Non-Integer Data Recovery Unit

XAPP1362 (v1.0) February 25, 2021

Summary

Contemporary applications require transceivers that can operate over a wide range of input data rates. Transceivers have a native operational data rate that prevents easy interfacing to low-speed client signals. For example, the GTM transceiver in the Versal™ ACAP has a lower operating line rate of 10.3 Gbit/s. Also, SelectIOs cannot usually recover data from a serial line. The non-integer data recovery unit (NIDRU) described in this application note extends the lower data rate limit to 0 Mb/s and allows SelectIOs to operate as clock and data recovery units. The NIDRU operational settings (data rate, jitter bandwidth, input PPM range, and jitter peaking) are dynamically programmable, avoiding the need for bitstream reload or partial reconfiguration. At any time, without affecting the data traffic, a live horizontal eye scan can be performed to measure the eye width as seen by the NIDRU.

Download the reference design files for this application note from the Xilinx® website. For detailed information about the design files, see Reference Design.

Introduction

The NIDRU described in this application note is optimized for the Versal device. This NIDRU operates on fractionally oversampled data and includes these features:

- Fully synchronous architecture based on DSP. It is based on a single clock tree even when multiple NIDRUs are instantiated and operating at different line rates.
- The input datapath width is programmable (DT_IN_WIDTH) and supports 4, 20, 32, 64, and 128 bits.
- The output datapath width is programmable (WDT_OUT) from 1 to 64 bits.
- The eye scan is fully DSP based. It is non-disruptive and does not use the eye scan logic that is built in the transceiver hardware.
- Rate, bandwidth, and jitter peaking are run-time programmable.
- Built in PPM meter, which measures run time difference between the incoming data rate and the local reference clock, with sub-PPM accuracy.
- Ability to synthesize the recovered clock on a separate transmitter (e.g., synchronization interfaces).

The application note is divided into the following sections:

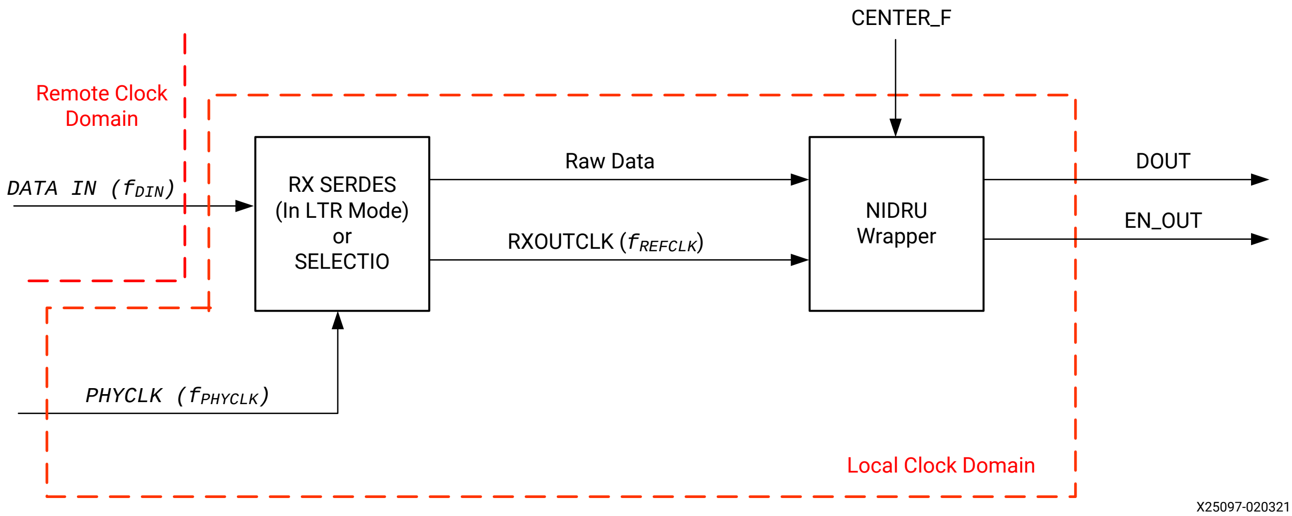
- Use Model: general operating principles.
- Block Diagram and Pinout: attributes and ports of NIDRU are described.

- Configuration: a description of how to configure NIDRU ports and attributes.
- Simulating the NIDRU: a description of the NIDRU simulation test bench, with the settings of multiple line rates ready to be used.
- Versal ACAP Test Bench: a walk through of the NIDRU fully-featured demonstration on the VCK190 evaluation board, highlighting fractional operation, on-the-fly rate change and non-disruptive eye scan.

Use Model

This section describes the operating principle of the NIDRU at a high level. The following figure illustrates the high-level application architecture for the NIDRU.

Figure 1: **NIDRU High-level Application Architecture**



The f_{DIN} data rate is the rate needed to receive and is generally lower than the rate the receiver can support normally. To receive data at a rate f_{DIN}, the transceiver is set to operate at a rate f_{SAMPL}, which is expected to be supported by the receiver and higher than f_{DIN}.

The ratio f_{SAMPL}/f_{DIN} is defined as the oversampling rate (O_R) and it is recommended but not mandatory to keep it equal to at least 3 for reliable operation.

$$\frac{f_{SAMPL}}{f_{DIN}} = O_R$$

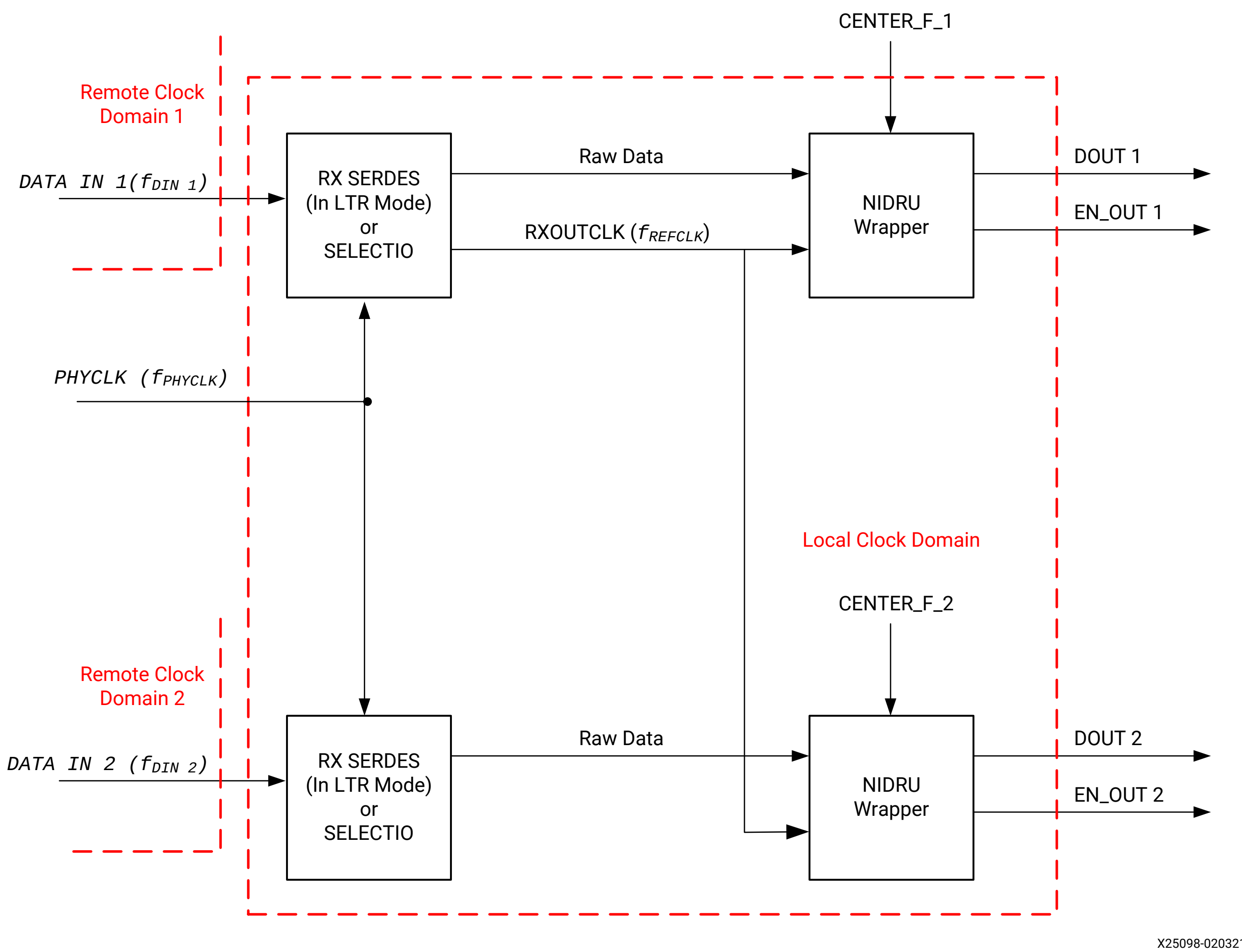
The transceiver is supposed to operate without tracking the data, a mode that is referred to as “Lock to Reference." In this mode, the transceiver operates as an A/D converter with a 1 bit resolution. For this reason, the NIDRU clock (f_{REFCLK}) is always locked to the PHY clock (f_{PHYCLK}) and is, consequently, part of the same clocking domain. As a result, a SelectIO can be used instead of a transceiver, as shown in the figure above.

EN_OUT is synchronized to the NIDRU clock. However, the rate at which EN_OUT is asserted by the NIDRU is locked to the remote clock domain (f_{DIN}).

