

Architectures for Multi-Gigabit Wire-Linked Clock and Data Recovery



© PHOTO F/X 2

Ming-ta Hsieh and Gerald E. Sobelman

Abstract

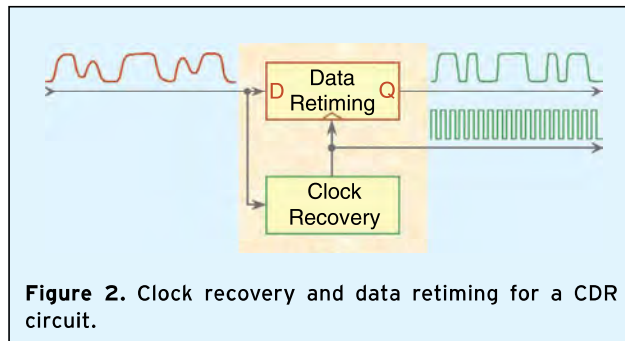
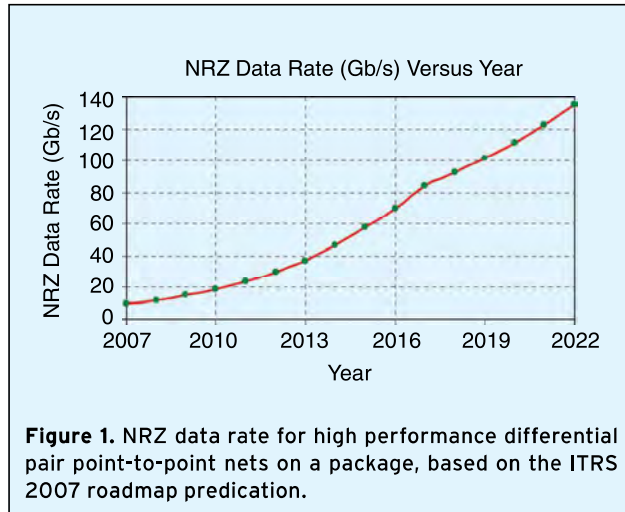
Clock and data recovery (CDR) architectures used in high-speed wire-linked communication receivers are often shown as PLL or DLL based topologies. However, there are many other types of CDR architectures such as phase-interpolator, oversampling and injection locked based topologies. The best choice for the CDR topology will depend on the application and the specification requirements. This paper presents an overview and comparative study of the most commonly used CDR architectures. This analysis includes the circuit structures, design challenges, major performance limitations and primary applications. Finally, the tradeoffs among the various CDR architectures are summarized.

I. Introduction

Data bandwidth for state of the art wire-linked communication systems is growing at an extremely fast rate. In 2007, the International Technology Roadmap for Semiconductors (ITRS) predicted that the non-return to zero (NRZ) data rate for high-performance differential pair point-to-point nets on a package would reach 100 gigabits per second (Gbps) by the year 2019, as shown in Figure 1 [1]. The data in such high-speed wire-linked communication systems often become severely distorted by both external and internal noise during transmission, which leads to jitter and skew in the received data.

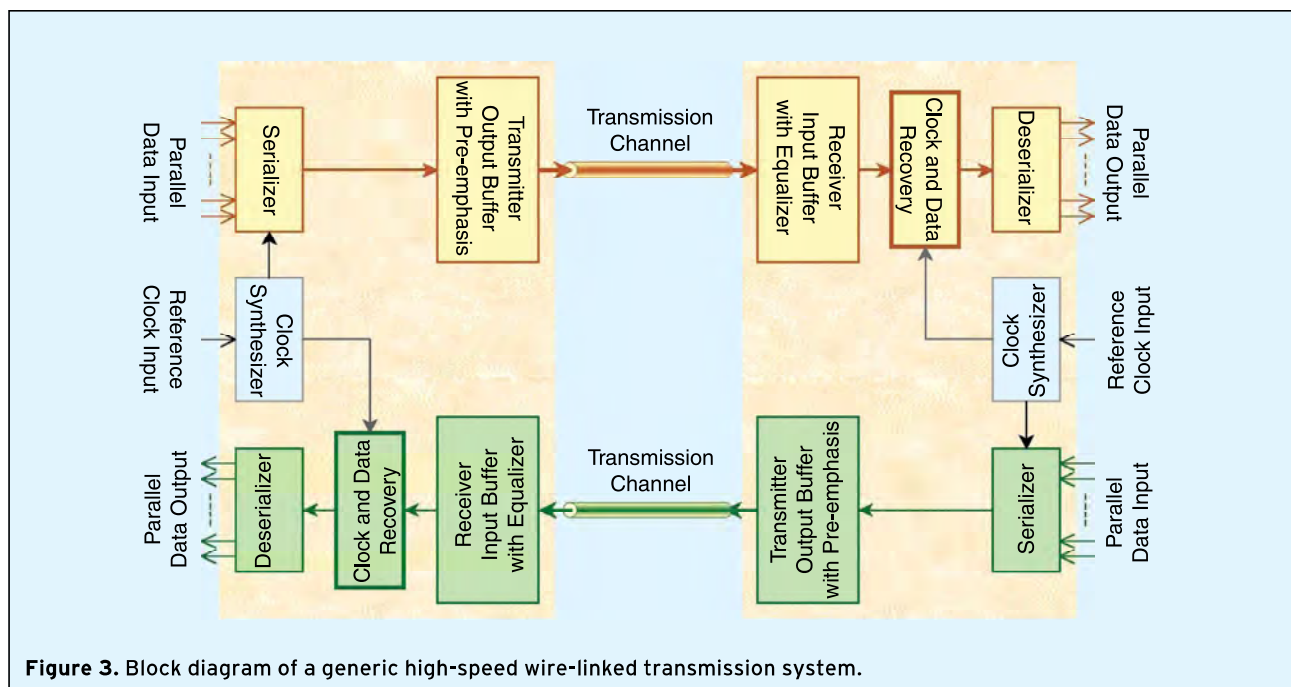
Digital Object Identifier 10.1109/MCAS.2008.930152

A clock and data recovery (CDR) circuit is an essential block in many high-speed wire-linked data



transmission applications such as optical communications systems, backplane data-link routing and chip-to-chip interconnection. The important role of a CDR is to extract the transmitted data sequence from the distorted received signal and to recover the associated clock timing information. Figure 2 illustrates a simplified functional diagram of clock recovery and data retiming using a CDR circuit. The clock recovery circuit detects the transitions in the received data and generates a periodic clock. The decision circuit often uses D-type Flip-Flops (DFFs) driven by the recovered clock to retime the received data, which samples noisy data and then regenerates it with less jitter and skew [2].

A generic block diagram of a high-speed wire-linked data transmission system is shown in Figure 3, where the received data is equalized in the receiver input buffer and retimed in the CDR module before proceeding into the deserializer module. A source-asynchronous system is shown, in which the transmitting and receiving sides use different clock sources. This results in a possible a frequency offset between the transmitted data and the local clock on the receiver side due to natural device mismatches, creating additional challenges for the CDR circuit. Most wire-linked communication systems fall into this category. In contrast to this, data transmission systems such as chip-to-chip interconnect in which both the transmitter and receiver use the same clock source are known



M.-t. Hsieh and G.E. Sobelman are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 (E-mail: sobelman@umn.edu).

as source-synchronous systems. A CDR for this type of system only needs to provide a finite phase capturing range.

The clock synthesizer in Figure 3 may also drive multiple transmitters and receivers (TXs/RXs), which is known as a multi-channel configuration. Having multiple TXs/RXs use the same clock synthesizer reduces the area and power overhead.

Many researchers have proposed a wide variety of CDR designs for high-speed wire-linked data transmission applications, such those based on an analog phase locked loop (APLL) [3]–[6], a digital phase locked loop (DPLL) [7], [8], a delay locked loop (DLL) [9], [10], a phase interpolator [11]–[13], injection locking [14], [15], oversampling [16]–[19], a gated oscillator [20]–[22], and a high-Q bandpass filter [23]–[25]. The goal of this paper is to provide a comprehensive overview and comparative performance analysis for all of these types of multi-gigabit rate CDRs.

