

Introduction

A main system bottleneck in high-speed communications equipment is data transmission from chip-to-chip and over backplanes. Stratix™ GX devices help remedy the problem by supporting 3.125-gigabit per second (Gbps) channels and integrating advanced functionality into the device's logic array. Versatile Stratix GX transceiver blocks support many emerging industry protocols that require high-speed differential I/O with clock data recovery (CDR) (e.g., 10 Gigabit Ethernet via the 10-Gbps attachment unit interface [XAUI]), SONET scrambled backplane, and custom implementations.

The Stratix GX transceiver block is designed to support XAUI. In addition, the other 10 Gigabit Ethernet interface protocols (i.e., 10-gigabit, 16-bit interface [XSBI] and 10-gigabit medium independent interface [XGMII]) are also supported by Stratix GX devices.

This application note discusses the following topics:

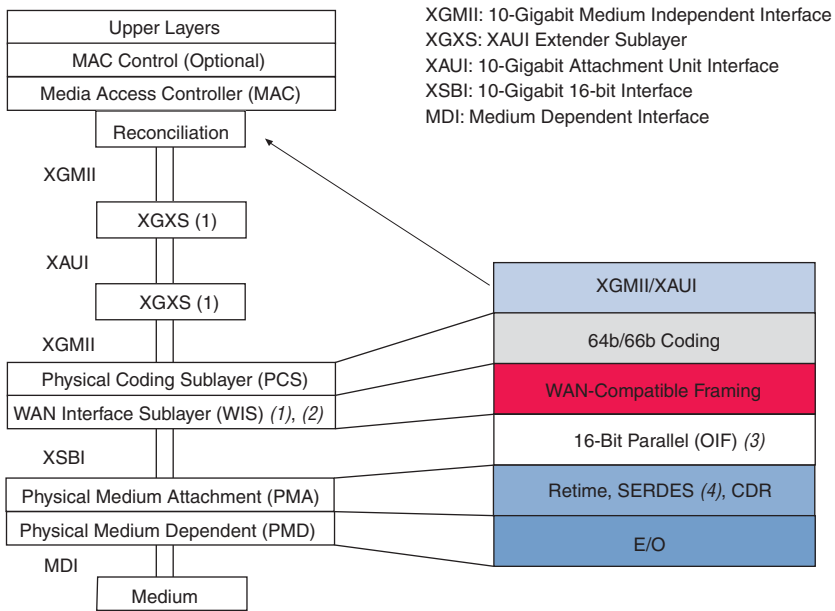
- Fundamentals of 10 Gigabit Ethernet & XAUI
- XAUI electrical specifications
- Implementing XAUI in Stratix GX devices
- Using Quartus II to implement XAUI
- Applications of Stratix GX and XAUI

Fundamentals of 10 Gigabit Ethernet & XAUI

Over the years, Ethernet speed has leapfrogged from the initial 10 megabits per second (Mbps), to 100 Mbps, and more recently to 1 Gbps. Today, the Ethernet is the dominant network technology in local area networks (LANs), and with the advent of the 10 Gigabit Ethernet, it is now competitive in the metropolitan area network (MAN) and wide area network (WAN) markets.

The purpose of the 10 Gigabit Ethernet standard is to extend IEEE 802.3 (Ethernet) operating speed to 10 Gbps and include WAN applications. These additions provide a significant increase in bandwidth while maintaining maximum compatibility with current IEEE 802.3 interfaces. [Figure 1](#) shows the 10 Gigabit Ethernet layer diagram.

Figure 1. 10 Gigabit Ethernet Layer Diagram



Notes to Figure 1:

- (1) Optional sublayer.
- (2) Adding the WIS makes the WAN a PHY layer.
- (3) OIF: Optical Interworking Forum.
- (4) SERDES: serializer/deserializer.

The Ethernet PHY layer—(i.e., layer 1 of the open systems interconnection [OSI] model)—connects the media (e.g., optical or copper) to the MAC layer (i.e., layer 2). The Ethernet architecture further divides the PHY layer into a PMD sublayer, a PMA sublayer, and a PCS. For example, optical transceivers are PMD sublayers. The PMA converts the data between the PMD sublayer and the PCS. The PCS is made up of coding (e.g., 8b/10b, 64b/66b) and serializer, or multiplexing functions.

The 10 Gigabit Ethernet has three different implementations for the PHY layer:

- 10GBASE-X
- 10GBASE-R
- 10GBASE-W

The 10GBASE-X implementation is a PHY layer that supports XAUI. XAUI, used in conjunction with the XGMII extender sublayer (XGXS), allows more separation in distance between the MAC and PHY layers. The 10GBASE-X PCS uses four lanes of 8b/10b coded data at a rate of 3.125 Gbps. The 10GBASE-X is a wide wave division multiplexing (WWDM) LAN PHY layer.

