Challenges in the Design of High-Speed Clock and Data Recovery Circuits

Behzad Razavi, University of California, Los Angeles

ABSTRACT

This article describes the challenges in the design of monolithic clock and data recovery circuits used in high-speed transceivers. Following an overview of general issues, the task of phase detection for random data is addressed. Next, Hogge, Alexander, and half-rate phase detectors are introduced and their trade-offs outlined. Finally, a number of clock and data recovery architectures are presented.

INTRODUCTION

Clock and data recovery (CDR) is a critical function in high-speed transceivers. Such transceivers serve in many applications, including optical communications, backplane routing, and chip-to-chip interconnects. The data received in these systems are both asynchronous and noisy, requiring that a clock be extracted to allow synchronous operations. Furthermore, the data must be "retimed" such that the jitter accumulated during transmission is removed. CDR circuits must satisfy stringent specifications defined by communication standards, posing difficult challenges to system and circuit designers.

This article presents the challenges in the design of high-speed CDR circuits, focusing on monolithic implementations in very large scale integrated (VLSI) technologies. The next section of the article addresses general issues in CDR design dealing with phase detectors for random data. We also describe a number of CDR architectures and their design trade-offs.

GENERAL CONSIDERATIONS

In order to perform synchronous operations such as retiming and demultiplexing on random data, high-speed receivers must generate a clock. As illustrated in Fig. 1a, a clock recovery circuit senses the data and produces a periodic clock. A D flipflop (DFF) driven by the clock then retimes the data (i.e., it samples the noisy data), yielding an output with less jitter. As such, the flipflop is sometimes called a *decision circuit*.

The clock generated in the circuit of Fig. 1a must satisfy three important conditions [1]:

- It must have a frequency equal to the data rate; for example, a data rate of 10 Gb/s (each bit 100 ps wide) translates to a clock frequency of 10 GHz (with a period of 100 ps).
- It must bear a certain phase relationship with respect to data, allowing optimum sampling of the bits by the clock; if the rising edges of the clock coincide with the midpoint of each bit, the sampling occurs farthest from the preceding and following data transitions, providing maximum margin for jitter and other timing uncertainties.
- It must exhibit a small jitter since it is the principal contributor to the retimed data jitter.

Simple CDR Circuit — We now consider a simple CDR circuit, analyzing its behavior and shortcomings.

As illustrated in Fig. 1a, to generate the clock waveform we employ a voltage-controlled oscillator (VCO), and to define its frequency and phase we phase-lock the VCO to the input data using a DFF operating as a phase detector (PD). The low-pass filter (LPF) suppresses ripple on the oscillator control line. Also, to retime the data, we add another DFF that is clocked by the VCO output. Note that the recovered clock, CK_{out}, drives the D input of the phase detector and the clock input of the retimer.

The circuit of Fig. 1b operates as follows. Upon turnon, the DFF multiplies the edge-detected data by the VCO output, generating a beat that drives the VCO frequency toward the input bit rate. If the initial difference between the VCO frequency and the data rate is sufficiently small, the loop locks, establishing a well-defined phase relationship between D_{in} and CK_{out}. In fact, with the bang-bang characteristic provided by the DFF phase detector, the data edges settle around the zero-crossing points of the clock. Even for a slight

0163-6804/02/$17.00 © 2002 IEEE

phase error, the PD generates a large output, driving the loop toward lock.

The simple CDR circuit of Fig. 1b suffers from a number of drawbacks. First, the PD may produce full digital outputs for run lengths greater than one, thereby creating substantial ripple on the oscillator control voltage and hence jitter at the output. Second, since the PD samples the clock by the data, whereas the decision circuit samples the data by the clock, data retiming exhibits significant phase offset at high speeds. Typical flipflops display unequal delays from the D input to the output and from the clock input to the output. Thus, if, for example, the CK-to-Q delay is longer than the D-to-Q delay by ΔT, the PD locks such that the data *leads* the clock by ΔT, sampling the clock closer to the zero crossing after the data experiences the intrinsic delay of the PD. The VCO output suffers from even more delay as it propagates through the decision circuit, sampling the data far from the middle of the eye. In other words, if the difference between the CK-to-Q and D-to-Q delays is equal to ΔT, the retiming suffers from a skew of $2\Delta T$.

The third drawback of the simple CDR architecture of Fig. 1b relates to the feedthrough of data to the VCO output through both flipflops. The output phase is disturbed on arrival of each data transition, requiring that the VCO be followed by a buffer stage providing significant reverse isolation.

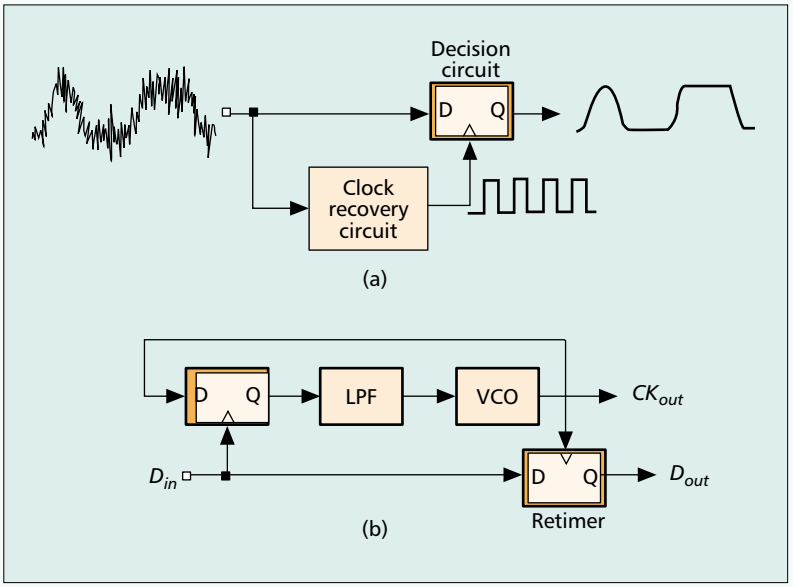

Figure 1. *a) The role of a CDR circuit in retiming data; b) an example of CDR implementation.*

