



XAPP250 (v1.3.2) May 2, 2007

## Clock and Data Recovery with Coded Data Streams

Author: Leonard Dieguez

### Summary

This application note and reference design outline a method to implement clock and data recovery in Virtex™-II devices. Although not limiting the implementation to a specific FPGA family, this reference design focuses on the Virtex-II architecture. With minor modifications, Clock and Data Recovery (CDR) is possible with Virtex-E and Spartan™-IIE devices. A implementation of CDR at 270 Mb/s with 8B/10B coded data is described herein.

**Note:** Designs not requiring a recovered clock should refer to a specific Data Recovery application note [XAPP224](#).

### Introduction

This application note describes Clock and Data recovery methods allowing a receiver to extract embedded clock information and retimed data from an incoming encoded data stream. Sometimes, the received clock is also used for onward data transmission. The speed of operation is limited by the maximum input clock frequency from an external VCO and internal routing resources. The maximum operational speed is 420 MHz using a Virtex-II device (-5 speed grade). [Figure 1](#) shows a CDR primitive.

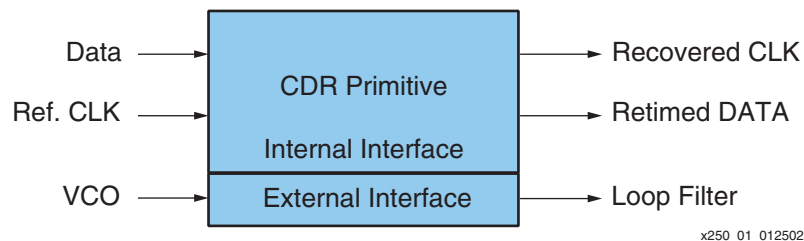


Figure 1: CDR Primitive

If a recovered clock is not required in a design, then XAPP224 (Data Recovery) is more appropriate. System requirements are as follows:

- Clock and data must be coded into a single data stream. This method is proven using 8B/10B coded data.
- A reference oscillator
- The availability of four FPGA pins in close proximity (two input LVDS pins and two output CMOS pins)
- An external Voltage Controlled Oscillator (VCO)

A block diagram describing the system is illustrated in [Figure 2](#). In practice, any VCO or VCXO can be used. As an example, a Maxim VCO was chosen. The Maxim 2605-2609 devices, operating in frequency ranges from 45 MHz to 650 MHz, come in a SOT-23 6-pin package. These integrated VCOs have tuning diodes and differential outputs. The only external circuit components are a tuning inductor, a bypass capacitor, and an output matching network. The goal is to have a trimless phase-locked loop (PLL) design operating across temperature and device variations. Working knowledge of PLL function, design, and implementation are important when using the accompanying reference design.

© 2002–2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

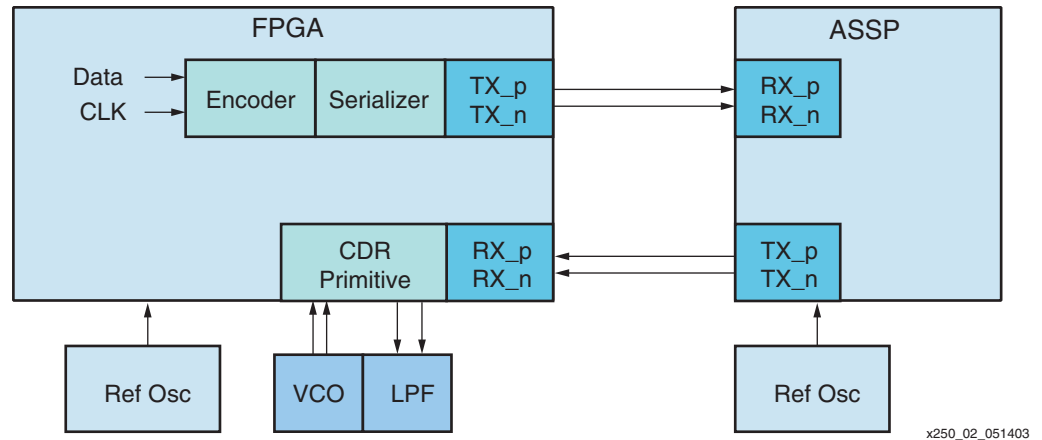


Figure 2: CDR In Virtex-II Devices

**CDR Function**

The clock and data recovery circuit shown in Figure 2 includes a delay-line phase detector, a standard phase and frequency detector (PFD), a VCO, a loop filter, and a control circuit.