The 10GBASE-R and 10GBASE-W are serial LAN PHY layers and serial WAN PHY layers, respectively. Unlike the 10GBASE-X implementation, the 10GBASE-R and 10GBASE-W implementations have an XSBI interface.

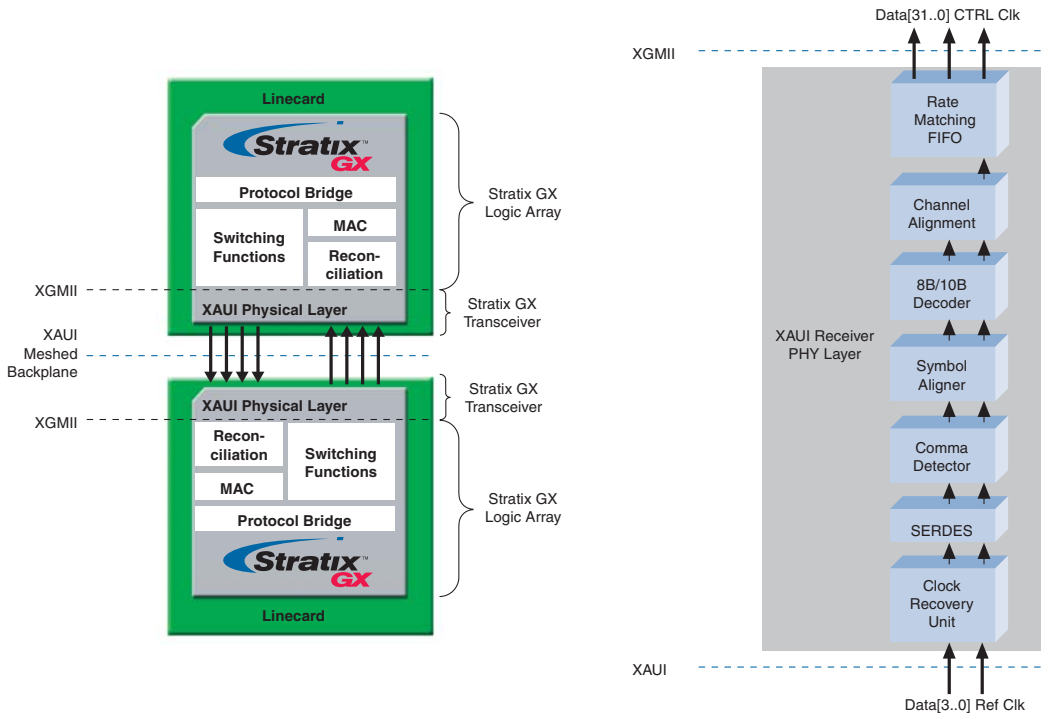
XAUI Fundamentals

XAUI is designed as an interface extender for the 10 Gigabit XGMII. You can use XAUI in various applications including 10 Gigabit Ethernet line cards, LAN-to-WAN bridges (i.e., Ethernet to SONET converter), and as a backplane and chip-to-chip interconnect.

Using 1.5-V PCML electrical levels, XAUI uses four full-duplex serial links at 3.125 Gbps in each direction. In aggregate, you can transfer a total of 12.5 Gbps in each direction using the XAUI standard. The 8b/10b encoding/decoding overhead supports a 10-Gbps throughput. At those rates, the protocol can also accommodate SONET OC-192 traffic.

Figure 2 shows that Stratix GX devices are designed for easy XAUI implementation. You can implement higher-level layers in the programmable logic section of Stratix GX devices, including reconciliation, MAC, switching functions, and a protocol bridge. In addition, the Stratix GX gigabit transceiver block contains the required components for implementing the XAUI physical layer (see **Figure 2**). The XAUI physical layer, also known as the XGXS, contains the XAUI sublayers PCS, PMA, and PMD.

Figure 2. Implementing XAUI in Stratix GX Devices



For more information on 10 Gigabit Ethernet and the XAUI, visit the 10 Gigabit Ethernet Alliance web site at www.10gea.org.

XAUI Electrical Specification

The XAUI electrical characteristics, a XGMII (10-gigabit media independent interface) extender, are described in this section according to the IEEE P802.3ae standard, clause 47 of the IEEE Draft P802.3ae/D4.1 document. Specifically, this section describes XAUI:

- Signal levels
- Signal path
- Driver characteristics
- Load impedance
- Amplitude & swing
- Driver template & jitter
- Receiver characteristics
- Skew/jitter budget
- Characteristic impedance
- Stratix GX support of the XAUI standard

Signal Levels

To allow maximum interoperability between different supply-voltage components, XAUI uses a low swing, AC-coupled differential interface. The low swing differential interface signals minimize noise and electromagnetic interference (EMI).