The remainder of this paper is organized as follows: Section II provides an overview of CDR architectures which are commonly used in modern high-speed wire-linked data transmission and discusses the design challenges and other considerations for each type of CDR. Section III gives the performance tradeoffs among these architectures. The paper concludes with a summary of appropriate CDR architectures for a wide range of applications having various performance requirements.

II. CDR Architectures

CDR architectures can be classified according to the phase relationship between the received input data and the local clock at the receiver. Commonly used CDR topologies may be divided into three major categories:

1. Topologies using feedback phase tracking, including, phase locked loop (PLL), delay locked loop (DLL), phase interpolator (PI) and injection locked (IL) structures.
2. An oversampling-based topology without feedback phase tracking.
3. Topologies using phase alignment but without feedback phase tracking, including gated oscillator and high-Q bandpass filter architectures.

In the following subsections, we will present the structure, operation, advantages and design challenges for each of these types of CDR architectures.

A. PLL-based CDR

CDR designs based on a PLL topology can be categorized according to whether or not they utilize a reference clock. They can be further categorized as analog or digital PLL-based CDR designs. PLL-based CDR designs inherently provide a tunable bit rate and are easily

integrated in a monolithic design. However, a frequency acquisition aid is typically required in order to prevent false locking.

1) PLL based CDR Designs without Reference Clock

Figure 4(a) shows an architecture without a reference clock [5], where a frequency tracking loop provides a frequency comparison through the frequency detector (FD) and a phase tracking loop leads to phase locking through the phase detector (PD). The FD module provides a frequency comparison between the input data, $D(in)$, and the voltage-controlled oscillator (VCO) output clock which eliminates the need for using an external reference frequency. During either CDR startup or loss of phase lock, the FD is activated to produce a control voltage through the charge pump (CP) and the loop filter (LF), which moves the VCO oscillation frequency toward the input data rate. Once the frequency difference falls within the phase tracking loop's capture range, the PD takes over and allows the VCO output clock phase to lock onto the input data phase.

There are two possible issues associated with the CDR architecture of Figure 4(a). First, the frequency tracking

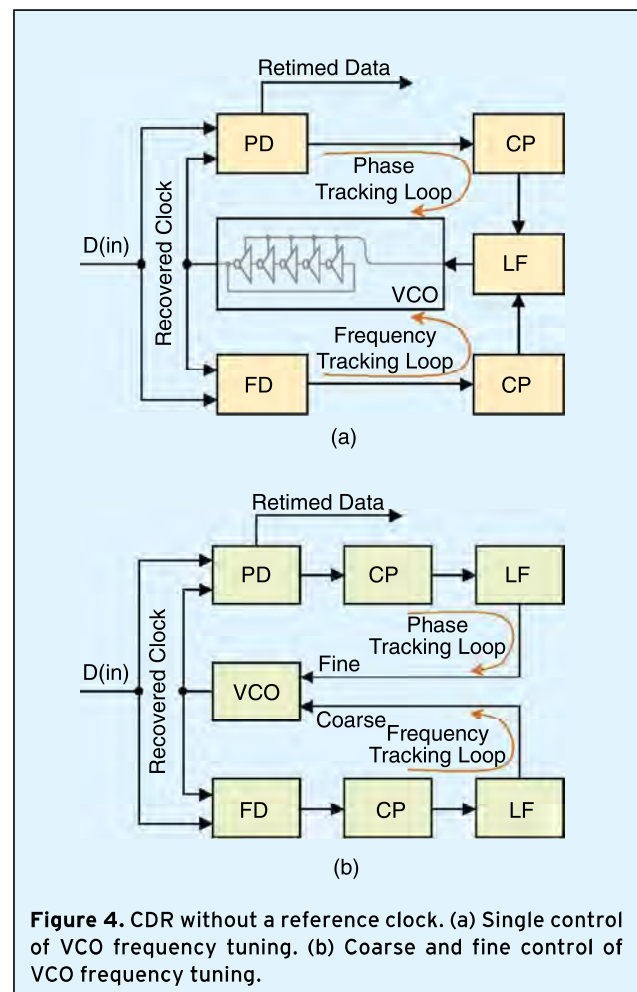


Figure 4. CDR without a reference clock. (a) Single control of VCO frequency tuning. (b) Coarse and fine control of VCO frequency tuning.

loop and the phase tracking loop may potentially interfere with each other during the interval when the FD transfers control to the PD, resulting in a failure to lock onto the phase [3] and/or ripple generation on the VCO control line. Second, the FD could become momentarily “confused” about the actual input data rate if the received input data consists of random consecutive identical digits (CIDs) [3] or if the received rising and falling edges are corrupted by external or internal noise during the transmission. Because of these issues, the loop bandwidth of the frequency tracking loop is typically chosen to be much smaller than that of the phase tracking loop [3].

In order to independently select the bandwidths of the frequency locking loop and the phase locking loop, one can modify the system such that each loop not only has its own charge-pump (CP) but also its own loop fil-

ter (LF). This is illustrated in Figure 4(b) [26], in which the frequency loop and phase loop drive the coarse control and fine control, respectively. However, this has the disadvantage of requiring a larger total layout area due to the presence of two LFs. Reference [8] has suggested using a hybrid analog/digital loop filter in order to reduce this area overhead.

2) PLL-based CDR with an External Reference Clock

An example of a PLL-based CDR design with an external reference clock input is shown in Figure 5(a) [6], which uses a similar scheme of coarse and fine tracking loops. The frequency tracking loop with the phase-frequency detector (PFD) locks the output clock phase of VCO_2 to that of the input reference clock, $F(\text{ref})$. This is a completely stand-alone clock multiplication with VCO_2 acting as a replica circuit of VCO_1 . The presence of the divide by M block in the frequency tracking loop allows the input reference clock to run at a low frequency. Since VCO_1 and VCO_2 are identical, the control voltage to VCO_2 can be used as a coarse control input to VCO_1 in such a way as to move the oscillation frequency of VCO_1 very close to or equal to the input data rate. Therefore, the frequency tracking loop provides a coarse control signal to VCO_1 . The phase tracking loop with the PD locks the VCO_1 output clock phase to the input data to create a fine control signal for VCO_1 . The gain of the phase tracking loop must be relatively low compared with that of the frequency tracking loop in order to maintain the fine control of VCO_1 .

There are two potential issues associated with the CDR architecture shown in Figure 5(a). First, any mismatch between VCO_1 and VCO_2 could lead to a difference in oscillation frequencies even though the two VCOs share the same coarse input [3]. Second, the data rate of a high-speed serial link in an asynchronous mode of operation will often allow a certain frequency offset between the transmitted data rate and the receiver’s local clock frequency, which leads to a frequency offset between VCO_1 and VCO_2 . A possible frequency pulling phenomena could move VCO_1 away from the received data rate and towards $M \times F(\text{ref})$. This could be especially problematic when a spread spectrum clocking (SSC) scheme is required, such as in Serial AT Attachment (SATA) applications [13]. Another general concern regarding the CDR architecture shown in Figure 5(a) is the excessive layout area needed for the two VCO designs, especially in the case of using an LC VCO based PLL for clock generation [1], [2]. However, the impact of the extra area for VCO_2 is less of a concern if the design is targeted for multi-channel applications since the frequency tracking loop is shared by multiple phase tracking loops.

