LogiCORE IP RXAUI v2.1

DS740 March 1, 2011 **Product Specification**

Introduction

The LogiCORE™ IP RXAUI core is a high-performance, low pin count 10-Gb/s interface intended to allow physical separation between the data-link layer and physical layer devices in a 10-Gigabit Ethernet system.

The RXAUI core implements a single-speed full-duplex 10-Gb/s Ethernet Reduced Pin eXtended Attachment Unit Interface (RXAUI) solution for Xilinx Virtex®-6, Virtex-7, and Kintex™-7 FPGAs that comply with the Dune Networks and Marvell RXAUI specifications.

Virtex-6 FPGAs in combination with the RXAUI core, enable the design of RXAUI-based interconnects whether they are chip-to-chip, over backplanes, or connected to 10-Gigabit optical modules.

Features

- Designed to Dune Networks and Marvell RXAUI specifications
- Uses GTX transceivers at 6.25 Gb/s line rate to achieve 10-Gb/s data rate
- Implements DTE XGXS, PHY XGXS, and 10GBASE-X PCS in a single netlist
- Uses Mixed-Mode Clock Managers
- Uses device-specific transceivers for the RXAUI interface
- *IEEE 802.3-2008* clause 45 MDIO interface (optional)
- Available under the Xilinx End User License Agreement

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family[1]	Virtex-6, Kintex-7, Virtex-7				
Supported User Interfaces	XGMII, MDIO				
	Resources[2]				**Frequency**
Configuration	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs**	**Max. Freq.**
Dune Networks, MDIO	775	875	0	0	156.25 MHz
Marvell Networks, MDIO	1500	1420	0	0	312.5 MHz
Provided with Core					
Documentation	Product Specification User Guide				
Design Files	NGC Netlist				
Example Design	Verilog/VHDL				
Test Bench	Verilog/VHDL				
Constraints File	UCF				
Simulation Model	UniSim-based Simulation Models				
Tested Design Tools					
Design Entry Tools	ISE 13.1 software				
Simulation	Mentor Graphics ModelSim v6.6d Cadence Incisive Enterprise Simulator (IUS) v10.2 Synopsys VCS and VCS MX 2010.06				
Synthesis Tools	XST				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.
2. For additional device performance numbers, see Table 9 and Table 10, page 11.

Overview

The RXAUI standard was developed as a means to improve the 10-Gigabit Ethernet port density. The number of XAUI interfaces that could be implemented was limited by the number of available transceivers, with capacity and performance still to be utilized. RXAUI halves the number of transceivers required compared with a XAUI implementation.

RXAUI is a two-lane, 6.25 Gb/s-per-lane serial interface. It is intended to work with an existing XAUI implementation and multiplexes/demultiplexes the two physical RXAUI lanes into 4 logical XAUI lanes. Each RXAUI lane is a differential pair carrying current mode logic (CML) signaling, and the data on each lane is 8B/10B encoded before transmission. The Dune Networks RXAUI implementation maintains 8B/10B disparity per RXAUI physical lane and the Marvell RXAUI implementation maintains 8B/10B disparity per XAUI logical lane. Special code groups are used to allow each lane to synchronize at a word boundary and to de-mux the two physical RXAUI lanes into four logical XAUI lanes at the receiving end. For full details on the RXAUI specification, contact Dune Networks or Marvell.

Applications

The applications of RXAUI have extended beyond 10-Gigabit Ethernet to backplane and other general high-speed interconnect applications. Figure 1 shows a typical backplane application.

