Clock Data Recovery Design Techniques for E1/T1 Based on Direct Digital Synthesis

Author: Paolo Novellini and Giovanni Guasti

XAPP868 (v1.0) January 29, 2008

Summary

Low data rates (less than 10 Mb/s) in a telecommunications environment can be terminated and regenerated by fully digital PLLs in today's Xilinx FPGAs. In a direct digital synthesis (DDS) based receiver, the jitter tolerance is reduced by the reference clock period. For example, a 10 Mb/s receiver that uses a 100 MHz clock has a 0.1 UI jitter tolerance reduction due to the DDS architecture. This document details the design aspects of digital PLLs used in telecommunications applications (digital oscillator, phase detector, and filter design criteria) and evaluates them against PLL performance (including bandwidth and jitter peaking) and loop stability.

The DDS approach to oversampling takes full advantage of the lookup table (LUT) structure. It is very resource efficient (60 Virtex-5 FPGA slices or 100 Spartan™-3 FPGA slices plus 1 BUFG) and recovers a non-stepping clock from each channel.

This application note provides a reference design that implements a fully optimized and compliant digital clock data recovery (CDR) circuit and jitter attenuator for 2.048 Mb/s (E1) and 1.544 Mb/s (T1) operation. This design can be implemented in both Virtex™ and Spartan devices, allowing for clock recovery and jitter attenuation functionality in the low frequency range using the SelectIO™ pins.

Introduction

In the telecommunications access industry, many channels often need to be terminated at low frequency:

- 2.048 Mb/s (E1) in Europe
- 1.544 Mb/s (T1) in North America, Japan, and Korea

These data rates are the first aggregation level for phone calls: 32 phone calls in an E1 line and 24 phone calls in a T1 line. Small companies often use E1 and T1 lines as data tributaries to carry data traffic (via the Internet). In comparison to ADSL, E1 and T1 lines offer symmetric downstream and upstream bandwidth. Thus E1 and T1 are preferred for broadband access by small companies. For details on E1 and T1 compliance, refer to ITU-T Recommendation G.703.

This reference design has two main functions for E1 and T1 lines:

- Clock data recovery when the input is data
- Jitter attenuation when the input is an E1 or T1 clock

The three main applications for this reference design are:

1. CDR for E1/T1 lines that are internal to the network node.

    In many cases, a line interface unit (LIU) is used to extract the clock. This LIU can be removed by using the code provided with the reference design.

2. CDR for E1/T1 lines that are external to the network node.

    The LIU can be replaced with a partial LIU (an LIU with no CDR or jitter attenuation functionality). Alternatively, the connection between the LIU and the FPGA carrying the extracted clock can be removed, saving one pin in the FPGA.

3. All serial lines with data transfer rates less than 10 Mb/s that can be synchronized with the reference design.

    Refer to "Reference Design Analysis," page 8 for how to customize the center frequency of the VCO.

For telecommunications access, many channels have to be terminated on the same board, so that they can be multiplexed to a higher level in the synchronous digital hierarchy (SDH) or plesiochronous[1] digital hierarchy (PDH) and transported. Thus it is important to optimize the cost of these interfaces.

The number of plesiochronous channels that can be terminated in a single FPGA is limited only by the amount of logic and the pins available. If n is the number of channels, the FPGA slices for all the channel (R_n) are given by Equation 1 for Virtex-5 FPGAs and Equation 2 for Spartan-3A FPGAs.

$$R_n = 60 \cdot n + 1 \text{ BUFG} \qquad\qquad \text{Equation 1}$$

$$R_n = 100 \cdot n + 1 \text{ BUFG} \qquad\qquad \text{Equation 2}$$

One BUFG is needed independent of the number of plesiochronous channels implemented in the single FPGA, with any mix of E1 and T1 data rates.