The key feature of the NIDRU is that the ratio between the remote clock domain and the local clock domain (nominally, this ration is O_R) can be fractional, and the nominal ratio is specified using the port CENTER_F, as described later in this application note.

The most relevant consequence of this capability is that one single clock tree is needed, even if multiple NIDRUs are working at different line rates, provided all receivers are working at the same f_{SAMPL}. See the following figure.

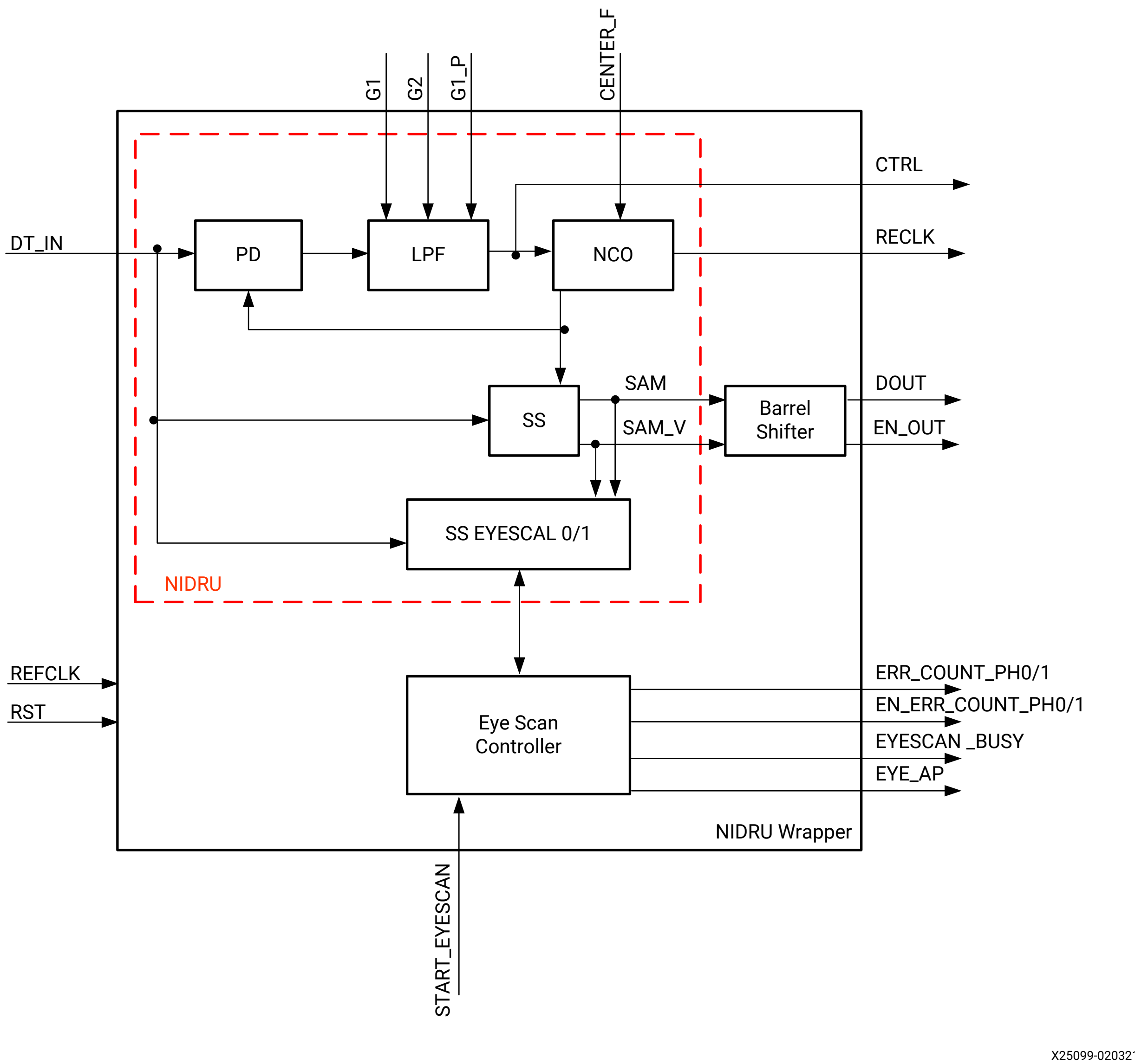
Figure 2: **Multiple NIDRUs Working at Different Rates and Sharing Same Clock Tree**



Block Diagram and Pinout

This section describes the structure of the NIDRU wrapper and its pinout. The wrapper structure is shown in the following figure. Only relevant ports are shown in the figure.

Figure 3: **Simplified Block Diagram**



The DT_IN port receives raw oversampled data from a SelectIO interface or a transceiver set in lock-to-reference mode. The width of the oversampled data is programmable by the DT_IN_WIDTH attribute. The value of the DT_IN_WIDTH attribute can be set to 4, 20, 32, 64, or 128 bits. The NIDRU bit-ordering convention is the same as the transceivers, where the LSB is the oldest bit or the bit that came in first.

Note: Starting with the Virtex®-4 devices, all transceivers follow the same ordering rule as the NIDRU described here. The SerDes in Virtex-II Pro devices use a different bit ordering.

The phase detector (PD) looks for transitions in the incoming data, continuously comparing the phase of the incoming data with the phase of the internal numerically-controlled oscillator (NCO). The digital error signal generated by the PD and filtered out by the low-pass (LP) filter corrects the NCO frequency to minimize the phase error, thus realizing the phase-locked loop (PLL) functionality of the NIDRU. See References.

Based on the NCO output, the sample selector (SS) block selects the samples that are more closely positioned to the middle of the eye. There can be up to 64 valid samples in each REFCLK cycle, which are placed by the SS on the SAM output. SAMV indicates the number of valid samples on SAM at each clock cycle. To simplify the connection between the NIDRU and the user application, a barrel shifter is provided in the wrapper, where the output data width can be programmed using the WDT_OUT attribute. All blocks in the NIDRU wrapper are synchronized to REFCLK. The NIDRU operates in parallel over the incoming data, generally producing more than one bit output for each clock cycle. The relationship between the operating frequency (REFCLK) and the incoming data rate dictates the maximum number of bits extracted per clock cycle, N_{MAX}, according to the following equation.

$$N_{MAX} = truncate\left[\frac{fDIN}{fREFCLK}\right] + 1$$

Three example user configurations of f_{REFCLK} and the oversampling rate are considered here with the assumption that a 20-bit NIDRU is used.

- Fast Ethernet with f_{REFCLK} = 125 MHz, oversampling at 2.5 Gb/s: NMAX = 2.
- STM1 with f_{REFCLK} = 125 MHz, oversampling at 2.5 Gb/s: NMAX = 2.
- Fast Ethernet with f_{REFCLK} = 155.52 MHz, oversampling at 3.1 Gb/s: NMAX = 1.

If WDT_OUT > 1, a barrel shifter is automatically inserted in the wrapper to ease the interfacing of the NIDRU to a fixed-width FIFO. If WDT_OUT = 1 (i.e., the user application has 1-bit width only) the barrel shifter is not needed and is not instantiated in the NIDRU wrapper.

WDT_OUT must satisfy the criteria in the following equation to make sure the output bandwidth of the NIDRU is compatible with the incoming throughput.

$$WDT\_OUT \geq N_{MAX}$$

The following table describes the NIDRU configuration attributes. The NIDRU ports are described in Table 2: NIDRU Ports.

Table 1: **NIDRU Configuration Attributes**

Attribute Name	Type/Range	Description	Comment
Configuration Section			
WDT_OUT	Integer from 2 to 64	Output data width	Output width for the bus DOUT.
DT_IN_WIDTH	Integer 4, 20, 32, 64, or 128	Input data width	Output width for the bus DT_IN.
EN_CENTER_F_ATTR	Standard logic	Enables use of CENTER_F_ATTR	When set to 1, CENTER_F_ATTR is used as CENTER_F. When set to 0, the CENTER_F port is used.
CENTER_F_ATTR	Standard logic vector 39 down to 0	Attribute configuration for CENTER_F	CENTER_F_ATTR can be used instead of CENTER_F depending on the value of EN_ CENTER_F_ATTR.
EN_G1_ATTR	Standard logic	Enables use of G1_ATTR	When set to 1, CENTER_F_ATTR is used as CENTER_F. When set to 0, the CENTER_F port is used.