Signal Path

XAUI signal paths are point-to-point connections. XAUI has four lanes, and each lane corresponds to one pair of transmit and one pair of receive differential traces. The signal paths are intended to operate across 50 cm with controlled impedance traces in FR-4 material. Stratix GX devices support one meter (40") of FR-4 trace and two connectors.

Driver Characteristics

The XAUI data rate is specified to be 3.125 Gbps \pm 100 PPM. The nominal unit interval is 320 ps. [Table 1](#) summarizes the XAUI driver characteristics.

<i>Table 1. XAUI Driver Characteristics</i>		
Parameter	Value	Unit
Data rate tolerance	3.125 Gbps \pm 100 PPM	Gbps PPM
Unit interval nominal	320	ps
Differential amplitude maximum	1,600	mV _{p-p}
Absolute output voltage limit (maximum)	2.3	V
Absolute output voltage limit (minimum)	-0.4	V
Output jitter near-end maximums (total jitter) (1)	\pm 0.175 from the mean	UI _{p-p}
Output jitter near-end maximums (deterministic jitter) (1)	\pm 0.085 from the mean	UI _{p-p}
Output jitter far-end maximums (total jitter) (2)	\pm 0.275 from the mean	UI _{p-p}
Output jitter far-end maximums (deterministic jitter) (2)	\pm 0.185 from the mean	UI _{p-p}

Notes to [Table 1](#):

- (1) Near-end is the transmitter side.
- (2) Far-end is the receiver side.

Load Impedance

The load is $100\ \Omega \pm 5\%$ differential. Stratix GX devices support programmable on-chip differential termination of $100\ \Omega$, $120\ \Omega$, and $150\ \Omega$.

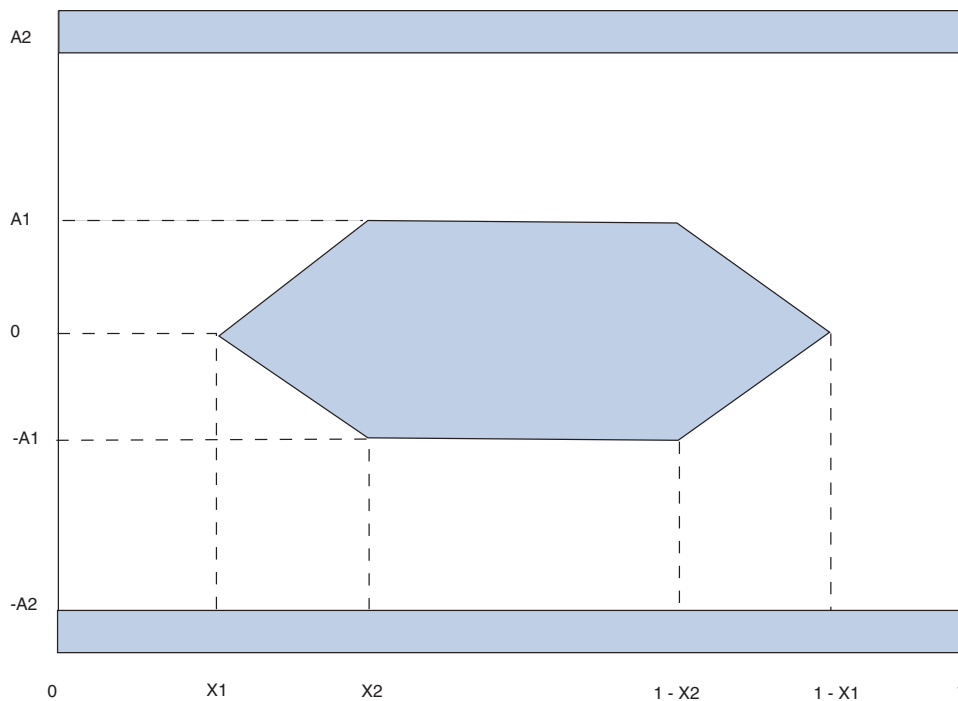
Amplitude & Swing

The maximum differential output amplitude is specified to be $1,600\ \text{mV}_{\text{p-p}}$ including any transmit equalization. The receiver is AC-coupled. The absolute output voltage of the driver should be between $-0.4\ \text{V}$ and $2.3\ \text{V}$ with respect to ground.

Driver Template & Jitter

Table 2 shows the driver template intervals. The driver should comply with either near-end, or far-end, eye template and jitter requirements. Jitter for a data signal is usually measured in terms of unit intervals (UIs). Figure 3 shows the driver template.