The PFD is used to lock the VCO output signal to the reference clock. Normally the reference clock is a sub-multiple of the bit rate. Therefore, a divided down version of the VCO clock is used for the comparison.

Once the VCO is locked to the approximate bit-rate frequency as defined by the reference clock using a simple PFD (see “3-State Phase Detector”), the logic circuitry switches the VCO correction values to the inputs of the delay line phase – frequency detector (DLLPFD). The DLLPFD further adjusts the VCO to match the exact frequency of the received data rate.

**CDR Architecture**

The Figure 3 illustrates the typical block diagram for implementing a PLL. The basic blocks for a PLL are:

- Phase detector
- Loop filter
- Voltage controlled oscillator
- Counters

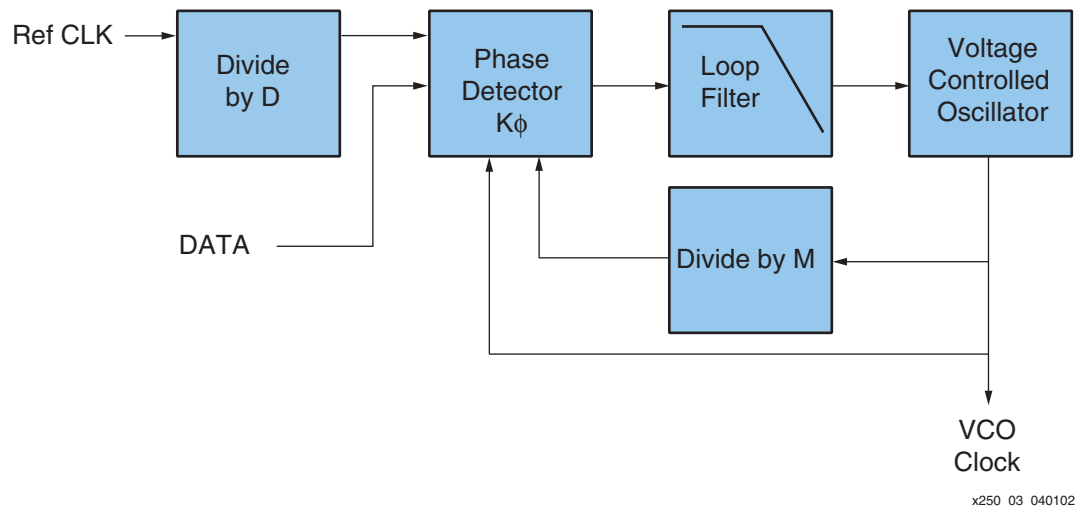


Figure 3: PLL Block Diagram

## Phase Detectors

A combination of a 3-state phase frequency detector and a delay line phase detector are used in this design for clock recovery and data synchronization.

There are two phase frequency detectors (PFDs) in the phase detector block. One PFD compares the phase and frequency of the VCO to the reference clock. Once the VCO is locked to the reference, the data input rate and the VCO rate should be within a small known difference (approximately 200 ppm from each other). The second PFD, the delay line PFD, further adjusts the VCO to lock to the data stream. Logic is used to determine when the VCO is locked to the reference and when data is present. If both conditions are met, then the second PFD is used to match the VCO to the data rate. The DL PFD is used to match the VCO to the data stream.

### 3-State Phase Detector

The 3-state phase detector is used because it is simple, has a linear range of  $\pm 2\pi$  radians (at the comparison frequency), and acts as both phase and frequency detector (PFD). A state diagram for the circuit is shown in [Figure 4](#).