Table 1: **NIDRU Configuration Attributes** *(cont'd)*

Attribute Name	Type/Range	Description	Comment
G1_ATTR	Standard logic vector 4 down to 0	Attribute configuration for G1	G1_ATTR can be used instead of G1 depending on the value of EN_ G1_ATTR.
EN_G2_ATTR	Standard logic	Enables use of G2_ATTR	When set to 1, CENTER_F_ATTR is used as CENTER_F. When set to 0, the CENTER_F port is used.
G2_ATTR	Standard logic vector 4 down to 0	Attribute configuration for G2	G2_ATTR can be used instead of G2 depending on the value of EN_G2_ATTR.
EN_G1_P_ATTR	Standard logic	Enables use of G1_P_ATTR	When set to 1, CENTER_F_ATTR is used as CENTER_F. When set to 0, the CENTER_F port is used.
G1_P_ATTR	Standard logic vector 4 down to 0	Attribute configuration for G1_P	G1_P_ATTR can be used instead of G1_P depending on the value of EN_G1_P_ATTR
EN_SHIFT_S_PH_ATTR	Standard logic	Enables use of SHIFT_S_PH_ATTR	When set to 1, CENTER_F_ATTR is used as CENTER_F. When set to 0, the CENTER_F port is used.
SHIFT_S_PH_ATTR	Standard logic vector 7 down to 0	Attribute configuration for SHIFT_S_PH	SHIFT_S_PH_ATTR can be used instead of SHIFT_S_PH depending on the value of EN_SHIFT_S_PH_ATTR.
EN_EN_INTEG_ATTR	Standard logic	Enable port EN_INTEG	When set to 1 enables EN_INTEG port. When set to 0, the EN_INTEG port is connected to EN_INTEG_ATTR.
EN_INTEG_ATTR	Standard logic	Attribute configuration for EN_INTEG	See EN_EN_INTEG_ATTR.
EN_EN	Standard logic	Enables the EN port	The EN port can be disabled by setting EN_EN=1, reducing the complexity of the circuit.
ENABLE_LTR_PORT	Standard logic	Enables lock to reference mode	When set to 1, the port LTR can be used to disable the tracking mechanism (LTR = 1).
Eye Scan Section			
PH_NUM	Integer from 0 to 2	Number of extra sampling phases	• 0: No eye scan logic is instantiated. • 1: Eye scan logic with one extra phase. • 2: Eye scan logic with two extra phases.
Logic Optimization Section			
S_MAX	Integer from 1 to 64	Expected maximum number of extracted samples per clock cycle	See Logic Optimization for configuration instructions for this port. Set S_MAX ≥ NMAX. Setting S_MAX to DT_IN_WIDTH/2 works for all cases although it might not be optimal from the resource point of view.
S_MAX_EYE	Integer from 1 to 64	Maximum number of samples extracted by the eye scan controller	See Logic Optimization for configuration instructions for this port.

Table 1: **NIDRU Configuration Attributes** *(cont'd)*

Attribute Name	Type/Range	Description	Comment
MASK_CG	Standard logic vector 15 down to 0	Mathematical precision of the generated coefficients	See Logic Optimization for configuration instructions for this port. Setting MASK_CG to all ones forces NIDRU to use maximum precision.
MASK_PD	Standard logic vector 15 down to 0	Mathematical precision of the PD calculations	See Logic Optimization for configuration instructions for this port. Setting MASK_PD to all ones forces NIDRU to use maximum precision.
MASK_VCO	Standard logic vector 36 down to 0	Mathematical precision of the NCO output	See Logic Optimization for configuration instructions for this port. Setting MASK_VCO to all- ones forces NIDRU to use maximum precision.

The following table describes the NIDRU ports.

Table 2: **NIDRU Ports**

Pin Name	Type	Description	Comment
Data Ports			
DT_IN	Input 4, 20, 32, 64, or 128 bits	Input data from SerDes or SelectIO interface	Bit 0 is the oldest.
EN	Input	Enable	Enables all processes of the NIDRU.
CLK	Input	Clock	Clock for all NIDRU processes.
RECCLK	Output DT_IN_WIDTH bits	Recovered clock	This is the recovered clock to be serialized by a TX SerDes or a SelectIO interface. In terms of serialization order, bit 0 has to be serialized first.
EN_OUT	Output	Output data valid	When data on DOUT is valid, the NIDRU sets EN_OUT to 1.
DOUT	Output WDT_OUT bits	Output data	Output data for the user application. The width of DOUT is programmable through the attribute WDT_OUT.
Configuration Ports			
CENTER_F	Input 40 bits	Center frequency at which the NIDRU operates	See Logic Optimization for configuration instructions for this port.
G1	Input 5 bits	Direct gain	See Logic Optimization for configuration instructions for this port.
G1_P	Input 5 bits	Integral pre-gain	See Logic Optimization for configuration instructions for this port.
G2	Input 5 bits	Integral post-gain	See Logic Optimization for configuration instructions for this port.
LTR	Input	Lock to reference mode	See attribute EN_LTR_PORT.
Eye Scan Ports			

Table 2: **NIDRU Ports** *(cont'd)*

Pin Name	Type	Description	Comment
AUTOM	Input	Auto/manual mode	Setting to 1 enables the embedded eye scan controller. Setting to 0 allows performing eye scan manually.
START_EYESCAN	Input	Start eye scan	Pulse for at least 1 clock cycle to request an eye scan. Any eye scan request while EYESCAN_BUSY = 1 is discarded.
EYESCAN_BUSY	Output	Eye scan is operating	When set to 1, eye scan is being acquired.
TRIGGER_MODE	Input	Eye acquisition mode	See One-Dimensional Eye Scan.
EYE_AP	Output 9 bits	The eye aperture can be read here.	This value is updated each time the signal `EYSCAN_BUSY` goes down.
RST_PH_0	Input	Phase 0 reset	See One-Dimensional Eye Scan.
RST_PH_1	Input	Phase 1 reset	See One-Dimensional Eye Scan.
RST_PH_SAMP	Input	Sampling phase reset	See One-Dimensional Eye Scan.
ERR_PH_0	Output 7 bits	Error number from phase 0	See One-Dimensional Eye Scan.
ERR_PH_1	Output 7 bits	Error number from phase 1	See One-Dimensional Eye Scan.
PH_0	Input 8 bits	Phase 0 position	See One-Dimensional Eye Scan.
PH_1	Input 8 bits	Phase 1 position	See One-Dimensional Eye Scan.
PH_0_SCAN	Output 8 bits	Current phase 0 being scanned	See One-Dimensional Eye Scan.
PH_1_SCAN	Output 8 bits	Current phase 1 being scanned	See One-Dimensional Eye Scan.
WAITING_TIME	Input 48 bits	Waiting time for each sample	See One-Dimensional Eye Scan.
ERR_COUNT_PH_0	Output 52 bits	Errors accumulated in PH_0	See One-Dimensional Eye Scan.
EN_ERR_COUNT_PH_0	Output	Valid signal for ERR_COUNT_PH_0	See One-Dimensional Eye Scan.
ERR_COUNT_PH_1	Output 52 bits	Errors accumulated in PH_1	See One-Dimensional Eye Scan.
EN_ERR_COUNT_PH_1	Output	Valid signal for ERR_COUNT_PH_1	See One-Dimensional Eye Scan.
Debug Ports			
PH_OUT	Output 21 bit	Output NCO phase	Debug output.
INTEG	Output 32 bits	Integral-branch output	
DIRECT	Output 32 bits	Direct-branch output	
CTRL	Output 32 bits	NCO control signal	

Table 2: **NIDRU Ports** *(cont'd)*

Pin Name	Type	Description	Comment
AL_PPM	Output	PPM alarm	Input data stream is at the limit of the NIDRU when AL_PPM =1. This signal is not latched.
RST	Input	Reset	Reset signal for all processes, except for the filter.
PH_EST_DIS	Input	Phase error estimation method	Debug input. Set to 0.
EN_INTEG	Input	Enable integral path	Debug input. Set to 1.
VER	Output 8 bits	Version	NIDRU version. The version delivered by this application note is 10.
SAMV	Output 7 bits	Number of samples out	At each clock cycle, NIDRU reports how many bits have been extracted. Connected to the barrel shifter in the wrapper.
SAM	Output, DT_IN_WIDTH/2	Samples out	At each clock cycle, NIDRU reports the SAMV bits which have been extracted. They are placed in the lowest portion of SAM. Connected in the wrapper to the barrel shifter.

Configuration

This section describes how to translate the incoming data rate and reference clock frequency into a valid NIDRU configuration.

The user configuration defines specifications for:

- Incoming data rate with associated tolerance ($f_{\_DIN}$± PPM)
- Available reference clock frequency with associated tolerance ($f_{\_REFCLK}$± PPM)

While $f_{\_DIN}$ is given, $f_{\_REFCLK}$ can be selected inside a valid range. The range upper limit comes from the necessity to close timing in the target device, and is thus device and speed grade dependent. The lower limit to $f_{\_REFCLK}$ might be imposed by the PHY. For example, a SerDes typically specifies a minimum reference clock frequency. A SelectIO interface does not typically impose a lower limit to $f_{\_REFCLK}$. The maximum $f_{\_DIN}$ is typically limited by the oversampling rate O_R as defined in the following equation.

$$O_R = \frac{DT\_IN\_WIDTH \times f_{REFCLK}}{f_{DIN}}$$

RECOMMENDED: *Xilinx® recommends keeping $O_R \geq 3$ to have enough high frequency jitter tolerance.*

CENTER_F, G1 and G2 should be set according to the following equations:

$$CENTER\_F = \frac{f_{DIN}}{f_{REFCLK}} \times 2^{32}$$

$$G_1 = G_2 \leq 32 - roundup\left[log_2 \frac{2^{33} \times (PPM_{DIN} + PPM_{REFCLK}) \times f_{DIN} \times 10^{-6}}{f_{REFCLK}} \right]$$

Using the equal value guarantees that the NIDRU operates in the lock-in region over the full PPM range of both the incoming data and the reference clock. Further reducing G_2 increases the NIDRU bandwidth. In general, increasing G_2 is not recommended, as the NIDRU would operate in pull-in region, where the automatic lock is not always guaranteed.

In general, G_(1_P) should be evaluated using the spreadsheet in the folder `/excel_plots`. The value should be increased up to when the ringing effect on the output phase becomes negligible. When G_1= G_2, setting G_(1_P)=16 always guarantees a negligible ringing effect. Consequently, G_(1_P)=16 is good for most of the cases.

The resulting transfer function can be obtained by using the excel file `nidru_transfer_function_v_1_0.xls` in the folder `/excel_plots`.