Figure 3. Driver Template



The maximum total jitter $\pm 0.175 U_{I_{p-p}}$ from the mean is required at the near-end. The maximum deterministic component is $\pm 0.085 U_{I_{p-p}}$. The maximum total jitter is $0.275 U_{I_{p-p}}$ from the mean at the far-end and maximum deterministic component is $0.185 U_{I_{p-p}}$ from the mean.

Table 2. Driver Template Intervals

Symbol	Near-End Value	Far-End Value	Unit
X1	0.175	0.275	UI
X2	0.390	0.400	UI
A1	400	100	mV
A2	800	800	mV

Receiver Characteristics

The peak-to-peak total jitter amplitude tolerance for the XAUI receiver should be at least $0.65 U_{I_{p-p}}$. The total jitter consists of three components: deterministic, random, and an additional sinusoidal jitter. [Table 3](#) shows the receiver characteristics.

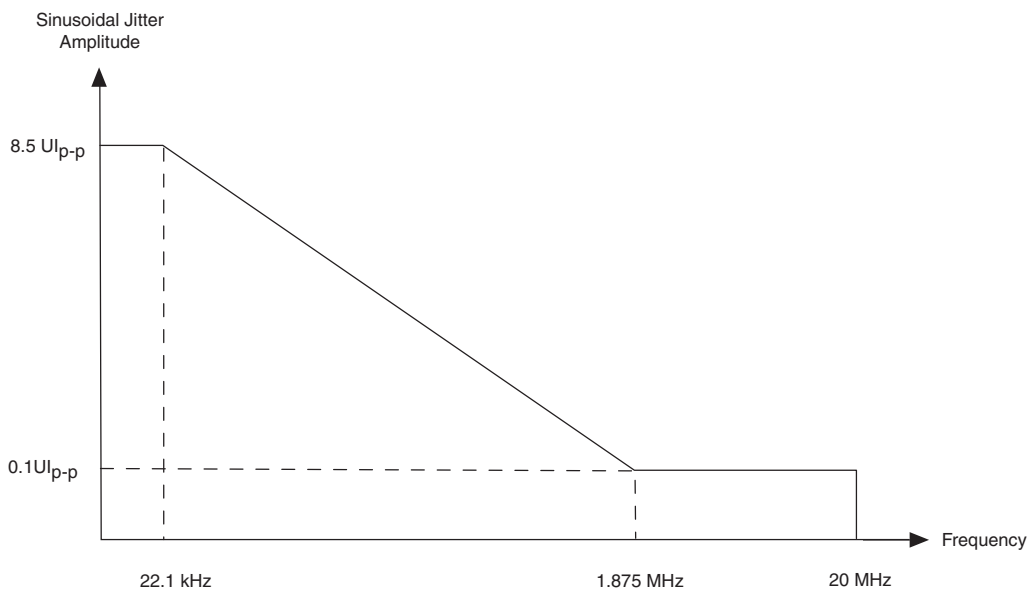
Table 3. Receiver Characteristics

Parameter	Value	Units
Data rate	3.125	Gbps
Tolerance	± 100	PPM
UI nominal	320	ps
Receiver coupling	AC	
Jitter amplitude tolerance	0.65	$U_{I_{p-p}}$

Skew/Jitter Budget

The peak-to-peak total jitter tolerance for the XAUI receiver must be at least $0.65 UI$. There are three components in the total jitter tolerance: deterministic jitter, random jitter, and sinusoidal jitter. As described in the IEEE P802.3ae standard, the deterministic jitter tolerance must be at least $0.37 U_{I_{p-p}}$. The sum of the deterministic and random jitter tolerance is $0.55 U_{I_{p-p}}$. [Figure 4](#) shows a single-tone sinusoidal jitter tolerance with any frequency and amplitude.

Figure 4. Single-Tone Sinusoidal Jitter Mask



Characteristic Impedance

The recommended differential characteristic impedance of circuit board trace pairs is $100 \Omega \pm 10\%$.

Stratix GX Support of the XAUI Standard

Stratix GX devices are ideal for the XAUI standard. [Table 4](#) summarizes Stratix GX device support for XAUI.