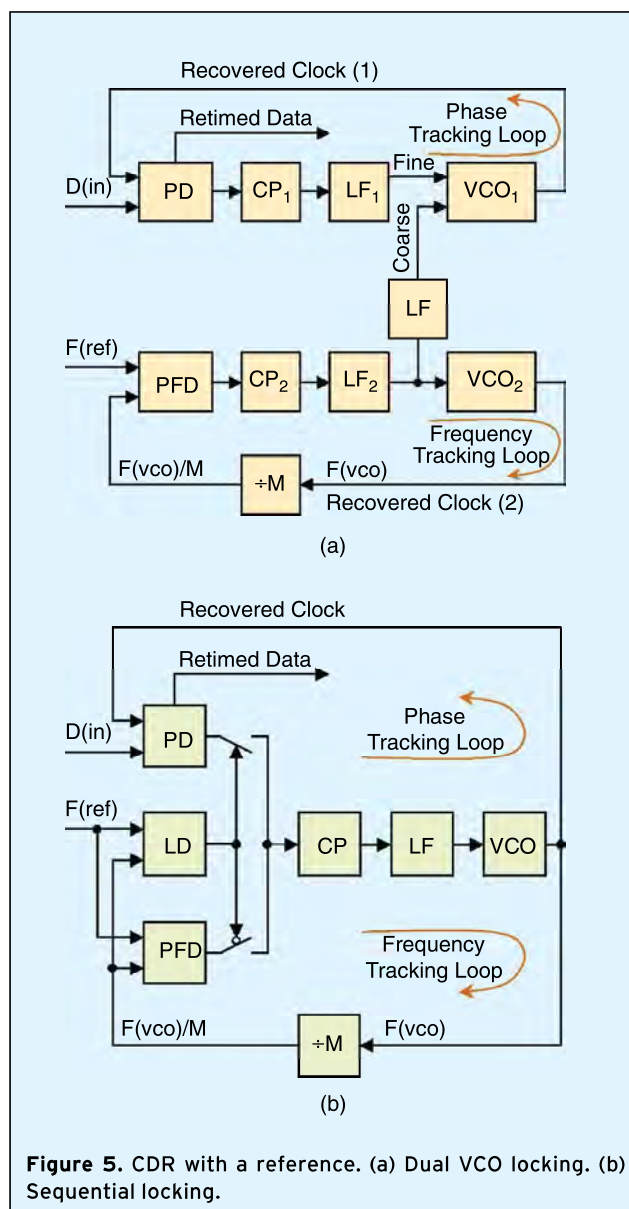


Figure 5. CDR with a reference. (a) Dual VCO locking. (b) Sequential locking.

On the other hand, having an independent clock multiplication from VCO_2 in Figure 5(a) makes it easier to satisfy the loop stability and bandwidth requirements. Furthermore, the availability of the coarse control signal from VCO_2 provides a large improvement in the acquisition time of the phase tracking loop.

One way to reduce the CDR design size in Figure 5(a) is through the sequential locking scheme of Figure 5(b). This uses a lock detector (LD) to sequentially enable the frequency loop and the phase loop, which eliminates the need for dual CPs, LFs and VCOs [27], [28]. During CDR startup, the LD first activates the frequency loop, and moves the VCO oscillation frequency towards $M \times F(\text{ref})$. Once the LD detects that the frequency of $F(\text{voc}) \div M$ is equal to $F(\text{ref})$, it disables the frequency tracking loop and enables the phase tracking loop. If loss of phase locking occurs as a result of unexpected noise, the LD re-activates the frequency tracking loop and the phase tracking loop sequentially [2], [3]. One potential issue with Figure 5(b) is that the transition from the frequency tracking loop to the phase tracking loop may disturb the VCO control signal and cause a VCO frequency shift when the FD transfers control to the PD. This, in turn, may result in a failure to lock phase due to a metastable transition and/or a large ripple on the VCO control signal, similar to the situation described earlier for the PLL-based CDR design without a reference clock [3].

3) Digital PLL (DPLL) based CDR Designs

CDR architectures in which the CP and LF are replaced by digital logic can minimize the required layout area and simplify the closed-loop stability analysis by minimizing the process, voltage and temperature (PVT) variations of the LF. Figure 6(a) shows an example of a partially DPLL-based CDR design [29], which implements the CP and LF in the form of a digital LF (DLF) but which includes digital-to-analog converters (DACs) in both the frequency and phase tracking loops. This CDR architecture is similar to the one shown on Figure 4(a). Another significant advantage of using the DLF is that it allows the CP and LF functions to be easily programmable.

There are two important issues associated with this CDR design. First, the potentially long loop latency from the DLF and the DAC may degrade the phase and frequency tracking capability, especially in an SSC operational mode which has reduced CDR jitter tolerance [13]. Second, the finite resolution of the DAC causes VCO frequency wandering between adjacent frequency steps, which increases jitter generation. The two DACs in Figure 6(a) can be eliminated by designing a VCO having digital switches to fine-tune the VCO frequency [8]. However, the issues of long loop latency and finite resolution still remain.

The DPLL architecture shown in Figure 6(b) utilizes a digital-to-multi-phase converter (DMPC) [7], generating m clock phases which are fed back to the PD array. The PD array consists of multiple bang-bang phase detectors which use the multi-phase clocks from the DMPC to sample multiple data bits. The data sampling process produces multiple early or late indication signals of phase error for data transitions and neutrals for non-transitions. The decimation block reduces the multiple early/late/neutral signals to an effective early, late, or neutral signal at a lower rate. The CDR design of Figure 6(b) faces the same issues as the one in Figure 6(a) with a potential long loop latency from the decimation in the DLF, and finite resolution from the DMPC. The DPLL-based CDR architecture of Figure 6(b) is generally used in low to moderate rate applications, but provides a programmable, all-digital design that is easily transferable between different process technologies.

B. DLL-based CDR

The DLL-based CDR architecture shown in Figure 7 [9] often shares a common PLL-based reference clock generator among multiple channels. This structure avoids

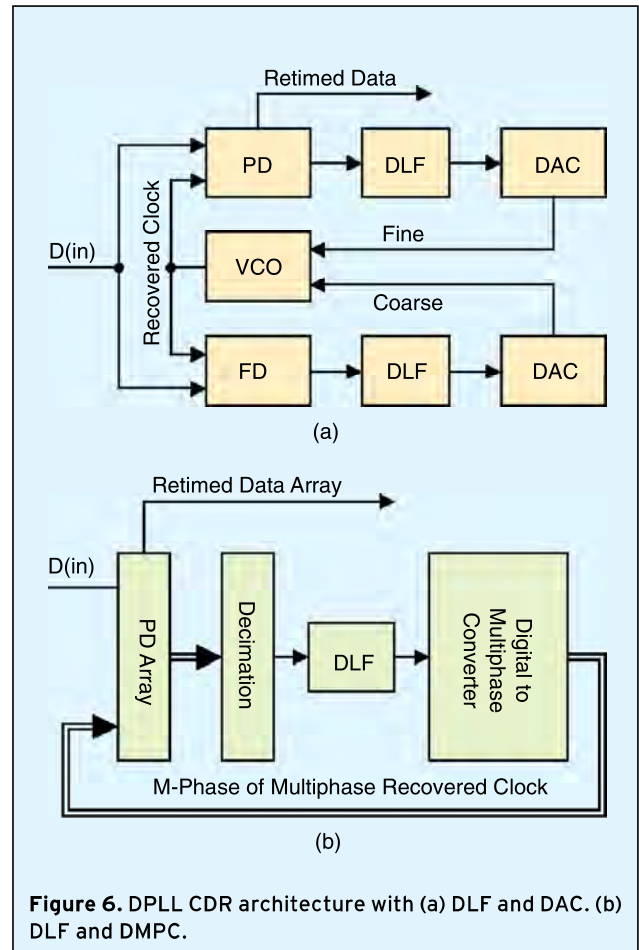


Figure 6. DPLL CDR architecture with (a) DLF and DAC. (b) DLF and DMPC.

the drawbacks of multi-VCO coupling/pulling, high power dissipation and large area. The DLL-based CDR design shown in Figure 7 is similar to the PLL-based CDR design of Figure 5(a). Here, however, the frequency tracking loop provides a reference clock rather than a control voltage signal. Also, the phase tracking loop uses a voltage-controlled delay line (VCDL) for phase synchronization instead of a VCO. The reference clock for the VCDL, $F(vco)$, must oscillate at the input data rate and is typically generated from a shared PLL-based clock multiplication which provides a low-pass filtering of the input reference clock, $F(ref)$, in order to reduce jitter transferred from $F(ref)$ [2], [3].

The primary benefit of using a DLL-based CDR is that it does not have the jitter accumulation issue [30] of a PLL-based CDR design [2]. Also, a DLL-based CDR provides a more stable system [30]. The VCDL control voltage directly alters the clock phase, whereas the VCO control voltage indirectly alters the clock phase through the integral of the dynamically changing clock frequency. Therefore, the VCDL does not introduce a pole in the loop transfer function. Furthermore, a DLL-based CDR design provides faster lock speed because there is no need for clock synthesis [9].