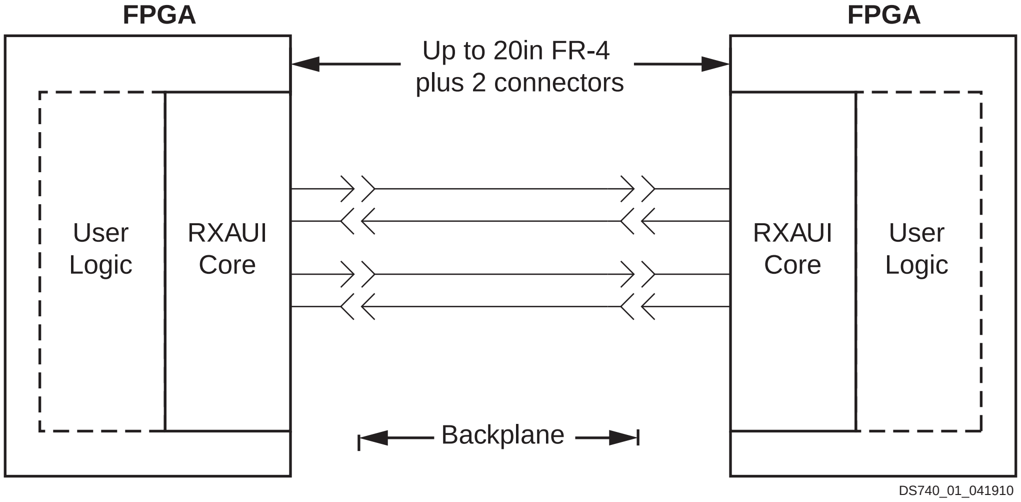

Figure 1: **Typical Backplane Application for RXAUI**

Functional Description

Figure 2 shows a block diagram of the Dune Networks RXAUI core implementation. The major functional blocks of the core include the following:

- **Transmit Idle Generation Logic** creates the code groups to allow synchronization and alignment at the receiver.
- **Demux Logic** separates the two physical RXAUI lanes into four logical XAUI lanes.
- **Synchronization State Machine (one per lane)** identifies byte boundaries in incoming serial data.
- **Deskew State Machine** monitors the deskew logic per the IEEE 802.3-2008 specification.
- **Optional MDIO Interface** is a two-wire low-speed serial interface used to manage the core.
- **Transceiver** (integrated in the FPGA) provides the high-speed transceivers as well as 8B/10B encode and decode, and elastic buffering in the receive datapath.