The usual three cases are considered as examples, yielding the following results:

1. Fast Ethernet (f_{DIN}=125 Mbit/s ± 100 PPM) with 125 MHz refclk (± 100 PPM):
   CENTER_F= `b0000100000000000000000000000000000000`
   $G_1 = G_2 \leq 11$
2. Fast Ethernet (f_{DIN}=125 Mbit/s ± 100 PPM) with 155.52 MHz refclk (± 20 PPM):
   CENTER_F= `b0000011001101110000101110010110101001`
   $G_1 = G_2 \leq 11$
3. OC3 (f_{DIN}=155.52 Mbit/s ± 20 PPM) with 125 MHz refclk (± 100 PPM):
   CENTER_F= `b0000100111110100000010100010100001110`
   $G_1 = G_2 \leq 11$

Logic Optimization

The NIDRU performs many run-time calculations. The precision of these calculations can be controlled by attributes, which can be used to trade off between precision and resource usage.

MASK_CG = `1111111111111111` means internal 16 bit precision.

MASK_CG = `1111111111111110` means internal 15 bit precision.

The same is true for MASK_PD and MASK_VCO.

Lowering the precision can be done to reduce resources and should be evaluated by simulation or in hardware. If in doubt, set MASH_CG, MASK_PD, and MASK_VCO to all ones.

N_{MAX} is the maximum number of samples extracted during one clock cycle. This parameter, defined in the following equation, is the reference to define the minimum value for other attributes of the NIDRU.

$$N_{MAX} = truncate\left(\frac{fDIN}{fREFCLK}\right) + 1$$

S_{MAX} must be set according to the following equation.

$$S_{MAX} \geq N_{MAX}$$

The NIDRU configuration can be changed during run time. Different configurations can have different N_{MAX}. S_{MAX} must be selected so that the previous equation is true for all configurations.

S_{MAX_EYE} is an internal parameter and has to be set according to the following equation.

$$S_{MAX\_EYE} \geq S_{MAX} + 1$$

WDT_OUT is the output width of the NIDRU and should be set according to the following equation.

$$WDT\_OUT \geq N_{MAX}$$

Setting S_{MAX}, S_{MAX_EYE}, and WDT_OUT above their minimum values has no impact on the functionality of the NIDRU, and only results in slightly more usage of resources.

One-Dimensional Eye Scan

The NIDRU allows plotting the eye diagram using live data, i.e., without affecting the data traffic. This feature is very useful during the debug phase, as it allows measuring the receiver margin. During normal operation, this feature allows detecting links that are degrading over time before they can result in a visible BER.

The resolution of the eye scan is 256 taps for a single UI, independently on the oversampling rate.

The operating principle is based on the creation of additional sampling phases (exploring phases), on top of the one (mission phase) that is sampling the eye in the position that is supposed to be the optimal one, which is always active.

The number of exploring phases is defined with the attribute PH_NUM. Because it is DSP based, there is no limit to the number of exploring phases that can be created, and the NIDRU allows a maximum of two exploring phases.

The PH_NUM attribute is used to activate and configure the eye scan logic.

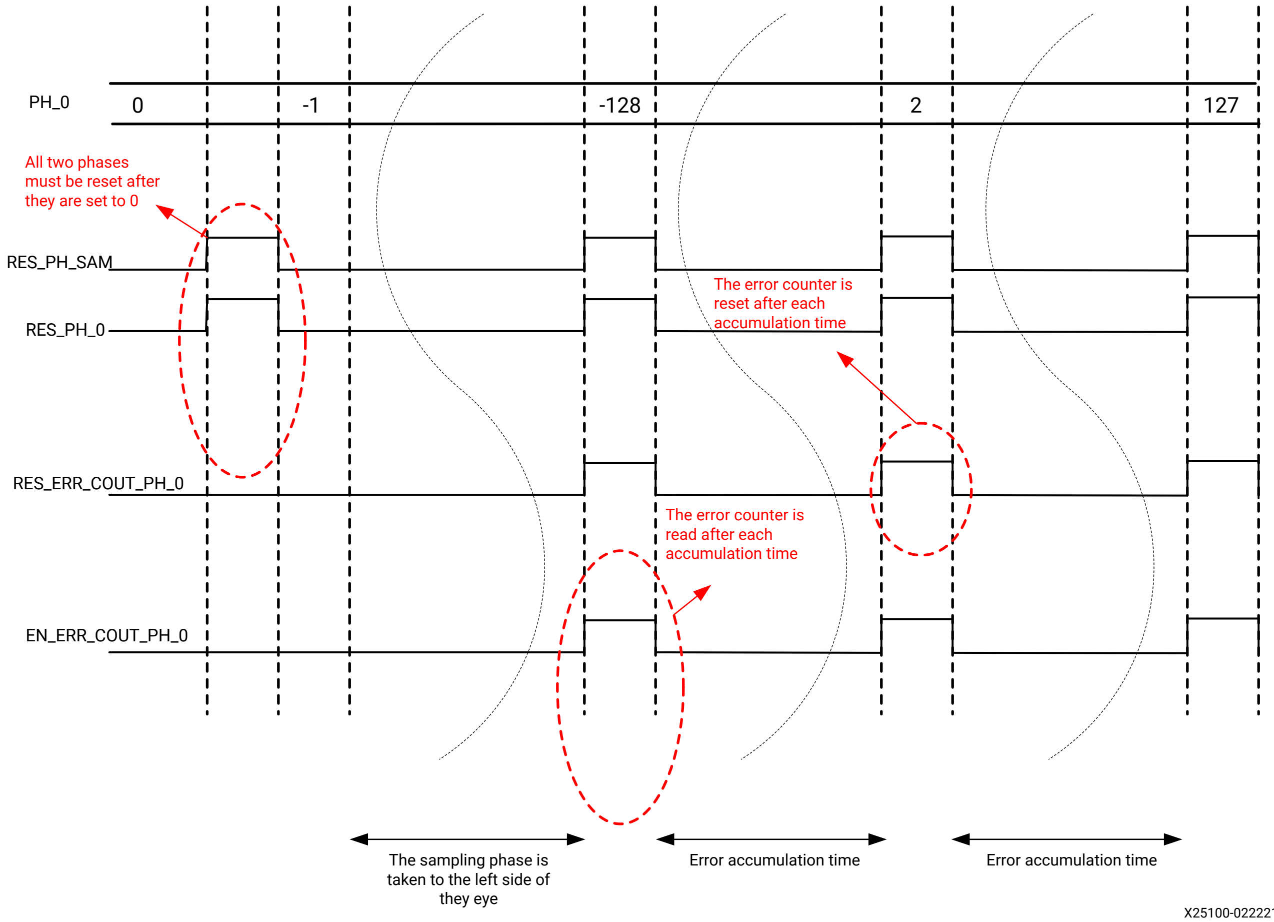
- PH_NUM=0 means no eye scan.
- PH_NUM=1 means that the eye scan operates with one exploring phase, ranging from -128 to 127, equivalent to -0.5UI to 0.5UI.
- PH_NUM=2 means that the eye scan operates with two exploring phases. Using the automatic eye scan controller, PH_0 sweeps the right portion of the eye, from 0 to 127 (i.e., from 0 to 0.5 UI). Symmetrically, PH_1 sweeps the left side of the eye, from -1 to -128.

Setting PH_NUM to 0 allows the saving of fabric resources. If the eye scan functionality is only used during debug, it can be removed after the debug phase, and the NIDRU functionality is unaffected.

In automatic mode (AUTOM=1), all steps to perform an eye scan are managed by the embedded eye scan controller and each sweep can be triggered by simply pulsing the START_EYESCAN port for at least one clock cycle. The output EYESCAN_BUSY is asserted by the NIDRU until all 256 points are swept. The port WAITING_TIME is used to specify how many clock cycles (f_{REFCLK}) should be devoted to each single sweeping point.