The primary drawback of the DLL-based CDR topology shown in Figure 7 is its limited phase capturing range, so that it is unable to handle any frequency offset between the transmitter and receiver. Therefore, the DLL-based CDR architecture shown in Figure 7 is most suitable for source-synchronous applications such as chip-to-chip interconnections [10].

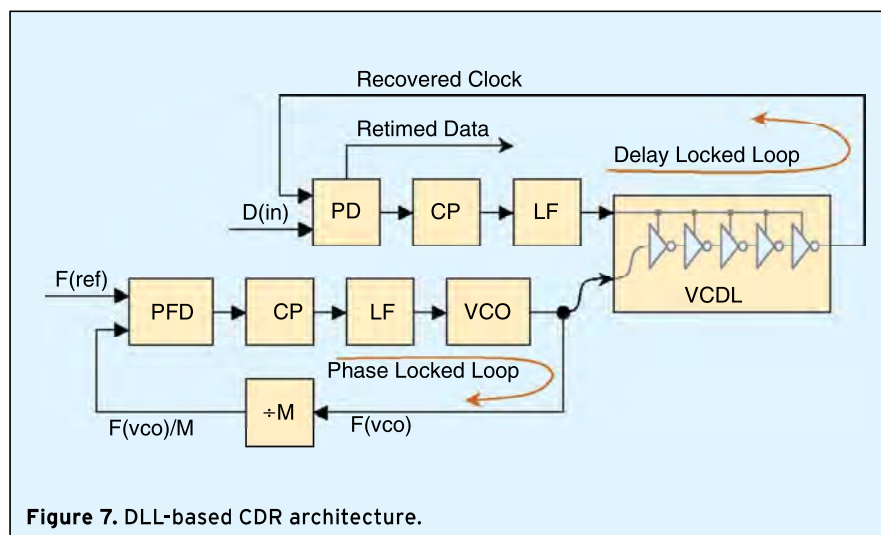
C. Combination of PLL/DLL based CDR

The dual loop CDR topology shown in Figure 5 can have good input jitter rejection as a result of a narrow loop bandwidth in the phase tracking loop while also having

a short acquisition time due to its frequency tracking loop. However, a PLL-based CDR topology with a second- or higher-order closed-loop frequency response often needs a closed-loop zero to stabilize the loop, which causes the PLL to exhibit jitter peaking in its input-to-output transfer function. This jitter peaking behavior is very undesirable, especially in an application such as SONET (Synchronous Optical Network) which cascades several CDRs as a string of repeaters, leading to the accumulation of jitter. Reducing the PLL loop bandwidth can also minimize jitter peaking but with an increase in the acquisition time.

One way to eliminate jitter peaking and allow the PLL to maintain a small loop bandwidth without compromising acquisition speed is to combine the DLL-based and PLL-based CDR architectures, as shown in Figure 8(a) [30]. Here, a requirement is that the PLL should not provide a closed-loop zero, which is accomplished by modifying the loop filter so that it only uses a capacitor [30].

The primary concern with the CDR topology of Figure 8(a) is that the loop can become unstable if the VCDL is driven to the edge of its delay range. This is due to the fact that both the DLL and the PLL share the same control voltage. The stabilizing zero for the PLL provided by the DLL of Figure 8(a) is no longer present once the DLL is driven to its delay range limit and acts as an open loop response. One way to eliminate this potential problem is to constrain the VCO tuning range to be a subset of the VCDL tuning range. Furthermore, both the VCO and the VCDL must be driven in the same phase direction. Figure 8(b) [31] shows an alternative design having all of the benefits from the design of Figure 8(a) together with independent tracking loops. Here, the DLL loop dynamics do not affect the PLL performance, at a cost of requiring dual CPs and LFs.



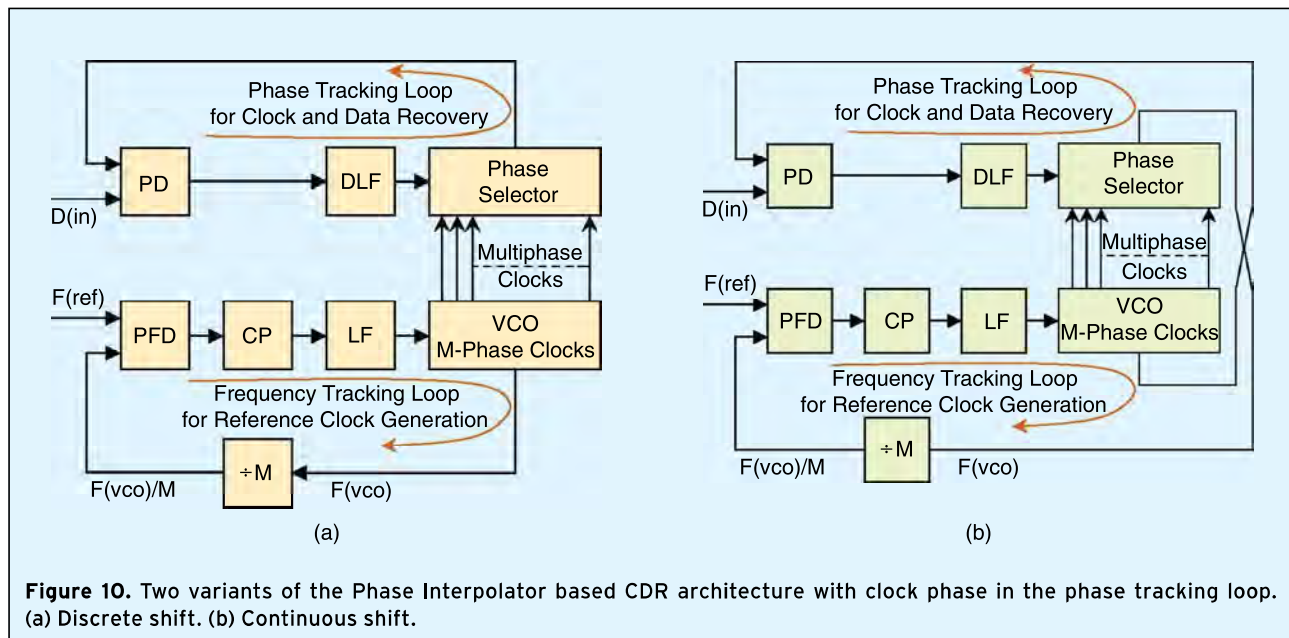
D. Phase Interpolator (PI) based CDR

The topology and operating mechanism of the Phase Interpolator (PI) based CDR architecture shown in Figure 9 [11]–[13] are similar to those of the DLL-based CDR design of Figure 7. In this structure, however, the CP and LF are replaced by a digital LF (DLF) and a current digital-to-analog converter (I.DAC), and the VCDL is replaced by a PI. The recovered clock phase from the PI is driven directly by the I.DAC using a function proportional to the control voltage. Both

issue with the design of Figure 10(a) is that the discrete clock phase shift step leads to larger cycle-to-cycle jitter. However, the smaller phase spacing produced by the VCO leads to lower VCO frequency, higher power

The diagram illustrates a Phase-Locked Loop (PLL) system. It consists of four main blocks: PD (Phase Detector), CP (Charge Pump), LF (Loop Filter), and VCO (Voltage-Controlled Oscillator). The PD block receives an input signal and a feedback signal. Its output goes to the CP block, which then feeds into the LF block. The LF block's output is connected to the VCO block. The VCO block's output is fed back to the PD block, forming a closed loop. Additionally, the output of the VCO block is also fed back to the PD block through a path labeled "Recovered Clock". The output of the PD block is also labeled "Retimed Data".

