XAPP589 (v2.3) April 29, 2015

All Digital VCXO Replacement for Gigabit Transceiver Applications (7 Series/Zynq-7000)

Authors: David Taylor, Matt Klein, and Vincent Vendramini

Summary

This application note delivers a system that is designed to replace external voltage-controlled crystal oscillator (VCXO) circuits by utilizing functionality within each serial gigabit transceiver.

Note: In this application note, *transceiver* refers to these types of transceivers:

Device Family	Transceiver Type
Artix®-7 FPGA	GTP transceiver
Kintex®-7 FPGA	GTX transceiver
Virtex®-7 FPGA	GTX and GTH transceivers
Zynq®-7000 All Programmable SoC	GTP and GTX transceiver

A common design requirement is to frequency or phase lock a transceiver output to an input source (known as loop, recovered, or slave timing). Traditionally, an external clock cleaning device or VCXO and PLL components are used to provide a high-quality clock reference for the transceiver, since FPGA logic-based clocks are generally too noisy. While effective, external clock components carry a power and cost penalty that is additive as each individual clock channel is generated. When using many channels or in low-cost systems, the cost can be significant. Additionally, adding many external clock sources provides more opportunity for crosstalk and interference at the board level.

The system described in this application note provides a method to effectively replace these external clock components using a combination of unique Xilinx transceiver features in conjunction with a high-performance FPGA logic based digital PLL (DPLL). Each transceiver has a phase interpolator (PI) circuit in the high-speed analog PLL output circuits that provides, on a individual transceiver channel basis, the ability to phase and frequency modulate the transmit clock operating the transceiver. Using a fully digital interface, the phase interpolator can be phase and frequency controlled from the FPGA logic resources under control of a high-resolution programmable DPLL. In conjunction with the FPGA logic DPLL, the phase interpolator provides the ability to phase or frequency modulate the transceiver data output directly locking to an input reference pulse or clock while providing an built-in clock cleaning filter function. Unlike conventional solutions, high-quality system results because the clocking components are contained within the transceiver.

The reference design circuit provides a fully integrated DPLL and transceiver phase interpolator system which can be instantiated for each transceiver channel used. The transceiver can phase or frequency lock to an input reference signal. The DPLL enables generation of a synchronous transceiver data output with run-time configurable parameters (e.g., gain, cutoff frequency, and

clock divider values) to enable you to set up the operation specifically for the end application. This particularly highlights the flexibility of the reference input signal and DPLL cleaning bandwidth.

The reference design circuit can lock an individual transceiver channel to up to ±160 ppm from the reference oscillator and programmatically provide jitter cleaning bandwidths in the range from 0.1 Hz to 1 KHz. In the 7 series FPGAs, the transceiver is capable of operating at up to 12.5 Gb/s. Typical applications for this circuit include video SD/HD, Sync E, IEEE1588, SDH, SONET, and OTN.

You can download the reference design files for this application note from the Xilinx® website. For detailed information about the design files, see Reference Design, page 28.

System Applications

A number of different applications need external VCXOs and PLLs or clock-cleaning components on a per transceiver transmitter basis.

Some application examples include:

- OTN muxponders for trunk to tributary slaving of outputs.
- Broadcast equipment including switchers and routers using SD, HD, and 3G SDI video outputs.
- Synchronous Ethernet.
- Recovered media clock generation (per IEEE specification 1588).

External components used to do this task are a high-impact expense for several reasons:

- Significant BOM cost, an estimated $10 to $15 per additional VCXO/PLL or clock cleaner.
- Significant power consumption (300 mW to 500 mW) per additional VCXO/PLL or clock cleaner.
- Board space and PCB complexity, both due to additional board area required and careful noise-reduction design layout requirements.

Figure 1 shows a general use case where inputs are received through any one of a number of types of links carrying data. There could be one input link per output link or a group of input links that data is striped across, which are de-multiplexed to form output links. While each input link can share one reference clock, the challenge is that each output link needs its own VCXO/PLL or clock cleaning system to provide a clean reference to the transceiver to serialize outgoing data and produce the expected low-jitter output signal.

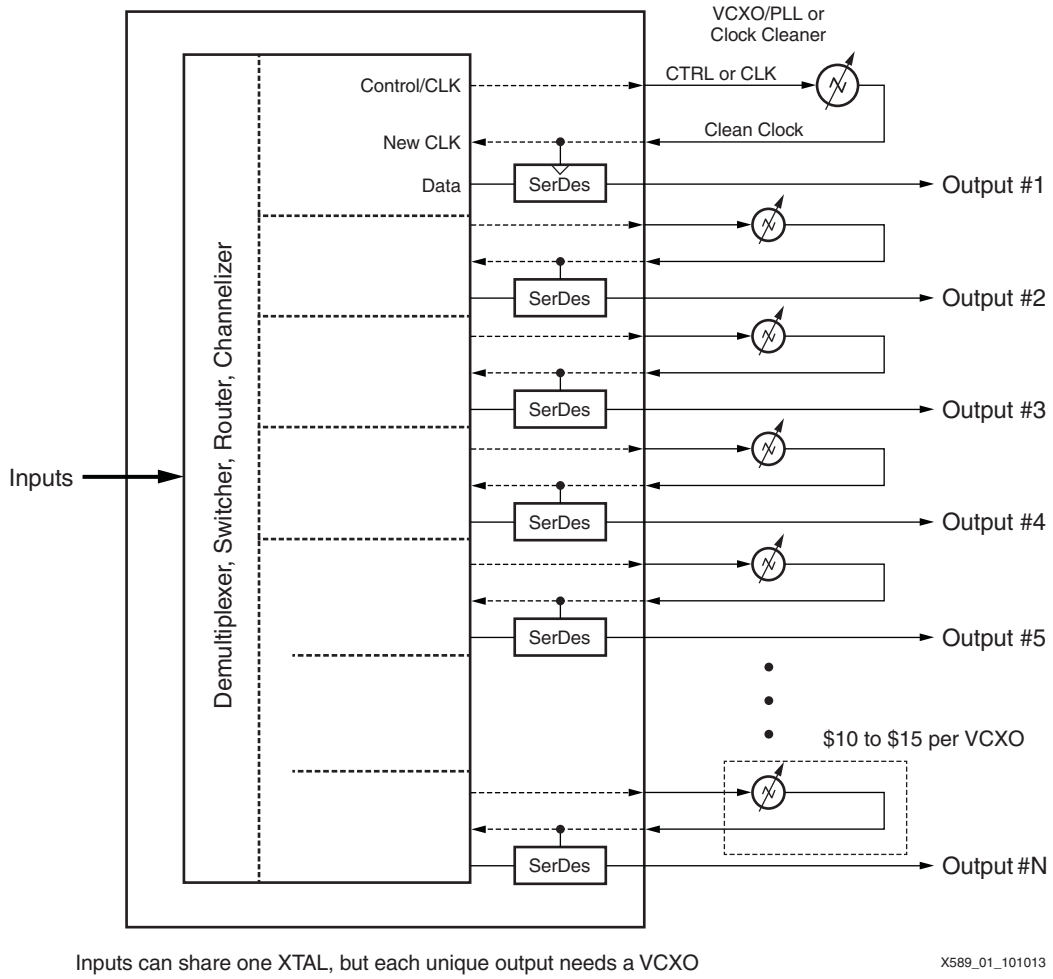


Figure 1: **Typical Design with Multiple VCXOs (One per Unique Output Rate)**

By using transceivers the need for external VCXO/PLLs and clock cleaners can be eliminated. The basics of the reference design method are:

- The 7 series FPGA transceiver has a transmit clock phase interpolator (TX PI) for the transmit serial/deserializer bit clock.
- Each phase interpolator in each transmit serial/deserializer can be independently, dynamically, and continuously changed in phase and hence, shifted in frequency.

There are a number of benefits to this implementation:

- Significant BOM cost reduction, an estimated $10 to $15 per additional VCXO/PLL.
- Significant power consumption savings (300 mW to 500 mW) per VCXO/PLL.
- Reduced board space and PCB complexity.
- Ability to have four unique differentiated transmission rates within a transceiver Quad.

An example block diagram of this new method is shown in Figure 2. The output VCXOs/PLLs or clock cleaners are brought into the FPGA using the transceiver Quad's phase-shifting functionality.

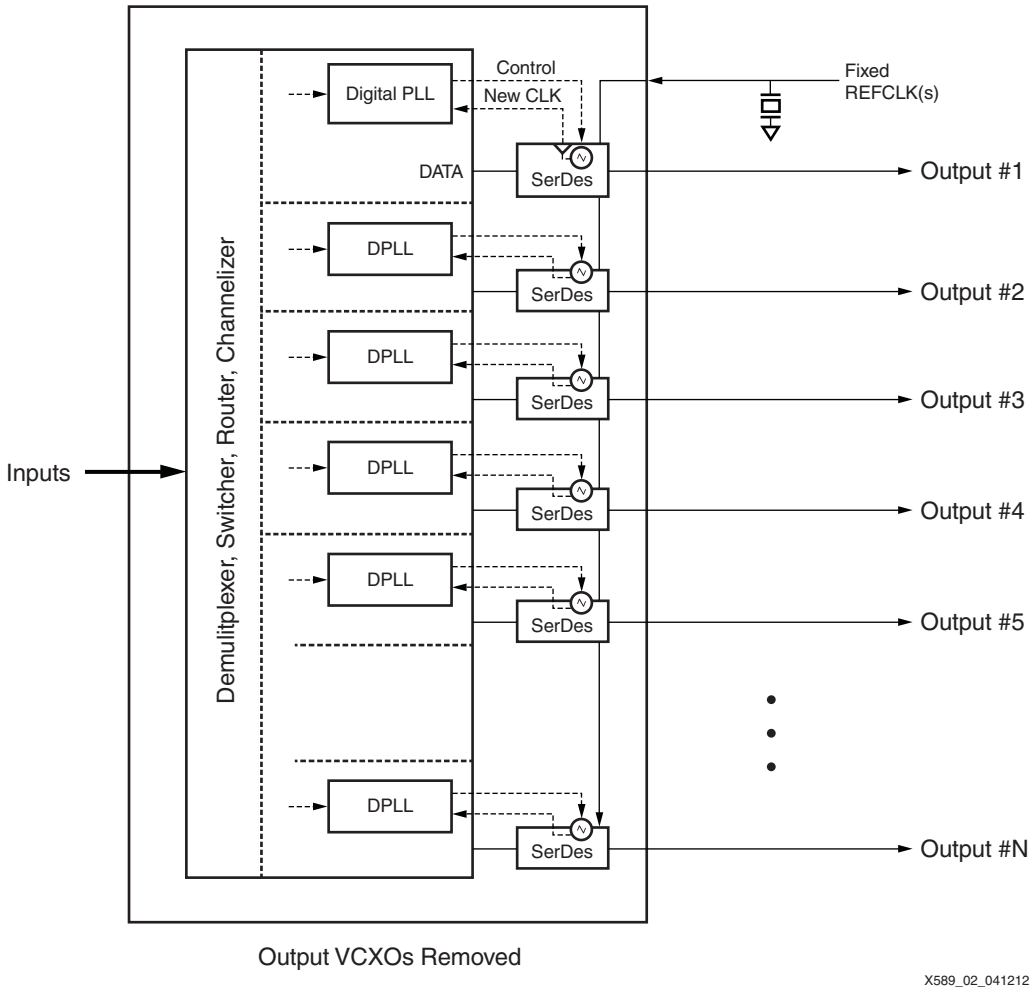


Figure 2: **Phase-Shifting Solution Block Diagram**

Solution Examples

This section describes solutions for broadcast switcher or router applications and OTN trunk to tributary demultiplexer applications. Normally, an external VCXO/PLL or clock cleaner is required for each unique output transmit serial/deserializer channel. This is very expensive because each parts per million (PPM) channel variation requires an additional circuit, even if the output base rate is the same (e.g., 1.485 Gb/s + 50 ppm and 1.485 Gb/s – 20 ppm).

