



ALL PROGRAMMABLE™

XAPP1294 (v1.0) August 30, 2016

Lightweight and Scalable 4x Oversampling Asynchronous Data Recovery Unit for Single-Ended or Differential Inputs

Authors: Catalin Baetoni, David Taylor, and Vincent Vendramini

Summary

This application note describes a method for capturing asynchronous communication using SelectIO™ interface primitives. The method consists of oversampling the data with a clock of similar frequency ($< \pm 10,000$ ppm), taking multiple samples of the input data at different phases, and processing these to get a sample of the data at the most ideal point to give error-free data recovery.

The SelectIO interface performs 4x asynchronous oversampling using an IDDR primitive. Clocks are generated from an MMCM or PLL primitive and are routed through BUFG clock networks and can operate on single-ended or differential signaling using any chosen input within the device.

The example design provided with this application note is designed for an Artix®-7 XC7A200T-1FBG676C FPGA (-1 speed grade) running on an AC701 evaluation board. It is delivered as an IP repository block to be added to the Vivado® Design Suite IP catalog.

The design uses about 20 LUTs per channel, operates at up to 200 Mb/s, and requires 400 MHz, 200 MHz, and 100 MHz clocks from an MMCM or PLL to provide data recovery at 200 Mb/s. The operating speed of the data recovery is determined by the applied clock speeds and can be readily modified provided the same overall clock ratios are maintained.

The design specifications are:

- Minimum duty cycle: 40%/60%
- Operating tolerance: 10,000 ppm
- Single-ended design with no placement restrictions on I/O banks

You can download the [reference design files](#) for the example design from the Xilinx® website. For detailed information about the design files, see [Reference Design](#).

Introduction

Synchronizing the clock and data is a common method of achieving communication between devices. This means that the clock is transmitted on one channel and the data on one or several other inputs (differential or single-ended). The clock at the receiver is used to capture the data after delay synchronization. This is known as source-synchronous communication.

When transmitting data without a separate accompanying clock signal, the clock used to capture the data must be recovered at the receiver side from the incoming data stream. This is called asynchronous communication, also known as data and/or clock recovery. Xilinx serial transceivers use this principle. Data recovery allows a receiver to extract data from the incoming clock/data stream and then move the data into a new clock domain. Sometimes, the recovered clock is used for onward data treatment or transmission.

The circuit described in this application note provides a solution where no clock is actually recovered, but the arriving data is fully extracted from asynchronous data.

Design Details

Asynchronous Oversampling

For signal processing, *oversampling* means sampling a signal using a sampling frequency significantly higher than twice the bandwidth (or highest frequency) of the signal being sampled. For the communication interface described in this application note, the significantly higher sampling frequency is obtained using different edges of faster clocks. It is called asynchronous oversampling because the clocks used to create the sampling frequency are nominally equal to the data stream frequency, but not exact or synchronous.

The circuit discussed here uses a clock (local oscillator) running at an integer multiple of the same *nominal* frequency as the data stream being captured. Nominal here means that the local oscillator is either slightly faster or slightly slower than the incoming clock/data stream.

The oversampling (PHY block) in the DRU_PHY uses a 400 MHz clock to capture data on rising and falling edges. This operates at a continuous sample rate of 800 MSPS and has a 4-bit raw output which feeds the data recovery unit (DRU block) on the 200 MHz clock.

The raw output is also available as a debug port and can be used if required.

Data Recovery Unit

The DRU block uses a 200 MHz clock for the raw oversample data and 100 MHz for data recovery. All clocks are edge aligned from the same clock wizard module.

The DRU has an XOR array to locate the data transitions in the raw oversample data—these are captured in array E4. The edge positions in E4 are then used by the data select FSM to determine which oversample bit is valid. That is, select a data bit away from the data transitions.

The data select finite state machine (FSM) is shown in [Figure 1](#). The FSM state determines which oversample bit is valid based on the E4 transition information.

