

DG0760
Demo Guide
RTG4 SerDes XAUI



a  **MICROCHIP** company



a  MICROCHIP company

Microsemi Headquarters

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

©2021 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 6.0	1
1.2	Revision 5.0	1
1.3	Revision 4.0	1
1.4	Revision 3.0	1
1.5	Revision 2.0	1
1.6	Revision 1.0	1
2	RTG4 SerDes XAUI	2
2.1	Design Requirements	2
2.2	Prerequisites	2
2.3	Demo Design	3
2.3.1	Design Implementation	4
2.3.2	IP Configuration	4
2.4	Clocking Structure	7
2.5	Reset Structure	7
2.6	Simulating the Design	8
2.7	Conclusion	8
3	Appendix 2: Running the TCL Script	9

Figures

Figure 1	SerDes XAUI Demo Design Top-Level Block Diagram	3
Figure 2	SerDes XAUI Design	4
Figure 3	High-Speed Serial Interface Configurator	5
Figure 4	CCC Configurator	6
Figure 5	Clocking Structure	7
Figure 6	Reset Structure	7
Figure 7	Simulating the Design	8
Figure 8	Simulation Waveform Window	8

Tables

Table 1	Design Requirements	2
---------	---------------------------	---

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 6.0

Replaced Figure 2, page 4 and Figure 4, page 6.

1.2 Revision 5.0

Information about input reference clock was added in Figure 5, page 7.

1.3 Revision 4.0

The following is a summary of the changes made in this revision.

- Replaced Figure 4, page 6.
- Added Appendix 2: Running the TCL Script, page 9.
- Removed the references to Libero version numbers.

1.4 Revision 3.0

Updated the document for Libero v11.9 SP1 software release.

1.5 Revision 2.0

Updated the document for Libero v11.8 SP2 software release.

1.6 Revision 1.0

Revision 1.0 is the first publication of this document.

2 RTG4 SerDes XAUI

RTG4™ FPGAs have embedded high-speed SerDes blocks that can handle data rates from 1 Gbps to 5 Gbps. The SerDes module integrates several functional blocks to support multiple high-speed serial protocols within the FPGA. XAUI is a standard for extending the 10 Gb media independent interface (XGMII) between the media access control (MAC) and PHY layer of 10 Gb Ethernet (10 GbE). RTG4 high-speed serial block implements hardened XAUI.

This document contains information with respect to the software simulations of the XAUI protocol.

2.1 Design Requirements

The following table lists the software requirements for this demo design.

Table 1 • Design Requirements

Requirement	Version
Operating system	64-bit Windows 7 and 10
Software	
Libero® System-on-Chip (SoC)	Refer to the <code>readme.txt</code> file provided in the design files for the software versions used with this reference design.

Note: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

2.2 Prerequisites

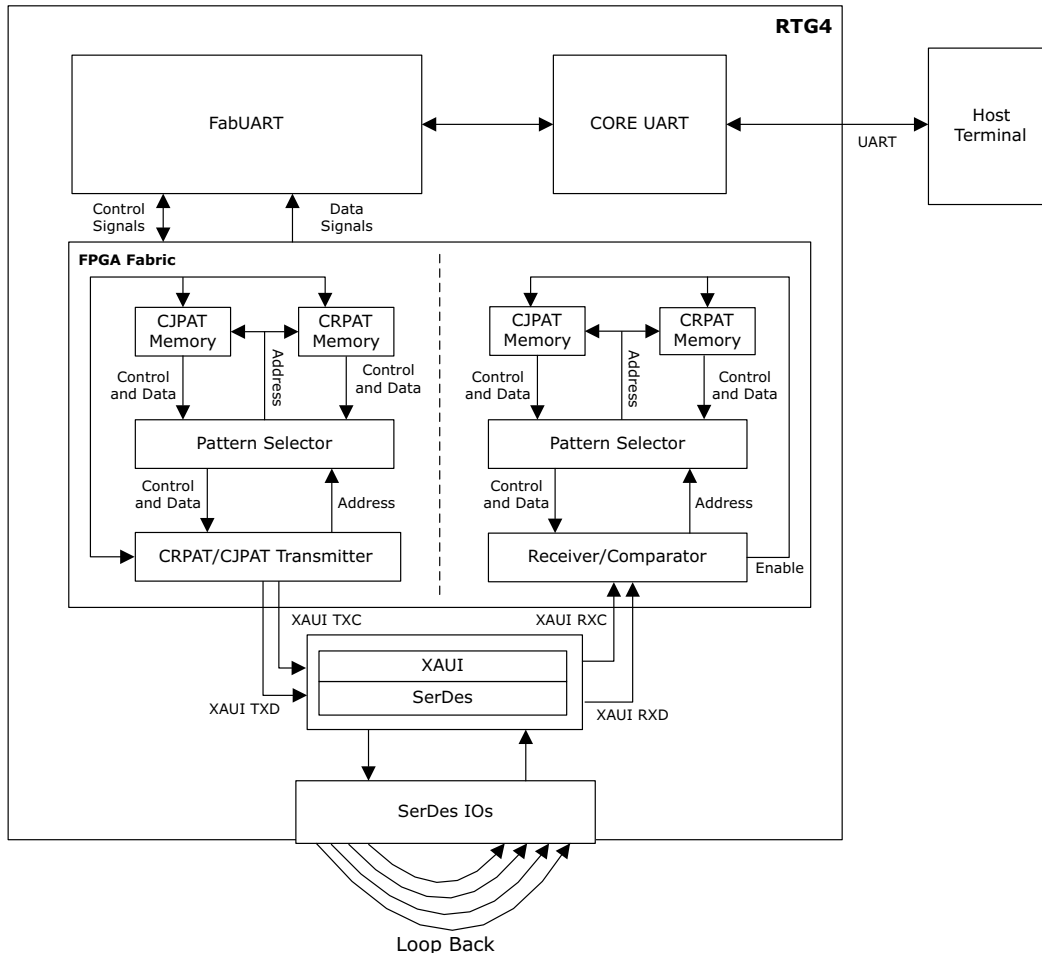
Before you start:

1. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location: <https://www.microsemi.com/product-directory/design-resources/1750-libero-soc>
2. Download the design files from the following link:
http://soc.microsemi.com/download/rsc/?f=rtg4_dg0760_df
3. Start Libero SoC software, and in the **Project** menu, click **Open Project**.
4. Browse the RTG4_XAUI Libero project folder and open the `Libero_Project.prjx` file. The RTG4 SerDes XAUI project opens.
5. Open the **Design Hierarchy** window and double-click the **RTG4_XAUI** component. The SmartDesign page opens on the right pane and displays the high-level design. You can view the design blocks and IP cores instantiated for the SerDes XAUI design.

2.3 Demo Design

This demo design demonstrates transmitting continuous random test pattern (CRPAT) and continuous jitter test pattern (CJPAT) over the RTG4 high-speed SerDes interface. SerDes is configured for 3.125 Gbps operational speed. The system block diagram for the design implemented in the RTG4 device, as shown in the following figure.

Figure 1 • SerDes XAUI Demo Design Top-Level Block Diagram



CRPAT_MEM: This block implements the IEEE 802.3ae CRPAT and generates a continuous stream of identical patterns, separated by a minimum inter-packet gap (IPG). The CRPAT is intended to provide broad spectral content and minimal peaking used for the measurement of jitter at either a component or system level.

Each packet in the CRPAT consists of eight octets of PREAMBLE/SFD, followed by 1488 data octets (124 repetitions of the 12-octet modified RPAT sequence), plus four CRC octets, followed by a minimum IPG of 12 octets of IDLE.

CJPAT_MEM: This block implements the IEEE 802.3ae CJPAT. The CJPAT is intended to expose a receiver CDR to large instantaneous phase jumps. The pattern alternates repeating low-transition density patterns with repeating high-transition density patterns. The repeating code-group duration should be longer than the time constants in the receiver clock recovery circuit. This assures that the clock phase has followed the systematic pattern jitter and the data sampling circuitry is exposed to large systematic phase jumps. This stresses the timing margins in the received eye.

Pattern selector: This module is used for the selection of either CRPAT or CJPAT to be sent onto the transmitter of XAUI.

Transmitter: It sends the pattern along with control signals onto the SerDes interface. The data is fed onto the transmitter from the pattern selector block. The transmitter sends 64 bits of data signals onto the TXD interface and eight bits of control signals onto the TXC interface of the SerDes.