Basic PLL Theory

The basic PLL shown in Figure 1 consists of three elements: a phase detector, a controlled oscillator, and a low-pass filter.
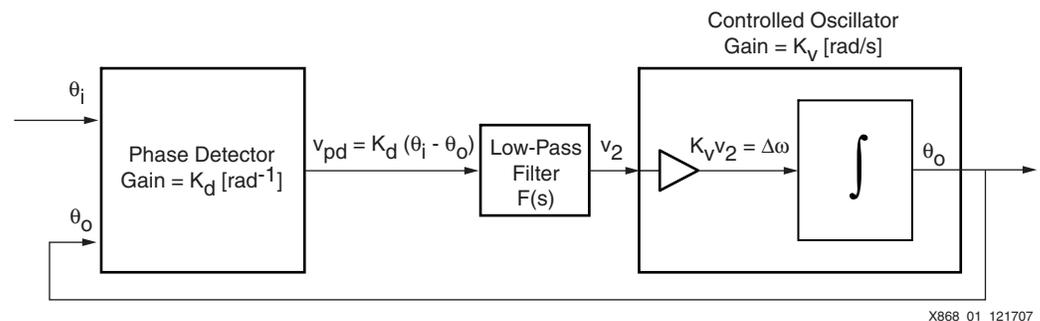
Figure 1: **Basic PLL/CDR Block Diagram**

In an analog PLL, the controlled oscillator is a voltage controlled oscillator (VCO), the output of the phase detector is a voltage or current signal, and the low-pass filter is built with resistors and capacitors. In a digital PLL, there are no resistors or capacitors, and the outputs of the phase detector and the controlled oscillator (which is an accumulator) are 0s and 1s. The frequency of the most-significant bits in the accumulator is determined by the value accumulated at each clock cycle.

The phase detector output (defined in Equation 3) is proportional to the difference between the input phase (θ_i) and the phase of the signal coming from the controlled oscillator (θ_o). K_d (the phase detector gain factor) is measured in units of rad^{-1}.

$$\upsilon_{PD} = K_d(\vartheta_i - \vartheta_o) \qquad\qquad \text{Equation 3}$$

---

1. Two or more signals are defined plesiochronous if their significant instants occur at nominally the same rate, with any variation in rate being constrained within specified limits.

When compared to the nominal frequency, the controlled oscillator's output frequency deviation ($\Delta\omega$) is proportional to the controlling signal at its input (υ_2) as shown in Equation 4. K_V (the controlled oscillator's gain constant) is measured in units of radians/s.

$$\Delta\omega = K_\upsilon \cdot \upsilon_2 \qquad\qquad Equation\ 4$$

Frequency is the derivative of the signal phase (see Equation 5). The controlled oscillator output is the Laplace transform of the signal phase as shown in Equation 6.

$$\frac{d\theta_o}{dt} = K_v \cdot \upsilon_2 \qquad\qquad Equation\ 5$$

$$L\left[\frac{d\theta_o}{dt}\right] = s\theta_o(s) = K_v \cdot V_2(s) \qquad\qquad Equation\ 6$$

Therefore the relationship between output phase and control is as defined in Equation 7.

$$\theta_o(s) = \frac{K_\upsilon \cdot V_2(s)}{s} \qquad\qquad Equation\ 7$$

The low-pass filter is described by its transfer function *F(s)*, so that

$$V_2(s) = F(s) \cdot V_d(s) \qquad\qquad Equation\ 8$$

Combining Equation 3 through Equation 8 gives the loop equation (Equation 9) and the error equation (Equation 10).

$$\frac{\theta_o(s)}{\theta_i(s)} = H(s) = \frac{K_\upsilon K_d F(s)}{s + K_\upsilon K_d F(s)} \qquad\qquad Equation\ 9$$

$$\frac{\theta_i(s) - \theta_o(s)}{\theta_i(s)} = \frac{\theta_e(s)}{\theta_i(s)} = \frac{s}{s + K_\upsilon K_d F(s)} \qquad\qquad Equation\ 10$$

Digital Low-Pass Filter in a Second-Order Loop PLL

The signal phase error (an output of the phase detector) must be low-pass filtered. The simplest low-pass filter is an integrator. The main benefit of the integrator is that the residual phase error in the steady state is 0. Unfortunately this system is always not stable, because the loop function crosses the 0 dB axis with a slope of –40 dB/decade, thus violating the Bode criterion for stability (Figure 2a).