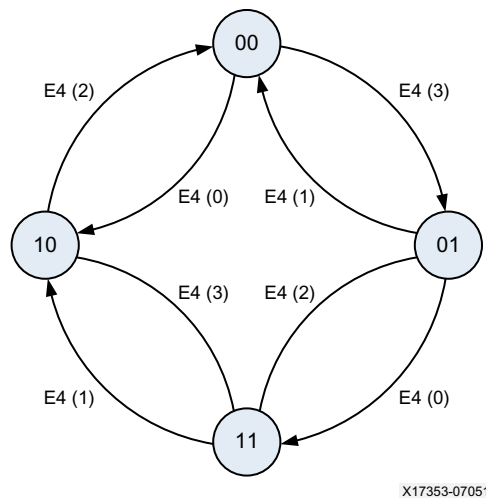


Figure 1: Data Select with Edge Information

When the edge is detected to move from the current oversample word to another, this is known as a bit skip and the FSM transitions a complete revolution, either positive or negative. This will then either insert or remove a sample bit on that output cycle to compensate for the asynchronous frequency difference between the received data stream and the local clock source. For example, when receiving 200 Mb/s data faster than the local clock, three valid data bits are set when a bit skip is introduced. When receiving data slower than the local clock, one valid data valid bit is set when a bit skip is introduced.

More information on a DRU x4 FSM and bit skip operation is available in the *Virtex-6 FPGA LVDS 4X Asynchronous Oversampling at 1.25 Gb/s Application Note* (XAPP881) [\[Ref 1\]](#).

The output data bus configuration is shown in [Table 1](#).

Table 1: Output Data Bus Configuration

Data Rate	Data and Data Valid Bit Widths 100 MHz Clock Domain	Nominal Output
200 Mb/s	3 + 3 bits	2 valid bits

Physical interface

[Table 2](#) describes the input and output signals of DRU_PHY IP core shown in [Figure 3](#).

Table 2: Physical interface of the DRU_PHY IP Core

Signal Name	Direction	Description
RxData	Input	Received data.
CLK1x	Input	Slowest clock, or half the nominal frequency. When the nominal data rate is 200 Mb/s, this clock is at 100 MHz.
CLK2x	Input	Nominal frequency clock, 200 MHz at data rate of 200 Mb/s.

Table 2: Physical interface of the DRU_PHY IP Core (Cont'd)

Signal Name	Direction	Description
CLK4x	Input	Oversampling clock, twice the frequency of the nominal clock, 400 MHz at data rate of 200 Mb/s
dout_raw[3:0]	Output	Output of the PHY, used for debug. Synchronous to CLK2X.
dout[2:0]	Output	Data output of the DRU. Synchronous to CLK1X.
dout_valid[2:0]	Output	Data output qualifier. 001 dout LSB is valid, 011 dout[1:0] is valid, 111 dout[2:0] is valid. Synchronous to CLK1x.

Resource Usage, DRU IP Core

Table 3 shows the resource usage for one DRU channel standalone.

Table 3: DRU IP Core Resource Usage

Description	Quantity
Target Device	Artix-7XC7A200T-1FBG676C FPGA
Slice LUTs	17
Slice Registers	44
Occupied Slices	12
Block RAM	0
BUFG	0
MMCM	0

Implementation and Deliverables

The DRU is delivered as an IP core. After adding it to the Vivado IP catalog, it can be instantiated in HDL code or as part of a Vivado IP Integrator project.

Requirements

Hardware



TIP: These three items are included with the AC701 evaluation kit:

- AC701 evaluation board
- Power supply: 100–240 VAC input, 12 VDC 5.0A output
- One USB cable, standard-A plug to micro-B plug
- One SMA cable

Computer

- X86-based computer with:
 - Two USB ports
 - OS supported by the Vivado Design Suite

Software

The reference design requires this software to be installed before building and running the example design:

- Vivado Design Suite 2016.2
- Silicon Labs Dual CP210x USB UART Drivers (UART drivers required for serial communication) See *Silicon Labs CP210x USB-to-UART Installation Guide* (UG1033) [Ref 3] for installation instructions.

Set Up the AC701 Evaluation Board

Referring to [Figure 2](#):

1. Place switch SW1 to the OFF position and connect the power supply to J49.
2. Connect the USB cables between the computer and the JTAG connector on the AC701 evaluation board and the computer.
3. Connect J33 (User Clk P) to J31(User GPIO P) using an SMA cable.