Table 4. Stratix GX Support for XAUI

Parameter	XAUI	Stratix GX Support	Unit
Data rate	3.125	0.622 to 3.125	Gbps
Tolerance	±100	±100	PPM
Driver differential amplitude maximum	1,600	1600	mV _{p-p}
Unit interval nominal	320	Yes	ps
Absolute output voltage limit (maximum)	2.3	Yes	V
Absolute output voltage limit (minimum)	-0.4	Yes	V
Drive capabilities	50 cm (~20")	100 cm (~40")	cm
Receiver coupling	AC-Coupled	Yes	
Receiver jitter tolerance	At least 0.65	Yes	UI
Receiver deterministic jitter tolerance	At least 0.37	Yes	UI
Receiver bit error ratio	Less than 10 ⁻¹²	10 ⁻¹²	
Receiver input amplitude tolerance	May be larger than 1,600	1,600	mV _{p-p}

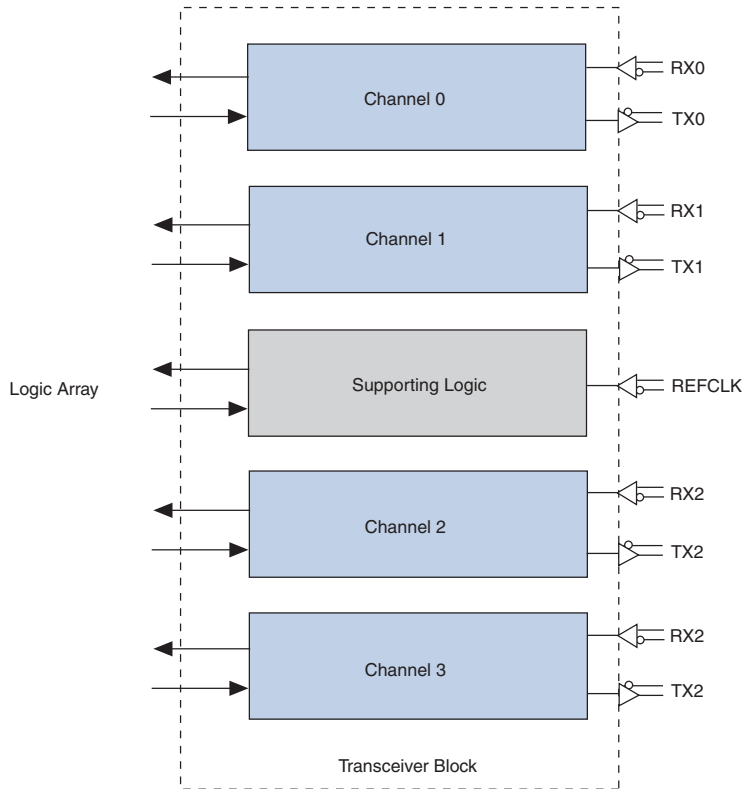
Implementing XAUI in Stratix GX Devices

[Figure 5](#) shows the Stratix GX gigabit transceiver block, made up of four channels (the term channel and lanes are interchangeable in the XAUI standard) and the supporting logic. Each channel contains a receiver and transmitter. The supporting logic contains a phase-locked loop (PLL), which generates clocks for the four channels' CDR reference.

The XAUI channel alignment state machine is included in the transceiver block's supporting logic and interacts with the XAUI channel alignment logic in the receiver.

- The XAUI mode receiver operates with J = 10, a full automatic synchronizer and word aligner, a channel aligner and first-in first-out (FIFO) buffer, and a rate-matching FIFO followed by the 8b/10b decoder.
- The XAUI mode transmitter operates with J = 10, a transmit FIFO followed by the 8b/10b encoder. All of the components of the transceiver are used to implement XAUI.

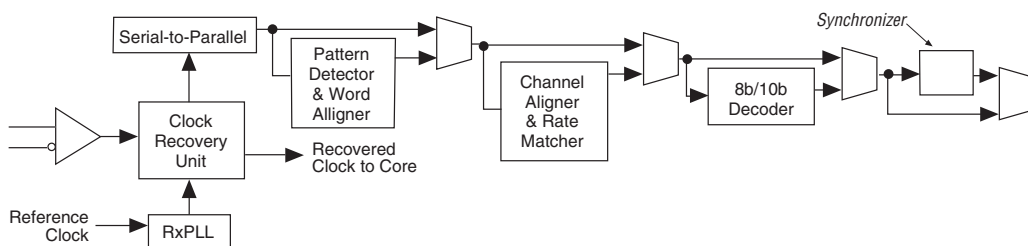
Figure 5. Architectural Overview of the Stratix GX Transceiver Block



Stratix GX Device at the Receiver Side of XAUI

Figure 6 shows a Stratix GX device at the XAUI receiver side, where 3.125 Gbps of serial data is deserialized into 10-bit parallel data. The pattern detector detects comma patterns (i.e., either K28.1, K28.7, or K28.5) from the parallel data, and the word aligner aligns the data based on the incoming K set. Because the data is going through four separate channels, it is likely that the channels are skewed with respect to each other. To solve this potential problem, the decoded data is processed by the rate matching block and then by the channel alignment block. The aligned data is then fed into the 8b/10b decoder.

Figure 6. Stratix GX Device at the XAUI Receiver Side



The following section describes the duties of each module (shown in [Figure 6](#)) as the data is processed and transmitted to one of the Stratix GX device's receiver channels.

