage drop between the base of $Q2$ and the emitter of $Q3$. Therefore v_e equals v_{in} exactly.

In practice, finite β and Early voltage degrade the performance of the negative resistor somewhat; these effects may also be cancelled if desired [3]. Even without such additional compensation, however, increases of a factor of 50 in the effective load resistance are routinely achievable, assuming a resistor match of 0.1% and a β of 100.

B. Phase Shifter

The phase shifter is based on the cascaded current-starved differential-amplifier cell shown in Fig. 21. The slew-limited outputs of $Q1$ and $Q2$ drive a differential comparator formed by $Q3$ - $Q6$. Control of the delay is by variation of the total differential voltage (by varying $I1$) that the input pair must slew before the comparator changes state, leading to an approximately linear delay-versus-control characteristic. Additionally, the positive

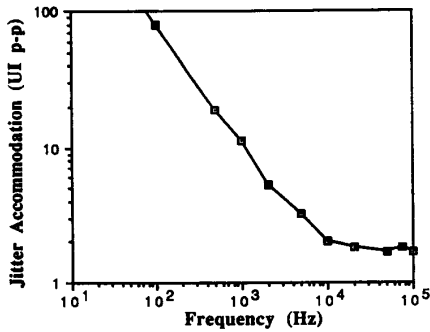


Fig. 23. Measured D/PLL jitter accommodation.

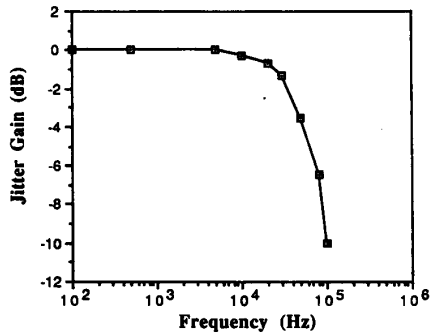


Fig. 24. Measured D/PLL jitter transfer function.

feedback action of the comparator ensures rapid switching once the thresholds are reached, reducing both the window of time over which noise could induce jitter and any pattern sensitivity that could arise if node capacitances were not restored to consistent initial conditions.

Fig. 22 shows idealized waveforms for the delay cell, where exponential transients have been coarsely approximated as straight-line segments. If we further assume that the comparator switches as soon as the differential voltage applied to it becomes zero, the delay can be expressed as follows:

$$\Delta T \approx \frac{RC}{2} \left(1 + \frac{I_1}{I_2} \right)$$

where C is the effective capacitance seen at each collector of the input pair.

As stated in Section III-A, the phase shifter must be able to provide at least ± 0.5 unit intervals (UI's) of shift range, and a somewhat greater value is desirable to improve acquisition in the presence of jitter as well as to absorb inevitable system offsets. This DLL employs a delay chain that provides a typical measured total phase-shift range of approximately 2.2 UI. This value is also

consistent with the requirements of telecommunication standards such as SONET OC-3 specifications.

V. EXPERIMENTAL RESULTS

In addition to avoiding jitter peaking, a practical clock recovery circuit must be able to accommodate the jitter that is always present on real input signals. Jitter accommodation in the D/PLL is governed by different factors in three distinct frequency regimes [5].

Fig. 23 shows the measured jitter accommodation of the D/PLL formed by combining the DLL with an external VCXO that possesses a useable tuning range of approximately 335 ppm. As can be seen, the low-frequency characteristic has a $1/f$ shape. In this region of operation, the VCXO's tuning range controls the jitter accommodation. The $1/f$ shape is due to the integral relationship between phase and frequency, which may be expressed as

$$\#UI_{pp} = \frac{\Delta f_{pk}}{\pi f_{jitter}}$$

where Δf is the peak tuning deviation from center of the VCXO in hertz and f_{jitter} is the frequency of the sinusoidal input phase modulation.

At midrange frequencies (the DLL regime of operation), bounded roughly at the lower end by where the accommodation of the VCXO equals the range of the phase shifter and at the higher end by the loop crossover frequency, the phase shifter controls jitter accommodation. Hence, the midrange accommodation is roughly constant with frequency and equal to the phase shift range of 2.2 UI. In this design, the midrange extends to approximately 3 MHz to accommodate nulling of phase errors in approximately 40 clock periods.

At frequencies above the loop crossover frequency, jitter accommodation necessarily drops below unity, to a value determined by the width of the eye opening and the static phase error. Because of test limitations, measurements in this regime are absent from Fig. 23.

Fig. 24 shows the measured jitter transfer gain of the D/PLL. There is no evidence of jitter peaking anywhere within the 40-kHz bandwidth. Note that this closed-loop bandwidth is only approximately 250 ppm of the bit rate, while the loop crosses over at approximately 3 MHz, in contrast with conventional loops where both bandwidths must be the same.