Phase Detectors for Random Data

As mentioned earlier, PDs for random data must provide two essential functions:

- Data transition detection
- Phase difference detection

Furthermore, the skew effect described above makes it desirable to retime the data inside the phase detector, thus eliminating the explicit decision circuit and the skew associated with it. This observation immediately leads to another: if the data is to be retimed by the VCO output, then the flipflop(s) in the phase detector must be strobed by the latter rather than the former. In other words, unlike the DFF PD of Fig. 1b, the phase detector must sample the incoming data by the VCO signal.

The Hogge Phase Detector

How can a PD detect data transitions if it samples the data by the VCO output? A single DFF fails to operate as a phase detector if it is used in such a mode. However, recognizing that a DFF produces a *delayed* replica of the input data, we can arrive at a synchronous edge detector (Fig. 2a). Since sample B changes only on the CK edges, $Y = D_{in} \oplus B$ contains pulses whose width represents the phase difference between D_{in} and CK. It is important to note that (a) the circuit produces a pulse for each data transition, providing edge detection, and (b) the width of the output pulses varies *linearly* with the input phase difference, suggesting that the circuit can operate as a linear PD. We call this type of output *proportional pulses*.

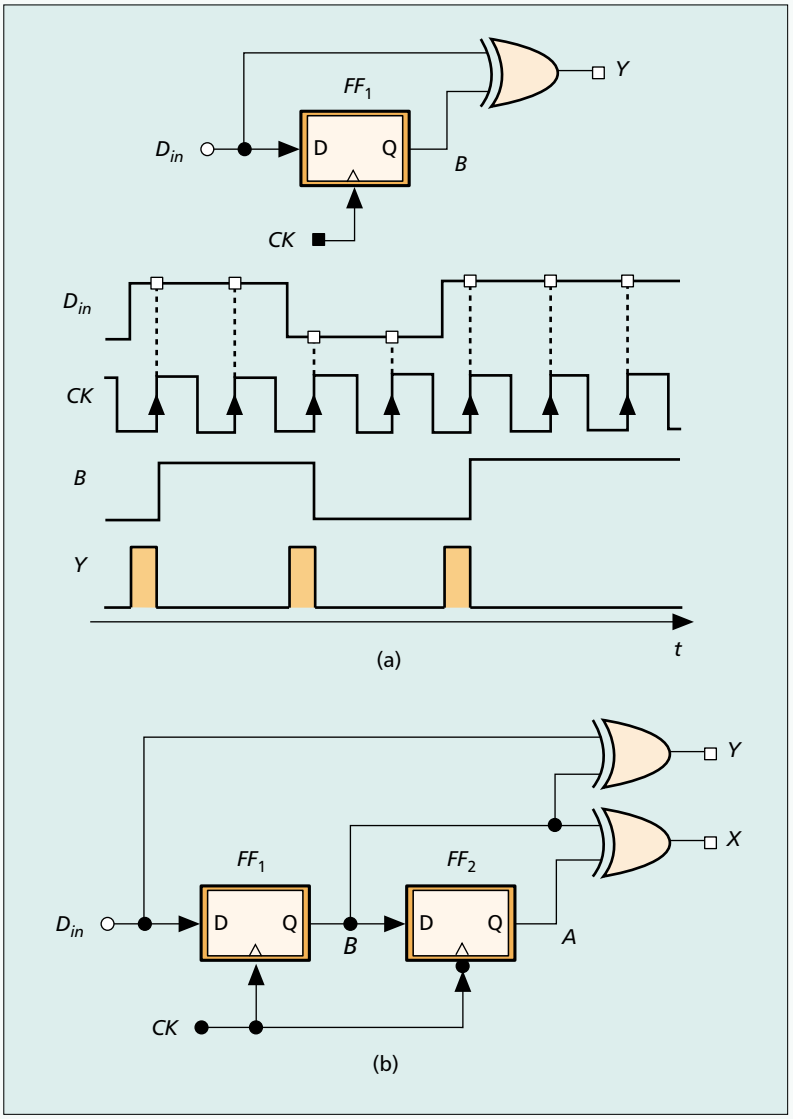

Figure 2. *a) A simple PD using synchronous edge detection; b) a Hogge phase detector.*

The Hogge topology is a linear phase detector, generating a small average as the phase difference approaches zero. Thus, a charge pump driven by a Hogge PD experiences little "activity" when the CDR loop is locked.