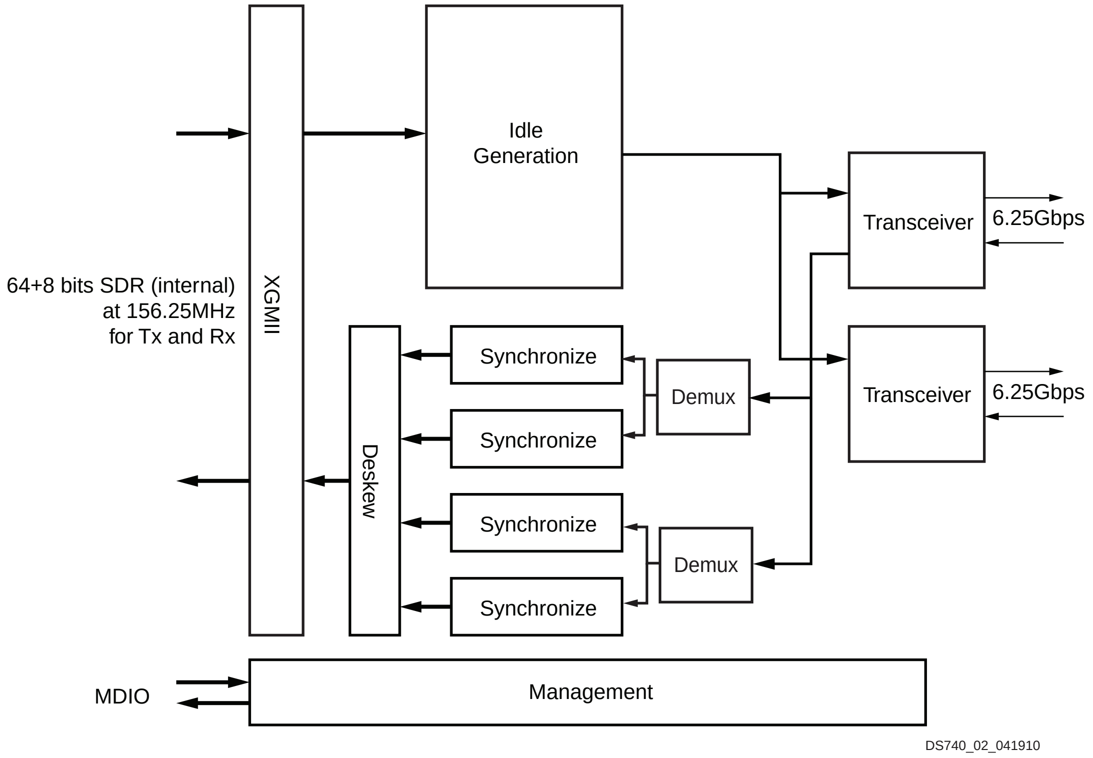

Figure 2: **Implementation of Marvell RXAUI Core**

Figure 3 shows a block diagram of the Marvell RXAUI core implementation. The major functional blocks of the core include the ones previously described in the Dune Networks implementation, and the following:

8B/10B encoder / decoder: Performs 8B/10B data conversion

Deskew: deskews the four received lanes into alignment

Clock Correction: Elastic buffer and clock correction manipulation to handle difference in clock rates

The core is implemented with the transceiver instantiations in the source code rather than in the netlist. This gives the user more flexibility in their particular application to use additional device-specific transceiver features and resolve placement issues.

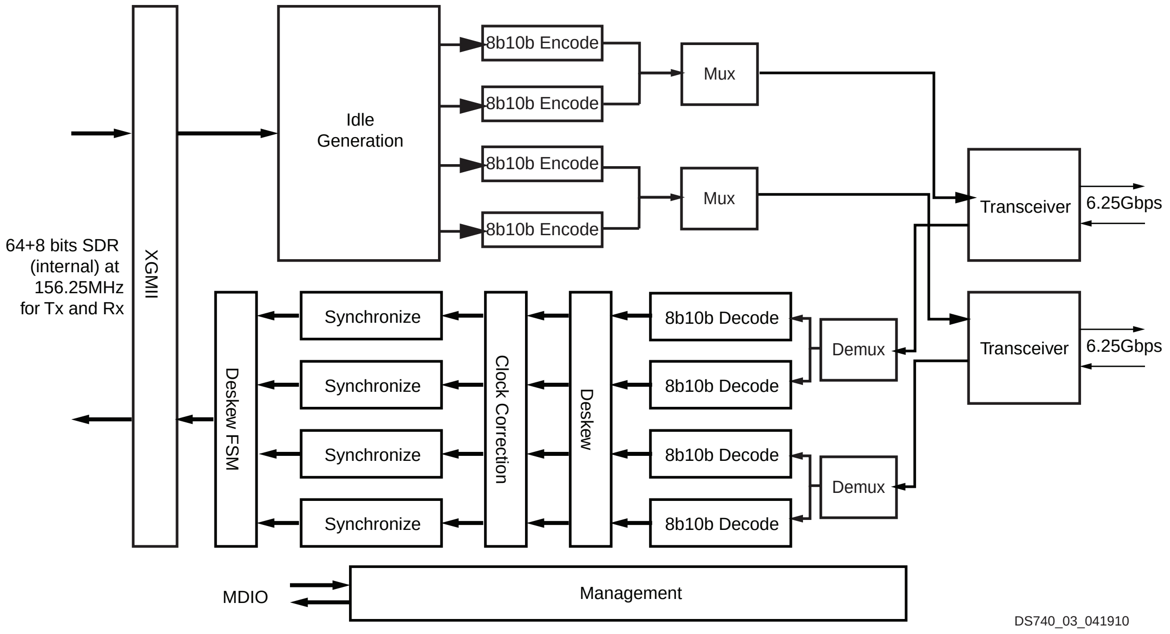

Figure 3: **Implementation of Marvell RXAUI Core**

Core Interfaces

RXAUI Interface

The RXAUI interface consists of two differential transmit and receive pairs plus two low-speed control inputs to indicate the status of an attached optical module. The differential pairs are generated by the transceiver. The control input signals are connected directly to the core. These signals are described in Table 1.