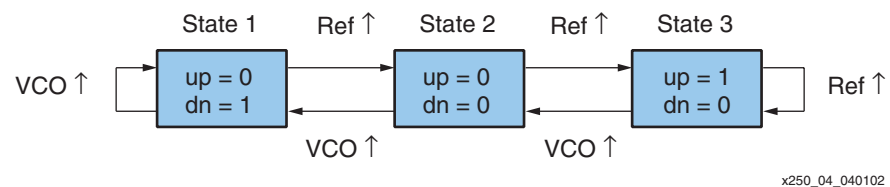


Figure 4: State Diagram for a 3-State PFD

One implementation of the 3-state PFD is shown in [Figure 5](#). State 2 is the assumed initial state (both up and down are zero). A rising edge on the reference causes "up" to be asserted, therefore, forcing a transition to the third state. Then when a rising edge of VCO occurs, both up and down are high for an instant. After the gate propagation delay, the registers are both reset. The PFD is now in state 2. The maximum useful frequency for this PFD is limited by the minimum duration of state 2. This PFD is not sensitive to the duty cycle of the VCO or the reference oscillator.

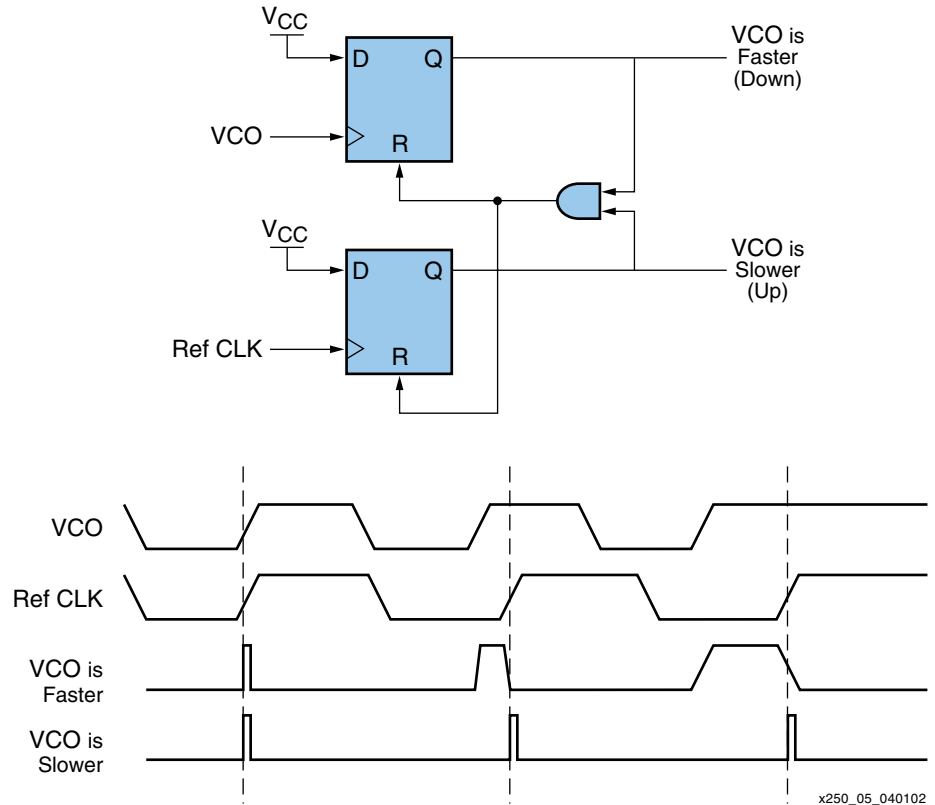


Figure 5: 3-State Phase-Frequency Detector

When the VCO is running at a faster frequency or is leading the reference frequency in phase and needs to be slowed down or delayed, is indicated by a signal "VCO is Faster". A "VCO is Slower" signal indicates a slower VCO than the reference frequency. The phase detector gain is the following:

$$K_D = \frac{V_H - V_L}{2\pi}$$

**Delay Line Phase Detector**

The delay line phase detector is composed of three main blocks:

- Delay line
- Registers on the outputs of the delay line
- Phase encoder.

The two basic techniques for detecting the phase between the VCO output and the input data are:

1. Pass the data through a delay line. Then sample the delayed elements with the VCO.
2. Pass the clock through a delay line. Then sample the delayed elements with the DATA from the IOB.