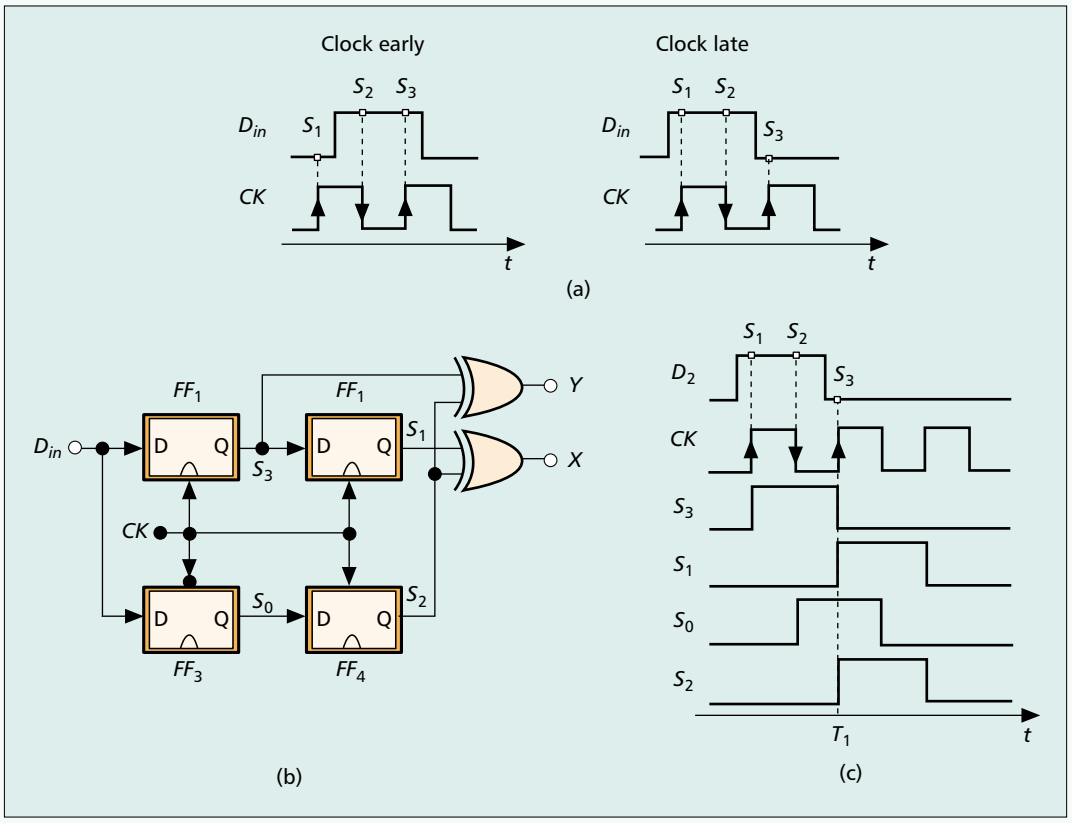

Figure 3. *a) Three-point sampling of data by clock; b) an Alexander phase detector.*

It may appear that the topology of Fig. 2a satisfies the requirements of a phase detector and can therefore be used as such. Unfortunately, however, the average value of the output is a function of the data transition density, failing to uniquely represent the phase difference for various data patterns. For example, the average output remains unchanged if the transition density falls by a factor of two and the phase difference rises by the same factor. In other words, two different phase errors may result in the same dc output, leading to false lock.

To overcome the above ambiguity, the proportional pulses must be accompanied by *reference pulses*. The latter are pulses that appear only on data edges but exhibit a *constant* width, eliminating the pattern dependency.

How can reference pulses be generated? We note that if the retimed data (at point B) in Fig. 2a is delayed by half a clock cycle, $T_{CK}/2$, and XORed with itself, pulses of width $T_{CK}/2$ are produced for each data transition. As depicted in Fig. 2b, the difference between the areas under X and Y can be viewed as the PD output, eliminating the ambiguity due to transition density. Note that under locked condition, X and Y produce equal pulse widths. This circuit is called the *Hogge phase detector* [2].

Let us summarize our thought process thus far. In order to avoid skews in the decision circuit, we choose to sample the data by the clock even in the PD. This in turn requires explicit edge detection, carried out by a DFF and an XOR gate. Finally, we produce a reference pulse to eliminate ambiguity for different data transition densities.

Recall that the principal motivation behind the above development is to retime the data inside the PD. Does the Hogge PD accomplish this? Indeed, both flipflops in Fig. 2b operate as decision circuits as well, thereby providing retimed data.

It is instructive to examine the behavior of the Hogge PD in the presence of finite delays in the flipflops. Owing to the *CK*-to-*Q* delay, ΔT, of FF_1, B changes ΔT seconds after the clock rises, yielding a pulse at the Y output that is ΔT s wider than the actual phase difference between D_{in} and *CK*. On the other hand, the *CK*-to-*Q* delay of FF_2 simply *shifts* the pulses at A by ΔT, still producing a pulsewidth equal to one clock period. As a result, X continues to produce pulses of width $T_{CK}/2$ for each data transition. This means, with a zero input phase difference, the proportional pulses are wider than the reference pulses by ΔT s. Thus, under locked condition, D_{in} and *CK* must sustain a skew of ΔT to equalize the widths of the X and Y pulses.

The above skew effect becomes a serious issue at high speeds. Since ΔT can be a significant fraction of the clock period, a systematic phase offset of several tens of degrees may arise after the loop is locked, degrading the clock phase margin and hence the jitter tolerance. In order to resolve this difficulty, we can either narrow the proportional pulses by ΔT [2] or widen the reference pulses by the same amount.

A drawback of the Hogge PD stems from the half-cycle skew between the two XOR outputs [3, 4]. Under locked condition, the PD produces the reference pulse *after* the proportional pulse, thereby creating a skew of $T_{CK}/2$ between the

two. As a result, the VCO phase (which is proportional to the integral of the control voltage) is severely disturbed. The phase detector can be modified to ameliorate this issue [4].

The Hogge topology is a linear PD, generating a small average as the phase difference approaches zero. Thus, a charge pump driven by a Hogge PD experiences little "activity" when the CDR loop is locked. This behavior is in contrast to that of the bang-bang PD of Fig. 1b.

The Alexander Phase Detector

The Alexander configuration is another example of PDs providing inherent data retiming. Following our reasoning for the Hogge PD, we note that this property requires that the data be sampled by the clock, but a single DFF does not suffice. Nonetheless, if the clock strobes the data waveform at *multiple* points in the vicinity of expected transitions, the resulting samples can provide the necessary information.