Example 1: Broadcast Switcher or Router

In this example (Figure 3), a broadcast switcher or router is receiving SD/HD/3G streams from different cameras or unrelated sources. The video is processed inside of an FPGA, but the outputs of FPGA must be exactly locked to some of the input channels. Individual HD-SDI and 3G-SDI inputs, for instance, while nominally running at 1.485 Gb/s and 2.97 Gb/s might not be locked to each other. The inputs could be locked to their original sources, which could vary from the nominal frequency by up to 150 ppm. Normally, when individual outputs of the FPGA are meant to be locked to these PPM varied inputs, the design includes an external VCXO/PLL or clock cleaner for each PPM varied output. By using the FPGA's TX PI, the complex and expensive components are instead designed into the transmit serial/deserializer functionality.

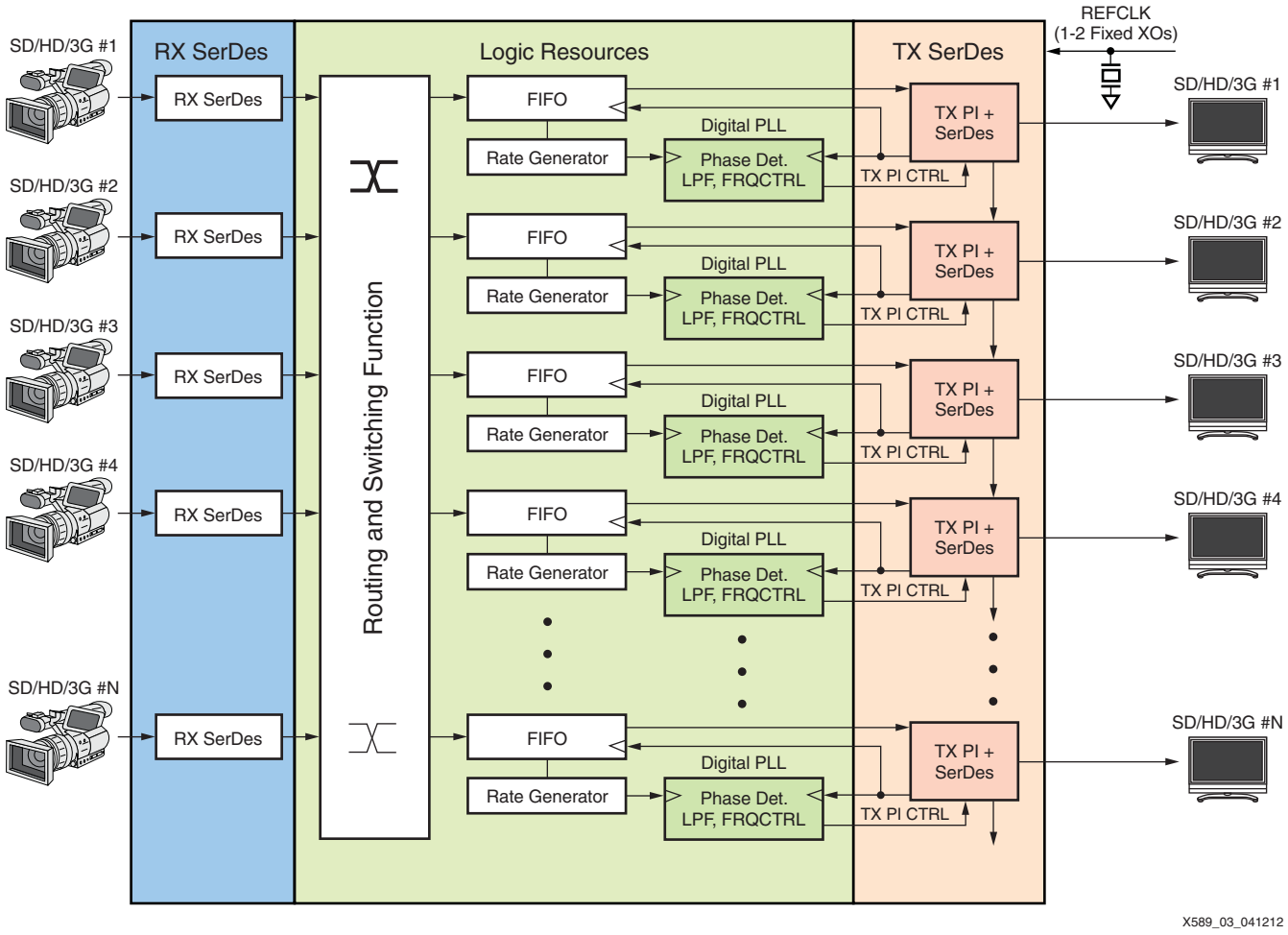


Figure 3: **Broadcast Switcher or Router without External VCXOs/PLLs or Clock Cleaners on the Outputs**

Example 2: OTN Muxponder

In this example (Figure 4), within an OTN trunk there are multiple streams of data. Each stream within the trunk has a unique rate. In a number of cases, an FPGA or other device receiving the trunk will demultiplex it into its component streams. Each source stream, even at approximately the same rate, has an oscillator or clock system that originally sourced the stream in the trunk. The timing of the recovered stream must be preserved on a stream by stream basis. For example, when there are multiple synchronous Ethernet streams that are each nominally at 1.25 Gb/s, but not from the same source, each stream (even if it is the same type) can vary by a few PPM. The recovered outputs must each be exactly locked to their original source, which requires a phase detector, low-pass filter, VCXO, and PLL external to the FPGA per unique output channel. The Xilinx FPGA's TX PI in the transmit serial/deserializer can take fixed nominal oscillator rates as REFCLKs and effectively slave them within the transmit serial/deserializer to the unique recovered rate with low jitter and without the need for external VCXOs/PLLs or clock cleaners.

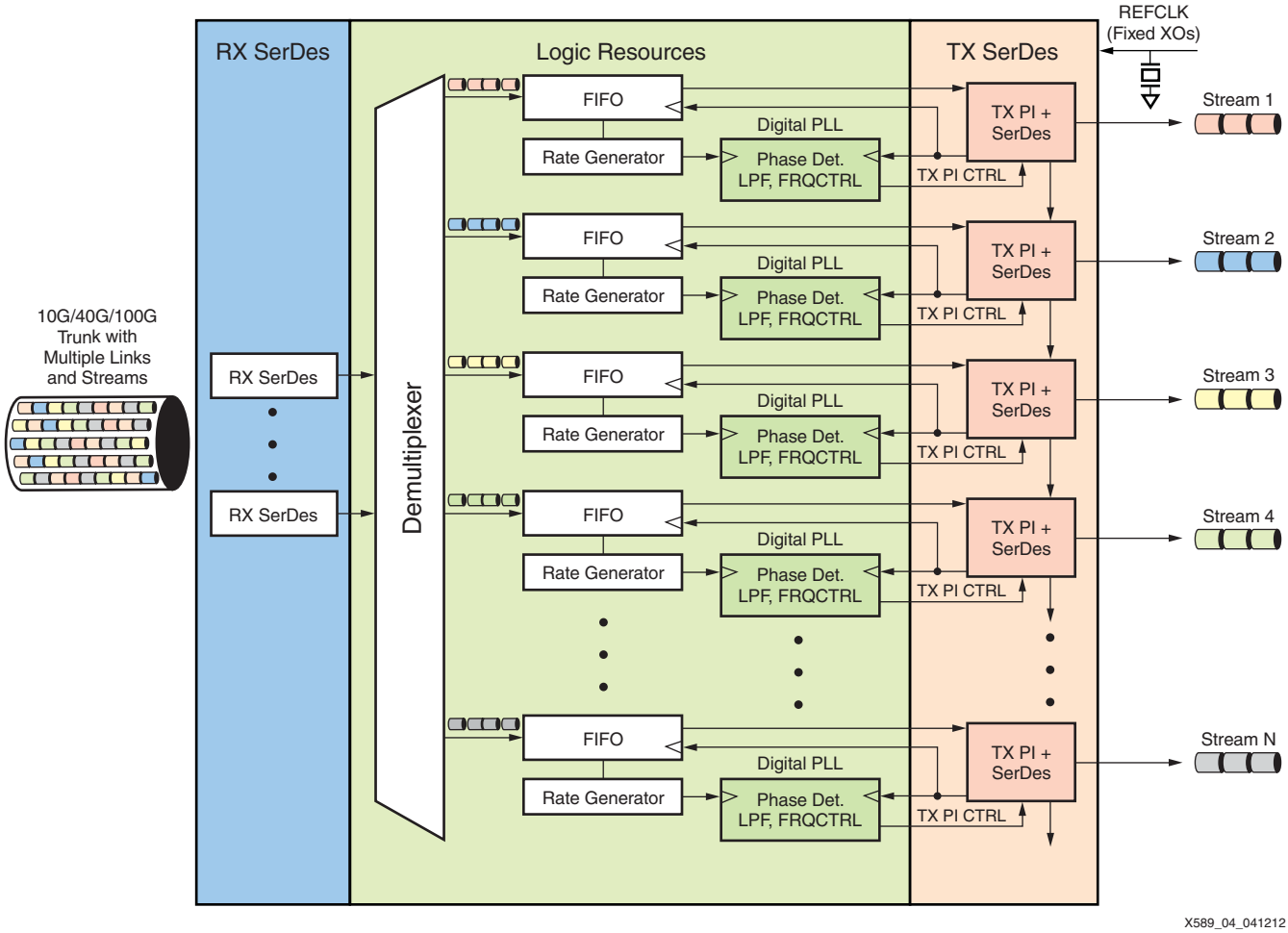


Figure 4: **OTN Trunk to Tributary Demultiplexor and Delivery without External VCXOs/PLLs or Clock Cleaners**

In these examples, and for many other cases, the unique Xilinx transmit clock phase interpolator functionality built into the transmit serial/deserializer and the FPGA-based phase detector, digital PLL, low-pass filter, and controlled transmit serial/deserializer phase interpolator replace expensive external VCXO/PLLs or clock cleaners.

VCXO Replacement Theory

The 7 series FPGAs contain a key functional block in the transceiver transmitter that enables functional replacement of a VCXO. This block, a phase interpolator, produces an output clock with a fine phase shift relative to an input clock. The fine phase shift is produced in response to a control word. The control word allows selection of a phase between 0° and 360°.

To create the equivalent of a VCXO with only a fixed frequency source, a phase is selected by the phase interpolator and the selected phase selection value is continuously updated with a linearly increasing or decreasing phase. This is equivalent to a positive or negative frequency shift proportional to the rate of change of the controlling phase. See Equation 1 through Equation 5.

$$f = \frac{d\Phi_{IN}(t)}{dt} \quad \text{Equation 1}$$

$$\Phi_{IN}(t) = \int f_{IN} dt = f_{IN} t \quad \text{Equation 2}$$

$$\Phi_{OUT}(t) = \Phi_{IN}(t) + \Phi_{CONTROL}(t) \quad \text{Equation 3}$$

$$\text{Differentiating} \frac{d\Phi_{OUT}(t)}{dt} = \frac{d\Phi_{IN}(t)}{dt} + \frac{d\Phi_{CONTROL}(t)}{dt} \quad \text{Equation 4}$$

$$f_{OUT} = \frac{d\Phi_{OUT}(t)}{dt}, f_{OUT} = f_{IN} + \frac{d\Phi_{CONTROL}(t)}{dt} \quad \text{Equation 5}$$

From these equations, the output frequency is shifted by the rate of change of the control phase $\Phi_{CONTROL}$ with respect to time.

Figure 5 shows a functional block diagram of the phase interpolator, which includes the inputs and outputs used in conjunction with the high-speed serial clock input from the SerDes transmit PLL and other circuits that participate in the complete solution. This block first produces a number of primary phases from the n-phase generation block. This block generates x phases separated by 360°/x. In the case of the transmit serial clock phase interpolator, there are eight primary phases (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°). The phase-select block in Figure 5 selects two adjacent phases based on the control logic.

The phase interpolation function is performed by the phase mixer, which functionally combines k parts of $\Phi 1$ and (1 – k) parts of $\Phi 2$ (resulting in an interpolated phase output of $\Phi 1k + \Phi 2(1 - k)$), where k is a fraction between 0 and 1.