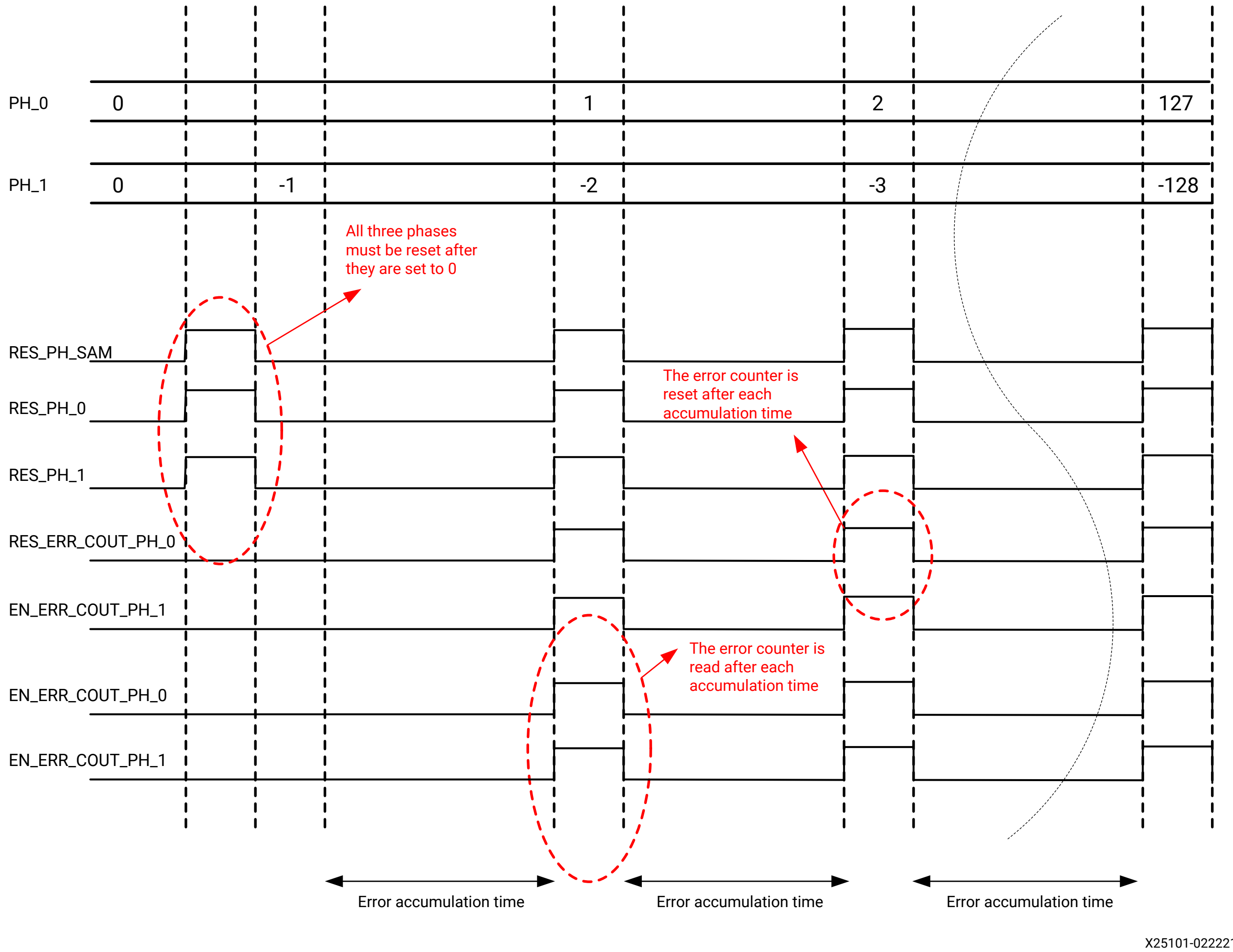
The following figure shows the timing diagram in automatic mode for PH_NUM=1.

Figure 4: **Timing Diagram of the Eye Scan Controller When PH_NUM=1**



The following figure shows the timing diagram in automatic mode for PH_NUM=2.

Figure 5: **Timing Diagram of the Eye Scan Controller When PH_NUM=2**



The measurement time for each point is user specified through the port WAITING_TIME, which specifies the error accumulation time in clock cycles of F_{REFCLK}. The following equation allows relating the value of WAITING_TIME to the target BER.

$$WAITINGTIME = \frac{O_R}{DT\_IN\_WIDTH \times BER}$$

To plot the eye scan, plot the ports ERR_PH_0 and ERR_PH_1 in the simulator viewer on an ILA core. These signals are valid only when the corresponding EN_ERR_PH_x signals are set to 1.

When PH_NUM=1, the full eye shape can be seen on one line. When PH_NUM=2, the two halves of the eye are simultaneously plotted on two independent lines.

The port TRIGGER_MODE is used only when AUTOM=1 and specifies if the eye scan should be performed repetitively (TRIGGER_MODE=1) or once (TRIGGER_MODE=0).

Keeping TRIGGER_MODE=1, the eye scan measurement is continuously updated on the output port EYE_AP, which can easily be used to define whether the NIDRU is locked to the incoming data.

Example plots are included in Simulating the NIDRU and Versal ACAP Test Bench.

RECOMMENDED: *Automatic mode is the recommended use model.*

In manual mode (AUTOM=0), both phases can be moved freely. To measure the eye aperture in any given point inside the eye, PH_0 is the port for phase 0 and PH_1 is the port for phase 1. The exploring phases should never be moved by more than 1 step per clock cycle.

The timing diagrams in automatic mode implemented in the eye scan controller can be used as examples to easily derive the timing diagram for any preferred eye exploration method.

Sampling Point Shift

Eye scan is a useful feature but it can be costly in terms of resources (see Resources). To save hardware resources, it is possible to shift the NIDRU sampling point inside the eye to measure the eye aperture. With this method, the eye scan can be performed only during the debug session, because bit errors are visible as soon as the sampling phase is pushed closer to the borders of the eye.

The port SHIFT_S_PH can be used to move the sampling point. The port is specified in twos complement and ranges from -128 to 127. 256 steps completely cover one UI. By setting SHIFT_S_PH to 0 (default), the NIDRU samples the incoming data stream in the middle of the eye.

The eye scan ports operate independently on the port SHIFT_S_PH. This means that each eye scan is always performed relative to the middle of the eye diagram and not to the setting of SHIFT_S_PH.

Simulating the NIDRU

The purpose of the TB_SIM_DRU_JITTER test bench is to simulate the ability of the NIDRU to operate at several popular data rates (see the following table) with synchronous and plesiochronous serial inputs. The simulation covers the nine cases shown in the following table in sequence. Any data rate can be added as an additional case. The cases show the ability of the NIDRU to operate at both fractional and integer oversampling rates.

Table 3: **Simulation Cases**

Protocol (Case #)	Data Rate	Ref Clock (Oversampling Rate)	Oversampling Rate	Minimum Width Supported
Proprietary (1)	250 Mbit/s	125 MHz	10	20
OC3 (2)	155.52 Mbit/s	125 MHz	16.075	4
SDI (3)	270 Mbit/s (+100 ppm)	148.5 MHz	11	20
OC3 (4)	155.52 Mbit/s	155.52 MHz	20	4
OC12 (5)	622.08 Mbit/s	125 MHz	4.019	20
FE (6)	125 Mbit/s	155.52 MHz	24.8832	4
Proprietary (7)	8 Gbit/s	229 MHz	3.664	128
Proprietary (8)	4 Gbit/s	229 MHz	7.328	64

Table 3: **Simulation Cases** *(cont'd)*

Protocol (Case #)	Data Rate	Ref Clock (Oversampling Rate)	Oversampling Rate	Minimum Width Supported
Proprietary (9)	10 Gbit/s	229 MHz	2.9312	128

To run the simulation script:

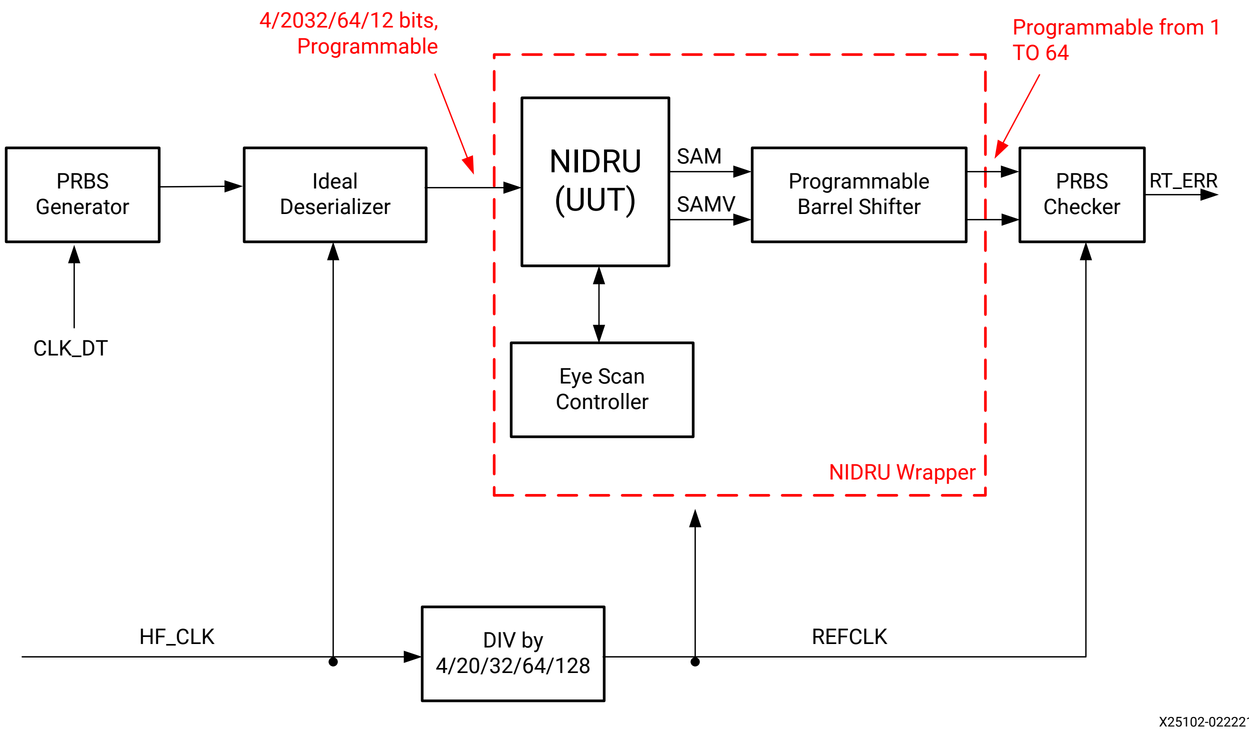
1. Open a DOS command window.
2. Change to the `/scripts` directory.
3. Open the Mentor Graphics Questa Advanced Simulator.
4. In the simulator, run the `run_sim_do` script.

Although the test bench has been designed for the Questa Advanced Simulator only, the NIDRU core is expected to operate with the following broader set of simulation tools.

- Vivado® Simulator
- ModelSim
- Synopsys VCS

The architecture implemented in the `tb_sim_nidru_v_3_0`.vhd test bench is shown in the following figure. An ideal deserializer and serializer are used instead of a full transceiver to minimize the simulation time. The serializer and deserializer datapath is 4, 20, 32, 64, or 128, programmable through the DTIN_WIDTH attribute.