Figure 3a illustrates the Alexander PD principle, also known as the *early-late* detection method. Utilizing three data samples taken by three consecutive clock edges, the PD can determine whether a data transition is present, and whether the clock leads or lags the data. In the absence of data transitions, all three samples are equal and no action is taken. If the clock leads (is *early*), the first sample, S_1, is unequal to the last two. Conversely, if the clock lags (is *late*), the first two samples, S_1 and S_2, are equal but unequal to the last sample, S_3. Thus, $S_1 \oplus S_2$ and $S_2 \oplus S_3$ provide the early-late information:

- If $S_1 \oplus S_2$ is high and $S_2 \oplus S_3$ is low, the clock is late.
- If $S_1 \oplus S_2$ is low and $S_2 \oplus S_3$ is high, the clock is early.
- If $S_1 \oplus S_2 = S_2 \oplus S_3$, no data transition is present.

The foregoing observations lead to the circuit topology shown in Fig. 3b [5]. Flipflops FF_1 and FF_2 sample their D inputs on the rising edge of CK, producing S_3 and S_1, respectively. Flipflop FF_3 samples D_{in} on the falling edge of CK, and flipflop FF_4 delays this sample by half a clock cycle, generating S_2. Note that the sampling points are defined by FF_1 and FF_3; the other two FFs merely serve as delay elements.

Let us examine the waveforms at various points in the Alexander PD to gain more insight into its operation. As depicted in Fig. 3c, the first rising edge of CK samples a high data level. The second rising edge of CK then accomplishes two tasks: it produces a delayed version of the first sample at the output of FF_2 and samples the low level on the input data. The values of S_1 and S_2 are therefore valid for comparison at $t = T_1$, remaining constant for one clock period.

On the first rising edge of CK in Fig. 3c, FF_1 samples a high level on the input data, and on the next rising edge, FF_2 reproduces this level. The key point here is that the choice of clock phases for the four FFs ensures that S_1, S_2, and S_3 reach valid levels for comparison at $t = T_1$, and remain at these levels for one clock period. As a result, the XOR gates always generate valid outputs simultaneously.

The Alexander PD is a bang-bang system, exhibiting a very high gain in the vicinity of $\Delta\phi = 0$. Consequently, a CDR loop utilizing this PD locks such that S_2 coincides with the data zero crossings. While exhibiting a bang-bang characteristic, the Alexander PD offers two critical advantages over a simple DFF PD. First, it retimes the data automatically, producing a valid data waveform at the output of FF_1 and FF_2 in Fig. 3b. Second, in the absence of data transitions, it generates a zero dc output, leaving the oscillator control undisturbed. As a result, for long data runs, the VCO frequency drifts only due to device electronic noise rather than due to a high or low level on the control line.

Half-Rate Phase Detectors

At very high speeds, it may be difficult to design oscillators that provide an adequate tuning range with reasonable jitter. For this reason, CDR circuits may sense the input random data at full rate but utilize a VCO running at half the input rate. This technique also relaxes the speed requirements of the phase detector and, in some CDR configurations, the frequency dividers. Called *half-rate* architectures, such CDR topologies require a phase detector that provides a valid output while sensing a full-rate random data stream and a half-rate clock.

It is important to note that none of the three PDs studied thus far can operate with a half-rate clock. However, data transitions may be detected properly if *both edges* of the half-rate clock are utilized to sample the input data. Consider the topology shown in Fig. 4a, where two D latches, L_1–L_2, operate on opposite edges of the clock. Note that each latch is transparent for half of the clock cycle, passing data transitions to its output. Assuming D_{in} leads CK by ΔT and L_1 is transparent when CK is high, we observe that A_1 goes high *before* CK falls and remains high until CK rises. In other words, L_1 produces a pulse width equal to $T_{CK}/2 + \Delta T$. On the other hand, if L_2 is transparent when CK is low, A_2 goes high when CK falls and remains high only until D_{in} falls. That is, L_2 generates a pulse width of $T_{CK}/2 \oplus \Delta T$. Thus, $A_1 \oplus A_2$ exhibits a pulse of width ΔT for each data transition.

The above study implies that the simple topology of Fig. 4a can indeed operate as a linear phase detector because it:

- Detects data edges
- Produces proportional pulses

However, as with the Hogge topology, this circuit must also provide a reference output to uniquely represent the phase error for different data transitions. To this end, let us follow the latches with two more (creating a master-slave flipflop in each path) and XOR the outputs (Fig. 4b). In the presence of data transitions, the outputs of L_3 and L_4 change on the falling and rising edges of the clock, respectively. As a result, $C \oplus D$ contains a pulse width of $T_{CK}/2$ for each input data edge, serving as the reference output [6].

How does a CDR loop employing the PD of Fig. 4b lock to random data? If the clock edge is to strobe the data in the middle of the eye, the proportional pulses are $T_{CK}/4$ s wide, whereas

At very high speeds, it may be difficult to design oscillators that provide an adequate tuning range with reasonable jitter. For this reason, CDR circuits may sense the input random data at full rate but utilize a VCO running at half the input rate.

A critical difficulty in modern CDR circuits stems from the use of low supply voltages. The gain of VCOs must increase as the supply is scaled down because the tuning range must remain a constant percentage of the center frequency. As a result, for a given ripple on its control line, the VCO suffers from greater jitter.