[illegible][illegible]



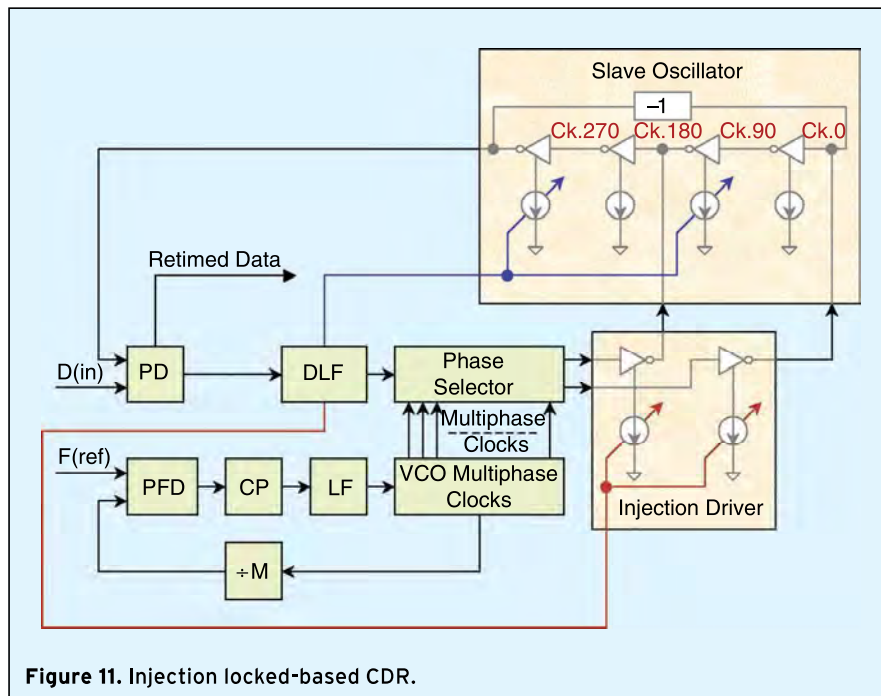
dissipation and a larger area to accommodate the increased number of clock phases. One way to smooth out the discrete phase shift step in Figure 10(a) is to swap the phase select and VCO output connections, as shown in Figure 10(b). The discrete change in phase selection from the DLF in the phase tracking loop is smoothed out by the LF and CP in the frequency tracking loop, which provides smooth frequency and phase drifting in the phase tracking loop. The major advantage of Figure 10(b) is that the loop bandwidths can be selected separately. However, it will not be able to support a multi-

channel application with a single frequency tracking loop for reference clock generation.

E. Injection Locked based CDR

The Injection Locked (IL) based CDR architecture shown in Figure 11 [14] is also a variant of the phase interpolator based CDR topology of Figure 9 and shares the same advantages of being a more stable system, having a faster acquisition time and an absence of jitter peaking, as compared with PLL-based CDRs. The phase selector, slave oscillator and injection driver in Figure 11 perform the

operations of the LDAC and PI of Figure 9. Here, the slave oscillator is locked by the frequency and phase injection from the injection driver. However, the slave oscillator acts like a low-pass filter and smoothes out duty-cycle distortion from the phase selector. This means that the recovered clock exhibits a much smoother phase shift compared to the phase interpolator based CDR design of Figure 9. Under the proper injection locked condition, the two clocks from the phase selector must be 180 degrees out of phase in order to maintain balanced injection into the differential slave oscillator. Furthermore, the adjustable current gain in the slave oscillator must be reduced during the activation of the injection driver



such that equal clock phase separation in the slave oscillator is maintained [14].

The injection locked based CDR design can exhibit a better duty-cycle balanced recovered clock and improved phase tracking jitter generation compared to the traditional phase interpolator based CDR design. (Phase tracking jitter is long-term jitter with respect to the ideal recovered clock.) However, these improvements trade off against the slave oscillator's lock range. Furthermore, a careful design and layout of the injection driver and the slave oscillator are needed in order to prevent any unwanted injection from such sources as the supply, substrate or any adjacent toggling signals.

F. Oversampling based CDR

A CDR design based on the oversampling architecture, as shown in Figure 12 [16], [17], provides data recovery without any time delay. Unlike the phase tracking based CDR design with its continuous adjustment of the recovered clock phase to track the received data phase, the oversampling based CDR circuit samples each received data bit at multiple points.

A minimum of 3 samples per received data bit are required for properly recovering the received data, as shown in Figure 12(b). The data recovery block in Figure 12(a) consists of a data register, bit boundary detector and data selector. The data register is a first-in first-out (FIFO) buffer which temporarily stores the sampled data from the multi-phase sampler while the data selector determines which ones will be retained. The bit boundary detector defines the data bit edge samples which allows the data selector to determine the proper data sample to retain as the recovered data.

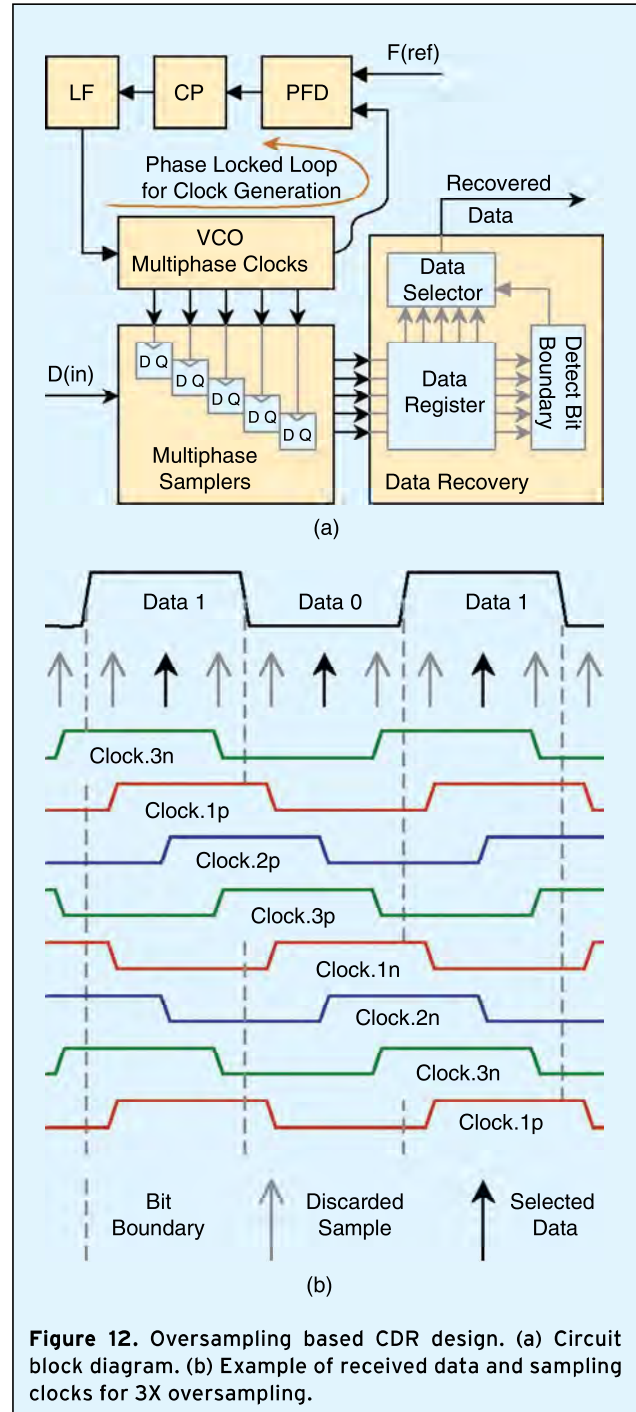
The primary advantages of the oversampling CDR design are its fast acquisition time and inherent stability. Furthermore, the feed-forward operation mechanism provides a very high data bandwidth. The oversampling CDR design technique is applicable in both burst-mode and continuous-mode data transmission because of the absence of feedback phase tracking and jitter transfer accumulation.

The drawbacks of an oversampling-based CDR are the need for high frequency data transitions to achieve high-frequency jitter rejection and the requirement of a large FIFO for sampled data storage, especially in a high-speed source-asynchronous system having a frequency offset between and receiver.

G. Gated Oscillator based CDR

Some applications such as passive optical networks (PONs) and optical packet routing systems impose no restrictions on the amount of jitter transfer but require a burst-mode operation to extract a synchronous clock and recover the received data immediately for each

asynchronous packet [20], [33]. The gated oscillator architecture shown in Figure 13 [20]–[22] is commonly used for such applications. The synchronous clock is derived from the gated oscillator which is triggered from the pulse generated in the edge detector and which follows the data transient edges. The frequency tuning for a gated oscillator is controlled through a replica gated oscillator from a PLL with its gated input tied to logic high. The variable delay buffer in the edge detector provides a data phase shift for the edge detector to determine the



data transition edge. It also allows the received data to be phase aligned with the recovered clock.