Clock Recovery Unit (CRU)

The CRU block is the unit that takes the serial data input at the receiver and generates a clock based on the data transitions. The CRU block is mostly an analog block and should be simulated simply as a clock generator that locks to the data-in rate. You cannot bypass the CRU block.

Deserializer

The deserializer is a serial-to-parallel data converter. It accepts serial data input clocked at a certain rate and generates parallel data (i.e., J bits wide, where J is programmable) at $1/J$ the serial rate. J is called the deserialization factor. XAUI uses the deserialization factor of 10 ($J = 10$) parallel. You cannot bypass the deserializer block.

Pattern Detector

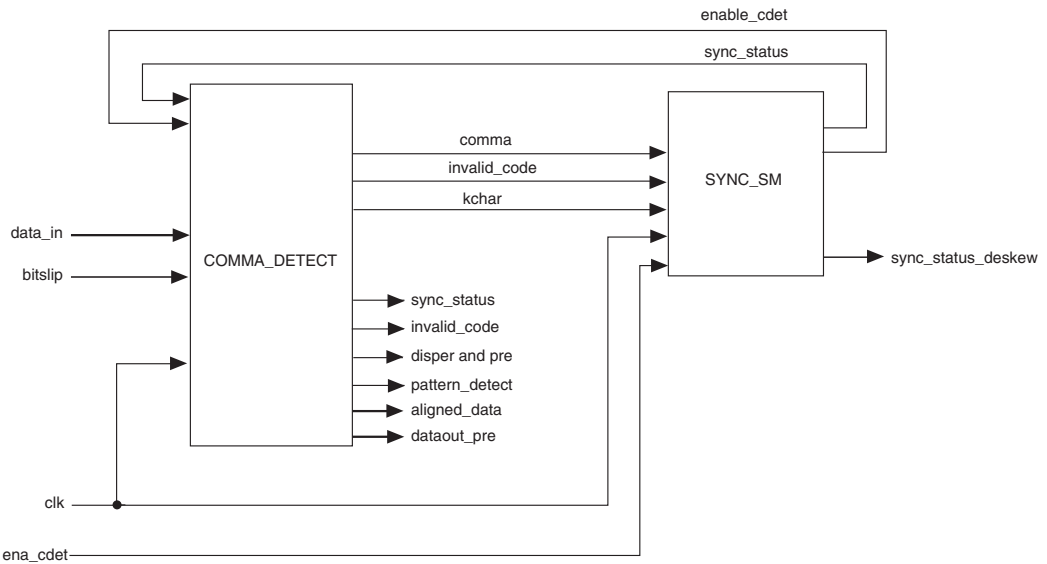
Data enters the pattern detector after leaving the deserializer. The pattern detector searches for a comma pattern across the entire incoming data that is word-aligned. The pattern detector checks for a pattern in every bit position, and is used to identify predetermined bit patterns in the transceiver. To ensure pattern recognition, a $1/K28.5$ pattern (i.e., 0011111010) is pre-programmed (via the Quartus® II software) into the pattern detector.

Word Aligner

The main function of the word aligner is to synchronize the word clock to the data stream word boundary by detecting and aligning a specific programmable pattern known as the COMMA character. In XAUI mode, a built-in synchronization state machine monitors word alignment.

The purpose of the synchronization state machine is to determine if the receiver is ready for the operation. The state machine informs the upper layer that the link is ready. If there is no signal, or a disparity error, or a word alignment error, the state machine tells the upper layer that the link is down. [Figure 7](#) shows the block diagram of the synchronization state machine.