Table 1: **RXAUI Interface Ports**

Signal Name	Direction	Description
SIGNAL_DETECT[1:0]	IN	Signals from an optical module indicating the optical receivers are illuminated by a signal. If unused, tie this bus to ’11.’

Client-Side Interface

The client-side interface is a 72-bit (64 data bits and 8 control bits) interface running at 156.25 MHz based on the XGMII standard. It is designed to be easily connected to user logic within the FPGA.

Table 2: **Client-Side Interface Ports**

Name	Direction	Description
XGMII_TXD[63:0]	IN	Transmit data, 8 bytes wide
XGMII_TXC[7:0]	IN	Transmit control bits, one bit per transmit data byte.
XGMII_RXD[63:0]	OUT	Received data, 8 bytes wide
XGMII_RXC[7:0]	OUT	Receive control bits, one bit per received data byte.

Figure 4 illustrates transmitting a frame through the client-side interface.

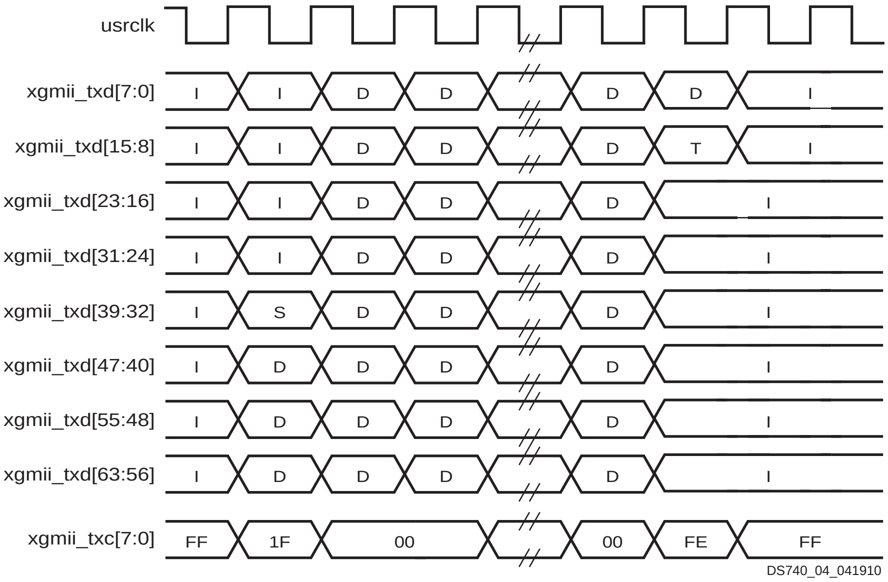

Figure 4: **Transmitting a Frame Through the Client-Side Interface**

Figure 5 illustrates receiving a frame through the client-side interface.