a) With an Integrator as a Low-Pass Filter   b) When the Filter in Figure 3 is Used
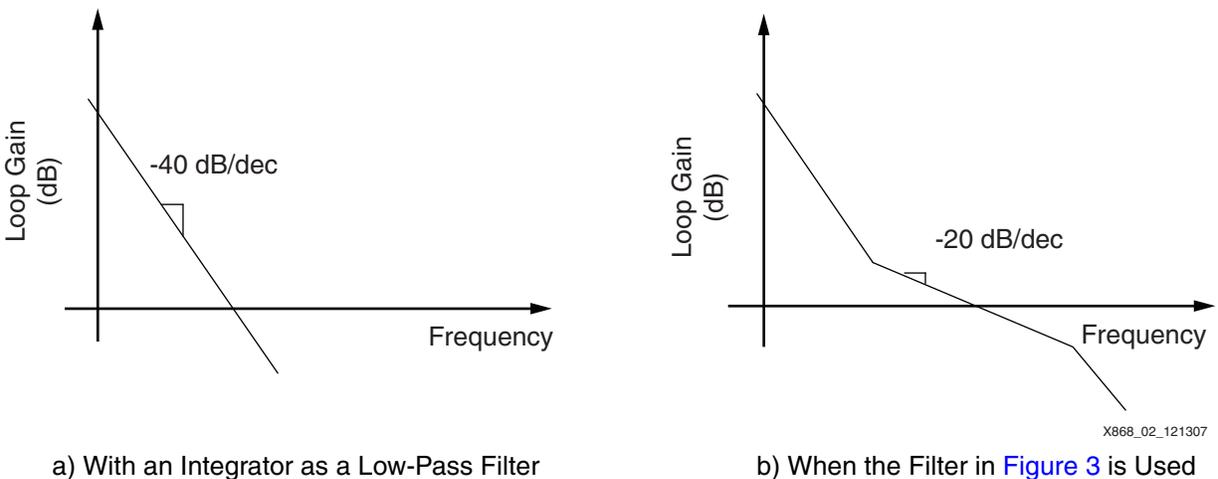
Figure 2:   **Loop Gains**

One requirement is that the overall loop must remain stable. For this reason, a second branch is added in the low-pass filter, yielding the loop function depicted in Figure 3. The new loop function crosses the 0 dB line with a slope of –20 dB/decade, which satisfies the Bode criterion for stability. The designer must position the zero and the pole in the loop function respectively above and below the 0 dB axis, to guarantee an adequate phase margin to the system. When $G_1 = 0$, the low-pass filter is always stable, but the residual phase error at the steady state is not 0. A filter with no integrator is often used in designs because the closed loop transfer function has no jitter peaking. With the integrator in the filter, jitter peaking can be minimized but is never eliminated.

X868_01_010808
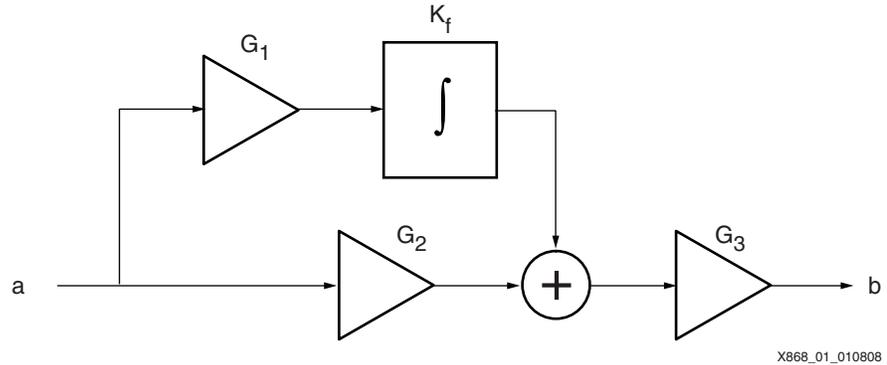
Figure 3: **Selected Low-Pass Filter for the Digital CDR for E1/T1 Lines**

Equation 11 lists the filter transfer function.

$$\frac{B(s)}{A(s)} = F(s) = G_3\left(G_2 + \frac{K_f \cdot G_1}{s}\right)$$

Equation 11

PLL Transfer Function

Combining Equation 9 and Equation 11, the PLL transfer function for the digital low-pass filter is defined in Equation 12.

$$H(s) = \frac{K_d K_\upsilon G_2 G_3 s + K_f G_1 G_3}{s^2 + K_d K_\upsilon G_2 G_3 s + K_d K_\upsilon K_f G_1 G_3}$$

Equation 12

By gathering some gains in the loop, H(s) can be written more simply, as indicated in Equation 13.