The application note covers the first method. The sampled delay line is shown in [Figure 6](#).

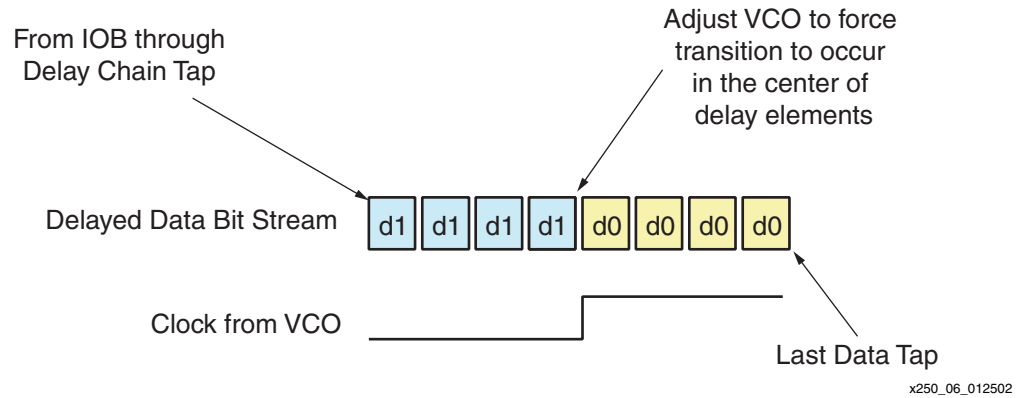


Figure 6: Sampled Delay Line

After the PLL has achieved lock to the reference oscillator, control circuitry attempts to recover the clock from coded data. Although other methods can be used, this application note uses 8B/10B encoded data streams. 8B/10B coded data has an NRZ format (i.e., there may be several bit times with no transitions). With 8B/10B coded data, the maximum number of bit times without transitions (maximum run length) is limited to five. The data stream is described as a reference clock with missing transitions (a punched clock). A data stream that only transitions once during a bit time is a single data rate (SDR) signal. This makes the data stream appear as a half-rate clock when an alternating one/zero pattern is present.

General architecture used for the delay line phase detector is depicted in [Figure 7](#). The delay line phase detector is similar to a quadrature phase detector because it outputs a frequency error by detecting phase slippage between the reference clock and the data. The delay line phase detector separates the phase differences in discrete intervals. In a Virtex-II or Virtex-II Pro design, a "MUXCY" carry chain is utilized for the discrete delay elements.

The phase detector provides the following functions.

- Detects if a data transition occurred within the current sample period.
- Produces a phase error output.
- Produces the retimed DATA output.

The delay line should have a delay of greater than one period, but not greater than two periods. The phase detector adjusts the VCO causing it to align the data transition to the center of the delay line. Outputs from the delay line are registered twice to remove any metastability.

The MUXCY carry chain is used for a delay line in this application note. Entry to the MUXCY carry chain is at the bottom of the die. Therefore the data input should be placed at the bottom or near the bottom to keep routing from the pin to the carry chain to a minimum.