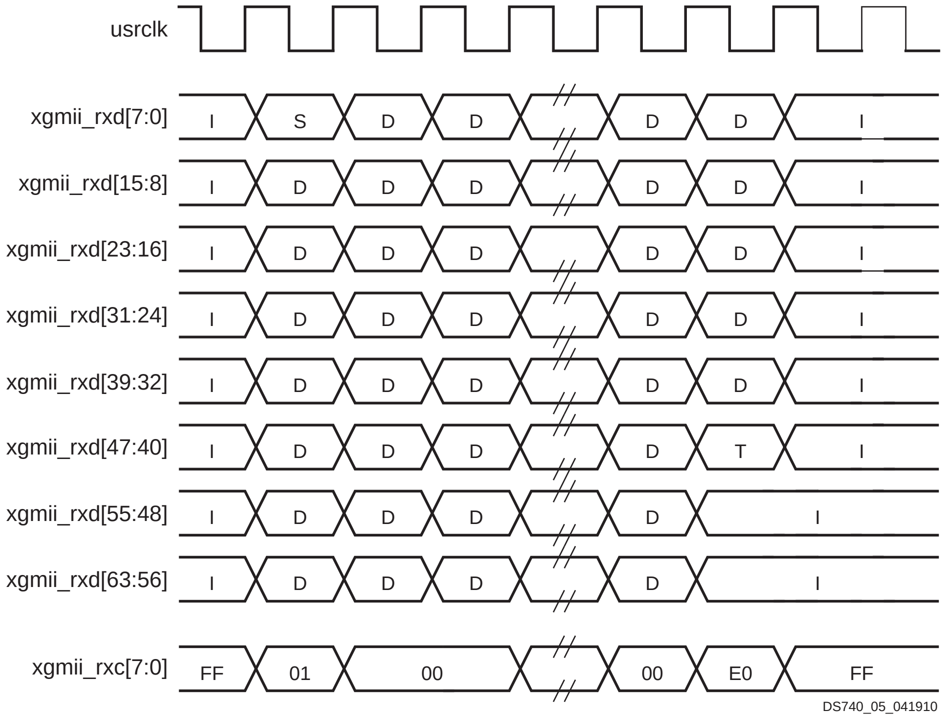

Figure 5: **Receiving a Frame Through the Client-Side Interface**

Management Interface (MDIO)

The MDIO interface is a simple low-speed 2-wire interface for management of the RXAUI core, consisting of a clock signal and a bidirectional data signal. The interface is defined in clause 45 of *IEEE 802.3-2008* standard.

In the RXAUI core, the MDIO interface is an optional block. If implemented, the bidirectional data signal MDIO is implemented as three unidirectional signals. These can be used to drive a 3-state buffer either in the FPGA IOB or in a separate device.

There are additional signals that control the behavior of the core MDIO interface, specifically to set its position in the MDIO memory map.

Table 3: **MDIO Management Interface Ports**

Signal Name	Direction	Description
MDC	IN	Management clock
MDIO_IN	IN	MDIO input
MDIO_OUT	OUT	MDIO output
MDIO_TRI	OUT	MDIO 3-state. ’1’ disconnects the output driver from the MDIO bus.

TYPE_SEL[1:0]	IN	Type select. Determines which MDIO Register addresses the core responds to.
		type_sel = ’00’ or ’01’ – 10GBASE-X PCS
		type_sel = ’10’ – DTE XS
		type_sel = ’11’ – PHY XS
		See the *LogiCORE IP RXAUI User Guide* for the MDIO Register addresses responded to in each case.

PRTAD[4:0]	IN	MDIO port address. When multiple MDIO-managed ports appear on the same bus, this address can be used to address each one individually.

Configuration and Status Signals

In addition to the pollable MDIO interface, the RXAUI core continuously indicates its status on ports as defined in Table 4.

Table 4: **Configuration and Status Vector Ports**

Signal Name	Direction	Description
ALIGN_STATUS	OUT	'1' when the RXAUI receiver is aligned across four logical XAUI lanes.
SYNC_STATUS[3:0]	OUT	Each pin is '1' when the respective XAUI logical lane receiver is synchronized to byte boundaries.
CONFIGURATION_ VECTOR[6:0]	IN	Configuration signals for the core. The bits are: Bit 0 - Loopback Bit 1 - Power down transceiver Bit 2 - Reset local fault status Bit 3 - Reset RX link status Bit 4 - Test Enable - '1' transmits test patterns on RXAUI TX Bits 6:5 - Test pattern select For a more comprehensive description of these signals, consult the *RXAUI User Guide*. This port only exists on the core if the MDIO interface is omitted.
STATUS_ VECTOR[7:0]	OUT	Status indicators for the core. The bits are: Bit 0 - TX local fault Bit 1 - RX local fault Bit 5:2 - Synchronization - identical to SYNC_STATUS[3:0] Bit 6 - Alignment - identical to ALIGN_STATUS Bit 7 - RX link status For a more comprehensive description of these signals, consult the *RXAUI User Guide*. This port only exists on the core if the MDIO interface is omitted.