$$H(s) = \frac{G_2 G'_3 s + G'_1 G'_3}{s^2 + G_2 G'_3 s + G'_1 G'_3}$$

Equation 13

Where:

$$G'_1 = G_1 \frac{K_f}{s}$$

$$G'_3 = G_3 K_d K_\upsilon$$

Figure 4 shows the location of the terms in Equation 13 within the digital CDR.
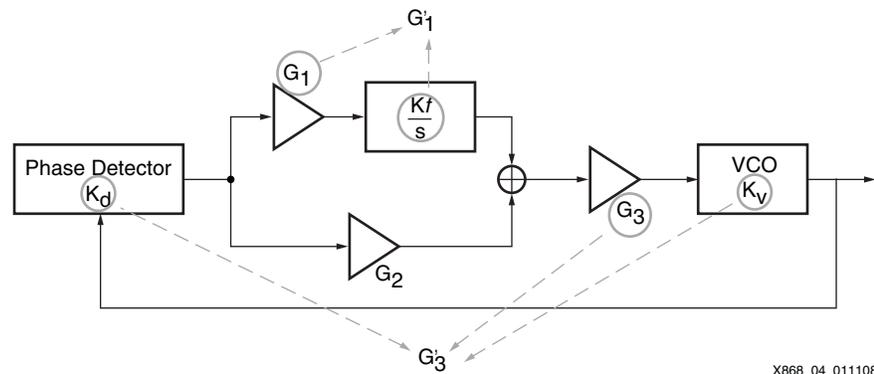
Figure 4: **Term Location within the Digital CDR**

Poles and Zeros in Loop Function: Sizing Criteria

The PLL loop is generally known as a second-order loop because the highest power of *s* in the denominator of the transfer function is 2. It is common to write the transfer function in terms of two factors: ω_n (the *natural frequency* of the loop) and ζ (the *damping factor*). Equation 14 and Equation 15 define ω_n and ζ for the PLL in terms of the hardware gains, respectively.

$$\omega_n = \sqrt{G'_1 G'_3} \qquad\qquad \textit{Equation 14}$$

$$\zeta = \frac{G_2}{2} \sqrt{\frac{G'_3}{G'_1}} \qquad\qquad \textit{Equation 15}$$

Equation 16 indicates the new form of the transfer function.

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad\qquad \textit{Equation 16}$$

ω_n is not the bandwidth of the PLL, although its value is close. The bandwidth of the PLL with no integrator is listed in Equation 17. If the bandwidth is set to 50 Hz (i.e., 314 rad/s), G_2 is equal to 2^{-3}.

$$\omega_{3dB} = G_2 \cdot K_D \cdot K_V \qquad\qquad \textit{Equation 17}$$

When the bandwidth is determined, ζ can be selected so that the jitter peaking is acceptable. The acceptance criteria depends on the application. Jitter peaking leads to a critical condition when many regenerators are cascaded, because jitter is amplified in the ω_n region.

As shown in Figure 5, $\zeta \geq 3$ guarantees a peaking that is better than 0.5 dB. $\zeta = 3$ can be obtained by setting $G_1 = 2^{-17}$ and $G_3 = 1$.
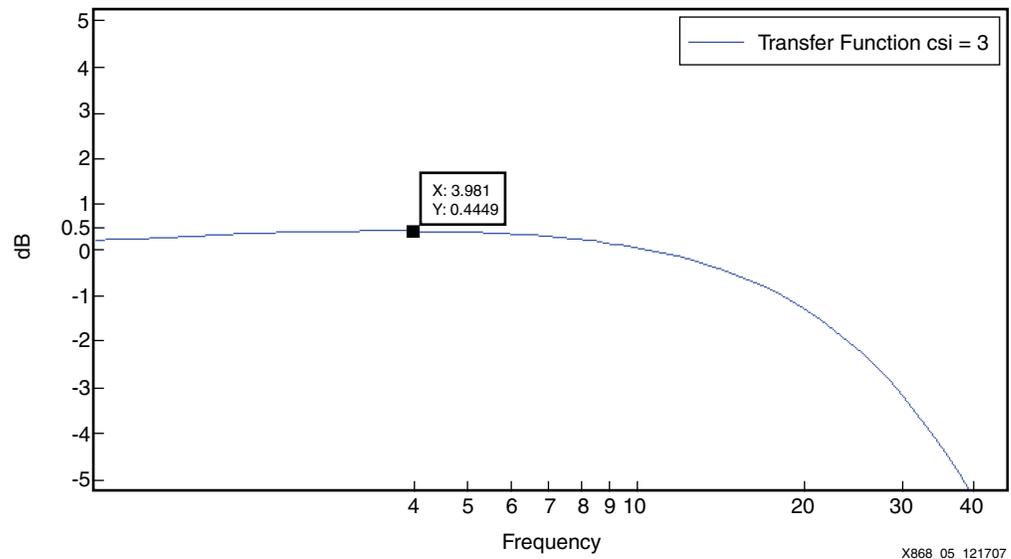
Figure 5: ζ **is Chosen Based on the Selected Maximum Jitter Peaking Tolerance**