Figure 6: **TB_SIM_NIDRU Block Diagram**



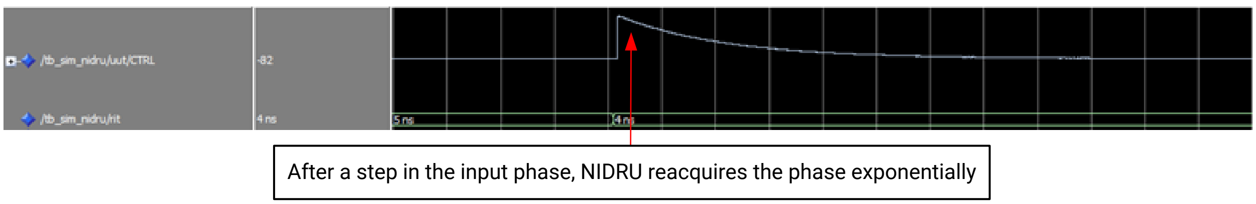
The test bench contains two clock domains:

- The clock domain of the line, synchronized to CLK_DT.
- The clock domain of the DRU, synchronized to REFCLK and to HF_CLK.

The pseudo-random binary sequence (PRBS) generator works at full speed on CLK_DT and can generate any kind of industry standard PRBS (see *An Attribute-Programmable PRBS Generator and Checker* (XAPP884)). The ideal deserializer, the NIDRU, and the PRBS checker all work on the local REFCLK domain, a divided version of HF_CLK.

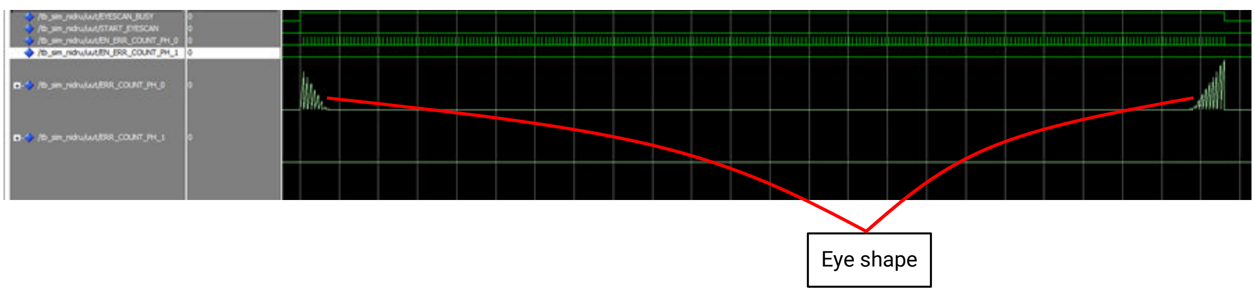
During test case 1, a 1 ns phase step datastream is applied to the input (see the following figure). The purpose of this is to show the ability of the NIDRU to respond with an exponential locking process to an input phase step and to prove the stability of the loop.

Figure 7: **NIDRU Response Following a 1 ns Step In the Input Phase**



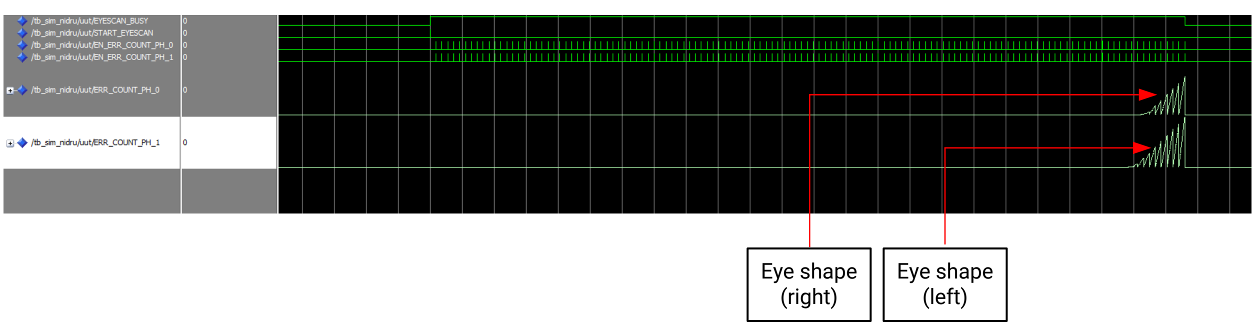
During test case 2 and 5, the eye scan is plotted with PH_NUM=1 (see the following figure). The eye is scanned with one single phase from left to right.

Figure 8: **Eye Scan with PH_NUM=1 (Simulation)**



An example of an eye scan with 2 phases (PH_NUM=2) is shown in the following figure. The left and right half of the eye are scanned at the same time.

Figure 9: **Eye Scan with PH_NUM=2 (Simulation)**

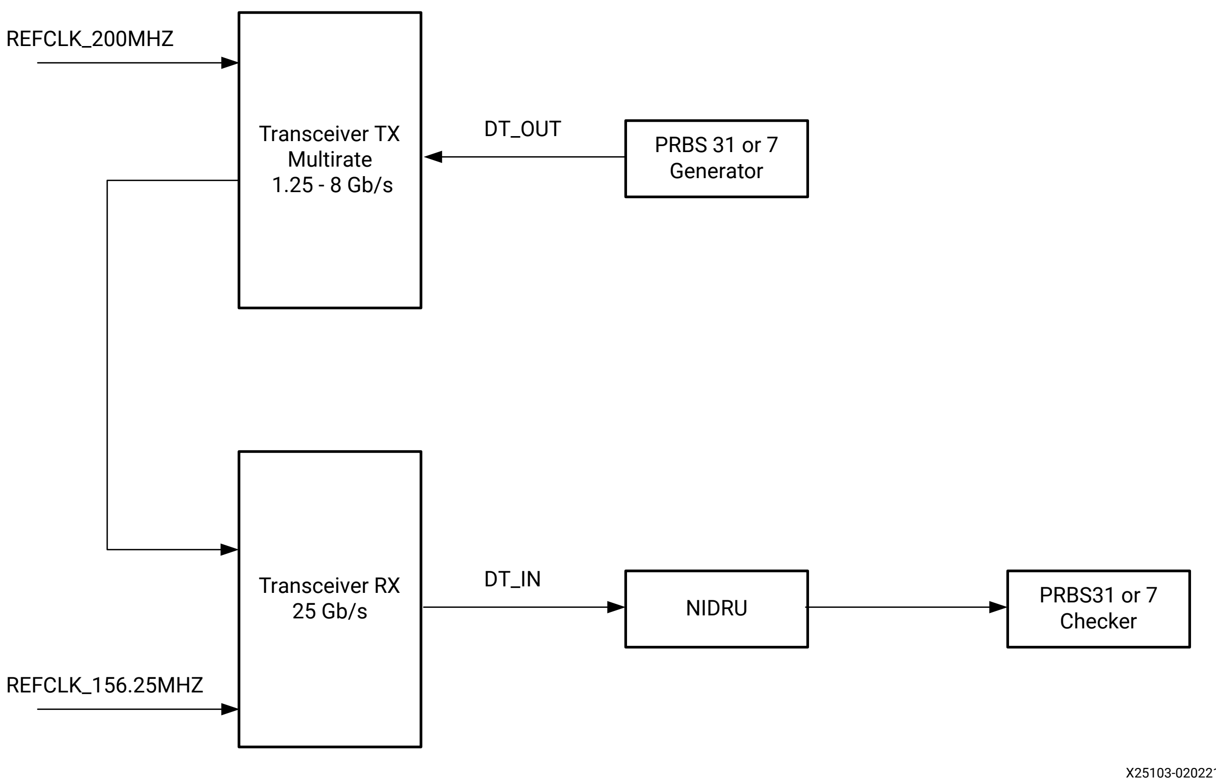


Versal ACAP Test Bench

The TB_HW_VERSAL test bench is available as part of the reference design, and it is designed for the VCK190 demonstration board. This test bench can be implemented to show the NIDRU data recovery capability with both synchronous and asynchronous inputs.

To compile the test benches, source the `nidru_design_versal`.tcl script from the Vivado Design Suite. The test bench architecture is shown in the following figure.

Figure 10: **TB_HW_DRU Test Bench Architecture**



The test bench includes:

- A 25G receiver, based on the NIDRU, operating on a 156.25 MHz reference clock.

  The TX channel transmits data synchronized with REFCLK 200 MHz. The channels are connected via a SFP cable so that the receiver receives data at a frequency not synchronized to its own reference clock.

  The two reference clocks are generated on board by using the system controller. The reference clock frequencies are configured through the serial interface on the VCK190 board.

- TX side

  A PRBS generator continuously sending a PRBS 7 or PRBS 31 pattern. Each of the two PRBS generators can be forced to generate an error using the Vivado Logic Analyzer to show error detection on the corresponding PRBS checker.