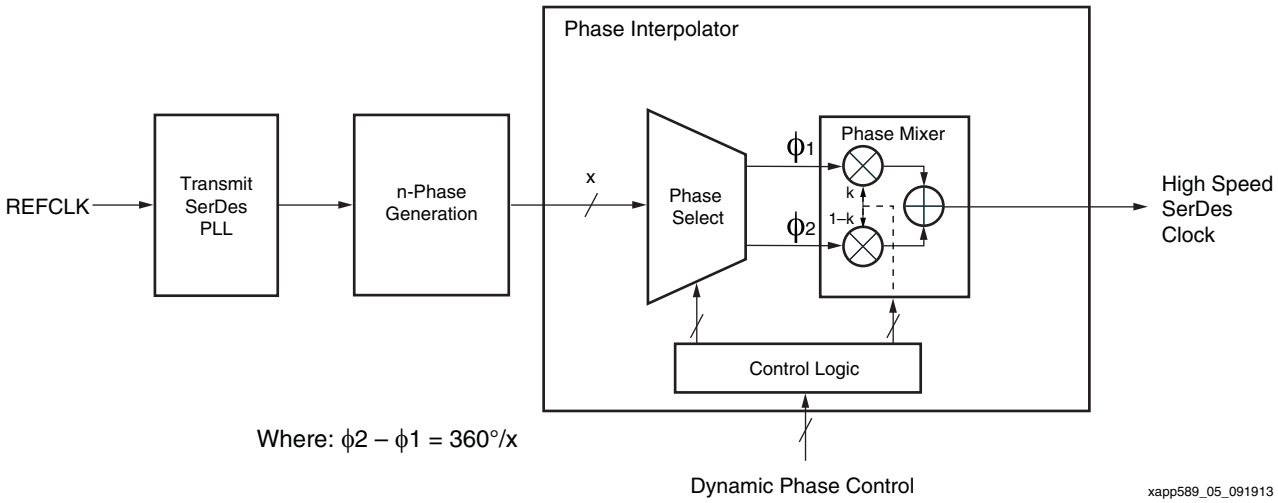


Figure 5: **Functional Block Diagram of a SerDes Transmit PLL Feeding a Phase Interpolator**

Figure 6 shows two primary phases (Φ1 and Φ2) selected from the n-phase generator that are interpolated by the phase mixer. The output is a clock with a phase that is in between Φ1and Φ2 with a resolution determined by the number of fractional steps allowed.

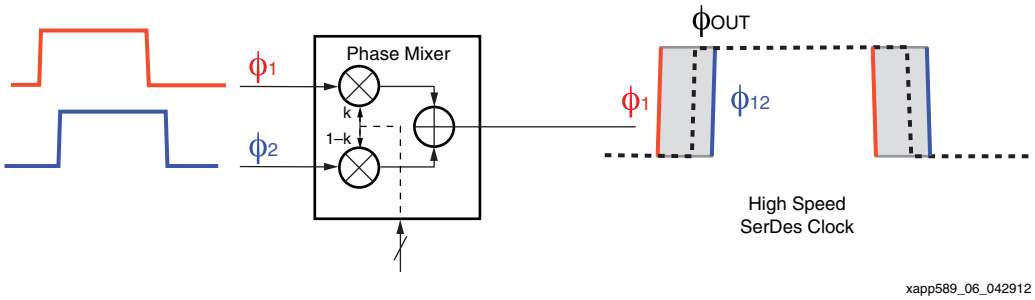


Figure 6: **Phase Mixer Producing an Interpolated Phase from Coarse Phases**

The primary phase generation and interpolated phase creation are all performed in a clean analog domain. The final result is a high-speed clock that can be any one of 128 phases of the high-speed transmit clock used for serialization of the parallel transmit data. This produces a very low jitter clock with very fine phase resolution.

A phase interpolator operates on the line rate and is present in all 7 series transceivers. It takes in an input clock and produces an output with a fine phase shift. The phase interpolator in the 7 series FPGA transceiver transmitter has a phase control port that can be dynamically accessed to enable fine frequency control. The phase control port update rate is dependent on the transceiver and clock speed, with greater than 200 MHz being achievable in Virtex-7 devices.

The maximum achievable frequency offset is a combination of phase interpolator step size and update rate. An Excel spreadsheet is provided with the design download to help you estimate achievable performance with your own system parameters.

PICXO DPLL

The phase interpolator controlled crystal or Xtal oscillator (PICXO) parameters must be set appropriately to generate a transceiver channel locked to a reference signal. The DPLL can be analyzed using standard methods from a derivation of the transfer function outlined in this section.

The PICXO DPLL circuit, for analysis purposes, is considered to be in three functional blocks:

1. Phase frequency detector

   The phase-frequency detector measures the phase difference between the reference (R) and the PICXO (V) clocks and produces an error output. As the DPLL is second order when locked, this error output is driven to zero. It has a transfer function that is defined in units of radian^{-1} and gain, G_{PD}.

2. Second order loop filter

   The second-order loop filter consists of proportional and integral paths with digital gains defined by the terms G_1 and G_2. The output represents the required tune value for the oscillator.

3. Numerically controlled oscillator

   The numerically controlled oscillator function is performed by the transmit phase transceiver's interpolator block, the phase accumulator, and the sigma-delta modulator. This has units of radians/s and gain G_{PICXO}.

These are configured in a standard DPLL configuration shown in Figure 7.

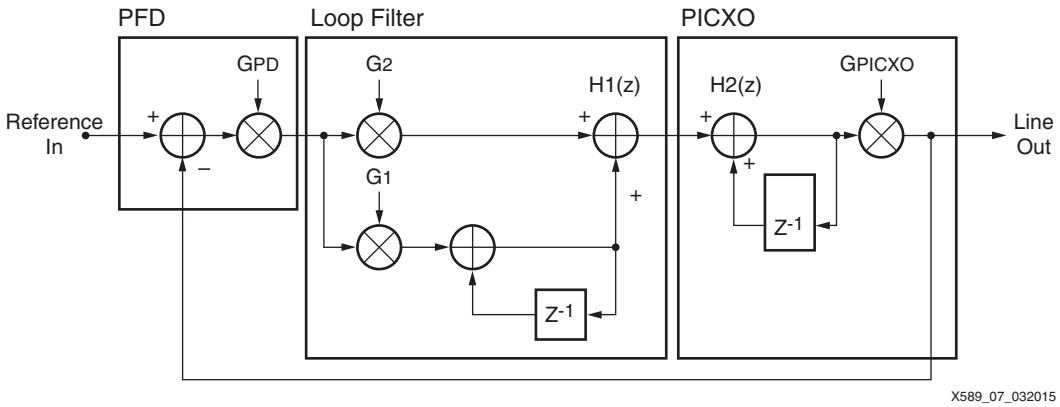


Figure 7: **PICXO DPLL Digital Equivalent**

The transfer of the reference input clock to the line output data is represented by the function in Equation 6. This allows the clock cleaning and tracking of the all-digital VCXO replacement to be exactly controlled by your application.

$$H(z) = \frac{H1(z)H2(z)G_{PD}}{1 + H1(z)H2(z)G_{PD}} \quad \text{Equation 6}$$

with:

$$H1(z) = \frac{(g1 + g2)z - g2}{(z - 1)} \quad \text{Equation 7}$$

$$H2(z) = \frac{z(G_{PICXO})}{(z - 1)} \quad \text{Equation 8}$$

The gain parameters g1 and g2 are defined as:

$$g1 = \frac{2^{(G1 - 2)}}{2^{28}} \quad \text{Equation 9}$$

$$g2 = \frac{2^{(G2 + 1)}}{2^{28}} \quad \text{Equation 10}$$

And G_{PD} and G_{PICXO} are defined as follows for GTX transceivers where the phase interpolation is controlled through the DRP:

$$G_{PD} = \frac{CLK(Hz) \times bitrate(bps) \times 0.4 \times 10^{-9}}{V \times CE_{DSP} \times 2\pi} \quad \text{Equation 11}$$

$$G_{PICXO} = \frac{CE_{PI} \times ACC\_STEP \times 2\pi}{64 \times TXOUT\_DIV \times 2^{21}} \quad \text{Equation 12}$$

Where:

$$CE_{PI} = \frac{CLK(Hz)}{wr_{TIME}} \quad \text{Equation 13}$$

$$CE_{DSP} = \frac{CE_{PI}}{\text{CE\_DSP\_RATE}} \quad \text{Equation 14}$$

The previous equations use these constants:

- For 7 series FPGAs and Zynq-7000 AP SoC GTX transceivers, PI_{res} = 64 and wr_{TIME} = 6
- For 7 series FPGAs and Zynq-7000 AP SoC GTH and GTP transceivers, PI_{res} = 64 and wr_{TIME} = 2

and these variables:

- bitrate: The final serial interface line rate
- CLK: The PICXO and/or the DRP clock sourced from TXOUTCLK
- TXOUT_DIV attribute: The main TXPLL divider setting that sets the operating rate range for the transceiver
- ACC_STEP: The TXPI step size of the system selected by the user

The Excel spreadsheet tool included in the PICXO design file package allows you to estimate the PICXO response when setting the configurable parameters listed above. The PICXO DPLL allows complete flexibility with settings. Therefore, you should understand the performance trade-offs in PLL operation.

For optimum jitter and cleaning performance, it is recommended that the PICXO DPLL bandwidth be less than 100 Hz. Higher tracking bandwidths can be achieved, however, with some increase in jitter. It might be desirable to have a high bandwidth to acquire lock, then switching subsequently to a lower cleaning bandwidth. This is known as fast acquisition for the DPLL.

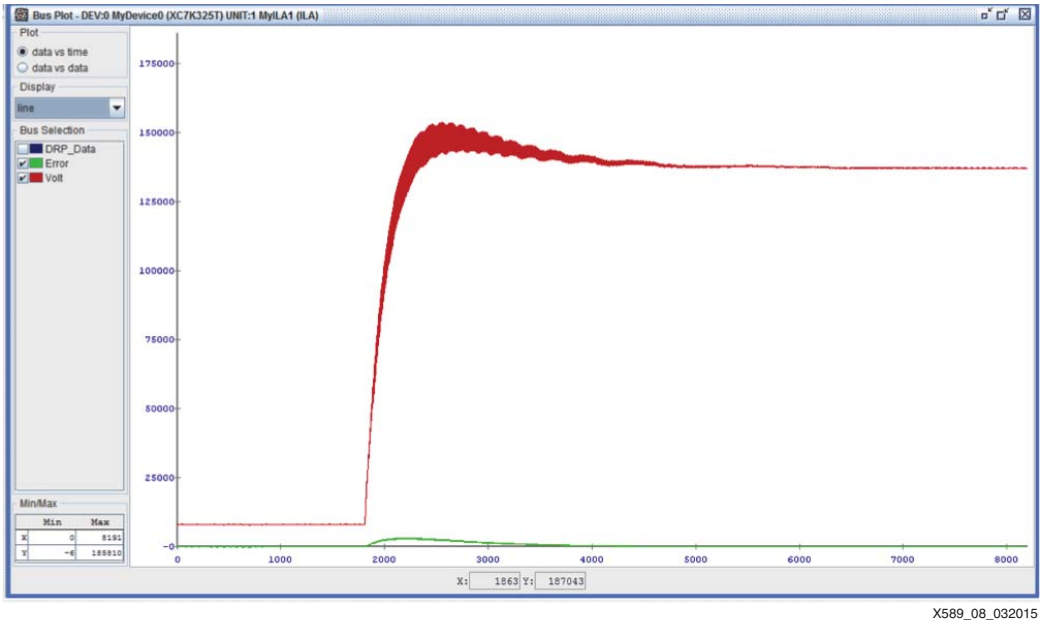
The DPLL architecture allows on-the-fly changes to the G1 and G2 values to support this while not losing phase lock. On-the-fly changes can be supported in user logic by applying variable G1 and G2 values. It might be appropriate to monitor the error output from the DPLL as one method to ascertain a suitable point at which to switch gain values.