Reference Design Description

Figure 6 shows the reference design symbol of the fully digital CDR.
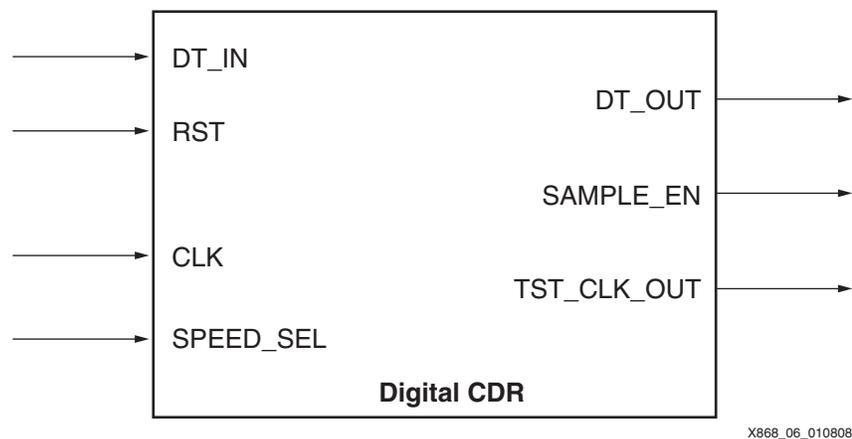
Figure 6: **Digital CDR Logic Symbol**

Data is input into DT_IN at 2.048 Kb/s or 1.544 Kb/s. It is considered valid on DT_OUT when the digital CDR is locked. The extracted clock appears on TST_CLK_OUT. The required reference clock (150 MHz) is input on CLK. SPEED_SEL determines whether the module operates at the E1 or the T1 rate.

Table 1 lists the pinout and the required frequency tolerances of the digital CDR.

Table 1: **Digital CDR Pinout Descriptions**

Pin Name	Type	Description	Comments
CLK	I	Input reference clock	The CLK frequency is required to be 150 MHz ± 40 ppm. Different reference clock frequencies can be used by changing the parameters of the CDR. Refer to "Digital VCO," page 8 for details on how to change the parameters.
DT_IN	I	Data enters the CDR on this pin	The data rate must be 2.048 Kb/s ± 20 ppm or 1.544 Kb/s ± 20 ppm.

Table 1: **Digital CDR Pinout Descriptions** *(Cont'd)*

Pin Name	Type	Description	Comments
DT_OUT	O	DT_OUT is valid on the edge of SAMPLE_EN	Use DT_OUT in combination with SAMPLE_EN to synchronously process the data out of the CDR.
RST	I	Reset	For simulation only. In hardware, the system has no unknown states.
SAMPLE_EN	O	The data on DT_OUT is valid when SAMPLE_EN is High	SAMPLE_EN is used to clock logic (clock enable). No additional BUFGs are required.
SPEED_SEL	I	• `1`: 2.048 Kb/s operating mode (E1) • `0`: 1.544 Kb/s operating mode (T1)	SPEED_SEL selects the operating mode. The reference clock does not need to be changed.
TST_CLK_OUT	O	Extracted clock	This clock is recommended only for scope plotting and testing. It is not recommended for clocking logic, because it consumes a BUFG.

This design provides a testbench that can be simulated (`TB_CDR`). The testbench generates a pseudo-random binary sequence (PRBS) data pattern and applies a step in frequency (about a 20 ppm increase) to show the tracking capability of the digital CDR.