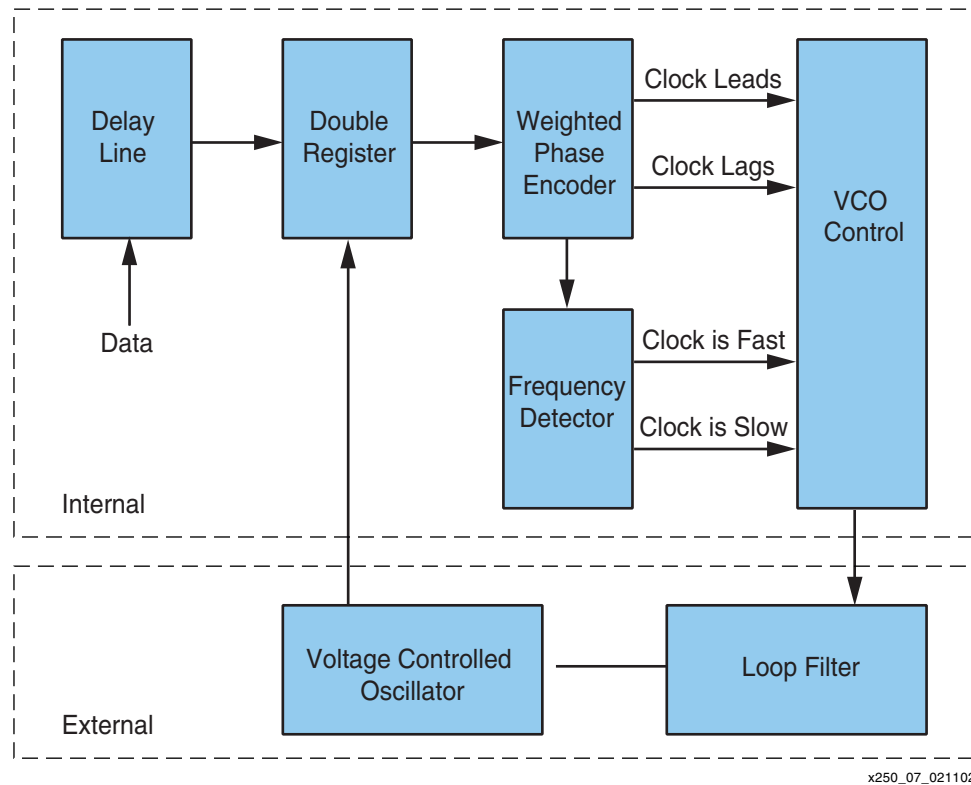


Figure 7: Delay Line Phase Detector

**Data Transition Detection**

Detection of data transitions for the current sample time can be accomplished by comparing strategically spaced sample points on the delay line. The sample points should be close enough to the adjacent sample point to guarantee that two transitions cannot occur. The sample points would be equal (all ones or all zeros) for sample time periods with no data transitions. A clock enable signal is constructed from the comparison of the sample points. The phase error signal is only updated when valid transitions occur in the data stream.

**Multiple Data Transition Detection**

The delay line phase detector should be designed for a maximum of two data transitions are allowed to occur in the delay line. A filtering mechanism is employed allowing the use of only one of the data transitions for phase error calculations.

For example, if two data transitions are detected at both the beginning and the end of the delay line, a phase correction is input to the VCO such that the phase of one transition is forced to the center of the delay line. The transition selected depends on the last known phase or whether the VCO is running a faster or slower frequency than the nominal bit rate frequency. The example given in this application note does not use the frequency error input from the delay line phase detector. In this application note the multiple transitions are avoided by only using the rising edges of the data for phase correction.

**Phase Error Output**

The phase error output is decoded in order to indicate the relative number of delay elements from the center of the delay line. Phase errors close to the center of the taped delay line indicate small phase errors and, hence, smaller corrections are needed. Phase errors further from the center of the taped delay line indicate larger corrections are needed. See [Figure 8](#).

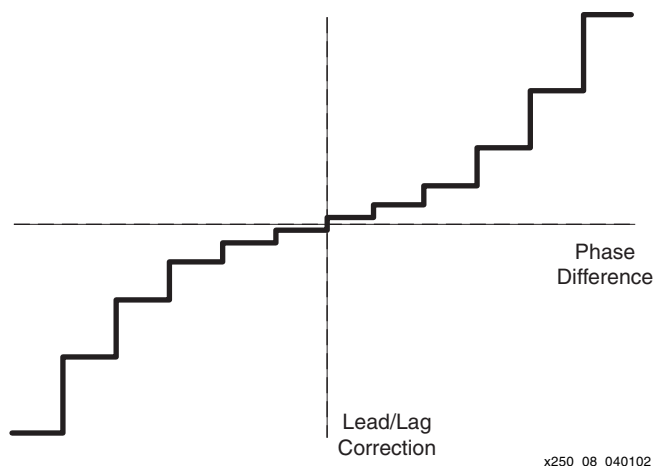


Figure 8: Phase Error Output

This application note uses phase accumulators to integrate phase noise over an update period of time. This is accomplished with two counters where the increments are adjusted by the amount of phase error. One counter is used for lead phase errors and the other is used for lag phase errors.