Figure 7. Block Diagram of the Synchronization State Machine



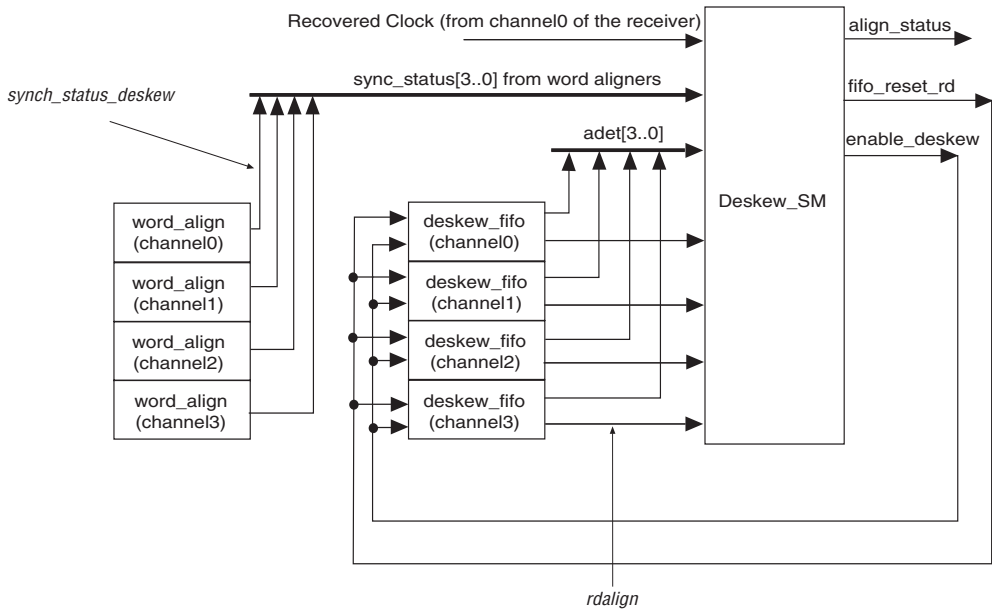
The synchronization state machine supports the XAUI protocol. In XAUI mode, the `sync_status` signal goes high upon detection of a /K/ pattern four times. You can follow each comma by any number of valid code-groups.

When receiving, each channel decodes its received 10b receiver code-group to its 8b code-group and control signals. When the decoder detects an invalid code, it generates an internal invalid code signal that can be used by the internal state machines.

Channel Aligner

The purpose of the channel aligner is to align the data for all channels (i.e., synchronize the channels) to the same recovered clock. The channel aligner is only active when the transceiver block is in XAUI mode, and it consists of a channel alignment symbol detector and a channel aligner FIFO buffer. The system relies on the common channel aligner state machine in the transceiver's common block. Each receiver has its own CDR block and the data arrives at each receiver pin at slightly different times. The channel aligner synchronizes the data from the FIFO buffer to the clock edge of channel 0's recovered clock; thus, assuring that the data from the four receiver channels is aligned correctly before the data is sent to the next layer. See [Figure 8](#).

Figure 8. Channel Aligner State Machine



The `adet` signal is sent to the channel aligner state machine when the industry-standard `/A/` pattern is detected at the channel aligner FIFO buffer (i.e., the current receiver has found the `/A/` pattern). The `/A/` pattern is the K28.3 10-bit string. When the `/A/` pattern is found in all channels, the XAUI channel aligner state machine enables the reading of data from the channel aligner FIFO buffer and the `align_status` signal is set. The XAUI channel aligner state machine now monitors the reception of the `/A/` pattern's columns that are not aligned, and responds as specified by the relevant 10 Gigabit Ethernet XAUI standard clause.

Rate Matcher

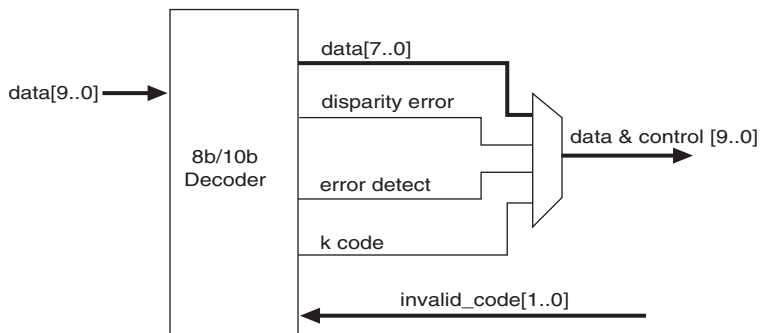
The receiver performs rate matching following the channel alignment. Stratix GX device transceiver blocks use rate matchers to adjust for ± 100 PPM clock fluctuation between the recovered clock and the clock within the logic array. The rate matching FIFO buffer is a XAUI-specific protocol for reading and writing from the FIFO buffer.

In XAUI mode, the rate matcher uses the FIFO counter register to keep track of write and read transactions to the FIFO buffer. The rate matching FIFO block detects skip or the `/R/` pattern (K28.0) code group from all the channels when the FIFO counter is above nine, and inserts the `/R/` pattern to all the channels when the FIFO counter is below four, which prevents FIFO overflow and underflow, respectively.

8b/10b Decoder (Receiver Side)

The 8b/10b decoder decodes a 10-bit parallel input stream into 8-bit data, and detects running disparity errors and invalid code groups. Because the decoder expects a 10-bit code, `J` must equal 10. Figure 9 shows the block diagram of the 8b/10b decoder.

Figure 9. 8b/10b Decoder Block Diagram



In XAUI mode, all four channels operate together and the XAUI state machine controls the decoding process where various idle code groups are mapped to a XGMII-specific 8b idle code, and error code groups are processed based on their location.