A second testbench is available in the design directory (`tb_top`). This testbench can be implemented in the Virtex-5 FPGA ML52X demonstration board. It comprises two digital CDRs working on two plesiochronous PRBS data inputs, which can be set independently at 2.048 Mb/s or 1.544 Kb/s. Each data input can also be modulated on-the-fly in the range ±100 ppm (0.03 Hz is the minimum resolution step). The user can see the tracking capability of the digital CDR and its limits (set at ±70 ppm) on a scope. Figure 7 shows the structure of the testbench.
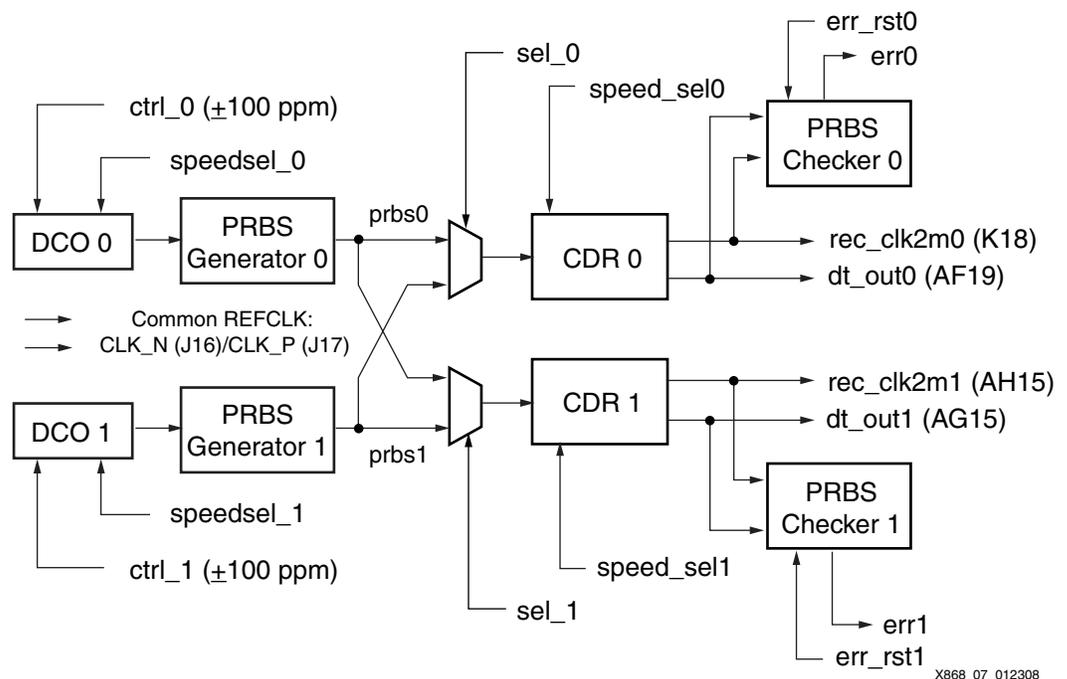
Figure 7: **Testbench Structure**

The reference design provides a ChipScope Pro™ analyzer project file (under the `CSPRO` folder) to fully control all the parameters of the testbench (E1/T1, frequency tuning, resets, etc.) on-the-fly.

The digital CDR can track a signal between ±70 ppm (20 ppm is allocated to the CDR and 40 ppm is allocated to the reference clock with a 10 ppm margin). The user can overdrive the

data source beyond the allowed 70 ppm, showing the limits of the digital CDR. The CDRs and the PRBS generators can be programmed independently to work at 1.544 Kb/s or 2.048 Kb/s via the ChipScope Pro interface.

The PRBS checker checks the recovered data with an error detector that can be read and reset via the ChipScope Pro interface.

Two analog recovered clocks that are available on two output pins (AF19 and AG15) of the ML52X board can be viewed with a scope. The minimum programmable frequency difference is 0.03 Hz. The two resampled data channels can be monitored on the K18 and AH15 pins. The 150 MHz reference clock must be provided differentially into the J16 and J17 pins. The user can design the application with single-ended clocks.