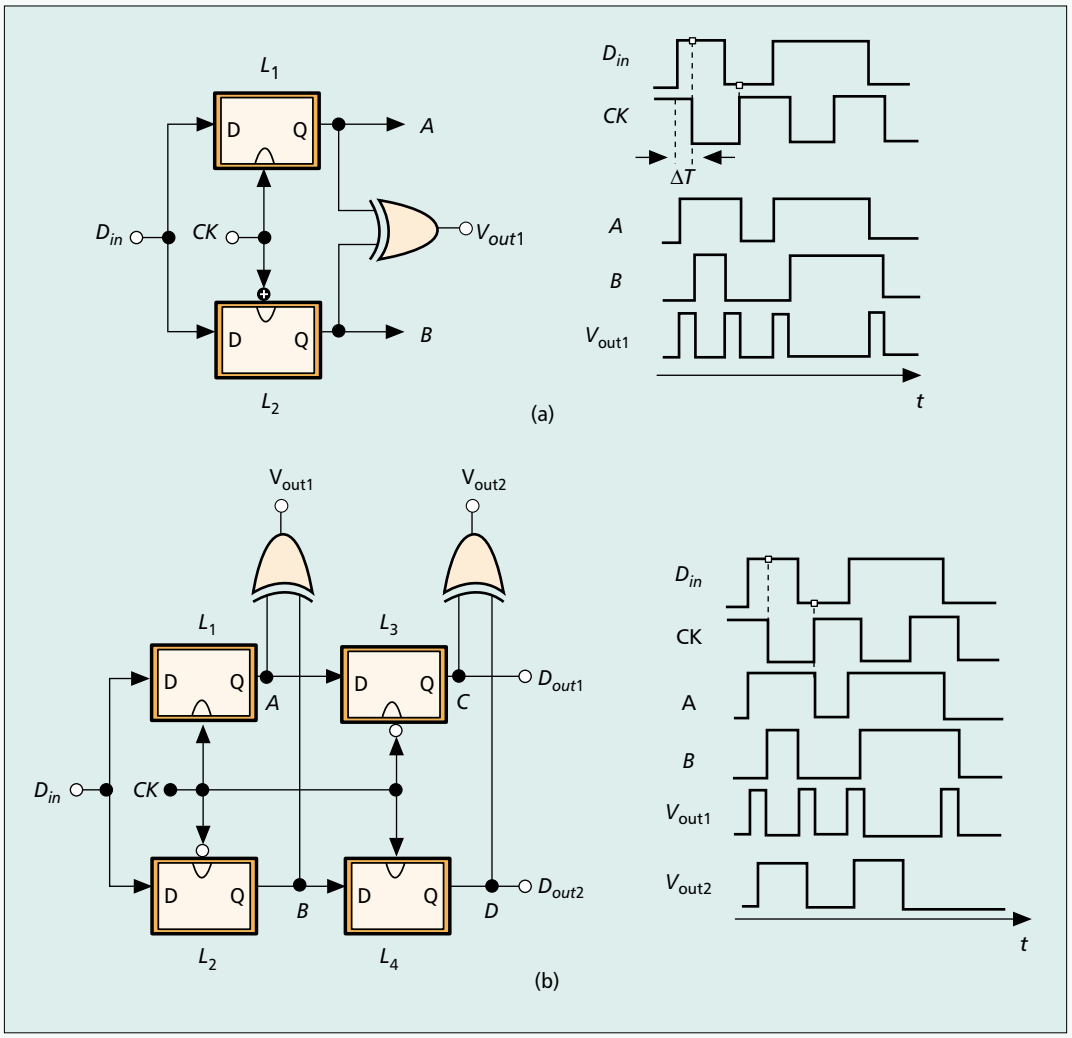

Figure 4. *a) A simple linear half-rate PD; b) a complete half-rate PD.*

the reference pulses are $T_{CK}/2$ s wide. The disparity between the average values of these outputs is removed by scaling down the *effect* of the output of the second XOR by a factor of two, that is, halving the corresponding current source in the charge pump.

The half-rate PD of Fig. 4b also retimes and demultiplexes the data, producing two streams at the outputs of L_3 and L_4. The linear characteristic of the circuit allows simple formulation of the loop dynamics.

Let us now consider the early-late method for half-rate operation. Since the Alexander PD already requires sampling on both clock edges for full-rate detection, it must employ additional phases of the clock if it is to operate in the half-rate mode. Shown in Fig. 5a, the solution involves sampling the data by both the in-phase and quadrature phases of the clock, CK_I and CK_Q, respectively. Now, A_1, A_2, and A_3 play the same role as the consecutive samples in a full-rate counterpart. As depicted in Fig. 5b, the implementation incorporates three FFs sampling the data by CK_I and CK_Q, and two XOR gates producing $A_1 \oplus A_2$ and $A_2 \oplus A_3$. Under the locked condition, the rising edge of CK_Q occurs in the vicinity of the data zero crossings.

CDR Architectures

With our foregoing study of PDs, we can now develop complete CDR architectures. Each architecture must include:

- Frequency and phase acquisition to ensure lock despite process and temperature variations of the VCO frequency
- Data retiming inside the PD to avoid systematic skews

A critical difficulty in modern CDR circuits stems from the use of low supply voltages. The gain of VCOs must *increase* as the supply is scaled down because the tuning range must remain a constant percentage of the center frequency. As a result, for a given ripple on its control line, the VCO suffers from greater jitter. A method of alleviating this issue is to decompose the VCO control into *fine* and *coarse* inputs, allowing the latter to remain quiet after the system is phase-locked. This concept is described in the context of some CDR architectures.