Clock and Reset

Table 5 describes the clock and reset ports present on the core.

Table 5: **Clock and Reset Ports**

Name	Direction	Description
RESET	IN	Synchronous reset for core. The reference clock must be running for the core to emerge from the reset state.
USRCLK	IN	FPGA logic system clock.
MGT_TX_RESET	IN	Connect this to the same signal used to drive the serial transceiver TXRESET signal.
MGT_RX_RESET	IN	Connect this to the same signal used to drive the serial transceiver RXRESET signal.
RXCLK	IN	Transceiver recovered clock (Present only in Marvell Mode).

MDIO Management Registers

The RXAUI core, when generated with an MDIO interface, implements an MDIO Interface Register block. The core responds to MDIO transactions as either a 10GBASE-X PCS, a DTE XS, or a PHY XS depending on the setting of the `type_sel` port (see Table 3).

10GBASE-X PCS Registers

Table 6 shows the MDIO registers present when the RXAUI core is configured as a 10GBASE-X PCS. For a more comprehensive description of the registers and their effect on core operation, see the *LogiCORE IP RXAUI User Guide*.

Table 6: **10GBASE-X PCS/PMA MDIO Registers**

Register Address	Register Name
1.0	PMA/PMD Control 1
1.1	PMA/PMD Status 1
1.2,1.3	PMA/PMD Device Identifier
1.4	PMA/PMD Speed Ability
1.5, 1.6	PMA/PMD Devices in Package
1.7	10G PMA/PMD Control 2
1.8	10G PMA/PMD Status 2
1.9	Reserved
1.10	10G PMD Receive Signal OK
1.11 TO 1.13	Reserved
1.14, 1.15	PMA/PMD Package Identifier
1.16 to 1.65 535	Reserved
3.0	PCS Control 1
3.1	PCS Status 1
3.2, 3.3	PCS Device Identifier
3.4	PCS Speed Ability
3.5, 3.6	PCS Devices in Package
3.7	10G PCS Control 2
3.8	10G PCS Status 2
3.9 to 3.13	Reserved
3.14, 3.15	PCS Package Identifier
3.16 to 3.23	Reserved
3.24	10GBASE-X PCS Status
3.25	10GBASE-X Test Control
3.26 to 3.65 535	Reserved

DTE XS Registers

Table 7 shows the MDIO registers present when the RXAUI core is configured as a DTE XS. For a more comprehensive description of the registers and their effect on core operation, see the *LogiCORE IP RXAUI User Guide*.

Table 7: **DTE XS MDIO Registers**

Register Address	Register Name
5.0	DTE XS Control 1
5.1	DTE XS Status 1
5.2, 5.3	DTE XS Device Identifier
5.4	DTE XS Speed Ability
5.5, 5.6	DTE XS Devices in Package
5.7	Reserved
5.8	DTE XS Status 2
5.9 to 5.13	Reserved
5.14, 5.15	DTE XS Package Identifier
5.16 to 5.23	Reserved
5.24	10G DTE XGXS Lane Status
5.25	10G DTE XGXS Test Control

PHY XS Registers

Table 8 shows the MDIO registers present when the RXAUI core is configured as a PHY XS. For a more comprehensive description of the registers and their effect on core operation, see the *LogiCORE IP RXAUI User Guide*.

Table 8: **PHY XS MDIO Registers**

Register Address	Register Name
4.0	PHY XS Control 1
4.1	PHY XS Status 1
4.2, 4.3	PHY XS Device Identifier
4.4	PHY XS Speed Ability
4.5, 4.6	PHY XS Devices in Package
4.7	Reserved
4.8	PHY XS Status 2
4.9 to 4.13	Reserved
4.14, 4.15	PHY XS Package Identifier
4.16 to 4.23	Reserved
4.24	10G PHY XGXS Lane Status
4.25	10G PHY XGXS Test Control

Verification

The RXAUI core has been verified using both simulation and hardware testing.