The reference design and all testbenches are available in both VHDL and Verilog. Figure 8 shows an annotated directory structure.
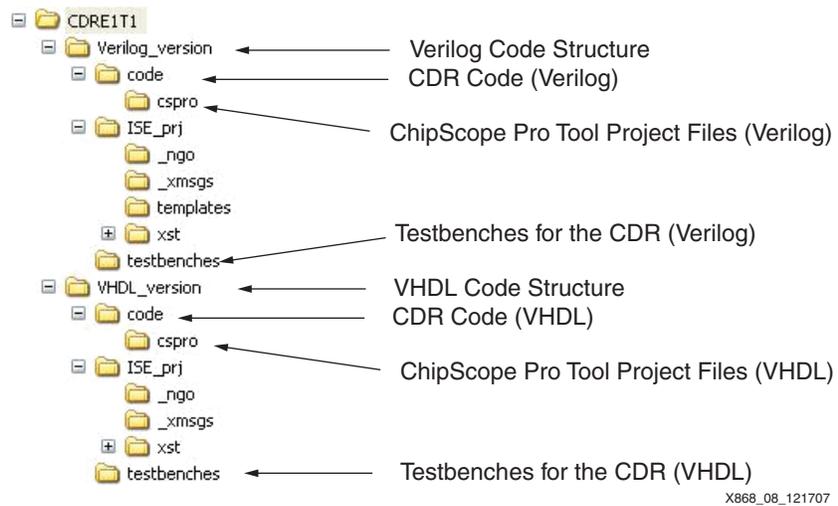
Figure 8: **Directory Code Structure**

Reference Design Analysis

This section provides details on the reference design implementation and how to customize it to change the data rate and/or the reference clock frequency. Details are provided on the digital VCO, phase detector, and digital filter as well as the hardware and software requirements.

Digital VCO

The digital VCO is implemented using a 32-bit accumulator and an adder, as shown in Figure 9.
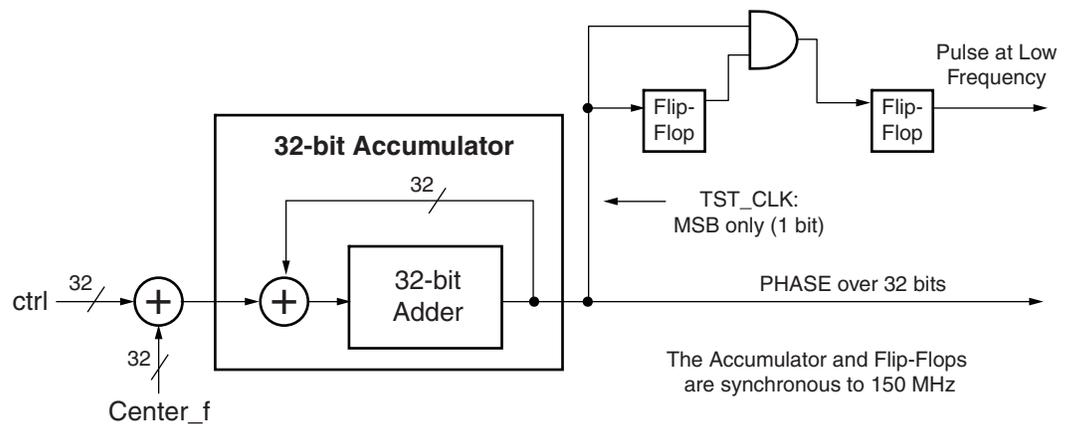
Figure 9: **Digital VCO Block Diagram**

Equation 18 defines *Center_f*, which controls the center frequency of the VCO. In Equation 18, f_W is the low frequency wanted on the output, and f_{CLK} is the reference clock frequency.

$$Center\_f = f_w \cdot \frac{2^{32}}{f_{CLK}}$$ *Equation 18*

The digital VCO can be tuned based on the settings for f_{CLK} and f_W, which calculate Center_f according to Equation 18. Table 2 shows example values for Center_f based on f_{CLK} and f_W. Any reference clock frequency can be used, provided that it is much higher than the data rate.