PICXO Measurements and Performance

This section includes sample measurements of an example PICXO design implemented on the KC705 board.

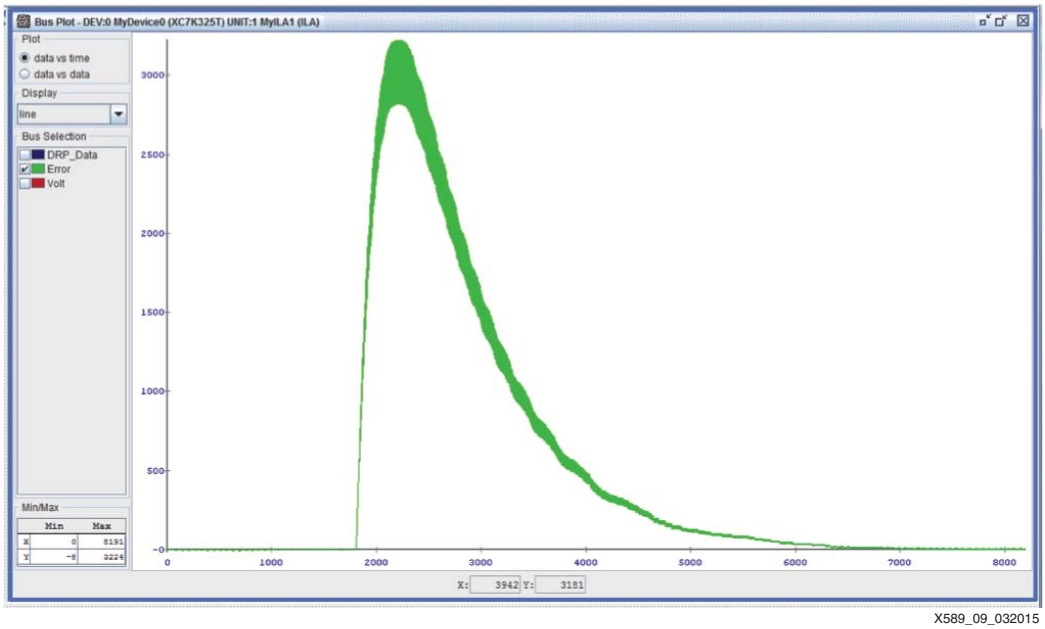
Figure 8 and Figure 9 demonstrate the DPLL error and virtual voltage during the locking process when a step change in frequency is applied. The error range is $\pm 2^{19}$, the virtual voltage $\pm 2^{20}$. In this case, the virtual voltage is settling ~140000, indicating the PICXO is generating a positive offset relative to the local GTX transceiver reference fixed source (GTX REFCLK frequency) of approximately +10 ppm. As the local GTX transceiver reference drifts in frequency, the output remains locked to the incoming data. This allows retransmission with no external VCXO and it performs jitter cleaning of the recovered signal. The time unit is CE_{DSP} clocks.

The plots in Figure 8 and Figure 9 show the Error and Volt outputs when a step frequency change of ~9 ppm is applied to the PICXO.



X589_08_032015

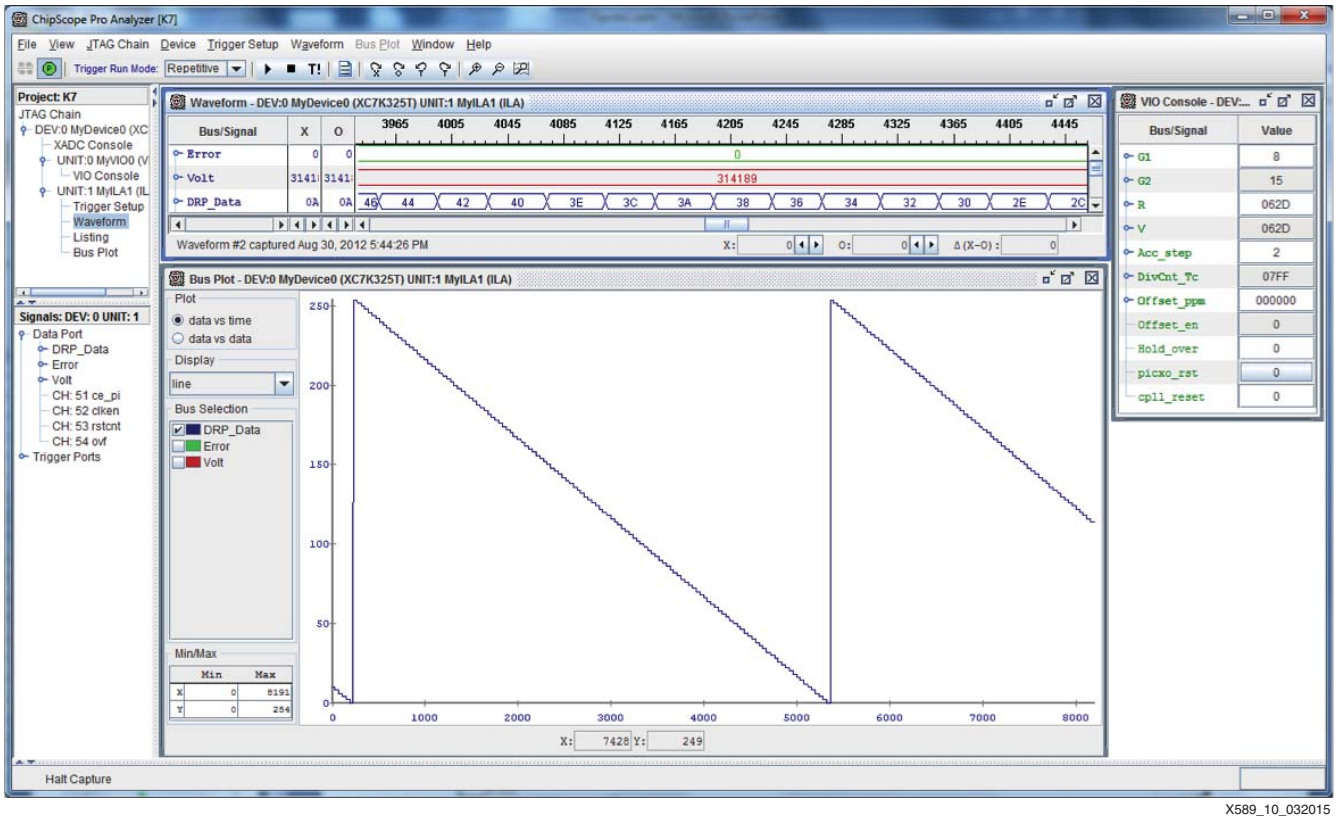
Figure 8: **Plot of PICXO DPLL Volt and Error During Step Change**



X589_09_032015

Figure 9: **Plot of PICXO DPLL Error During Step Change**

Figure 10 shows the transmit phase interpolator codes being written from the PICXO in the GTX transceiver. This demonstrates the frequency offset generation in progress. Figure 10 also reflects the direct phase shifting of the transmitter PLL at its operating frequency. The phase rotation shown results in a continuous phase ramp at the line rate. As a positive frequency is being generated, the phase is being continuously subtracted to generate shorter periods. The time unit is CE_{PI} clocks.



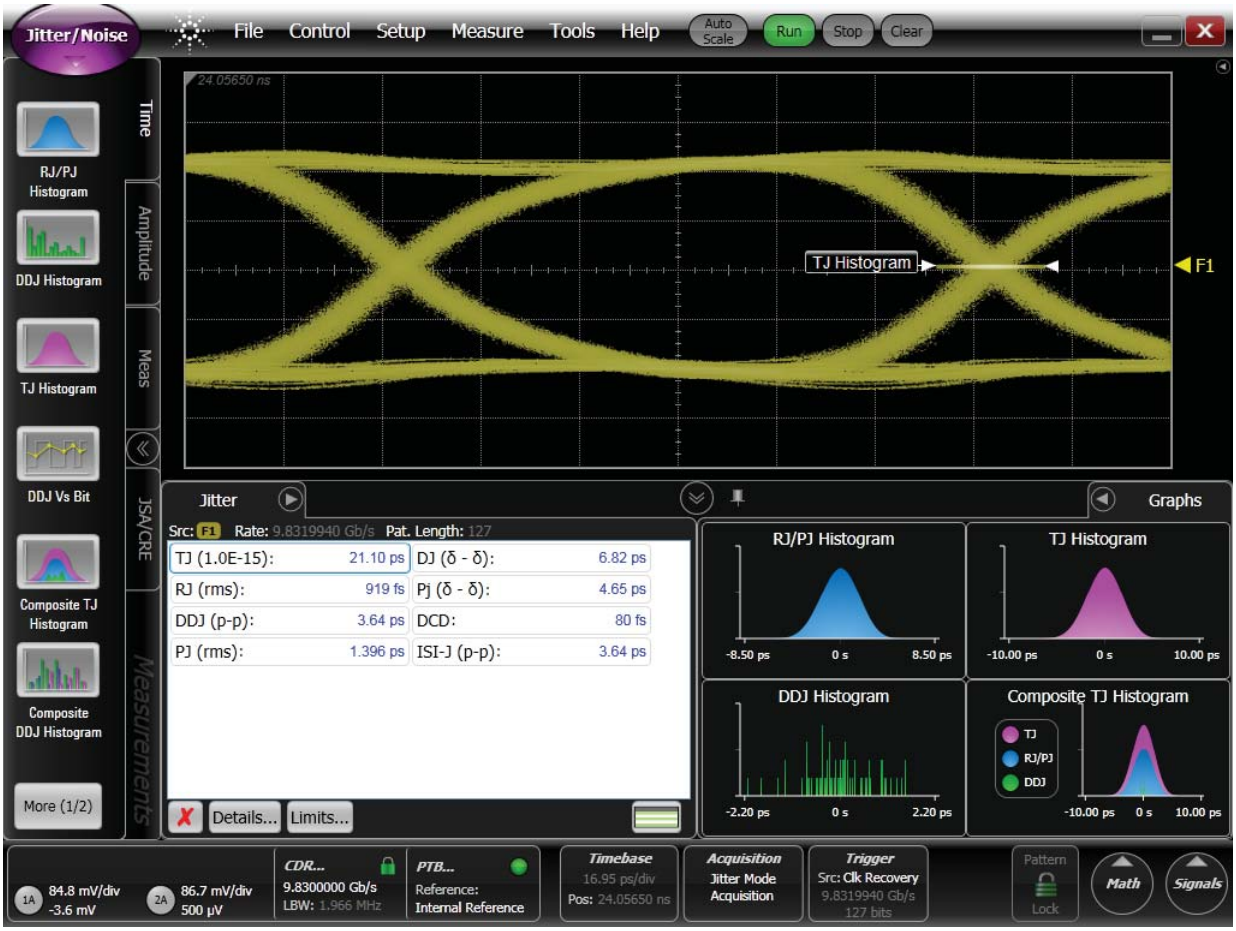
X589_10_032015

Figure 10: **Plot of PICXO Transmit Phase Interpolator Control When Locked**

When using the transmit phase interpolator in the GTX transceiver a general expectation is that the transmitter jitter will increase between 0.01 and 0.03 UI pk-pk due to the phase stepping and rotational nature of the modulator.

Figure 11 shows a Virtex-6 FPGA example waveform and Figure 12 shows a Kintex-7 FPGA waveform being operated in a jitter cleaning mode at a 9.83 GBs rate. When operating in systems where margins are reduced, Xilinx recommends performing some evaluations.

Figure 11: **Virtex-6 FPGA GTX Transceiver Data Output at 2.488 GB Generating +20 ppm Offset**



X589_12_032015

Figure 12: **Kintex-7 FPGA GTX Transceiver Data Output at 9.83 GB Operating as a PICXO-based Jitter Cleaner at +32 ppm Offset**

Figure 13, and Figure 14 demonstrate the transfer bandwidth of the PICXO. The loop filter programmability is exercised to show how the transfer function can be adjusted for varying user requirements for bandwidth and damping.