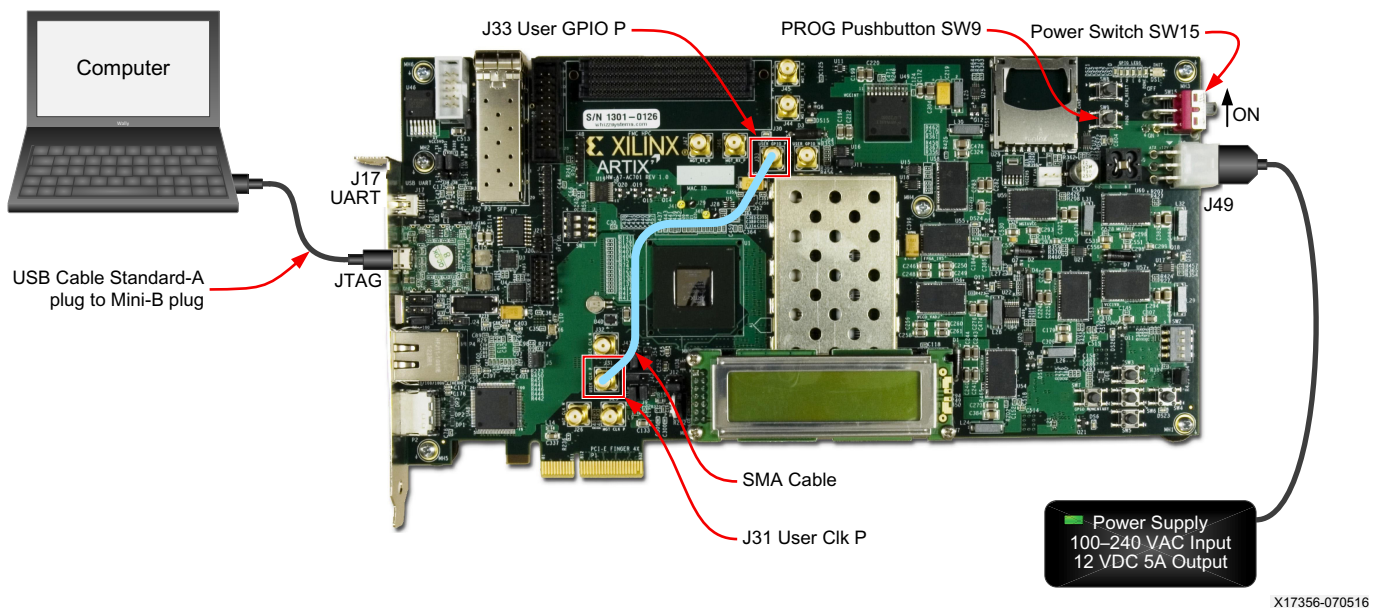


Figure 2: AC701 Evaluation Board Connections

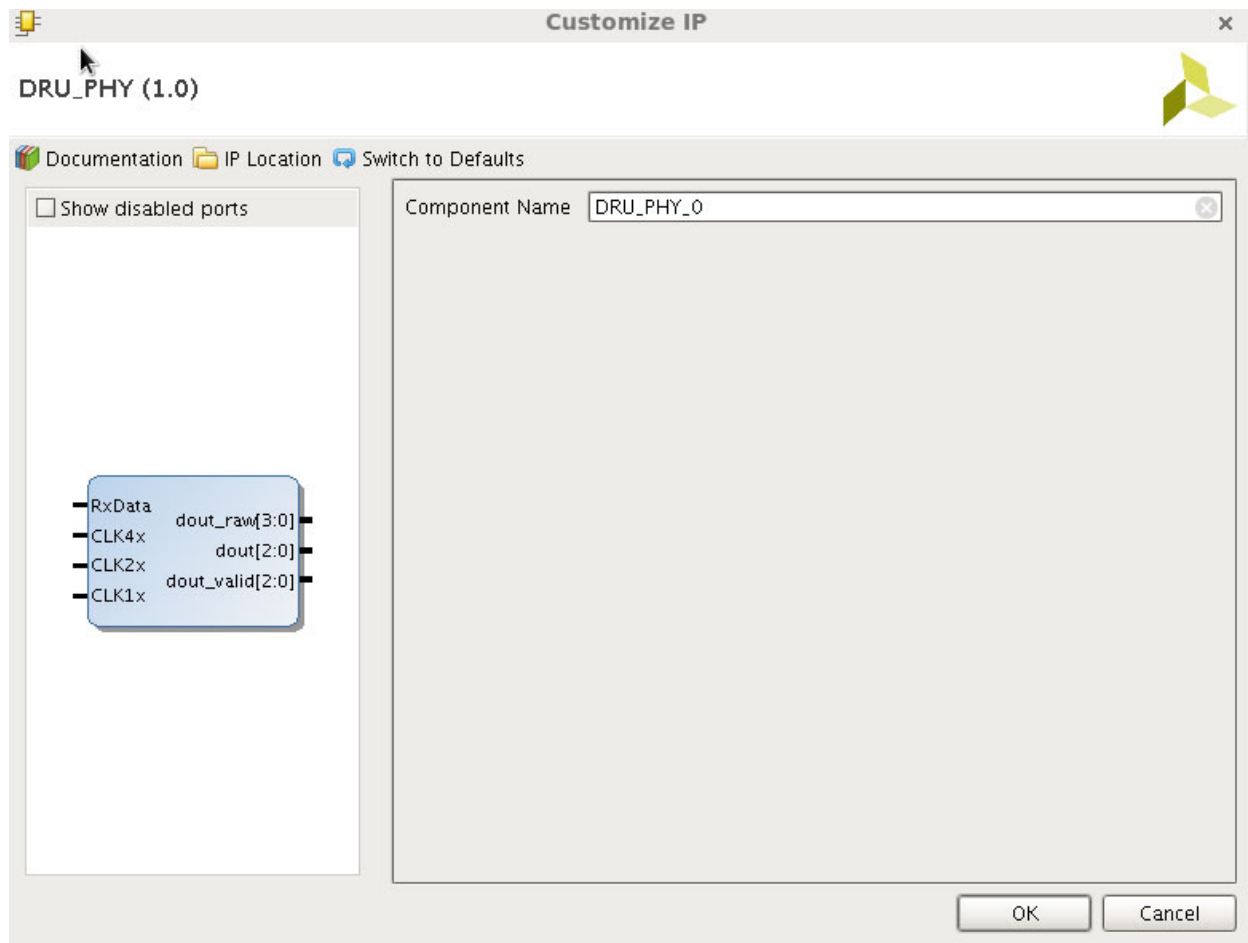
Download Reference Design Files

1. Download and unzip the [reference design files](#) to a <working directory> on the computer.

Add the DRU IP Core to a Project

This section describes the steps to add the DRU to a project:

1. In the Vivado Design Suite, add the IP repository to the project by selecting **Tools > Project Options**, select **IP** on the left window, click **Add Repository**, and select the <working directory>/DRU_PHY folder.
2. Select **Vivado IP catalog**. The DRU_PHY IP is under **User Repository > FPGA Features and Design > IO Interfaces**.
3. Right-click **DRU_PHY** and select **Customize IP**. There are no user-configurable options except the name of the DRU ([Figure 3](#)). Click **OK**.



X17354-070516

Figure 3: DRU_PHY IP Configuration

Example Design

Figure 4 shows the example design.