### Retimed Data Output

A tap at the center of the delay line is sampled with the register clocked by the opposite edge of the VCO clock source. This ensures that the sampling of the data is 180° out of phase with the jitter present on the data signal. In practice, the jitter tolerance of the CDR system above the loop filter roll off frequency is 0.6 UI. See Figure 9.

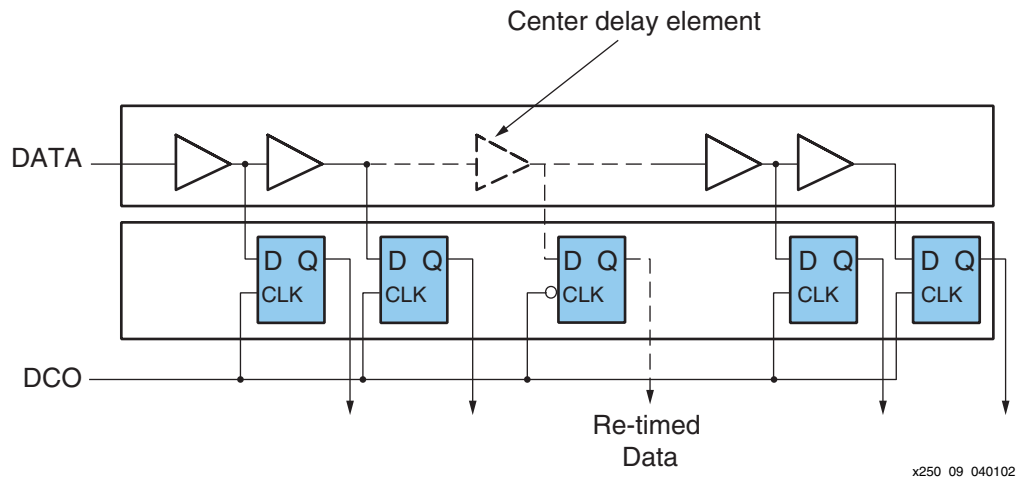


Figure 9: Retimed Data

### External Voltage Controlled Oscillator

Although this design uses the MAX2605-2609 for the external VCO, any other device with the same functionality can be substituted. The MAX2605-2609 was chosen because it is low cost, contains an integrated tuning diode, is available in a small package (SOT23-6), produces low phase noise, and improves noise immunity by placement physically away from the FPGA. The footprint of the external VCO solution is extremely small. Figure 10 shows the basic configuration for using an external VCO.