Fig. 25 is a histogram of jitter on the recovered clock for the D/PLL. With an input that possesses the maximum data transition density, the rms jitter is less than 1° . The rms jitter changes insignificantly even when the D/PLL is driven with a $2^{23}-1$ pseudorandom bit sequence (PRBS), indicating that there is negligible data-dependent jitter.

The $2.5 \times 4.3\text{-mm}^2$ die shown in Fig. 26 draws 70 mA from a standard ECL - 5.2-V power supply.

155 MHz

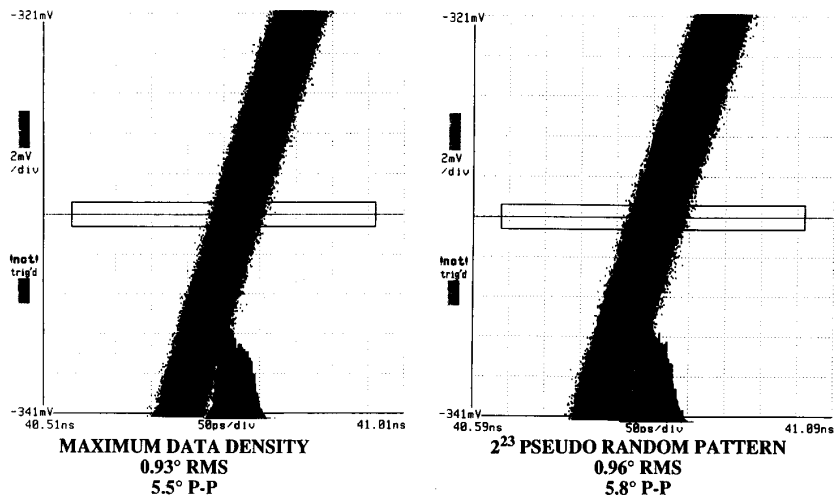


Fig. 25. Jitter histograms.



Fig. 26. DLL die photo.

VI. SUMMARY

By shifting the phase of the input data relative to the clock with a voltage-controlled phase shifter, the DLL and D/PLL avoid many of the trade-offs that limit the performance of conventional clock recovery circuits. Rapid acquisition can be achieved without compromising jitter filtering, and neither the DLL nor the D/PLL exhibits jitter peaking. These architectures thus offer a combination of jitter filtering and rapid acquisition not achievable with conventional clock recovery circuits.

ACKNOWLEDGMENT

The authors are grateful to B. Surette for assistance in testing; to T. Freitas for expertise in layout; to R.

Croughwell, E. Ferrari, and L. DeVito for invaluable help and enlightening discussions; and to S. Hubbard for aid in manuscript preparation.

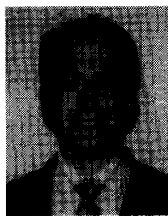
REFERENCES

- [1] J. Bulzacchelli, U.S. Patent 5 036 298, July 1991.
- [2] P. Trischitta and E. Varma, *Jitter in Digital Transmission Systems*. Dedham, MA: Artech House, 1989.
- [3] L. DeVito *et al.*, "A 52MHz and 155MHz clock-recovery PLL," in *ISSCC Dig. Tech. Papers*, Feb. 1991, pp. 142-143.
- [4] C. R. Hogge, "A self-correcting clock recovery circuit," *J. Light-wave Technol.*, vol. LT-3, no. 6, pp. 1312-1314, 1985.
- [5] J. Bulzacchelli, "A delay-locked loop for clock recovery and data synchronization," Master's thesis, Massachusetts Inst. of Technology, Cambridge, June 1990.
- [6] T. H. Lee and J. F. Bulzacchelli, "A 155MHz clock recovery delay- and phase-locked loop," in *ISSCC Dig. Tech. Papers*, Feb. 1992, pp. 160-161.



Thomas H. Lee (M'90) was born in Pittsburgh, PA, on July 2, 1959. He received the S.B., S.M., and Sc.D. degrees in electrical engineering, all from the Massachusetts Institute of Technology, Cambridge, in 1983, 1985, and 1990, respectively.

He joined Analog Devices in Wilmington, MA, in 1990 where he was primarily engaged in the design of high-speed clock recovery devices. He is now with Rambus Inc. in Mountain View, CA, where he is developing high-speed PLL's and DLL's for 500-megabyte/s DRAM's. He also maintains a strong interest in teaching, as well as in vacuum tubes and the early history of semiconductors.



John F. Bulzacchelli was born in New York, NY, in 1966. He received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology (M.I.T.), Cambridge, in 1990. He is currently a candidate for the doctorate in electrical engineering at M.I.T.

From 1988 to 1990 he worked on his S.M. thesis, "A delay-locked loop for clock recovery and data synchronization," at Analog Devices, Wilmington, MA, while he was a co-op student. His research into new techniques for clock recovery culminated in his invention of the delay-and-phase-locked loop, for which he recently received a patent.