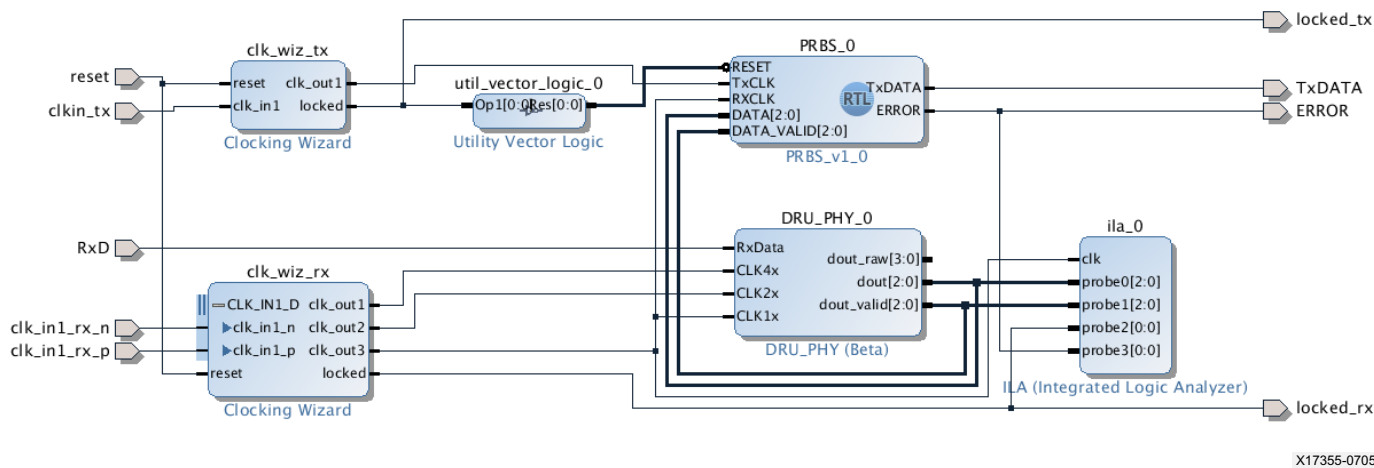


Figure 4: Vivado IP Integrator Example Design

After the IP is added to the project, an example design targeting the AC701 evaluation board can be created after the DRU_PHY IP core is added to the project.

To generate the example design shown in Figure 4:

1. In the Vivado source window, locate the **DRU_PHY** IP core.
2. Right-click **DRU_PHY** and select **Generate example design**.

The example design contains a synthesizable pseudo-random binary sequence (PRBS) generator and checker as well as the necessary clocking wizards and Integrated Logic Analyzer (ILA). The clocking wizard for the transmit data generates a 201 MHz clock having a 40/60 duty cycle, while the receiver clocking wizard generates clocks at 100 MHz, 200 MHz, and 400 MHz having a 50/50 duty cycle. The clocking wizards can be configured for other input clock rates.

Testbench and Verification

The reference design contains a testbench that provides functional simulation of the system. To start the simulation shown in Figure 5, click **Run simulation** in Vivado flow navigator. The testbench features include:

- 200 Mb/s line rates
- PRBS31 pattern generation and error checking
- $\pm 10,000$ ppm clock offsets
- 40/60 data duty cycle distortion

Figure 5 shows the simulation and illustrates the bit skip in operation. When the clock is fast with respect to the DRU reference clock an extra bit is inserted periodically in the data output, dout_valid = 111.



X17362-070816

Figure 5: Example Design Simulation Waveforms

Reference Design

You can download the [reference design files](#) for this application note from the Xilinx website. [Table 4](#) shows the reference design matrix.

Table 4: Reference design matrix

Parameter	Description
General	
Developer Names	Catalin Baetoni, David Taylor, and Vincent Vendramini
Target Device	Artix-7 XC7A200T-1FBG676C FPGA
Source code provided	Yes
Source code format	VHDL and Vivado IP Integrator
Design uses code/IP from existing Xilinx application existing Xilinx application note/reference designs, CORE Generator™ tool, or third party	Yes
Simulation	
Functional Simulation Performed	Yes
Timing Simulation Performed	Yes, post synthesis
Test bench used for functional and timing simulations	Yes
Test bench format	VHDL
Simulator software tools and version used	Vivado simulator version 2016.2
SPICE/IBIS Simulations	N/A
Implementation	
Synthesis software tools/version used	Vivado Design Suite 2016.2
Implementation software tools/version used	Vivado Design Suite 2016.2
Static timing analysis performed	Yes
Hardware Verification	
Hardware verified	Yes
Hardware platform used for verification	AC701 evaluation board [Ref 2]

Reference Design Resource Usage

[Table 5](#) shows the resource usage for the example design with one DRU channel.

Table 5: Reference Design Resource Usage

Description	Quantity
Target Device	Artix-7XC7A200T-1FBG676C FPGA
Slice LUTs	1,202
Slice Registers	1,793
Occupied Slices	571
Block RAM	1
BUFG	8
MMCM	2

References

1. *Virtex-6 FPGA LVDS 4X Asynchronous Oversampling at 1.25 Gb/s Application Note* ([XAPP881](#))
 2. *AC701 Evaluation Board for the Artix-7 FPGA User Guide* ([UG952](#))
 3. *Silicon Labs CP210x USB-to-UART Installation Guide* ([UG1033](#))
 4. *AC701 Evaluation Kit web page* (www.xilinx.com/AC701)
-

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/30/2016	1.0	Initial Xilinx release.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2016 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.