Table 2: **Center_f Values**

f_W	f_{CLK}		
	150 MHz	**155.520 MHz**	**125 MHz**
2.048 MHz	0x37EC8EC	0x35F0688	0x431BDE8
1.544 MHz	0x2A2957A	0x28AA3EC	0x2A2957A

Phase Detector

Figure 10 is a block diagram of the phase detector.
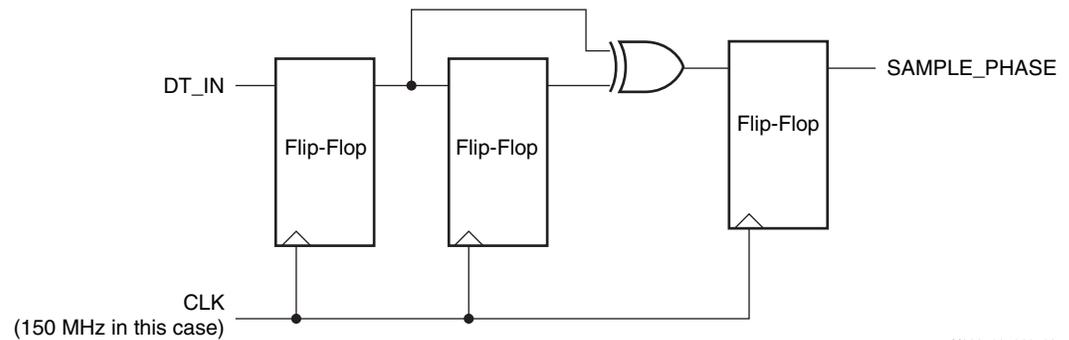
Figure 10: **Phase Detector Block Diagram**

When a transition is present on the DT_IN input, a pulse on SAMPLE_PHASE is generated for 1 clock period. The SAMPLE_PHASE signal samples the 32-bit phase information of the digital VCO, which is expected to be 0, if the VCO output is aligned with the data. If the value is not 0, the phase error is inverted algebraically (to implement the negative feedback) and is loaded into the digital filter.

The phase sampling and inversion is performed in the VCO block by this code fragment:

```
PROCESS(CLK,RST)
begin
if RST='0' then
    PHASE<=x"0000";
  elsif CLK='1' and CLK'event then
     if SAMPLE_PHASE='1' then
        PHASE<=NOT(phase_int(31 downto 16));
       else null;
     end if;
end if;
end PROCESS;
```

Only the 16 most-significant bits of the phase information are used. These 16 bits indicate that the phase resolution is better than 10^{-4} [radiant], which is an improvement over the time resolution used to sync the incoming transition (better than 1.3×10^{-2}).

Inverting a signed number (`phase_int`) yields the mathematical opposite with an error of 1. This error, which is equivalent to less than 10^{-4} [radiant], is simply ignored to avoid the insertion of an additional adder.

Digital Low-Pass Filter

The digital filter implementation closely follows the structure in Figure 3, page 4. The ctrl output is limited to 13 bits signed. Thus the digital VCO can be tuned by ±143 Hz.

As with all other blocks, the filter is fully synchronous to f_{CLK}, and the operation is pipelined to easily meet the timing. When customizing the digital CDR, the designer must pay attention to the stability of the loop. G_1 and G_2 need to be checked and tuned as described in "Poles and Zeros in Loop Function: Sizing Criteria," page 5.

Hardware Requirements

The digital CDR described in this application note can be implemented in a Virtex or a Spartan device. The reference design example is provided in both VHDL and Verilog for Virtex-5 FPGAs on the ML52X demonstration board.

The required hardware for this reference design is:

- Xilinx ML52X demonstration board, revision C or higher
- Programmable USB cable (Model DLC9 or newer)

Software Requirements

The required software for this reference design is:

- ChipScope Pro analyzer, version 9.1i or higher
- ISE™ tool, version 9.1i or higher
- Mentor ModelSim, version 6.1a or higher (to perform simulations)
- A real-time scope (to probe data and extracted clocks from the demonstration board)

Reference Design

The DDS based CDR reference design for E1 and T1 can be downloaded at:

https://secure.xilinx.com/webreg/clickthrough.do?cid=103271

References

The following material provides additional information related to this application note:

- ITU-T Recommendation G.703, *Physical/electrical characteristics of hierarchical digital interfaces*
  http://www.itu.int/ITU-T/index.html
- UG225, *ML52x User Guide*.

Conclusion

The reference design presented in this application note allows integration of the CDR functionality of E1/T1 lines into a Xilinx FPGA, leaving on the outside of the FPGA just the electrical line termination. E1 and T1 lines can be mixed on-the-fly by the end user, keeping the BUFG usage limited to one for any configuration, independent of the number of E1/T1 lines.

Acknowledgment

Xilinx wishes to thank Silvio Cucchi for his key contributions to this reference design and document.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
01/29/08	1.0	Initial Xilinx release.

Notice of Disclaimer