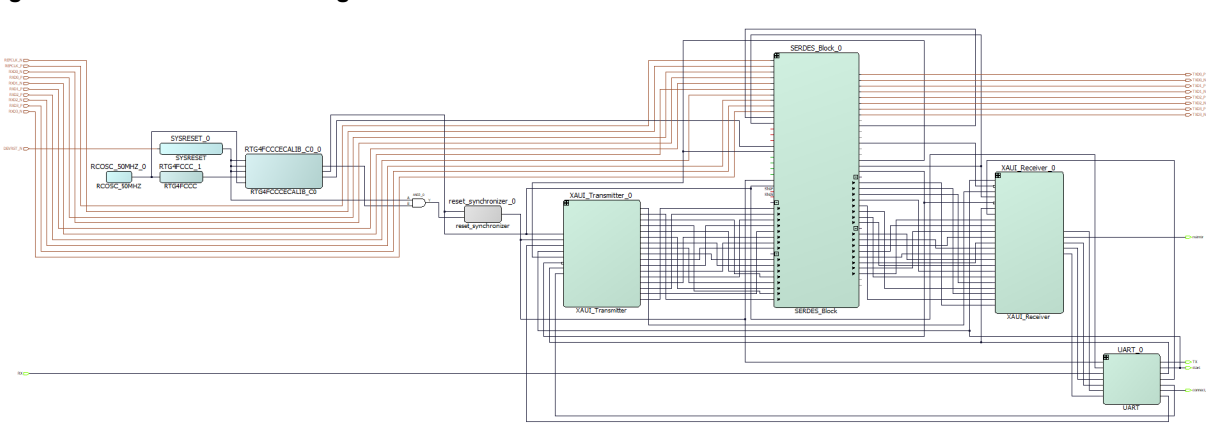
Receiver: This block checks for corresponding CRPAT or CJPAT sequences depending on the pattern selection done and sent over the SerDes interface. If the received sequence does not match with the one transmitted by the transmitter, the receiver indicates an error. The receiver also implements an error counter, which is incremented for each error in the received pattern.

Debounce: This logic is written for detecting the stable pulse. On detecting a stable pulse, the logic raises a flag which is used to initiate the XAUI demo.

2.3.1 Design Implementation

The following figure shows the Libero SoC software design implementation of the SerDes XAUI.

Figure 2 • SerDes XAUI Design



2.3.2 IP Configuration

The following IPs and macros need to be configured before simulating the demo design:

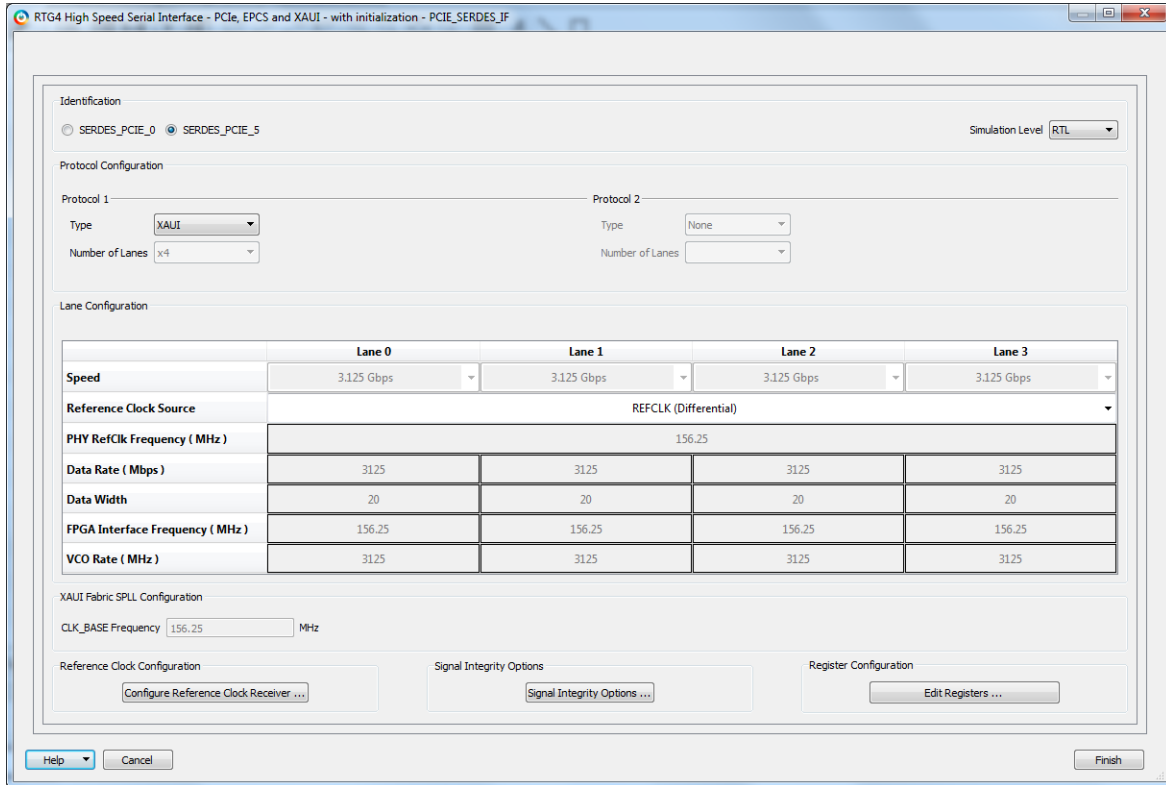
- High-speed serial interface
- Clock conditioning circuitry (CCC)