Full-Rate Referenceless Architectures

Frequency detectors capable of handling random data obviate the need for external reference frequencies. Figure 6a depicts a referenceless architecture [7], where loop I employs a frequency

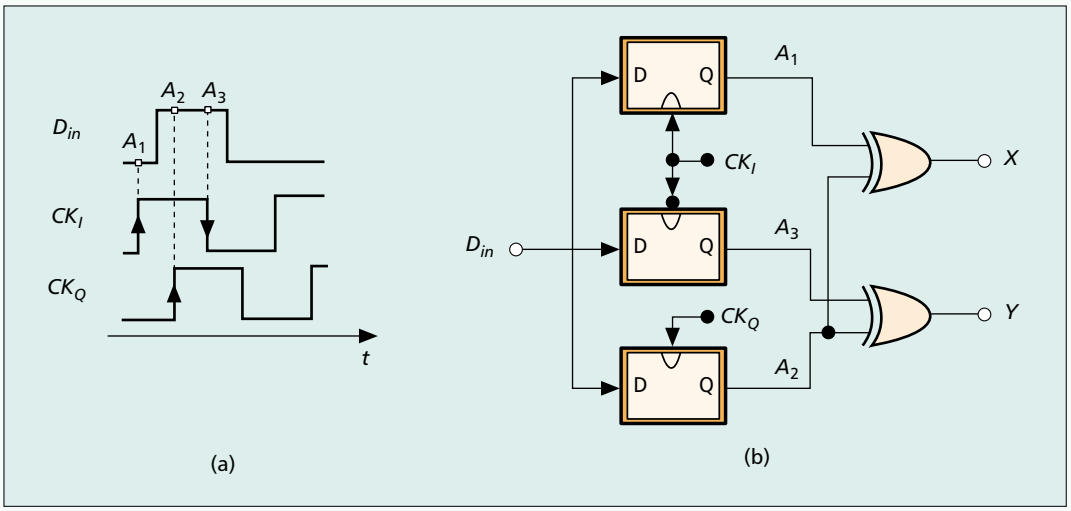

Figure 5. *a) The use of quadrature clocks for half-rate phase detection; b) a half-rate binary PD.*

detector (FD) [7] and loop II incorporates one of the PDs studied in an earlier section. Upon startup or loss of phase lock, the FD produces a dc level that drives VCO frequency toward the input data rate. When the frequency error falls within the capture range of loop I, the PD takes over, phase-locking the clock to the data.

The above architecture entails two issues. First, as the CDR circuit transfers the control of the VCO from the FD to the PD, the two loops may interact so heavily that the overall system fails to phase-lock. In this case, the two loops continue to "fight" indefinitely. Second, with the actual random data produced in a network, short-term spectral lines close but unequal to the nominal data rate may appear occasionally, possibly confusing the frequency detector. For these reasons, the bandwidth of the frequency-locked loop (FLL) is typically chosen to be much smaller than that of the PLL. These issues must be studied carefully for each specific design and application.

Note that this architecture employs only one control line for the oscillator and must therefore ensure a very small ripple. It is possible to modify the system as shown in Fig. 6b, where loop II drives only the coarse control. This modification also permits independent choice of the PLL and FLL bandwidths.

Upon startup or loss of phase lock, the FD produces a dc level that drives VCO frequency toward the input data rate. When the frequency error falls within the capture range of loop I, the PD takes over, phase-locking the clock to the data.

Architectures with External References

Recall that it is desirable to decompose the VCO control into fine and coarse inputs. A CDR architecture utilizing such a scheme is shown in Fig. 7a [8]. Here, loop I phase-locks VCO_1 to the input data through fine control. Since the gain of VCO_1 with respect to fine input is relatively low, ripple on this line translates to a small jitter at the output. Of course, the fine control may not provide enough tuning range to encompass process and temperature variations. Loop II is therefore added to lock VCO_2, a replica of VCO_1,to Nf_{REF}, with the resulting control voltage applied to the coarse input of VCO_1 as well. If Nf_{REF} is exactly equal to the input data rate and the two VCOs match perfectly, the fine control of VCO_1 stabilizes at a voltage equal to that on the coarse control. The low-pass filter consisting of R_1 and C_1 suppresses the ripple generated in loop II, thereby presenting a low-noise control to VCO_2.

While reducing the effect of ripple, the architecture of Fig. 7a faces two issues. First,