In addition to its fast synchronous clock recovery and data acquisition, the gated oscillator based CDR design is also a simpler and smaller design having lower power for multi-channel operation compared to oversampling based CDR designs.

Its major drawback is that it has no jitter rejection due to its broadband open loop design without loop bandwidth filtering. Furthermore, the phase alignment between the received data and the recovered clock is sensitive to process, temperature, data rate and supply voltage variations [4]. Finally, a gated oscillator based CDR design is more difficult to transfer from one process technology to another.

H. High-Q Filter based CDR

A simple open-loop based CDR design is shown in Figure 14 [4], [23], [25], [34], which uses a high quality factor (high-Q) bandpass filter to replace the gated

oscillator and PLL of Figure 13. The topology shown in Figure 14 is a technique that has been traditionally used in non-monolithic CDR designs [4]. The combination of the variable delay buffer and XOR logic gate in the edge detector operate as a pulse generator based on the received data transitions. The high-Q bandpass filter extracts the transition frequency, thereby recovering the clock at the received data rate.

In addition to being a simple design with a low cost of development, the high-Q bandpass filter has the same advantages of fast synchronous clock recovery and data acquisition as the gated oscillator based CDR. The filter designs are often based on an LC tank, surface acoustic wave (SAW) filter, dielectric resonator or PLL [4]. However, a PLL based filter [31] will not be able to perform an instantaneous clock extraction due to its long feedback phase tracking settling time. Both of the CDR implementations in [25], [34] use an off-chip SAW filter. The major limitation of this architecture is the difficulty of implementing a high-Q bandpass filter in a monolithic design. Also, there is no input jitter rejection and the clock-data phase alignment is sensitive to process, temperature, data rate and supply voltage variations as was the case for the gated oscillator based CDR designs.

III. Performance Comparison and Tradeoffs

CDR applications can be categorized as being either burst-mode or continuous-mode. A burst-mode system is often used in a point-to-multipoint application, where different senders transmit bursts of packet data with a silence time slot between bursts [36]. The data transmission link is re-activated whenever a packet of data is requested to be transmitted and remains inactive at other times in order to leave the data transmission link available for other users. Burst-mode data transmission often requires very fast acquisition time in order to meet the low network latency requirement which is usually within a few bytes of a preamble period [16]. Examples of burst-mode applications are the Fiber-To-The-Home (FTTH) Network,

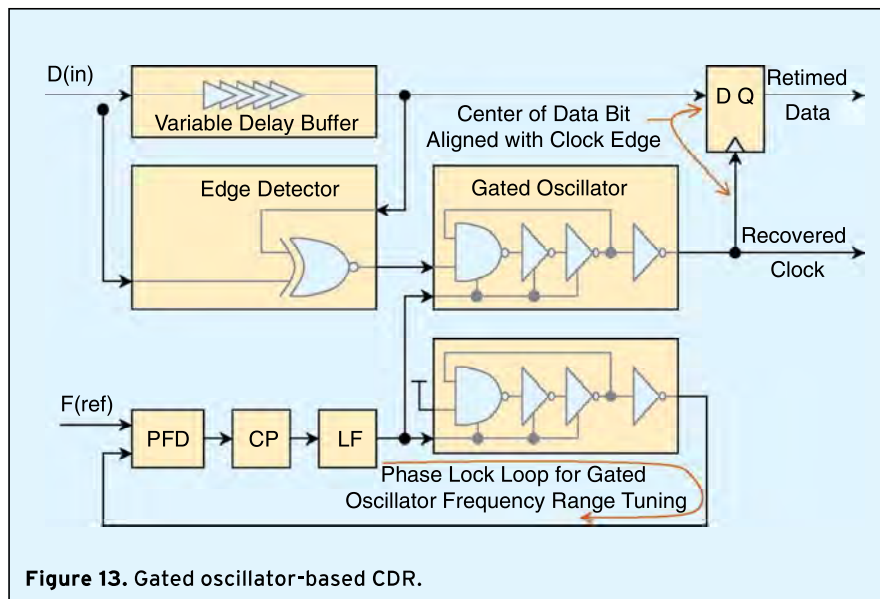


Figure 13. Gated oscillator-based CDR.

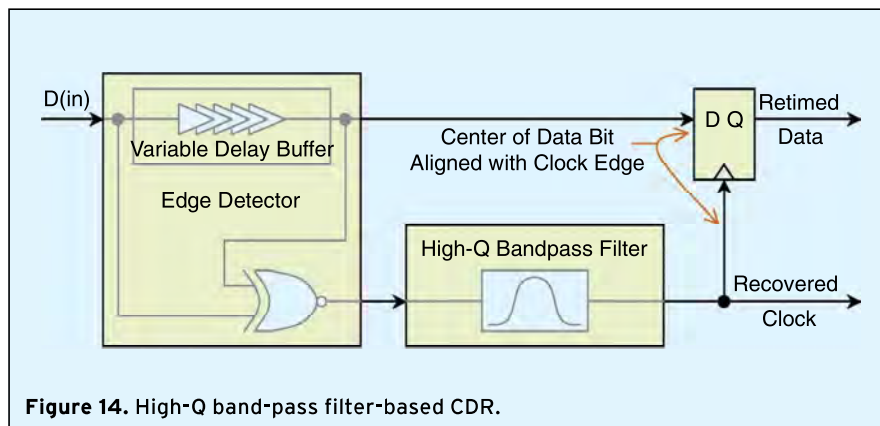


Figure 14. High-Q band-pass filter-based CDR.

Asynchronous Transfer Mode (ATM) Network, Ethernet Passive Optical Network (EPON), Gigabit Passive Optical Network (GPON) and Local Area Networks (LANs). The commonly used CDR architectures for burst-mode receivers are topologies without feedback phase tracking such as the gated oscillator and oversampling techniques in order to meet the low network latency requirement [22].

A continuous-mode system is often used in point-to-point applications, in which a steady and uninterrupted stream of bits is transmitted [36]. A fast acquisition time is often not required in such systems. However, some applications such as SONET have a stringent jitter transfer specification in order to avoid jitter accumulation from

repeaters, which requires the CDR to have very low or no jitter generation [33]. Furthermore, SONET applications must also tolerate a long sequence of consecutive identical digits (CIDs) [35], which leads to fewer transitions in the transmitted data pattern and provides less frequency content for retrieving the clock. In addition to SONET, other examples of continuous-mode applications are Fiber Channel and Gigabit Ethernet. The commonly used CDR architectures for continuous-mode receivers are PLL, DLL and combined PLL/DLL based topologies. Recently, the phase interpolator, injection locked and oversampling techniques have also been used in continuous-mode CDR designs.

Table 1.
CDR architecture comparison.