2.3.2.1 High-Speed Serial Interface

The RTG4 FPGA high-speed SerDes is a hard IP block on chip that supports rates up to 5 Gbps. The SerDes block offers embedded protocol support for PCIe, SRIO, XAUI, SGMII, EPCS, and so on. For more information about the SerDes block, refer to *UG0567: RTG4 FPGA High Speed Serial Interfaces User Guide*.

In this demo, the SERDESIF_Block block is configured for the XAUI mode, with reference clock from REFCLK0 coming from an external source.

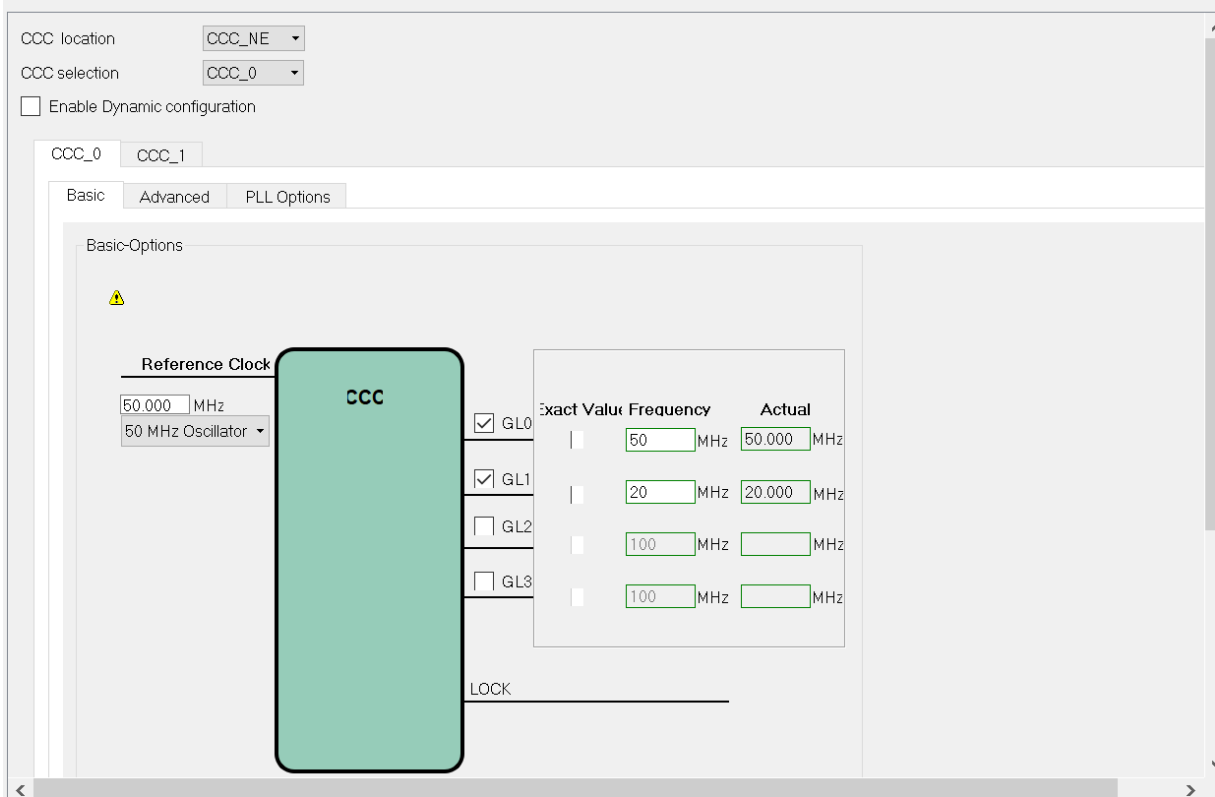
Figure 3 • High-Speed Serial Interface Configurator



2.3.2.2 Clock Conditioning Circuitry

RTG4FCCCECALIB provides reference clock needed for a clock for the CoreUART, FabUART, and clock for management block in the SerDes XAUI mode. For more information about CCC blocks, refer to *UG0586: RTG4 FPGA Clocking Resources User Guide*.