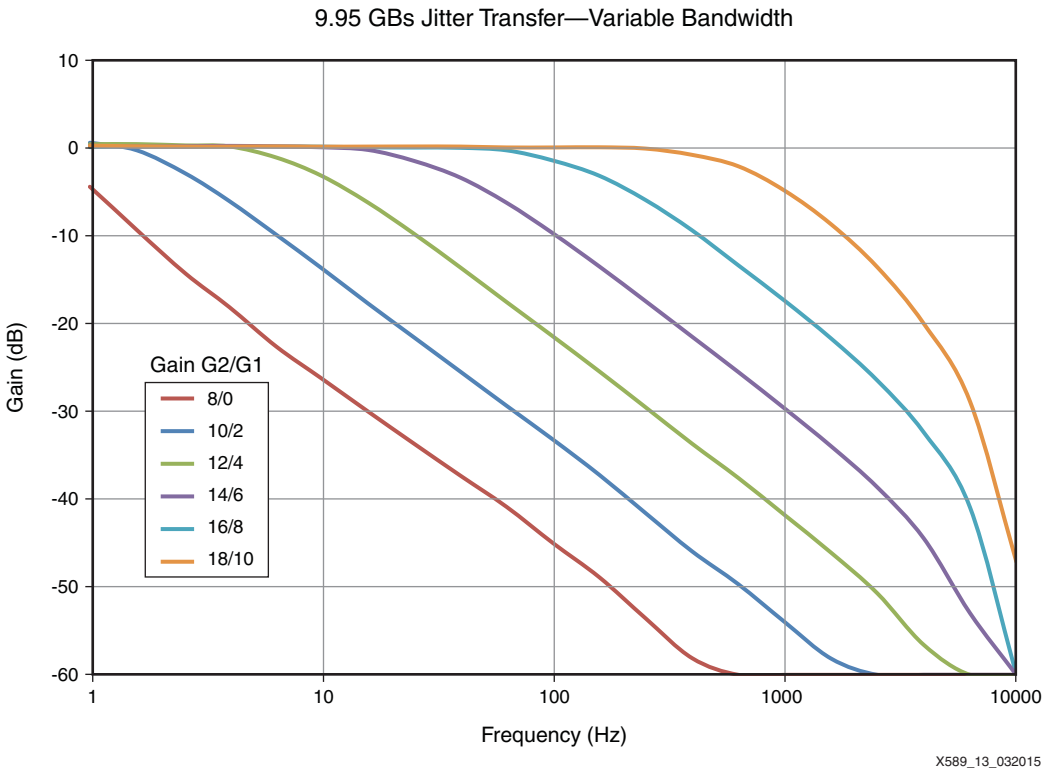


Figure 13: **Kintex-7 FPGA GTX Transceiver 10 GB Jitter Transfer Measurements—Variable Bandwidth**

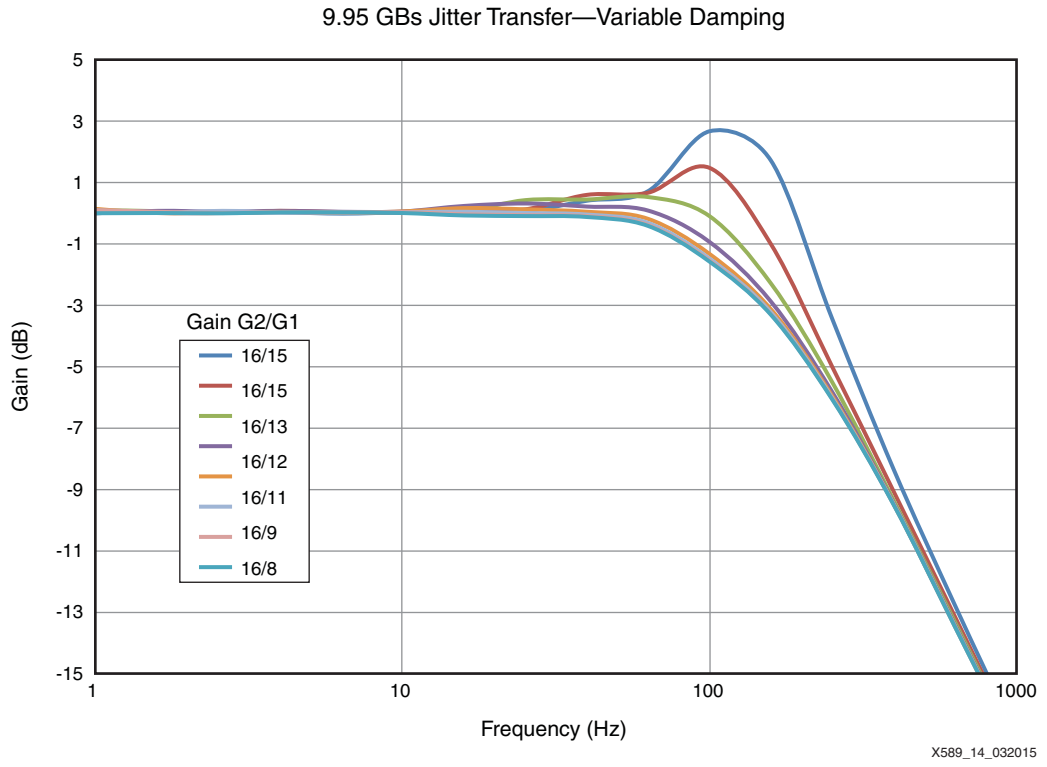
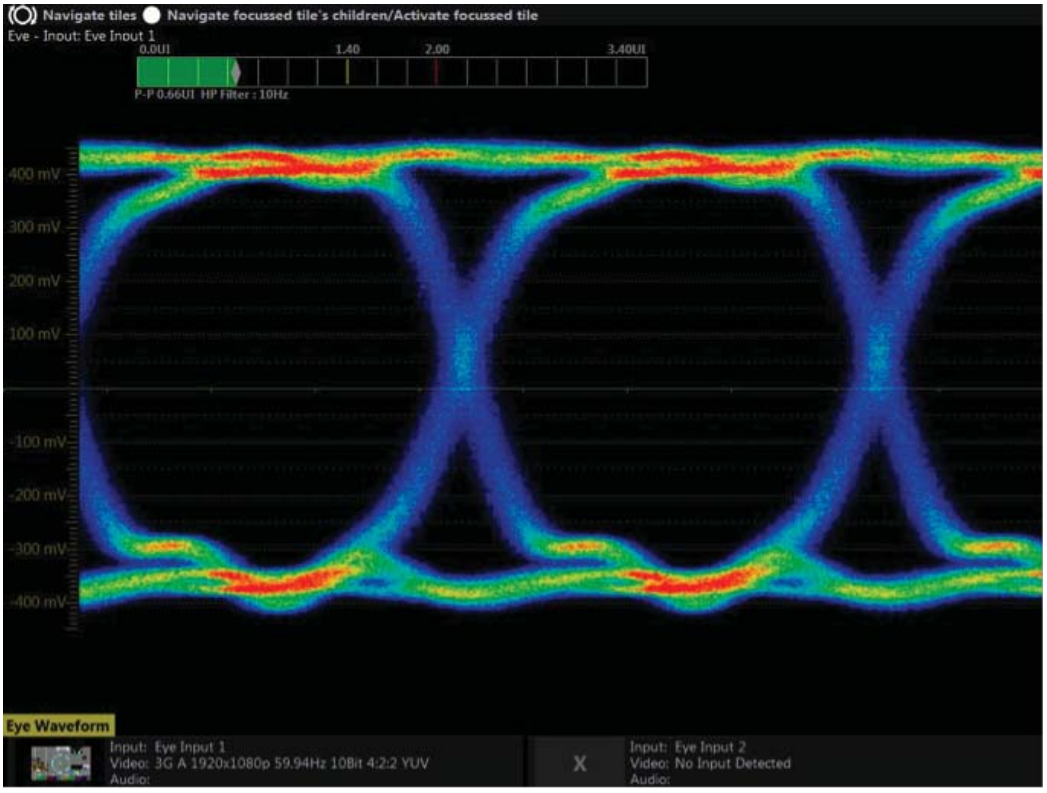


Figure 14: **Kintex-7 FPGA GTX Transceiver 10 GB Jitter Transfer Measurements—Variable Damping**

For broadcast equipment, 270 Mb/s, 1.485 Mb/s, and 2.97 Mb/s are standard rates for SD-SDI, HD-SDI, and 3G-SDI. It is a challenge to meet the broadcast jitter requirements of all 3G-SDI formats. One of the 3G-SDI formats is 3G Level A. The examples in Figure 15 and Figure 16 are measurements showing the system passing with margin for 3G Level A SDI (3G Level A 1920 x 1080p at 59.94 Hz) in both 10 Hz and 100 KHz jitter measurement bandwidths, respectively. In this design, the VCXO and re-clocking function is incorporated completely in the FPGA using the PICXO scheme.



X589_15_032015

Figure 15: **ML605 with SDI FMC Board for a 3G Level A SDI Output of a Triple-Rate SDI Pass-Through Design (at 10 Hz Jitter)**

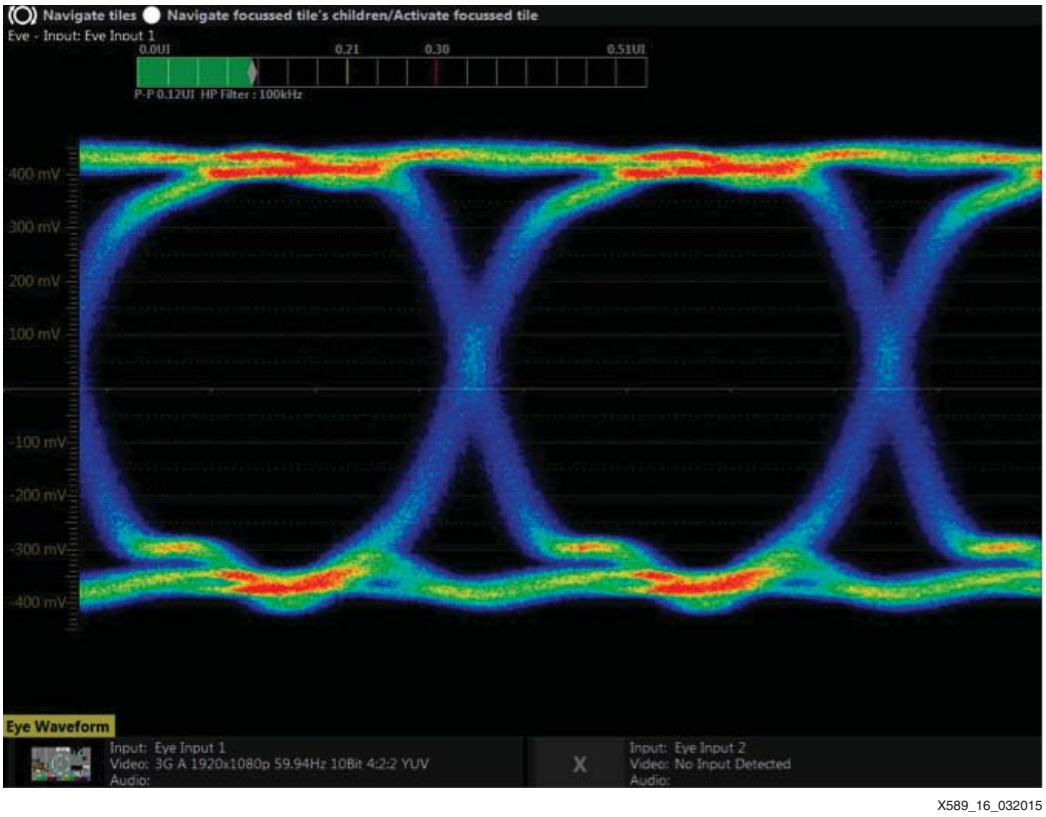


Figure 16: **ML605 with SDI FMC Board for a 3G Level A SDI Output of a Triple-Rate SDI Pass-Through Design (at 100 KHz Jitter)**

PICXO Architecture Overview

Instead of using an external VCXO, as described in VCXO Replacement Theory with its challenges, a complete digital PLL and clock cleaner can be created using the phase interpolator in the transmit serial/deserializer as a phase interpolator controlled crystal or Xtal oscillator (PICXO). The PICXO macro operation for GTX transceivers is shown in the functional block diagram of Figure 17.