Architecture based on	Advantages	Disadvantages	Applications (Suggested/Reported)	References
PLL	Input Jitter Rejection Input Frequency Tracking Good for SSC Frequency Tracking	Jitter Peaking Large Loop Filter Area (Analog) Multichannel Crosstalk/Pulling Long Acquisition Time	Continuous-Mode Source-Asynchronous/Synchronous SONET/SDH/Gigabit Ethernet High Speed Serial Link	[6,8,27,28] [29]
DLL	Stable/First Order System No Jitter Peaking Multichannel Share Input Clocks	Source-Synchronous Only Large Loop Filter Area (Analog) Limited Phase Capturing Range	Continuous-Mode Source-Synchronous High Speed Serial Link Chip-to-Chip Interconnection	[9] [10]
PLL/DLL	No Jitter Peaking Small Loop BW/Fast Acquisition Track Frequency/Good for SSC	Multichannel Crosstalk/Pulling Require Analysis for Two Loops	Continuous-Mode Source-Asynchronous/Synchronous SONET/Ethernet/Fibre Channel Multi-Gbps Link/Optical Receiver	[30, 39] [31]
Phase Interpolator	Multichannel Share Input Clocks	Quantization Phase Error Multiphase Clock Routing Cycle-to-Cycle Jitter	Continuous-Mode Source-Asynchronous/Synchronous SONET/SDH/OIF CEI Serial AT Attachment (SATA)	[11, 12] [13]
Injection Locked	Good Jitter Tolerance Duty-Cycle Correction Multichannel Share Input Clocks	Quantization Phase Error Multiphase Clock Routing Cycle-to-Cycle-Jitter Large Oscillator Range, Large Jitter	Continuous-Mode Source-Asynchronous/Synchronous Cross-Point Switch/SONET Multi-Gbps Serial Link I/Os	[14] [15]
Oversampling	No Feedback Phase Tracking Fast Acquisition Easy Transferred in Technologies No Stability Concern	Digital Circuit Complexity Large FIFO Size Possible Long Data Latency Require Multiphase Clocks	Burst/Continuous-Mode Source-Asynchronous/Synchronous Serial AT Attachment (SATA) Fiber Channel/Backplane/PON	[18] [19, 37]
Gated VCO	No Feedback Phase Tracking Fast Acquisition Small-Area	Data/Clock Phase Aligning No Input Jitter Rejection Possible Multichannel Crosstalk / Pulling	Burst-Mode Source-Asynchronous/Synchronous Fiber Channel/PON Short-Haul Data Transmission	[20, 22, 33] [21]
High-Q Bandpass Filter	No Feedback Phase Tracking Fast Acquisition Low Power Fast Time/Low Cost Development	Data/Clock Phase Aligning Difficult to Design in Monolithic No Input Jitter Rejection	Burst-Mode Source-Asynchronous/Synchronous SONET/SDH Fiber-to-the-Desk (FTTD)/LAN	[25] [34, 38]

PLL-based CDR designs have very good input jitter rejection but suffer from jitter peaking and stability concerns. On the other hand, a DLL-based CDR topology has no jitter peaking or stability concerns. Furthermore, it is well suited to multi-channel applications due to the lack of crosstalk injection or frequency pulling among VCOs. However, it is generally restricted to source-synchronous systems due to its limited phase capturing range. The combined PLL/DLL-based CDR architecture has the benefits of both of the PLL and DLL. However, its design complexity is larger because of the need to analyze the behaviors of two loops.

A phase interpolator based CDR design does not have jitter peaking or stability concerns and it has an unlimited phase capturing range but it suffers from quantization errors. The injection locked CDR design provides duty cycle correction but forces a tradeoff between its tracking jitter performance and the slave oscillator lock range.

The oversampling, gated oscillator and high-Q bandpass filter based CDR designs all provide a rapid data recovery capability. The oversampling-based CDR topology offers a complete digital design solution which is easily transferable between different process technologies. However, it has long data latency and it requires a large FIFO. The gated oscillator and high-Q bandpass filter based CDR designs provide rapid clock and data recovery but have no input jitter rejection and no intrinsically aligned clock-data phase for optimum data sampling points. The high-Q bandpass based CDR has the lowest design cycle time but it is difficult to integrate into a monolithic design.

A listing of the advantages and disadvantages, including suggested or reported applications for each type of CDR architecture, is given in Table 1. This table provides a comparison and tradeoff summary amongst all of these CDR topologies so that appropriate candidate architectures for a given application of interest can be determined.

IV. Conclusions

An overview of commonly used CDR architectures has been presented which discusses the applications and design challenges along with their advantages and limitations. PLL, DLL, Phase Interpolator and Injection Locked based CDR designs are suitable for continuous-mode communication. On the other hand, gated oscillator and high-Q bandpass filter based CDR designs are more applicable in burst mode systems. The oversampling based architecture is capable of handling both burst- and continuous-mode data. The DLL-based CDR is not applicable in a source-asynchronous system due to its limited phase capturing range. The strengths and weaknesses for each type of CDR design have been discussed in detail and a summary of the tradeoffs and applications for each type has been provided.

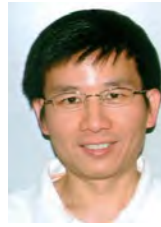
Acknowledgment

The authors would like to thank Toshiba America Electronic Components (TAEC), Inc. for their support.

References

- [1] ITRS, "International Technology Roadmap for Semiconductors 2007 Edition: Assembly and Packaging," *International Technology Roadmap for Semiconductors (ITRS)*, <http://www.itrs.net>, 2007.
- [2] B. Razavi, "Design of Integrated Circuits for Optical Communications," *McGraw-Hill*, New York, 2003.
- [3] B. Razavi, "Challenges in the Design of High-Speed Clock Data Recovery Circuit," *IEEE Communication Magazine*, pp. 94–101, Aug. 2002.
- [4] R. C. Walker, "Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems (Book style with paper title and editor)," *Phase-Locking in High-Performance Systems from Devices to Architectures*, B. Razavi, Ed. New Jersey: John Wiley & Sons, Inc., pp.34–45, 2003.
- [5] A. Pottbäcker, U. Langmann, and H.-U. Schreiber, "A Si Bipolar Phase and Frequency Detector IC for Clock Extraction Up to 8 Gb/s," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 12, pp. 1747–1751, Dec. 1992.
- [6] 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," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1935–1943, Dec. 2003.
- [7] J. L. Sonntag and J. Stonick, "A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1867–1875, Aug. 2006.
- [8] M.H. Perrott, Y. Huang, R.T. Baird, B.W. Garlepp, D. Pastorello, E.T. King, Q. Yu, D.B. Kasha, P. Steiner, L. Zhang, J. Hein, and B. Del Signore, "A 2.5-Gb/s Multi-Rate 0.25- μ m CMOS Clock and Data Recovery Circuit Utilizing a Hybrid Analog/Digital Loop Filter and All-Digital Referenceless Frequency Acquisition," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2930–2944, Dec. 2006.
- [9] X. Maillard, F. Devisch, and M. Kuijk, "A 900-Mb/s CMOS Data Recovery DLL Using Half-Frequency Clock," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 6, pp. 711–715, Dec. 2002.
- [10] H.-H. Chang, R.-J. Yang, and S.-I. Liu, "Low Jitter and Multirate Clock and Data Recovery Circuit Using a MSADLL for Chip-to-Chip Interconnection," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 51, no. 12, pp. 2356–2364, Dec. 2004.
- [11] R. Kreienkamp, U. Langmann, C. Zimmermann, T. Aoyama, and H. Siedhoff, "A 10-Gb/s CMOS Clock and Data Recovery Circuit with an Analog Phase Interpolator," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 736–743, Mar. 2005.
- [12] M.Y. He and J. Poulton, "A CMOS Mixed-Signal Clock and Data Recovery Circuit for OIF CEI-6G + Backplane Transceiver," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 597–606, Mar. 2006.
- [13] M. Hsieh and G.E. Sobelman, "Clock and Data Recovery with Adaptive Loop Gain for Spread Spectrum SerDes Applications," *IEEE International Symposium on Circuits and Systems*, pp. 4883–4886, May 2005.
- [14] H.-T. Ng, R. Farjad-Rad, M.-J. E. Lee, Member, W.J. Dally, T. Greer, J. Poulton, J.H. Edmondson, R. Rath, and R. Senthinathan, "A Second-Order Semidigital Clock Recovery Circuit Based on Injection Locking," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2101–2110, Dec. 2003.
- [15] R. Farjad-Rad, A. Nguyen, J. M. Tran, T. Greer, J. Poulton, W. J. Dally, J.H. Edmondson, R. Senthinathan, R. Rath, M.-J.E. Lee, and H.-T. Ng, "A 33-mW 8-Gb/s CMOS Clock Multiplier and CDR for Highly Integrated I/Os," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1553–1561, Sept. 2004.
- [16] J. Kim and D.-K. Jeong, "Multi-Gigabit-Rate Clock and Data Recovery Based on Blind Oversampling," *IEEE Communication Magazine*, pp. 68–74, Dec. 2003.
- [17] S.I. Ahmed and T.A. Kwasniewski, "Overview Of Oversampling Clock and Data Recovery Circuits" *Canadian Conference on Electrical and Computer Engineering*, pp. 1876–1881, May 1–4, 2005.
- [18] M. Ierssel, A. Sheikholeslami, H. Tamura, and W.W. Walker, "A 3.2 Gb/s CDR Using Semi-Blind Oversampling to Achieve High Jitter Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 10, pp. 2224–2234, Oct. 2007.
- [19] S.-H. Lee, M.-S. Hwang, Y. Choi, S. Kim, Y. Moon, B.-J. Lee, D.-K. Jeong, W. Kim, Y.-J. Park, and G. Ahn, "A 5-Gb/s 0.25- μ m CMOS Jitter-Tolerant Variable-Interval Oversampling Clock/Data Recovery Circuit," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1822–1830, Dec. 2002.