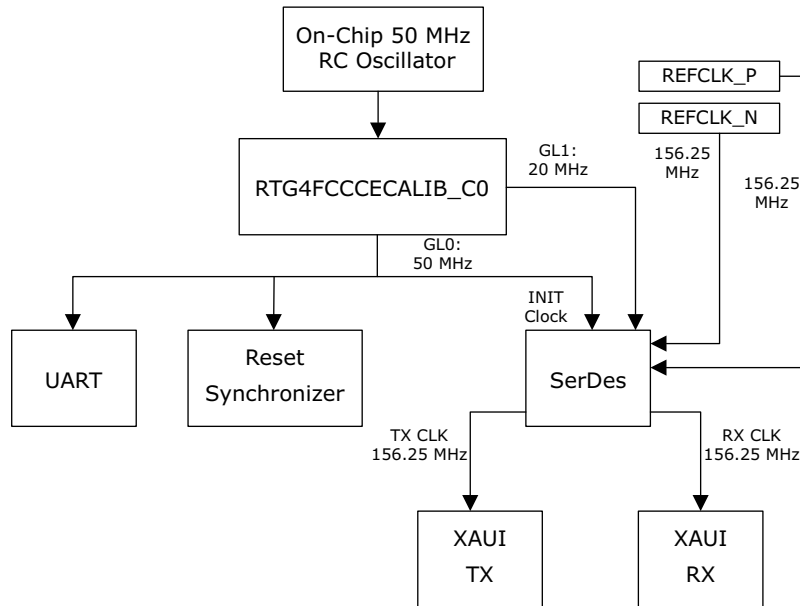
Figure 4 • CCC Configurator



2.4 Clocking Structure

The on-chip 50 MHz oscillator provides the reference clock to the RTG4FCCCECALIB_C0 block. The RTG4FCCCECALIB_C0 block provides two clocks, GL0 (50 MHz) and GL1 (20 MHz). The GL0 clock drives the SerDes block, UART, and Reset Synchronizer blocks. The GL1 clock drives the SerDes block. The following figure shows the clocking structure of the design. For more information about RTG4FCCCECALIB_C0, refer to *UG0590: RTG4 FPGA Clock Conditioning Circuit with PLL Configuration User Guide from Libero*.

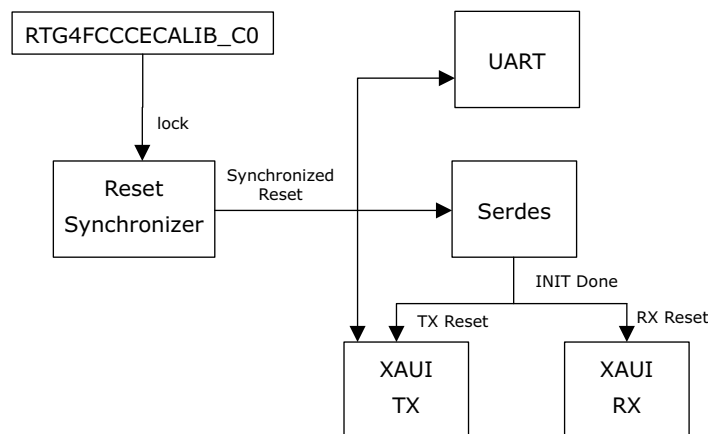
Figure 5 • Clocking Structure



2.5 Reset Structure

The LOCK signal from RTG4FCCCECALIB_C0 is synchronized by using Reset Synchronizer block. The synchronized LOCK signal is then provided to UART, SerDes, and XAUI transmitter. The INIT Done signal from SerDes block is fed to XAUI TX and XAUI RX.