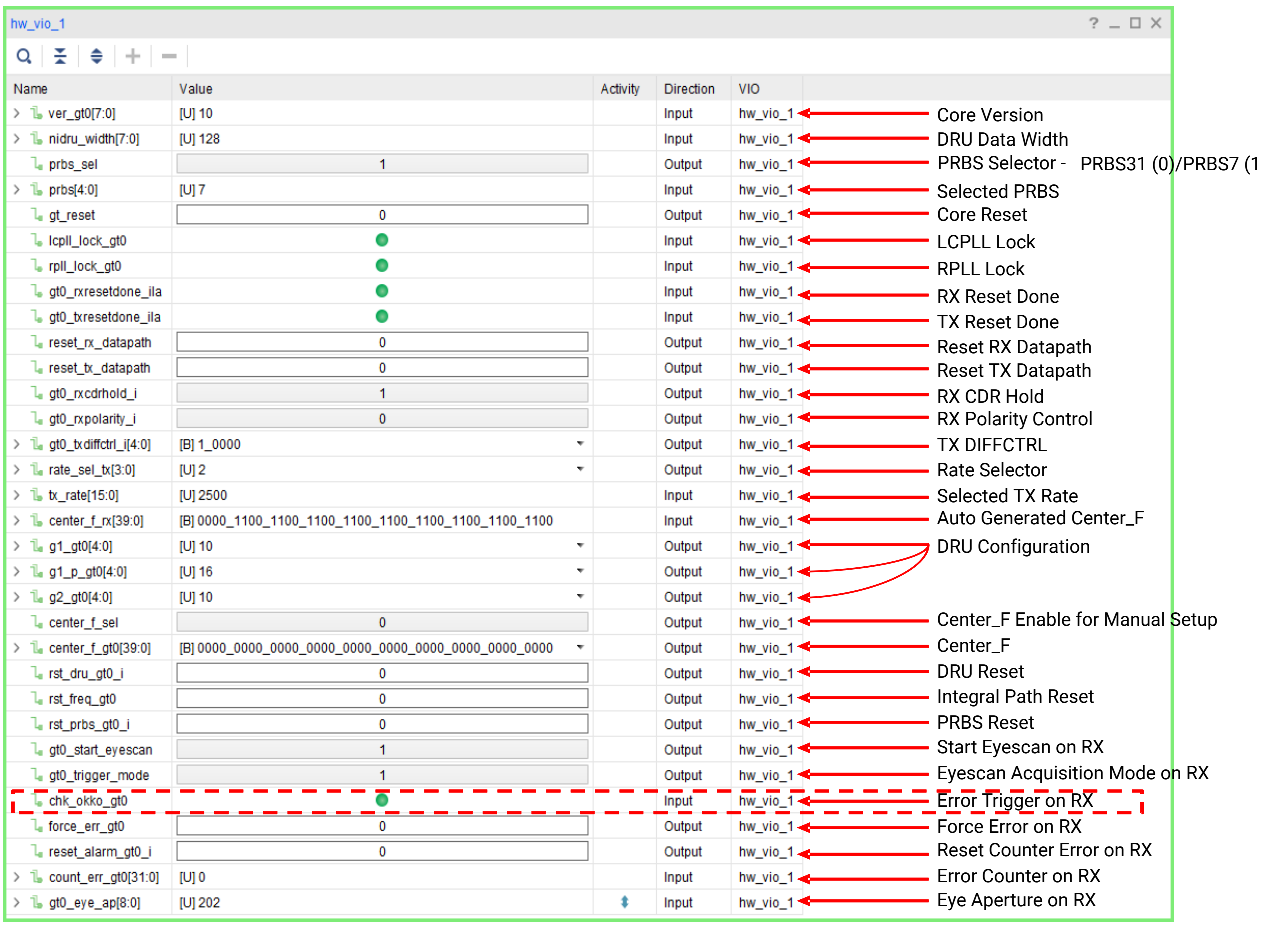
- RX side

A PRBS checker continuously checking the incoming PRBS 7 or PRBS 31 pattern. The ERR output indicates detection of at least one error from the last ERR_RST. ERR is connected to the virtual input/output (VIO) and checked in real time. An error counter is also provided.

The specific PRBS pattern used in this application note for both the generator and the checker is based on the polynomial x31+x28+ 1 for PRBS 31, x7+x6+ 1 for PRBS 7 and can be changed to any other industry standard PRBS type.

Each PRBS checker works on the data delivered by the barrel shifter, which is instantiated right after each NIDRU block. The following figure reports the detailed description of all signals of the test bench that are controlled by the Vivado Logic Analyzer. The pin names are consistent across the VHDL code, the logic analyzer project, and this application note.

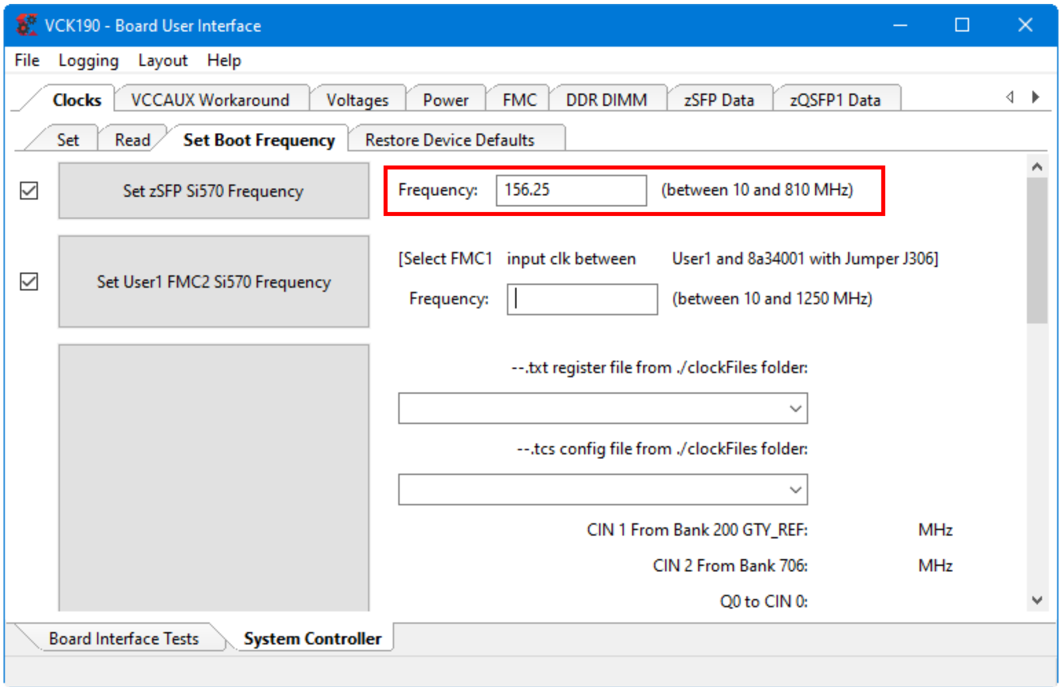
Figure 11: **Vivado Logic Analyzer Controlling the Demonstration Test Bench**



The transmitter can be set to generate a PRBS pattern, as described previously. When the application works properly, all LEDs are green. In case of an error in the datapath, the corresponding LED (highlighted with dashes in the previous figure) for the `chk_okko_gt0` signals is red. The example design needs two asynchronous and independent clocks. For the receiver, the frequency is 156.25 MHz. The clock for the transmitter can be a different frequency accordingly to the selected data rate.

The VCK190 board provides the required clocks through the system controller. In the example design supplied with this application note, the system controller can be programmed by the system controller user interface and using the serial interface on the VCK190 board. The system controller GUI software can be downloaded from the VCK190 lounge page. The following figure shows the settings for generating the correct frequency of 156.25 MHz.

Figure 12: **Clock Setup for Receiver**



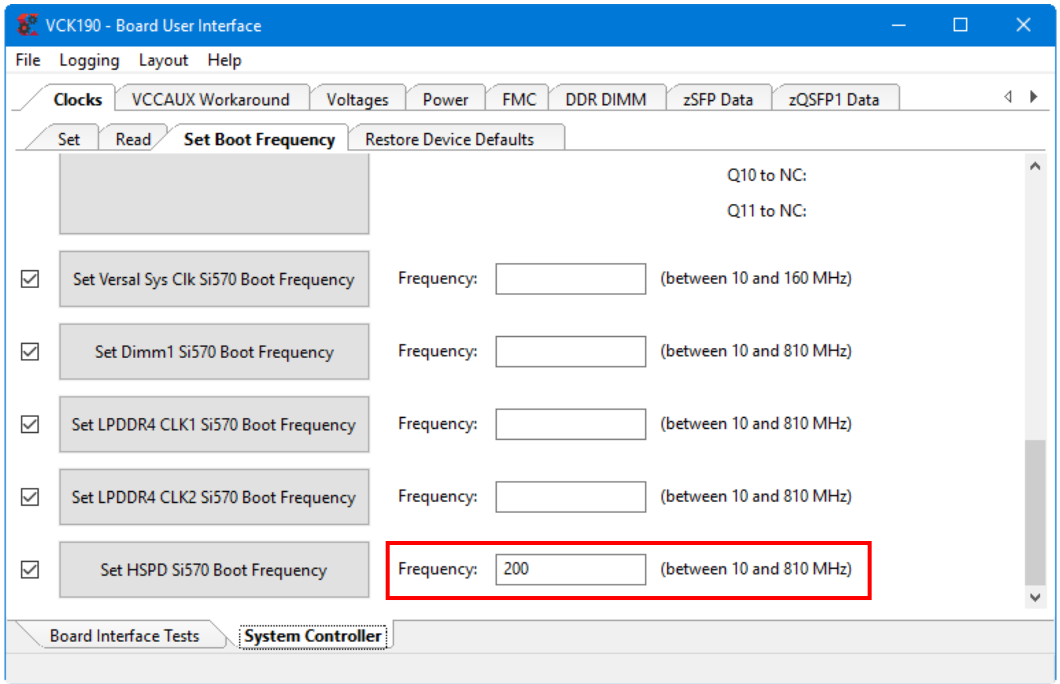
The following table shows the data rate and the frequency value to be set in the system controller GUI.