- [20] M. Nogawa, K. Nishimura, S. Kimura, T. Yoshida, T. Kawamura, M. Togashi, K. Kumozaki, and Y. Ohtomo, "A 10 Gb/s burst-mode CDR IC in 0.13 μm CMOS," *IEEE International Solid-State Circuits Conference*, Feb. 2005.
- [21] A. Tajalli, P. Muller, M. Atarodi, and Y. Leblebici, "A Multichannel 3.5mW/Gbps/Channel Gated Oscillator Based CDR in 0.18 μm Digital CMOS Technology," *Proceedings of the 31st European Solid-State Circuits Conference*, Sept. 2005, pp. 193–196.
- [22] A. Tajalli, P. Muller, and Y. Leblebici, "A Power-Efficient Clock and Data Recovery Circuit in 0.18 μm CMOS Technology for Multi-Channel Short-Haul Optical Data Communication," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 10, pp. 2235–2244, Oct. 2007.
- [23] R. Walker, "Clock and Data Recovery for Serial Digital Communications," *IEEE International Solid-State Circuits Conference—Short Course*, <http://www.omnisterra.com/walker/pdfs.talks/ISSCC2002.pdf>, Feb. 2002.
- [24] K. Ishii, K. Kishine, and H. Ichino, "A Jitter Suppression Technique for a 2.48832-Gb/s Clock and Data Recovery Circuit," *IEEE Transactions on Circuits and Systems II: Analog And Digital Signal Processing*, vol. 49, no. 4, pp. 266–273, Apr. 2002.
- [25] K. Yamashita, M. Nakata, N. Kamogawa, O. Yumoto, and H. Kodera, "Compact-Same-Size 52- and 156 Mb/s SDH Optical Transceiver Modules," *IEEE Journal of Lightwave Technology*, vol. 12, no. 9, pp. 1607–1615, Sept. 1994.
- [26] H. Wang and R. Nottenburg, "A 1 Gb/s CMOS Clock and Data Recovery Circuit," *IEEE International Solid-State Circuits Conference*, Feb. 1999.
- [27] L. Henrickson, D. Shen, U. Nellore, A. Ellis, J. Oh, H. Wang, G. Cpriglione, A. Atesoglu, A. Yang, P. Wu, S. Quadri, and D. Crosbie, "Low Power Fully Integrated 10-Gb/s SONET/SDH Transceiver in 0.13- μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 10, pp. 1595–1601, Oct. 2003.
- [28] H. S. Muthali, T.P. Thomas, and I.A. Young, "A CMOS 10-Gb/s SONET Transceiver," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1026–1033, Jul. 2004.
- [29] P. K. Hanumolu, M.G. Kim, G.-Y. Wei, and U.-K. Moon, "A 1.6 Gbps Digital Clock and Data Recovery Circuit," *IEEE Custom Integrated Circuits Conference*, Sept. 2006, pp. 603–606.
- [30] D. Dalton, K. Chai, E. Evans, M. Ferriss, D. Hitchcox, P. Murray, S. Selvanayagam, P. Shepherd, and L. DeVito, "12.5-Mb/s to 2.7-Gb/s Continuous-Rate CDR with Automatic Frequency Acquisition and Data-Rate Readback," *IEEE Journal of Solid-State Circuits*, vol. 40, no.12, pp. 2713–2725, Dec. 2005.
- [31] W. Rhee, H. Ainspan, S. Rylov, A. Rylyakov, M. Beakes, D. Friedman, S. Gowda, M. Soyuer, "A 10-Gb/s CMOS Clock and Data Recovery Circuit Using a Secondary Delay-Locked Loop," *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference*, Sept. 2003, pp. 81–84.
- [32] P. Larsson, "A 2-1600-MHz CMOS Clock Recovery PLL with Low-Vdd Capability," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1951–1999, Dec. 1999.
- [33] J. Lee and M. Liu, "A 20Gb/s Burst-Mode CDR Circuit Using Injection-Locking Technique," *IEEE International Solid-State Circuits Conference*, Feb. 2007.
- [34] I. Radovanovic, W. Etten, and H. Freriks, "Ethernet-Based Passive Optical Local-Area Networks for Fiber to the Desk Application," *IEEE Journal of Lightwave Technology*, vol. 21, no. 11, pp. 2534–2545, Nov. 2003.
- [35] B. Analui, and A. Hajimiri, "Instantaneous Clockless Data Recovery and Demultiplexing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 8, pp. 437–441, Aug. 2005.
- [36] K. Schneider and H. Zimmermann, "Highly Sensitive Optical Receivers," *Springer Berlin Heidelberg*, New York, 2006.
- [37] C.-H. Liang, S.-C. Hwu, and S.-I. Liu, "A 2.5 Gbps Burst-Mode Clock and Data Recovery Circuit," *IEEE Asian Solid-State Circuits Conference*, Nov. 2005, pp. 457–460.
- [38] P. Wallace, R. Bayruns, J. Smith, T. Laverick, and R. Shuster, "A GaAs 1.5Gb/s Clock Recovery and Data Retiming Circuit," *IEEE International Solid-State Circuits Conference*, Feb. 1990.
- [39] J. Kenney, D. Dalton, M. Eskiyeili, E. Evans, B. Hilton, D. Hitchcox, T. Kwok, D. Mulcahy, C. McQuilkin, V. Reddy, S. Selvanayagam, P. Shepherd, W. Titus, and L. DeVito, "A 9.95 to 11.1Gb/s XFP Transceiver in 0.13 μm CMOS," *IEEE International Solid-State Circuits Conference*, Feb. 2006.



Ming-ta Hsieh (S'97-M'07) received the B.S. and M.S. in electrical engineering from University of Minnesota, Minneapolis, in 1996 and 1998, respectively. He is currently working toward the Ph.D. in the electrical engineering at University of Minnesota, Minneapolis, MN. From Jan. 2001 to Aug. 2002, he was with Applied Micro Circuit Corporation (AMCC), Edina, MN, and involved with the fiber optical receiver and driver design. From Sept. 2002 to present, he is with Toshiba America Electronic Components, Inc. (TAEC), Bloomington, MN, and works in the area of Multi-Gbps SerDes physical layer design. His research interests are in the mixed-signal CMOS circuits systems for multi-Gbps wirelink applications.



Gerald E. Sobelman (M'81, SM'03) received a B.S. degree in physics from the University of California, Los Angeles. He was awarded M.S. and Ph.D. degrees in physics from Harvard University. He was a postdoctoral researcher at The Rockefeller University, and he has held senior engineering positions at Sperry Corporation and Control Data Corporation. He is currently a faculty member in the Department of Electrical and Computer Engineering at the University of Minnesota. He also serves as the Director of Graduate Studies for the Graduate Program in Computer Engineering at the University of Minnesota. Prof. Sobelman is a Senior Member of IEEE and serves on the technical program committees for IEEE ISCAS, IEEE SOCC and IEEE ICCSC. He is currently Chair of the Technical Committee on Circuits and Systems for Communications (CASCOM) of the IEEE Circuits and Systems Society. He has also served as an Associate Editor of IEEE Signal Processing Letters. He was Local Arrangements Chair for the 1993 IEEE International Conference on Acoustics, Speech and Signal Processing. In addition, he has chaired many sessions at international conferences in the areas of communications and VLSI design, and he is a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2008–2009. He has developed and presented short courses on digital VLSI design at several industrial sites. He has also given invited lectures at many universities, and he has been a consultant to a number of companies. His current research interests are in the areas of VLSI circuit and system design for applications in communications and signal processing. He has authored or co-authored more than 100 technical papers and 1 book, and he holds 11 U.S. patents.