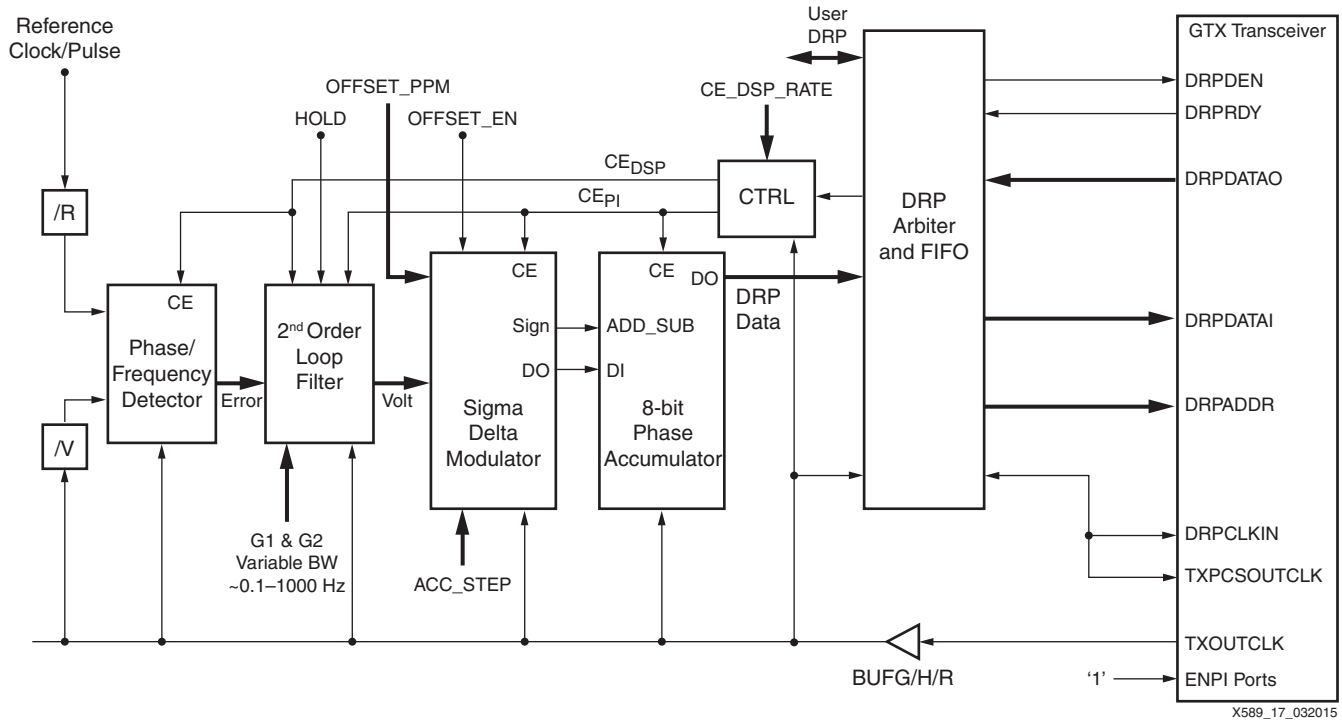


Figure 17: **PICXO Macro Functional Block Diagram for GTX Transceivers**

For Kintex-7 FPGAs, the dynamic reconfiguration port (DRP) arbiter/FIFO and control blocks manage the clock and DRP data interfaces between the GTX transceiver, the PICXO DPLL, and the User DRP.

A typical use model for a DRP operation is that a user application can, if required, program GTX transceiver DRP parameters prior to operation. The detailed operation is described in Designing with PICXO, page 21.

The phase in the GTX transceiver is under direct DRP control from the PICXO circuit consisting of phase accumulator, sigma delta modulator and loop filter, and phase detector components.

The phase accumulator tracks the current phase of the phase interpolator and increments or decrements the phase based on input from the sigma-delta modulator block. Incrementing or decrementing phases directly results in a negative or positive frequency offset.

The required fine frequency control is achieved by the sigma-delta modulation block driven by a second order DPLL consisting of filter and phase detector with user-configurable loop parameters and comparison frequencies for maximum flexibility.

The PICXO operation is synchronous with the DRP clock. The maximum phase interpolator update rate (DRP CLK/6) is the clock enable rate for the sigma-delta modulator and accumulator CE_{PI}, shown in Figure 17. The DPLL runs at a sub-rate CE_{DSP}, the clock enable rate for the phase/frequency detector and second-order loop filter (in Figure 17). This allows the sigma-delta modulator to run with high resolution and allows usable DPLL coefficients for low-frequency clock cleaning.

The reference design circuit uses one BUFG/BUFH/BUFR per line rate generated. When locked, this clock is synchronous with the reference clock and can be used for other user downstream logic.

The PICXO macro operation for GTH and GTP transceivers is shown in the functional block diagram of Figure 18.

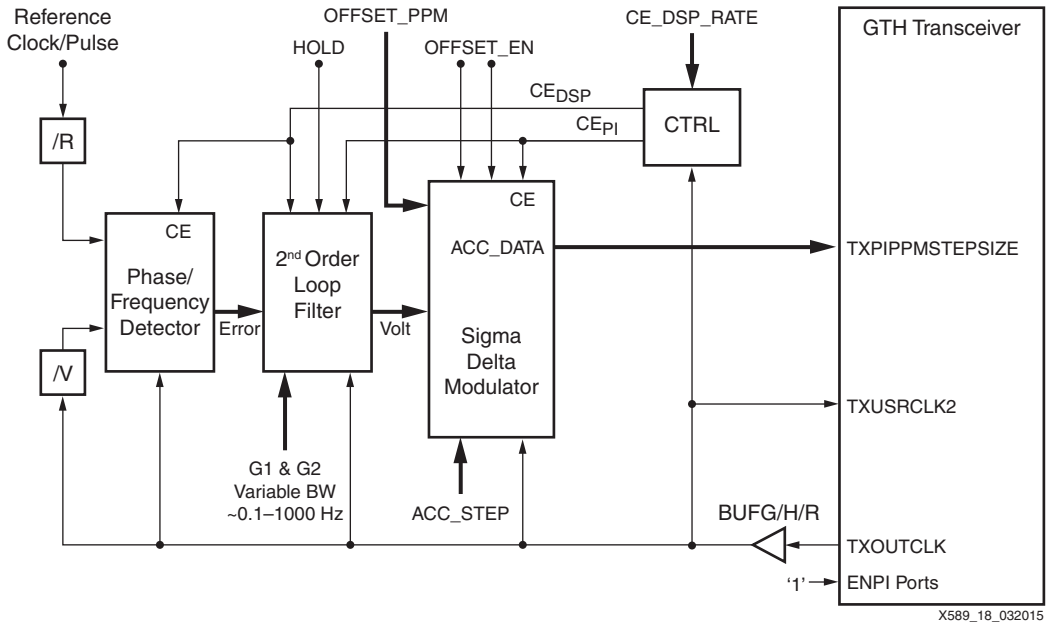


Figure 18: **PICXO Macro Functional Block Diagram for GTH and GTP Transceivers**

For Virtex-7 FPGAs with GTH transceivers and Artix-7 FPGAs with GTP transceivers, the principle of the PICXO operation is the same as for GTX transceivers. However, the control of the GTH and GTP TX PI is through dedicated ports.

The PICXO macro provides phase increment information directly to the GTH/GTP transceiver TXPPMSTEPSIZE input ports. This bus applies phase increment or decrement information to the internal GTH/GTP TXPI phase accumulator. This offers several advantages over the DRP access method:

- The DRP does not need to be arbitrated between the user functions and the PICXO.
- The dedicated ports allow a faster update rate for the TXPI.

Designing with PICXO

Physical Interface

Table 1 through Table 5 show the port definitions.

Table 1: **Clocks, Reset, and Interface to the Transceiver Ports**

Signal Name	Direction	Description
RESET_I	Input	Synchronous reset. Active-High.
REF_CLK_I	Input	Reference clock, can be any clock (local, BUFG, pulse, etc.)
TXOUTCLKPCS_I	Input	Unused. Connect to 0, 1, or TXOUTCLK_I.
TXOUTCLK_I	Input	Connects to TXOUTCLK of the GTX serial transceiver via a BUFG/BUFH/BUFR.
DRPEN_O	Output	Connects to the DRPEN port of the GTX serial transceiver.
DRPWEN_O	Output	Connects to the DRPWE port of the GTX serial transceiver.
DRPDO_I [15:0]	Input	Connects to DRPDO of the GTX serial transceiver. Internally connects directly to DRPDATA_USER_O[15:0].
DRPDATA_O [15:0]	Output	Connects to the DRPDI port of the GTX serial transceiver.
DRPADDR_O [8:0]	Output	Connects to the DRPADDR port of the GTX serial transceiver.
DRPRDY_I	Input	Connects to the DRPRDY port of the GTX serial transceiver.

Table 2: **DRP User Port (GTX Transceivers Only)**

Signal Name	Direction	Description
DRP_USER_REQ_I	Input	Asserted to request DRP port access. Active-High signal.
DRP_USER_DONE_I	Input	Reset DRP arbiter.
DRPEN_USER_I	Input	Same functionality as DEN GTX serial transceiver port [Ref 1].
DRPWEN_USER_I	Input	Same functionality as DWEN GTX serial transceiver port [Ref 1].
DRPADDR_USER_I [8:0]	Input	Same functionality as DADDR GTX serial transceiver port [Ref 1].
DRPDATA_USER_I [15:0]	Input	Same functionality as DI GTX serial transceiver port [Ref 1].
DRPRDY_USER_O	Output	This signal is a mirror of DRDY in the TXOUCLK domain. Indicates data present on DRPDO is valid [Ref 1].
DRPDATA_USER_O[15:0]	Output	Directly connected to DRPDO_I[15:0].
DRPBUSY_O	Output	Indicates that the DRP port is not available to the user. Active-High signal.

Table 3: **TXPI Port (GTH and GTP Transceivers Only)**

Signal Name	Direction	Description
ACC_DATA[4:0]	Output	Connects to TXPIPPMSTEPSIZE[4:0] of the GTH or GTP transceiver.

Table 4: **Debug Ports**

Signal Name	Direction	Description
ERROR_O [20:0]	Output	Output of phase detector. Signed number.
VOLT_O[21:0]	Output	Output of low-pass filter. Signed number.
DRPDATA_SHORT_O[7:0]	Output	Output of the accumulator. Unsigned number.
CE_PI_O	Output	Clock enable for accumulator.
CE_PI2_O	Output	Clock enable for low pass filter and DAC.
CE_DSP_O	Output	Reset phase detector counters, load phase detector error into the low-pass filter.
OVF_PD	Output	Overflow in the phase detector.
OVF_AB	Output	Saturation of the low-pass filter inputs.
OVF_INT	Output	Saturation of the low-pass filter integrator.
OVF_VOLT	Output	Saturation of the low-pass filter output.

Table 5: **PICXO Loop Parameters**

Signal Name	Direction	Description
G1[4:0]	Input	Filter linear path gain: Range `0` to `x12h`.
G2[4:0]	Input	Filter integrator path gain: Range `0` to `x14h`. (1)
R[15:0]	Input	Reference divider: Range `0` to `65535`. Divides by R+2.
V[15:0]	Input	TXOUTCLK_I divider: Range `0` to `65535`. Divides by V+2.
ACC_STEP[3:0]	Input	PICXO step size: Range `1` to `15` (`0` = no step).
CE_DSP_RATE[15:0]	Input	DSP divider: Default `07FF`. Controls CE_DSP rate.
VSIGCE_I	Input	Clock enable of the TXOUTCLK_I divider. Connects to 1 for normal operation.
VSIGCE_O	Output	Reserved: Floating.
RSIGCE_I	Input	Clock enable of Reference divider. Connects to 1 for normal operation.
C_I[7:0]	Input	Reserved: Connects to `0`.
P_I[9:0]	Input	Reserved: Connects to `0`.
N_I[9:0]	Input	Reserved: Connects to `0`.
OFFSET_PPM[21:0]	Input	Direct frequency offset control. Signed number. OFFSET_PPM overwrites the output of the low-pass filter (VOLT_O) when OFFSET_EN is High.
OFFSET_EN	Input	Enables direct frequency offset control input. Active-High: Enables OFFSET_PPM input to overwrite output of low-pass filter (Volt).
HOLD	Input	Holds low-pass filter output value (Volt). Clock enable of Volt that stops Volt to the latest known ppm.
DON_I	Input	Potential jitter reduction. Active-High.