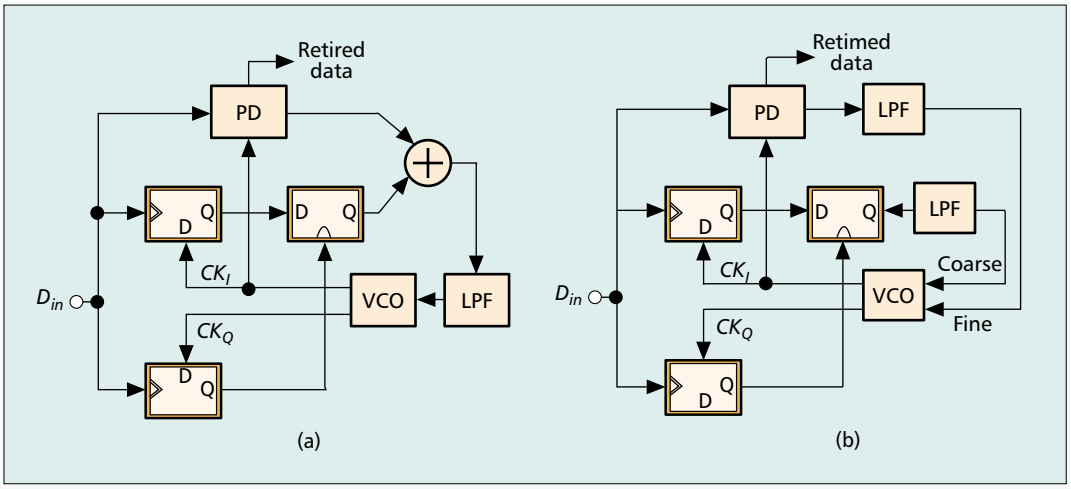

Figure 6. *CDR architectures with a) FD and PD; b) coarse and fine VCO control.*

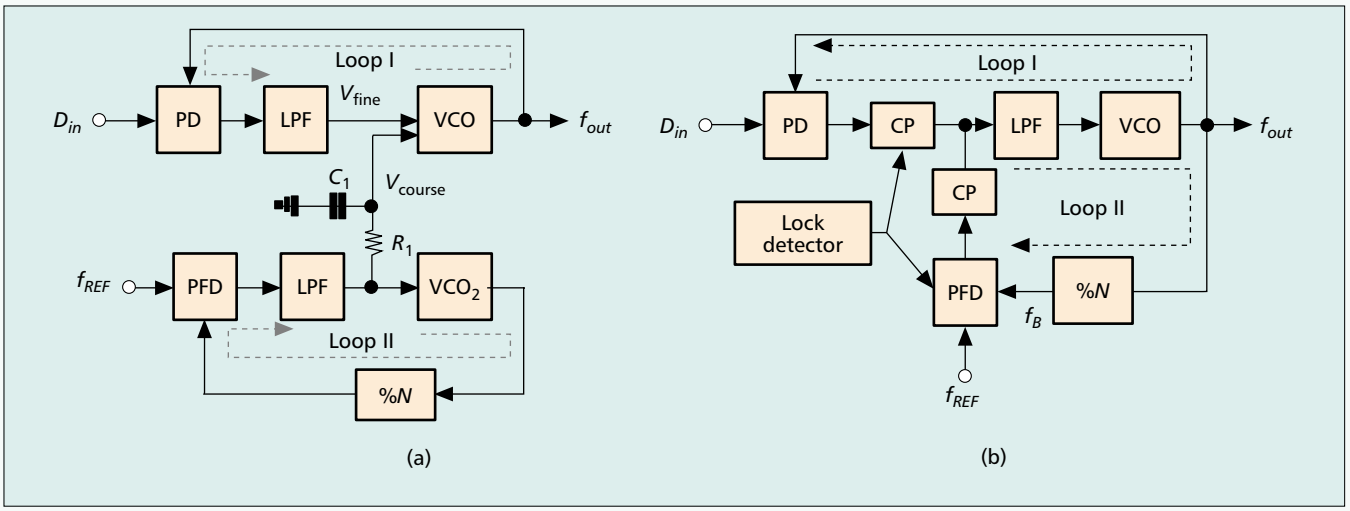

Figure 7. *CDR architecture using a) two VCOs; b) sequential locking.*

VCOs lead to a substantial center frequency mismatch even though the oscillators share the same coarse input. For this reason, loop I must still achieve a sufficiently wide capture range to guarantee lock despite the initial frequency error. With a typical frequency mismatch of a few percent, random data PDs may not provide adequate capture range.

Second, even if the oscillators match perfectly, the incoming data rate is not exactly equal to Nf_{REF} because the reference frequency in the far-end transmitter is derived from a crystal oscillator that may suffer an error of 5–10 ppm with respect to the f_{REF} generator in the receiver. Consequently, VCO_1 and VCO_2 operate at slightly different frequencies, possibly pulling each other through the substrate or supply lines. The use of differential swings for both VCOs alleviates this issue, but asymmetries in each circuit still give rise to a finite amount of crosstalk. It is interesting to note that the large bandwidth of loop II partially corrects for the pulling experienced by VCO_2. Loop I, on the other hand, has a bandwidth commensurate with the communication standard and hence cannot counteract pulling effects on VCO_1.

Another general concern in the architecture of Fig. 7a relates to the layout of the two VCOs. If both oscillators incorporate LC tanks, the large area occupied by on-chip inductors creates difficulties in routing the signal and power lines.

Figure 7b shows a relatively simple CDR architecture that acquires frequency and phase in two steps. Using a single VCO, the circuit first enables loop II, thereby beginning to lock the oscillator to Nf_{REF}. The lock detector monitors the difference between f_B and f_{REF}, disabling loop II and enabling loop I when the frequency error drops to a sufficiently small value (e.g., 0.1 percent). Thus, loop I begins with a frequency error well within its capture range, locking the VCO to the data. The lock detector continues to operate so that loop II can be activated again if loop I loses lock as a result of unexpected noise.

It is instructive to compare the CDR architectures of Figs. 7a and 7b. Both topologies require an external reference, but the latter need not deal with frequency mismatches or oscillator pulling. However, the former allows the use of fine and coarse controls for the oscillator, whereas the latter does not.

Another issue in the architecture of Fig. 7b relates to the transition from loop II to loop I. If the switches that perform this transition disturb the control voltage significantly, the VCO frequency may jump by a large amount, falling out of the capture range of loop I. Thus, the charge injection and clock feedthrough of the switches must be examined carefully.

TOPICS IN CIRCUITS FOR COMMUNICATIONS
PUBLISHED IN COLLABORATION WITH THE
IEEE CIRCUITS AND SYSTEMS SOCIETY
SERIES EDITORIAL BOARD

EDITOR

Mihai Banu
Agere Systems
600 Mountain Ave., Room 2D-328, Murray Hill, NJ 07974, USA
tel: 908 582 5380; fax: 908 582 6000
e-mail: mihai@agere.com

ASSOCIATE EDITORS

Young-Kai Chen
Bell Labs, Lucent Technologies
600 Mountain Ave.
Room 1C-325
Murray Hill, NJ 07974, USA
tel: 908 582 7956
fax: 908 582 6322
e-mail: ykchen@lucent.com

Bram Nauta
University of Twente
Electrical Engineering
EL/TN 3242, P.O. Box 217
7500 AE Enschede
The Netherlands
tel: 31 53 489 2655 (2644)
fax: 31 53 489 1034
e-mail: B.Nauta@el.utwente.nl

Jan Sevenhans
Alcatel
Francis Wellesplein 1
2018 Antwerpen, Belgium
Tel: 32 3 240 8752
Fax: 32 3 240 9947
e-mail: jsev@sh.bel.alcatel.be

Ernesto H. Perea
STMicroelectronics
Wireless Communication Division
12, rue Jules Horowitz - B.P. 217
38019 Grenoble Cedex, France
tel: (33) 4 76 58 65 03
GSM: (33) 6 87 86 11 99
fax: (33) 4 76 58 51 73
email: ernesto.perea@st.com

CONCLUSION

High-speed CDR design must deal with many circuit and architecture issues, including jitter, skews, and acquisition of lock. This article has focused on the difficulties in the design of phase detectors and CDR architectures, presenting a number of high-speed topologies. In particular, Hogge, Alexander, and half-rate PDs prove useful as the data rates continue to increase. The design of VCOs also presents many interesting challenges, and the reader is referred to [1] and various papers on the subject for details.