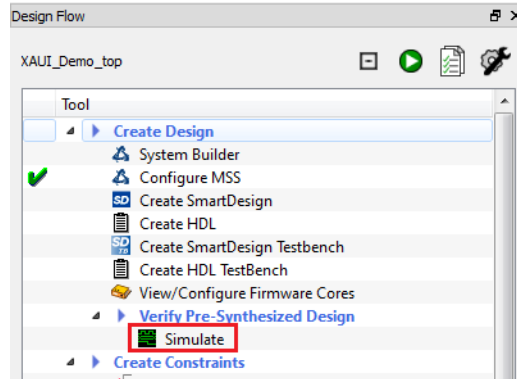
Figure 6 • Reset Structure



2.6 Simulating the Design

In the **Design Flow** tab, double-click **Simulate** under **Verify Pre-Synthesized Design** to simulate the design, as shown in the following figure. The ModelSim tool takes about 3 to 5 minutes to complete the simulation.

Figure 7 • Simulating the Design

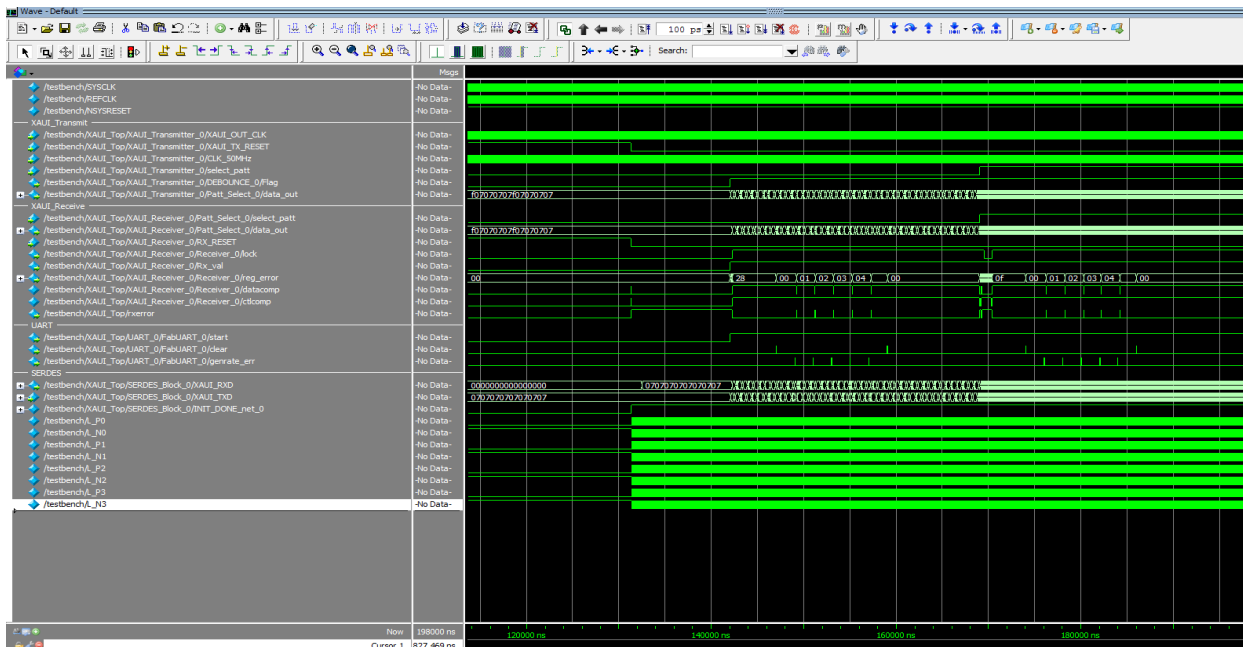


The demo design consists of following signals:

- 50 MHz and 156.5 MHz clock
- Reset
- XAUI transmitter signal
- XAUI receiver signal
- UART signal
- SerDes signal

When the simulation is successful, the waveform window appears as shown in the following figure.

Figure 8 • Simulation Waveform Window



2.7 Conclusion

This demo described how to simulate the SerDes XAUI IP cores and how to implement the Libero design flow using Libero SoC software.

3 Appendix 2: Running the TCL Script

TCL scripts are provided in the design files folder under directory TCL_Scripts. If required, the design flow can be reproduced from Design Implementation till generation of job file.

To run the TCL, follow the steps below:

1. Launch the Libero software
2. Select **Project > Execute Script...**
3. Click Browse and select `script.tcl` from the downloaded TCL_Scripts directory.
4. Click **Run**.

After successful execution of TCL script, Libero project is created within TCL_Scripts directory.

For more information about TCL scripts, refer to [rtg4_dg0760_df/TCL_Scripts/readme.txt](#).

Refer to [Libero® SoC TCL Command Reference Guide](#) for more details on TCL commands. Contact Technical Support for any queries encountered when running the TCL script.