Simulation

A highly parameterizable transaction-based simulation test suite was used to verify the core. Verification tests include:

- Register access over MDIO
- Loss and regain of synchronization
- Loss and regain of alignment
- Frame transmission
- Frame reception
- Clock compensation
- Recovery from error conditions

Hardware Verification

The core has been tested on the ML623 Virtex-6 LXT test family. The design comprises the Xilinx 10 Gb/s MAC, RXAUI, a *ping* loopback FIFO, and a test pattern generator all under embedded processor control. This design was used for conformance and inter operability testing with Dune Networks devices.

Device Utilization

Table 9 provides approximate utilization for the various core options on Virtex-6 LXT FPGAs.

Table 9: **Device Utilization – Virtex-6 LXT FPGAs**

MDIO Management	RXAUI Mode	Slices	LUTs	FFs	BUFRs
FALSE	Dune	360	672	832	0
TRUE	Dune	432	818	927	0
FALSE	Dune-Alternate	393	738	832	0
TRUE	Dune-Alternate	457	876	927	0
FALSE	Marvell	669	1306	1435	1
TRUE	Marvell	739	1453	1530	1

Table 10 provides approximate utilization for the various core options on Virtex-7 and Kintex-7 FPGAs.

Table 10: **Device Utilization – Virtex-7 and Kintex-7 FPGAs**

MDIO Management	RXAUI Mode	LUTs	FFs
FALSE	Dune	631	780
TRUE	Dune	775	875
FALSE	Dune_Alternate	680	780
TRUE	Dune_Alternate	830	875
FALSE	Marvell	1364	1294
TRUE	Marvell	1506	1389

Marvell RXAUI requires one additional BUFR for Virtex-7 devices or one additional BUFG for Kintex-7 devices.

References

1. *IEEE Std. 802.3-2008*, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.
2. *Dune Networks DN-DS-RXAUI-Spec v1.0*, RXAUI - Reduced Pin XAUI.
3. Marvell MV-S 105386-00 RXAUI Interface and RXAUI Adapter Specifications.
4. UG693, *LogiCORE IP RXAUI User Guide* on the RXAUI product page.
5. RXAUI core release notes.

Support

Visit www.xilinx.com/support for technical support. Xilinx provides technical support for this LogiCORE IP product when used as described in product documentation.

Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not listed in the documentation or if customized beyond that allowed in the product documentation, or if any changes are made in sections of the design marked *DO NOT MODIFY.*

Ordering Information

This version of the RXAUI IP core does not require a license key. Previous versions of the RXAUI IP core released in ISE® v12.1 software and earlier did require a license key; see the documentation for the version of the core you are using for information.

List of Acronyms

Acronym	Definition
CML	Current Mode Logic
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DDR	Double Data Rate
DTE	Data Terminal Equipment
FCS	Frame Check Sequence
FF	flip-flop
FIFO	First In First Out
FPGA	Field Programmable Gate Array.
GBIC	Gigabit Interface Converter
Gb/s	Gigabits per second
GFC	Gigabit Fibre Channel
GMII	Gigabit Media Independent Interface
IO	Input/Output
IP	Intellectual Property
ISE	Integrated Software Environment
IUS	Incisive Unified Simulator
LUT	Lookup Table
MAC	Media Access Controller
Mb/s	Megabits per second
MDIO	Management Data Input/Output
MHz	Mega Hertz
NGC	Native Generic Circuit
PCS	Physical Coding Sublayer
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
RXAUI	Reduced Pin eXtended Attachment Unit Interface
UCF	User Constraints File
VHDL	VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits)
XAUI	eXtended Attachment Unit Interface
XGMII	10-Gigabit Ethernet Media Independent Interface
XGXS	XGMII Extender Sublayer
XS	Extender Sublayer

Revision History

Date	Version	Revision
9/16/09	1.1	Initial Xilinx release
4/19/10	1.2	Updated to v1.2 for 12.1 release; added Marvell RXAUI Specification support.
3/01/10	1.3	Updated to v1.3 for 13.1 release.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.