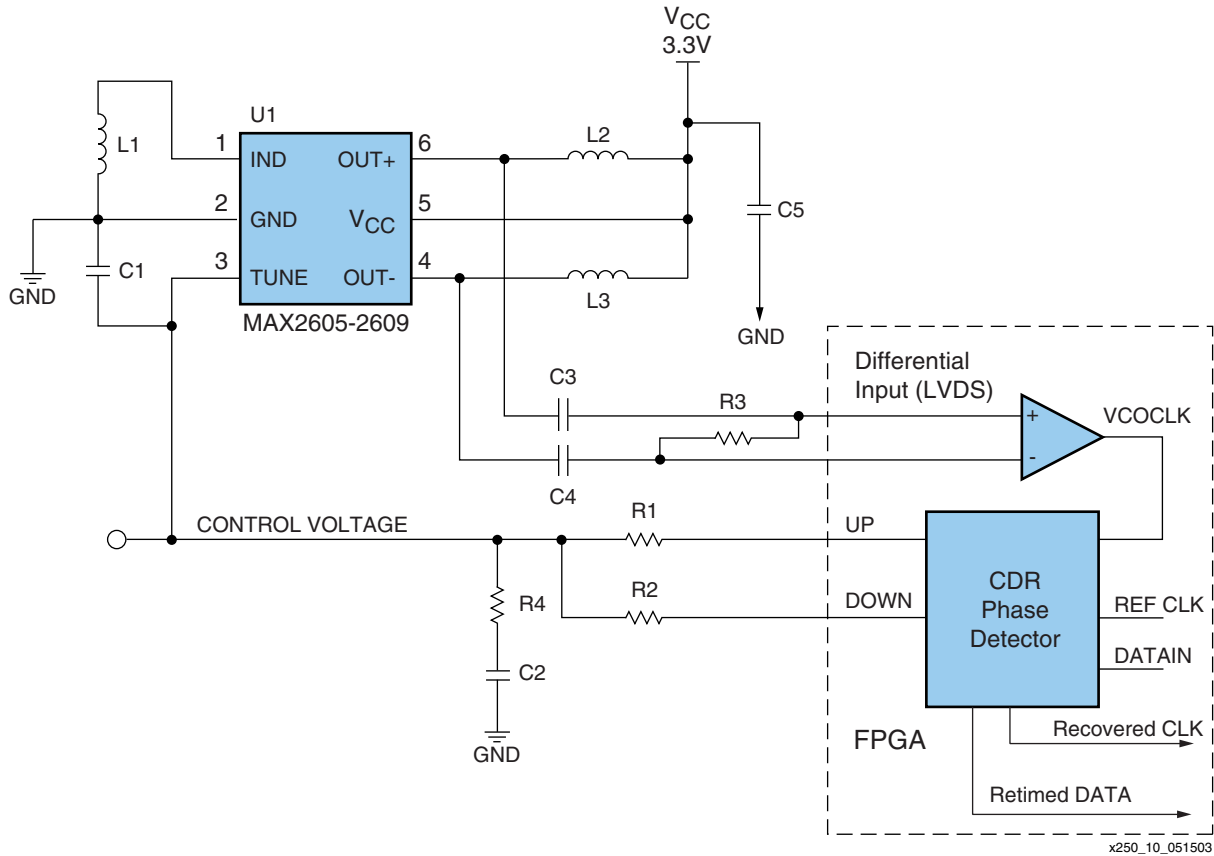


Figure 10: External VCO Configuration with Passive Loop Filter

### Passive Loop Filter

The loop filter described in this section is a passive-lag loop filter. The designer is encouraged to explore using an active loop filter. To quote Gardner from *Phaselock Techniques*<sup>[1]</sup>, "Better performance almost always will be obtained through use of an active filter." Assume the transmission of the data stream is made with a stable 100 ppm crystal oscillator when selecting devices to build the loop filter. The design only has to track temperature variations in a noisy environment. Figure 11 depicts the loop filter used.

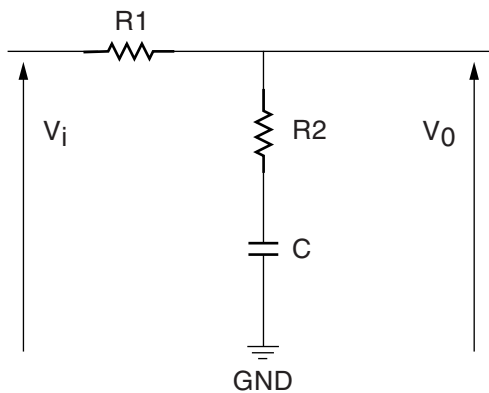


Figure 11: Loop Filter

The transfer function  $F(s)$  is given by:

$$F(s) = \frac{1 + s\tau_2}{1 + s(\tau_1 + \tau_2)}$$

If  $\tau_2$  is much smaller than  $\tau_1$ , the following approximation can be used:

$$F(s) = \frac{1 + s\tau_2}{1 + s\tau_1}$$

$$\tau_1 = (R_1 + R_2)C$$

$$\tau_2 = R_2 \times C$$

### Active Loop Filters

The PLL bandwidth is determined by:

$$\omega_{3dB} = K_d K_h K_o = K$$

Where  $K_d$  is the gain of the phase detector,  $K_h$  is the high-frequency gain of the loop filter and  $K_o$  is the gain of the VCO or VCXO. A large DC gain in the loop filter is desirable. For a passive loop filter, the maximum gain is unity. The DC gain for the active loop filter shown in [Figure 12](#), the gain is essentially infinite.