Table 4: **Data Rate and TX Frequency Setup**

	CONFIG 0	CONFIG 1	CONFIG 2	CONFIG 3	CONFIG 4	CONFIG 5	CONFIG 6
Rate Gb/s	1.25	2	2.5	3	3.5	4	8
TX ref clock MHz	200	200	200	150	175	200	200

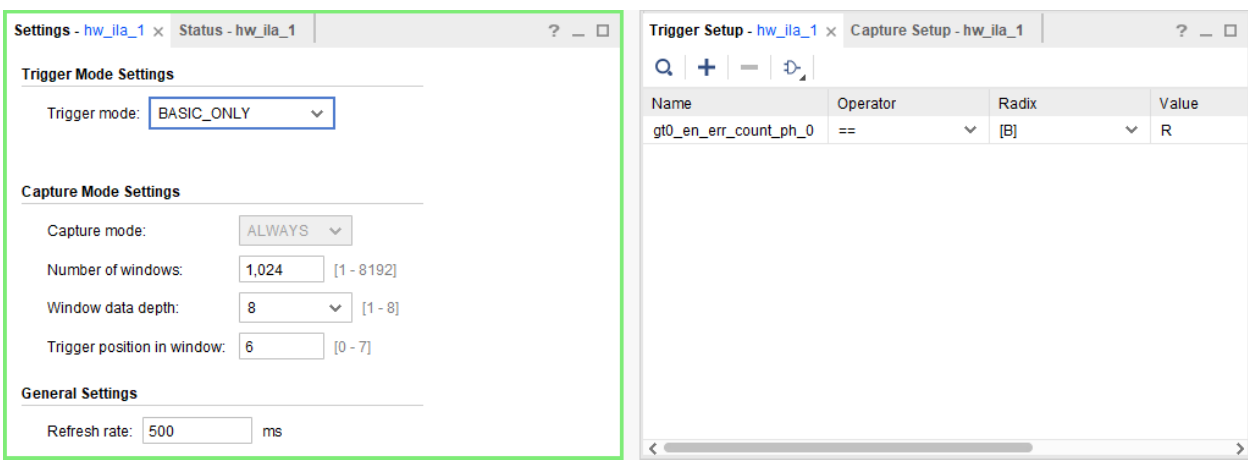
The following figure shows the settings for generating the correct frequency for the transmitter.

Figure 13: **Clock Setup for Transmitter**



To test the eye scan feature, configure the ILA Capture Mode Settings and Trigger Mode Settings as shown in the following figure.

Figure 14: **Vivado Logic Analyzer ILA Settings**



The trigger is done on the rising edge of the signal `EN_ERR_COUNT_0` asserting the signal `START_EYESCAN` available in the VIO window.

The following figures show a hardware eye scan with PH_NUM = 1 and PH_NUM = 2, respectively.

Figure 15: **Eye Scan with PH_NUM=1 (this is a Hardware Measurement)**

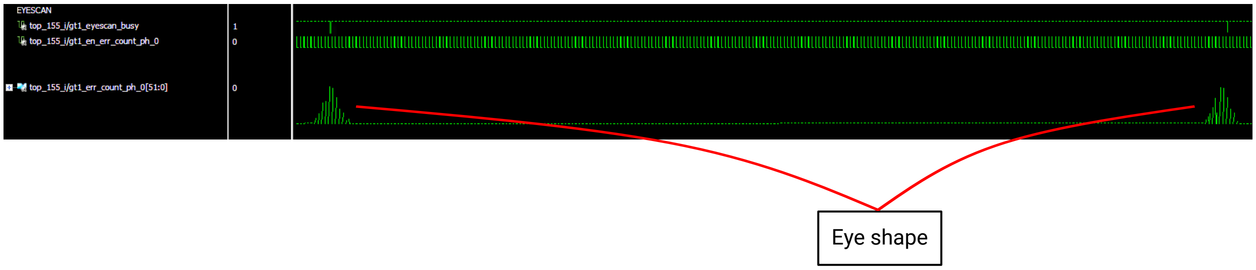
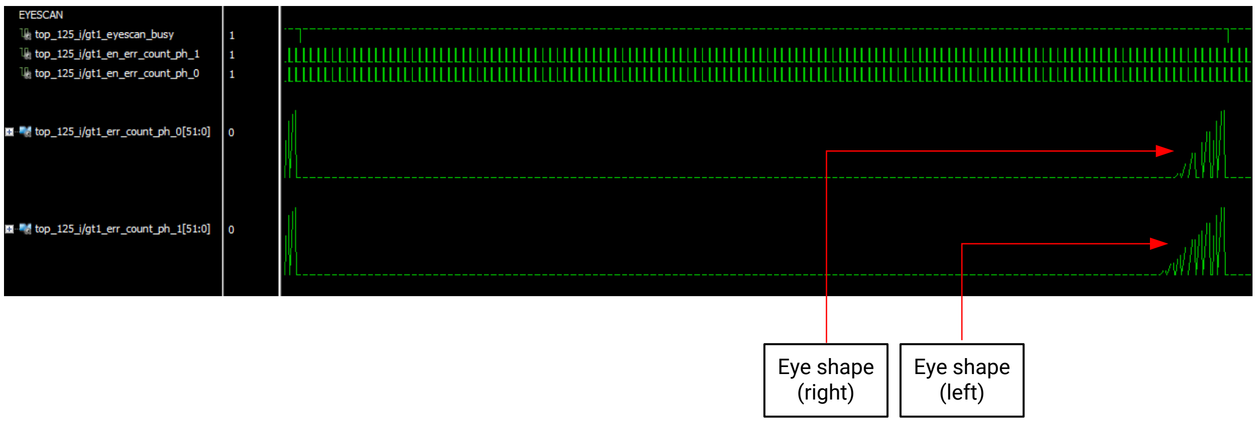


Figure 16: **Eye Scan with PH_NUM=2 (this is a Hardware Measurement)**



PHY Configuration

This section provides recommendations for correctly configuring the PHY.

The NIDRU processes oversampled data from a PHY, which is usually a SelectIO interface or a transceiver. In this second case, the receiver has to be configured in lock to reference mode, and its auto-adapting equalizer should be disabled by setting these ports to these values:

- RXCDRHOLD = 1
- RXLPMEN = 1

Resources

The NIDRU is designed with efficient structures only (adders, multipliers, accumulators, and shifters). The resource requirements for the Versal ACAP are summarized in the following table.

Table 5: **Hardware Resources Required for the NIDRU in Versal ACAPs**

Synthesis Type (NIDRU_DATA_WIDTH)	Eye Scan (PH_NUM)	Flip-Flops	LUTs	BUFG
32	0	1191	2308	1
	1	2483	4143	
	2	3105	5072	

Table 5: **Hardware Resources Required for the NIDRU in Versal ACAPs** *(cont'd)*

Synthesis Type (NIDRU_DATA_WIDTH)	Eye Scan (PH_NUM)	Flip-Flops	LUTs	BUFG
64	0	2336	4653	1
	1	5048	9148	
	2	6380	11451	
128	0	4481	9378	1
	1	9575	18488	
	2	12098	23055	

Notes:

1. Only one BUFG is required even if many channels are being set up, and even if all are working at different data rates.
2. These results were obtained using the Vivado Design Suite, version 2020.2. The strategy used for the synthesis and implementation was Default and the CLK period was 6.4 ns.

Software Requirements

The software required for the design is as follows:

- Vivado Design Suite, version 2020.2 or later.
- Mentor Graphics QuestaSim software, version 10.0c or later (for simulation).

Reference Design

Download the reference design files for this application note from the Xilinx website.

Reference Design Matrix

The following checklist indicates the procedures used for the provided reference design.

Table 6: **Reference Design Matrix**

Parameter	Description
General	
Developer name	Xilinx
Target devices	Versal ACAP
Source code provided?	Yes, partially encrypted
Source code format (if provided)	VHDL
Design uses code or IP from existing reference design, application note, 3rd party or Vivado software? If yes, list.	This reference design uses code from *An Attribute-Programmable PRBS Generator and Checker* (XAPP884).
Simulation	
Functional simulation performed	Yes
Timing simulation performed?	No
Test bench provided for functional and timing simulation?	Yes
Test bench format	VHDL
Simulator software and version	Mentor Graphics Questa Advanced Simulator 10.4c

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

Parameter	Description
SPICE/IBIS simulations	No
Implementation	
Synthesis software tools/versions used	Vivado synthesis 2020.2
Implementation software tool(s) and version	Vivado implementation 2020.2
Static timing analysis performed?	Yes
Hardware Verification	
Hardware verified?	Yes
Platform used for verification	VCK190 evaluation board

Conclusion

The NIDRU is a fully fractional over sampler, which allows extending the operating line rate of the GTY transceiver down to 0 Mbit/s. The NIDRU comes with a simulation and a HW test bench, which highlight the following:

- Multirate support
- DSP-based and hitless eye scan
- Run-time PPM

References

These documents provide supplemental material useful with this guide:

1. Best, Roland E. 1999. Fourth edition. *Phase-Locked Loops: Design, Simulation, and Applications*. McGraw-Hill Professional Publishing.
2. Gardner, Floyd M. 2005. Third edition. *Phaselock Techniques*. Wiley-Interscience.
3. *An Attribute-Programmable PRBS Generator and Checker* (XAPP884).

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
02/25/2021 Version 1.0	
Initial release.	N/A

Please Read: Important Legal Notices