Notes:

1. Incorrect G1 and G2 values (either too high or low) can stop the DPLL from locking. Normally the target loop bandwidth should be in the region of a few to several hundred Hz with G2 ≥ 1. It should also be understood G values are 2^N the set value.

Interface Operation

General Operation

The PICXO parameters (V, R, ACC_STEP, CE_DSP_RATE) can affect the PICXO lock if changed, therefore they are considered pseudo-static inputs. The gains G1 and G2 can be changed without loss of lock. All input and output signals to/from the PICXO are synchronous to TXOUTCLK_I except REF_CLK_I and R. Figure 19 shows the timing dependency between TXOUTCLK_I and the main debug outputs.

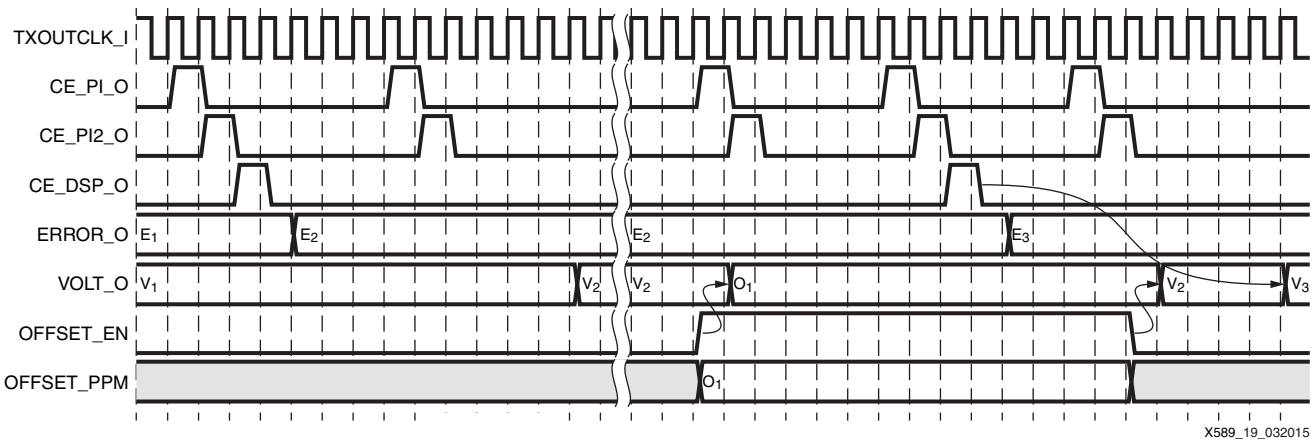


Figure 19: **Timing Waveforms of Main Debug Outputs**

Reset Considerations

The PICXO main reset RESET_I requires a minimum of eight TXOUTCLK_I cycles to reset the PICXO correctly. When applied, RESET_I resets all blocks, including the phase detector, low-pass filter, and DRP arbiter. When releasing RESET_I, the first phase detector output (ERROR_O) is zero, and the first word written in the transceiver phase interpolator is zero.

The PICXO second reset DRP_USER_DONE_I resets the DRP arbiter only. In 7 series FPGAs, a reset of the DRP arbiter is not necessary to restart DRP operation.

The transceiver TX PMA reset sequence must be completed before the PICXO reset is released for operation.

7 Series FPGA Transceiver Clocking

The primary clocking scheme is detailed in Figure 20. The transceiver TXOUTCLK connects to a BUFG that drives the PICXO inputs clocks TXOUTCLK_I and TXOUTCLKPCS_I, as well as the GTX DRP clock DCLK.

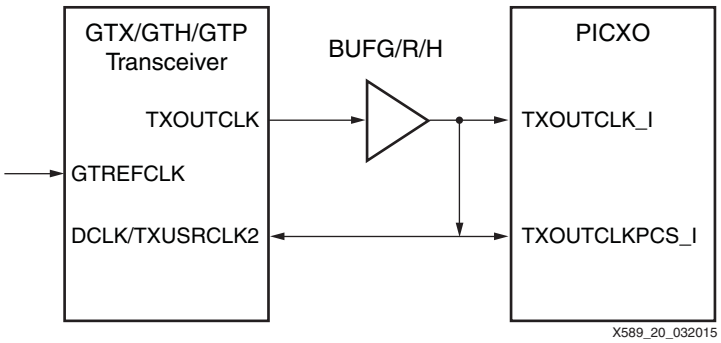


Figure 20: **7 Series FPGA Primary PICXO Clocking Scheme**

Notes relevant to Figure 20:

1. For 7 series FPGA GTX transceivers, TXOUTCLK_I must be the same clock as DCLK.
2. For 7 series FPGA GTH/GTP transceivers, TXOUTCLK_I must be the same clock as TXUSRCLK2.

A secondary clocking scheme is detailed in Figure 21. This clocking scheme can be used when TXOUTCLK exceeds the GTX DRP clock specifications. In this case, the GTX DRP clock must be an integer divisor of the TXOUTCLK frequency.

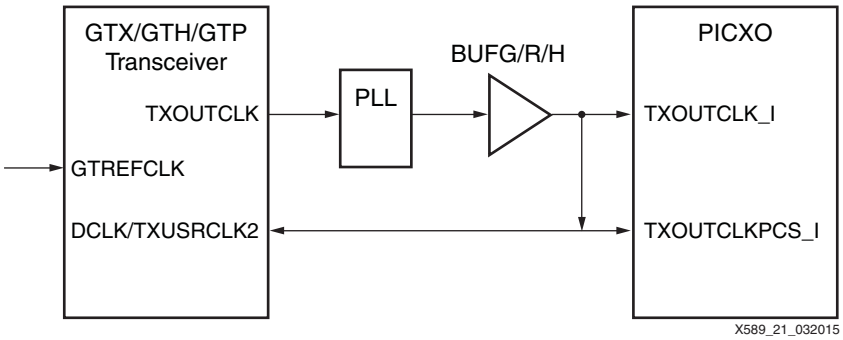


Figure 21: **7 Series FPGA Secondary PICXO Clocking Scheme**

Notes relevant to Figure 21:

1. For 7 series FPGA GTX transceivers, TXOUTCLK_I must be the same clock as DCLK.
2. For 7 series FPGA GTH/GTP transceivers, TXOUTCLK_I must be the same clock as TXUSRCLK2.

7 Series FPGA GTX DRP User Port Operation

The 7 series FPGA PICXO DRP user interface (Table 2) must be synchronous to TXOUTCLK_I. The 7 series FPGA GTX DRP interface (Table 1) is synchronous to TXOUTCLK_I. The DRP arbiter is implemented with a finite state machine (FSM). The FSM reset is triggered by DRP_USER_DONE_I or RESET_I. However, the user DRP port can still operate as normal when DRP_USER_DONE_I or RESET_I are High.

To operate the 7 series FPGA PICXO DRP user port, the application asserts the DRP_USER_REQ_I signal and waits for DRP_BUSY_O to transition Low. After DRP_BUSY_O is Low, the application can operate the DRP USER port as per the GTX DRP specification [Ref 1]. The application must keep DRP_USER_REQ_I asserted during a DRP transfer. After the application is finished with the DRP accesses, it drives DRP_USER_REQ_I Low. The PICXO regains control of the DRP port and DRPBUSY_O goes High (Figure 22).

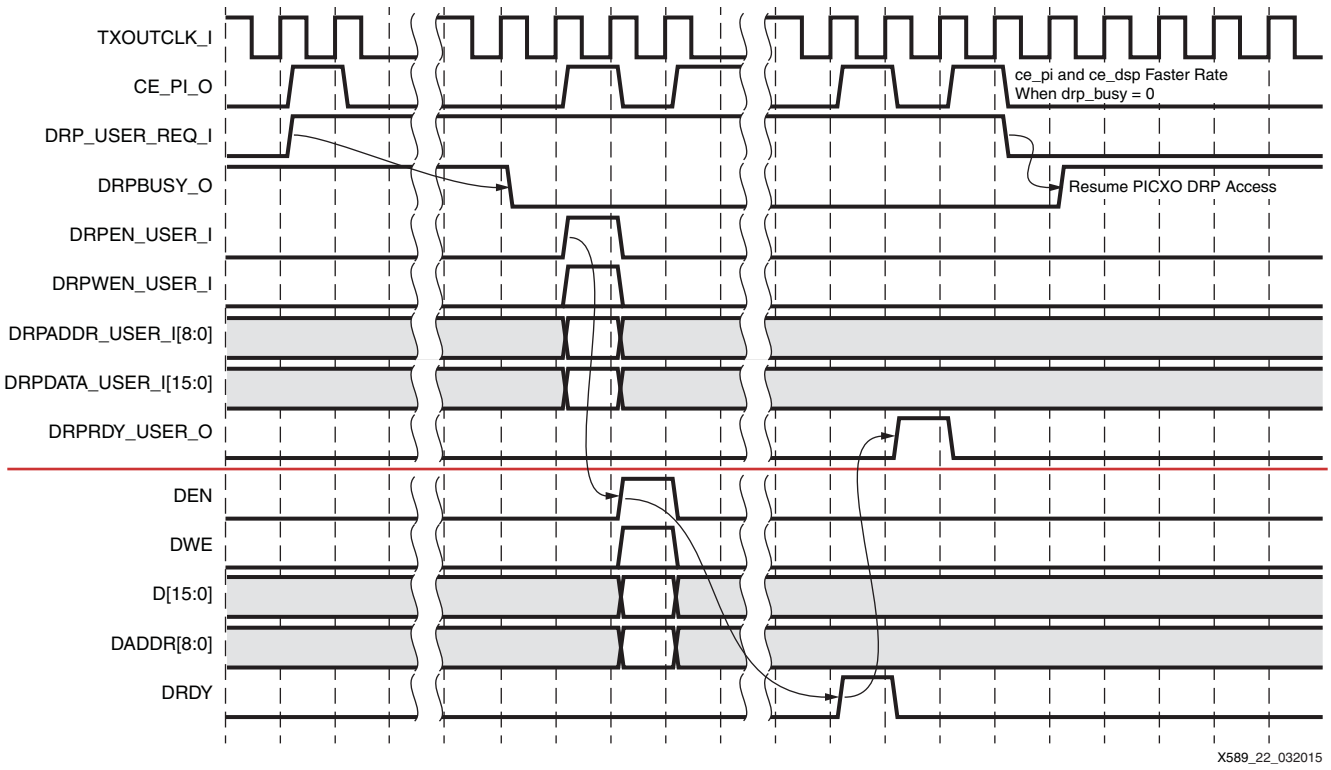


Figure 22: **7 Series FPGA GTX DRP User Port Operation**

HOLD Input Operation

The HOLD input is a clock enable to the low-pass filter integrator and output (VOLT_O). While HOLD is High, the phase detector continues to operate as normal. When HOLD returns to Low, the low-pass filter output is not synchronized anymore with the phase detector. Figure 23 illustrates this behavior.