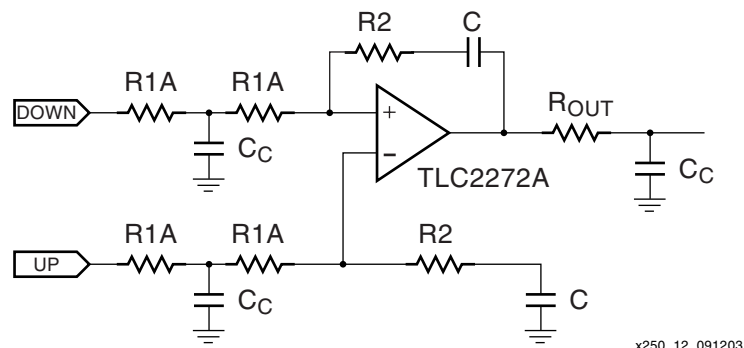


Figure 12: Active Loop Filter

### Design Example

This design example is needed to track temperature related voltage changes. Normally these changes occur over long periods of time. The components were chosen to give an overdamped response and a cut off frequency greater than a factor of 10 below the comparison frequency. A damping factor ( $\zeta$ ) of 0.707 is considered to be critically damped while values over 0.707 are considered to be overdamped. An overdamped response was chosen to avoid frequency ringing or hunting. In this design, the comparison frequency (1 MHz) is greater than 10 times the cutoff frequency ( $\omega_n = 15.1K$  Rad/sec), providing adequate filtering for the PLL operation.

This design example values are not intended to restrict the designer to a set of parameters. Rather, component values chosen are easy to obtain for the SMT footprints and also produce the desired results. The Xilinx SDV demo board has this design example implemented on it.

$$R_1 = 10.2\Omega$$

$$R_2 = 1K\Omega$$

$$C = 0.1\mu F$$

$$K_o = 125.6 \times 10^6$$

$$K_d = 5 \times 10^{-3}$$

$$\tau_1 = 1020 \times 10^{-6}$$

$$\tau_2 = 100 \times 10^{-6}$$

$$\omega_n = 15.1 \times 10^3$$

$$\zeta = 0.75$$

Where  $K_o$  is the VCO gain:

$$K_o = \frac{\Delta\omega}{\Delta V_C} = \frac{2\pi \times 20MHz}{1 Volt}$$

$$K_{OEFF} = \frac{K_o}{N}$$

$N = 270$  for a comparison frequency of 1 MHz.

$K_d$  is the PD gain:

$$K_d = \frac{V_H - V_L}{2\pi} = \frac{3.3 - 0}{2\pi}$$

The natural frequency of the loop is:

$$\omega_n = \sqrt{\frac{K_o K_d}{\tau_1}}$$

The damping factor is:

$$\zeta = \frac{1}{2} \left( \sqrt{\frac{K_o K_d}{\tau_1}} \right) \times \left( \tau_2 + \frac{1}{K_o K_d} \right)$$

## Reference Design

Verilog versions of the reference design are available on the Xilinx web site at <http://direct.xilinx.com/bvdocs/appnotes/xapp250.zip>.

## Conclusion

Virtex-II devices are used for performing CDR functions by extracting DATA and CLOCK from encoded serial bit streams. A Virtex-E device can run at link speeds up to 200 Mb/s (SDR rates) and a Virtex-II or Virtex-II Pro device can run at 400 Mb/s with additional components added to the board.

## References

1. Floyd M. Gardner, *Phaselock Techniques*, 2nd Edition, August 1979, John Wiley & Sons, ISBN 0-471-04294-3, page 92.
2. Dan H. Wolaver, *Phase-Locked Loop Circuit Design*, 1991 Prentice Hall, ISBN 0-13-662743-9, Chapter 3.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/01/02	1.0	Initial Xilinx release.
04/25/02	1.1	Corrected equation for $\tau_1$ on page 8.
09/12/03	1.2	Added new section on Active Loop Filters. Clarified CDR Function.
09/19/03	1.3	Added new section on Active Loop Filters. Clarified CDR Function.
08/17/04	1.3.1	Updated hyperlinks.
05/02/07	1.3.2	Removed reference to VHDL design files. Design is in Verilog only.