REFERENCES

[1] B. Razavi, *Design of Integrated Circuits for Optical Communications*, Chicago: McGraw Hill, 2002.
[2] C. R. Hogge, "A Self-Correcting Clock Recovery Circuit," *IEEE J. Lightwave Tech.*, vol. 3, Dec. 1985, pp. 1312–14.
[3] L. DeVito *et al.*, "A 52MHz and 155MHz Clock Recovery PLL," *ISSCC Dig. Tech. Papers*, Feb. 1991, pp. 142–43.
[4] L. DeVito, "A Versatile Clock Recovery Architecture and Monolithic Implementation," *Monolithic Phase-Locked Loops and Clock Recovery Circuits*, B. Razavi, Ed., New York: IEEE Press, 1996.
[5] J. D. H. Alexander, "Clock Recovery from Random Binary Data," *Elect. Lett.*, vol. 11, Oct. 1975, pp. 541–42.
[6] J. Savoj and B. Razavi, "A 10-Gb/s CMOS Clock and Data Recovery Circuit with a Half Rate Linear Phase Detector," *IEEE J. Solid-State Circuits*, vol. 36, May 2001, pp. 761–68.
[7] A. Pottbacker, U. Langmann, and H. U. Schreiber, "A Si Bipolar Phase and Frequency Detector for Clock Extraction up to 8 Gb/s," *IEEE J.Solid-State Circuits*, vol. 27, Dec. 1992, pp. 1747–51.
[8] J. C. Scheytt, G. Hanke, and U. Langmann, "A 0.155, 0.622, and 2.488 Gb/s Automatic Bit Rate Selecting Clock and Data Recovery IC for Bit Rate Transparent SDH Systems," *ISSCC Dig. Tech. Papers*, Feb. 1999, pp. 348–49.

BIOGRAPHY

BEHZAD RAZAVI (razavi@ee.ucla.edu) received his B.Sc. degree in electrical engineering from Sharif University of Technology in 1985, and his M.Sc. and Ph.D. degrees in electrical engineering from Stanford University in 1988 and 1992, respectively. He was with AT&T Bell Laboratories and Hewlett-Packard Laboratories until 1996. Since September 1996, he has been associate professor and subsequently professor of electrical engineering at the University of California, Los Angeles. His current research includes wireless transceivers, frequency synthesizers, phase-locking and clock recovery for high-speed data communications, and data converters. He was an adjunct professor at Princeton University, New Jersey, from 1992 to 1994, and at Stanford University in 1995. He served on the Technical Program Committee of the International Solid-State Circuits Conference (ISSCC) from 1993 to 2002 and is presently a member of the Technical Program Committee of Symposium on VLSI Circuits. He has also served as guest editor and associate editor of *IEEE Journal of Solid-State Circuits*, *IEEE Transactions on Circuits and Systems*, and *International Journal of High Speed Electronics*. He received the Beatrice Winner Award for Editorial Excellence at the 1994 ISSCC, the best paper award at the 1994 European Solid-State Circuits Conference, the best panel award at the 1995 and 1997 ISSCC, the TRW Innovative Teaching Award in 1997, and the best paper award at the IEEE Custom Integrated Circuits Conference in 1998. He was the co-recipient of both the Jack Kilby Outstanding Student Paper Award and the Beatrice Winner Award for Editorial Excellence at the 2001 ISSCC. He is an IEEE Distinguished Lecturer and the author of *Principles of Data Conversion System Design* (IEEE Press, 1995), and *RF Microelectronics* (Prentice Hall, 1998), among other books.

High-speed CDR design must deal with many circuit and architecture issues, including jitter, skews, and acquisition of lock.

CALL FOR PAPERS
IEEE COMMUNICATIONS MAGAZINE
FEATURE TOPIC - GMPLS

Scope:

The Generalized MultiProtocol Label Switching (GMPLS) is a very hot topic, which is currently under standardization by IETF and ITU. GMPLS is the next step of MPLS; it provides a generalized signaling control protocol standard for the multiple types of switching. It is our expectation that GMPLS is going to shape a new control plane for the future IP-networks and to simplify the control functions supporting multiple types of switching. In particular, the so-called GMPLS-oriented Label Switching Router (LSR) architecture is concreted for better performance, and impacts many aspects of future IP-networks.

This Feature Topic of IEEE Communications Magazine seeks to present a series of state-of-the-art articles on GMPLS-related topics. In particular, we are interested in tutorial and survey articles on, but not limited to, the following topics:

GMPLS Architecture
GMPLS-oriented Label Switching Router (LSR) System
GMPLS-oriented QoS
GMPLS-oriented Link Management
GMPLS-oriented OAM
GMPLS-oriented Mobile IP-Networks
GMPLS-oriented Wireless Handoff
GMPLS-oriented All-Optical IP-Networks
GMPLS-oriented Scalability
GMPLS-oriented New Issues

Schedule:

Submission Deadline:	November 1, 2002
Acceptance Notification:	January 15, 2003
Final Manuscript Due:	March 31, 2003
Publication Date:	June 1, 2003

Submission:

Please submit your complete paper in a .doc or .pdf file to our Editor-in-Chief, gskuo@ieee.org for paper review.