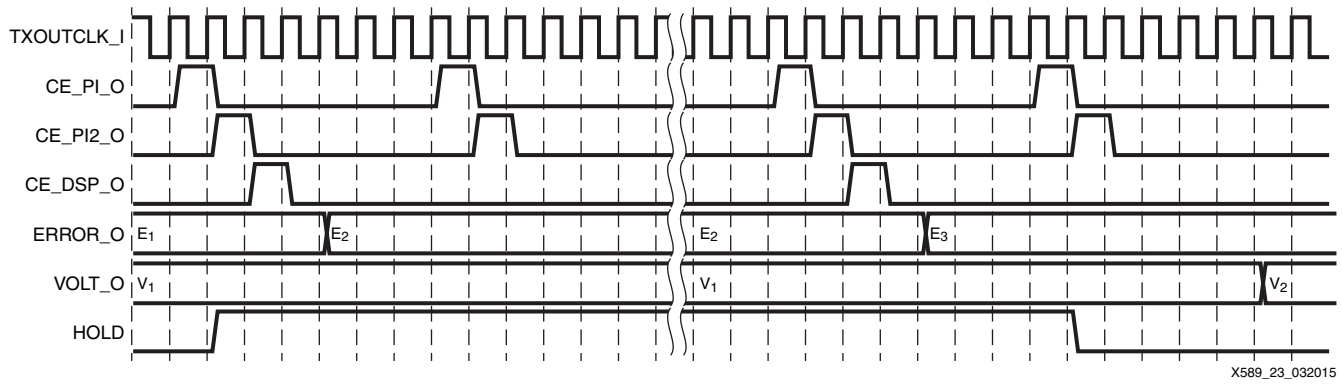


Figure 23: **HOLD Input Operation**

Direct Offset Control

OFFSET_PPM and OFFSET_EN allow direct control of the frequency offset. When OFFSET_EN is High, the output of the low-pass filter (VOLT_O) takes the OFFSET_PPM value. During this time, the phase detector and low-pass filter integrator operate normally. When OFFSET_EN returns to Low, the output of the low-pass filter (VOLT_O) takes the current value calculated by the phase detector and low-pass filter. Figure 24 illustrates this behavior.

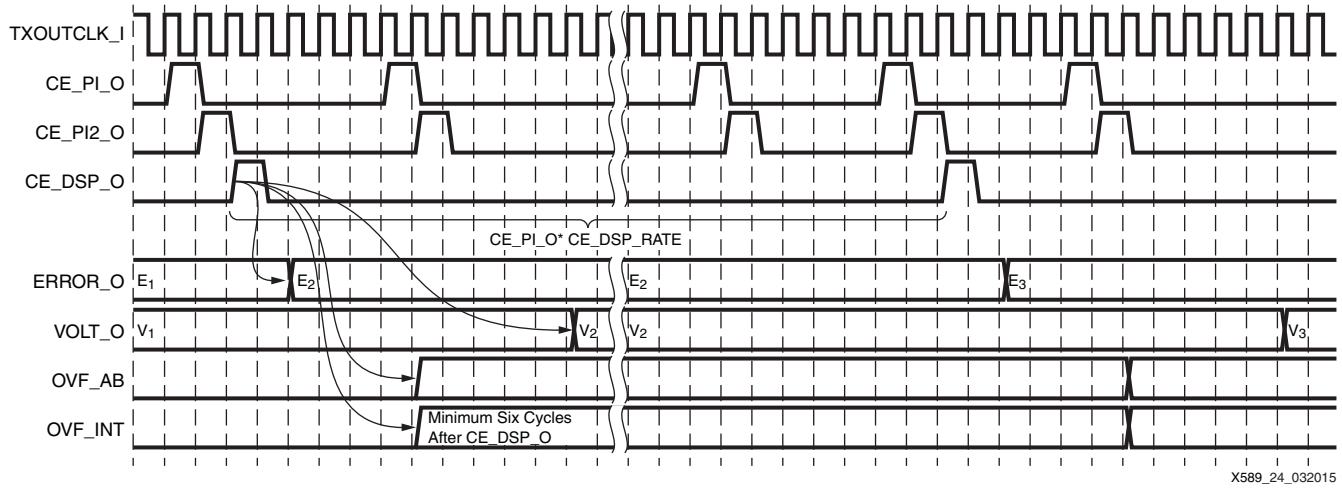


Figure 24: **Direct Offset Control**

Implementation

Vivado Tools Implementation

The PICXO design is delivered as a custom IP. To add the design to a project:

1. Unzip the file in a location.
2. Add the IP repository to the project: **Tools > Project Options**, select **IP** on the left pan, click **Add Repository**, and select the **XAPP589_picxo** folder.
3. Select the IP catalog. The PICXO IP is under **FPGA Features and Design > IO Interfaces**.
4. Right-click **XAPP589_picxo** and select **Customize IP**.
5. Select the IP module name and the type of GT. Click **OK.**
6. The example design can be generated by selecting the IP source, doing a right-click, and **generate example design.**

The GTP/GTX/GTH transceiver associated with the PICXO must be constrained to a specific location. Period constraints are necessary on TXOUTCLK_I and REFCLK_I.

Mandatory Conditions and Limitations

7 Series FPGA GTX Transceiver

- Transmit buffer bypass is not supported.
- TXDLY_LCFG[2] and PCS_RSVD_ATTR[1] must be set to 1.
- The Kintex-7 FPGA ports TXPHALIGN, TXPHALIGNEN, and TXPHOVRDEN must be connected to 1.
- TXPHDLYPD must be connected to 0.
- TXOUTCLKSEL must be set to TXOUTCLKPMA (010)

7 Series FPGA GTH/GTP Transceiver

- Transmit buffer bypass is not supported.
- GTH/GTP port TXPIPPMEN must be connected to 1.
- GTH/GTP port TXPIPPMSEL must be connected to 1.
- GTH/GTP attribute TXPI_SYNFREQ_PPM must be set to 001.
- TXOUTCLK and TXUSRCLK2 must be the same clock. TXOUTCLKSEL must be set to 010 or 001.

Reference Design

The reference design files are based on the 7 series transceiver wrapper v2.1 [Ref 2]. The designs target the KC705, AC701, ZC706, and VC709 development platforms. They loopback the receive data to the transmitter. The PICXO instance locks the transmitter to the recovered clock RXRECLK.

The output error_o of the phase/frequency detector can be captured when CE_DSP_O is High to monitor the PICXO response. When locked, ERROR_O should oscillate around 0 (see Figure 9).

Simulation of the example design is not supported. The drp_arbiter source code is provided to enable functional simulation of DRP user access.

You can download the reference design files for this application note at www.xilinx.com/member/vcxoff/index.htm (see *Download the VCXO Removal Reference Design for 7 Series FPGAs)*.

Table 6 shows the reference design matrix.

Table 6: **Reference Design Matrix**

Parameter	Description
General	
Developer names	David Taylor, Matt Klein, Vincent Vendramini
Target devices	Kintex-7 XC7K325T FFG900-1 Virtex-7 XC7VX690T FFG1761-2 Artix-7 XC7A200T FBG676C-2 Zynq-7000 XC7Z045 FFG900-2
Source code provided	Yes
Source code format	VHDL
Design uses code/IP from existing Xilinx application note/reference designs, CORE Generator™ tool, or third party	Yes
Simulation	
Functional simulation performed	No
Timing simulation performed	No
Test bench used for functional and timing simulations	No
Test bench format	N/A
Simulator software tools/version used	N/A
SPICE/IBIS simulations	N/A
Implementation	
Synthesis software tools/version used	Vivado® Design Suite 2015.1
Implementation software tools/version used	Vivado Design Suite 2015.1
Static timing analysis performed	Yes

Table 6: **Reference Design Matrix *(Cont'd)***

Parameter	Description
Hardware Verification	
Hardware verified	Yes
Hardware platform used for verification	KC705, VC709, ZC706, and AC701

Table 7 shows the device utilization table for the reference design.

Table 7: **Device Utilization and Performance for Reference Design (Vivado Design Suite 2015.1)**

Target Devices	Artix-7 FPGAs (One GTP Transceiver)	Kintex-7 FPGAs (One GTX Transceiver)	Virtex-7 FPGAs (Four GTH Transceivers)
	Full Design	Full Design	
Slice LUTs	4,350	2,993	12,203
Slice registers	6,054	3,966	15,696
Occupied slices(1)	2,077	1,447	5,533
Block RAM	14	14	56
BUFG/BUFHCE	6/0	4/0	10/3
GTP/GTX/GTH	1	1	4
MMCM	0	0	0

Notes:

1. The number of occupied slices can vary depending on packing results.

Table 8 shows the statistics and performance expectations for a standalone PICXO.

Table 8: **Statistics and Performance Expectations for a Standalone PICXO (Vivado Design Suite 2015.1)**

Target Devices	7 Series FPGA GTP Transceiver	7 Series FPGA GTX Transceiver	7 Series FPGA GTH Transceiver
LUTs	862	940	873
Registers	950	992	950
SRLs	17	17	17
Occupied slices	348	355	357
Maximum PICXO clock rate	Speed grade dependent, matches TXUSRCLK2 maximum frequency	Speed grade dependent, matches DRP port limit	Speed grade dependent, matches TXUSRCLK2 maximum frequency

References

1. *7 Series FPGAs Configuration User Guide* (UG470). See the "Dynamic Reconfiguration Port" chapter.
2. *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/08/2012	1.0	Initial Xilinx release.
06/19/2012	1.1	Updated to include Verilog version. Minor changes to the VHDL code in the Reference Design. Updated Table 5. Revised Block RAM in Table 7.
10/16/2012	2.0	Added support for Kintex-7 FPGAs throughout document, including adding figures and revising tables. See new Reference Design information with updated design file.
10/15/2013	2.1	Updated Equation 9 through Equation 13. Updated bulleted list of constants after Equation 14. Removed Figure 8: "Estimated Transfer Function using Function H(z) in Equation 6" and Table 1: "Kintex-7 FPGA PICXO Sample Parameters for a Nominal 50 Hz Cleaning Bandwidth." Updated PICXO Architecture Overview. Replaced DIVCNT_TC with CE_DSP_RATE throughout. Renamed, relocated, and updated PICXO Architecture Overview. In Table 1, replaced Kintex-7 with 7 series FPGAs, added DRPDO_I [15:0], and updated DRPADDR_O [8:0]. In Table 2, updated title, added DRP_USER_DONE_I, and updated DRPADDR_USER_I [8:0]. Added Table 3. In Table 4, replaced CKLEN_O with CE_PI2_O and RSTCNT_O with CE_DSP_O. In Table 5, replaced DIVCNT_TC[15:0] with CE_DSP_RATE[15:0] and updated descriptions of V[15:0], ACC_STEP[3:0], VSIGCE_I, RSIGCE_I, OFFSET_PPM[21:0], and HOLD. Updated Interface Operation, Implementation, Mandatory Conditions and Limitations, and Reference Design. Updated Table 8 and moved to Reference Design.
09/12/2014	2.2	Updated for Vivado Design Suite 2014.1. Removed information about Virtex-6 FPGAs and the ChipScope™ tool. Defined the transceiver types covered in the application note on page 1. Removed application support for 3G-SDI from the Summary, page 1. The "Phase Resolution and Frequency Shift Example" section was deleted. Added OVF_VOLT in Table 4 and DON_I in Table 5. The Table 6 title changed from Reference Design Checklist to Reference Design Matrix. Added support for AC701 and ZC706 to Reference Design, page 28 and to the Table 6 Hardware Verification section. Virtex-6 LXT XC6VLX240T FF1146-1 was removed from Table 6 target devices and Artix-7 XC7A200T FBG676C-2 and Zynq-7000 XC7Z045 FFG900-2 were added. XST, ISE Design Suite 13.4 and 14.4 were removed from Implementation software tools used. Removed ChipScope IP data from Table 7. The MUXFXes row was removed from Table 8, and the Virtex-6 LXT and SXT column was replaced with 7 series FPGA GTP transceiver information. Values in Table 7 and Table 8 were updated. Removed the "ISE Tools Implementation" section from Implementation, page 27. Removed Virtex-6 documents from References, page 29.
04/29/2015	2.3	Updated for Vivado Design Suite 2015.1. Updated constants for the equations in PICXO DPLL. Updated Table 1, Table 2, Table 3, Table 5, Table 6, Table 7, and Table 8. Added GTP to Table 7 and Table 8. Updated Figure 20 and Figure 21.

Please Read: Important Legal Notices