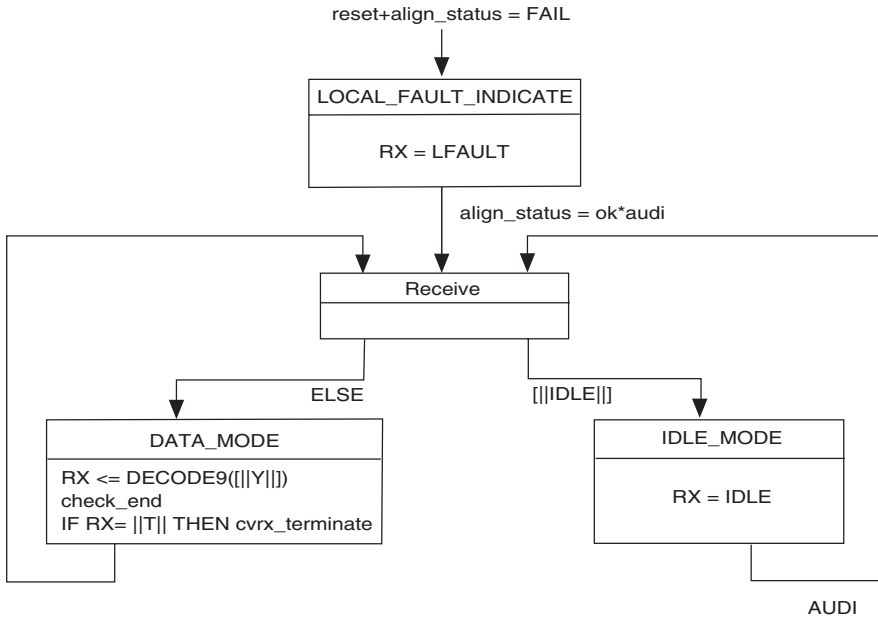
Receiver State Machine

The receiver state machine checks the word alignment and disparity status. There is one receiver state machine per Stratix GX transceiver block and it controls mapping codes from the PCS of the Ethernet PHY layer to the XGMII. The XAUI receiver state machine translates 8b/10b characters to XGMII characters. At reset, the XAUI receiver state machine transmits a link fault (LFAULT) condition (see [Figure 10](#)). There are two modes after alignment:

- Data mode: code groups are mapped to XGMII data or control characters
- Idle mode: Maps |I| (idle code group) to XGMII idle characters

The XAUI receiver state machine translates /A/, /K/, and /R/ characters to XGMII, idles and detects disparity error propagation to //T// (terminate code group) and the column next to //T//, and substitutes /E/ (error) characters.

Figure 10. Receiver State Machine



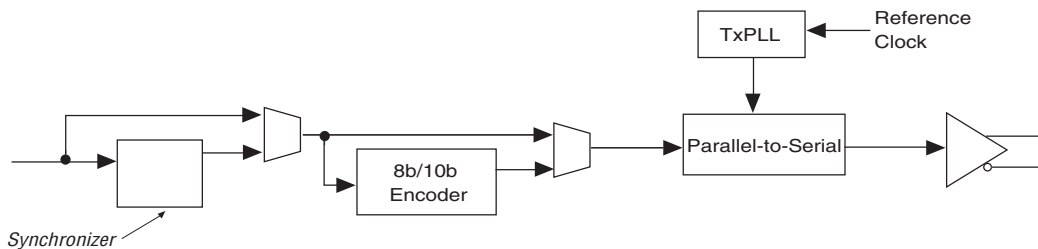
De-Multiplexer

At this point, the data is sent to the core of the Stratix GX device. The data can be transmitted either “as is” synchronized to the clock that is sent to the Stratix GX device’s logic array, or it can be de-serialized and transmitted at half the rate—but double the amount of bits in parallel.

Stratix GX Device at the XAUI Transmitter Side

Figure 11 shows a Stratix GX device at the XAUI transmitter side, where 32-bit data is split into 8-bit groups and each 8-bit group is transmitted on one Stratix GX transmitter channel. Next, before the data is sent out, the 8b/10b encoder encodes each 8-bit group into a 10-bit group. The transceiver feeds the encoded 10-bit parallel data into the serializer, and then the data is sent out serially at 3.125 Gbps.

Figure 11. Stratix GX Device at the XAUI Transmitter Side



The following section describes the duties of each module (shown in [Figure 11](#)) as the data is processed and transmitted to one of the Stratix GX device's transmitter channels.

Byte Serializer/Synchronizer

The byte serializer is only used in “double-width” mode. This process takes a 20-bit input and decodes it into two, 10-bit inputs. The serializer is clocked either by the PLL reference clock if the FIFO buffer is enabled, or by the Stratix GX device's core clock if the FIFO buffer is bypassed.

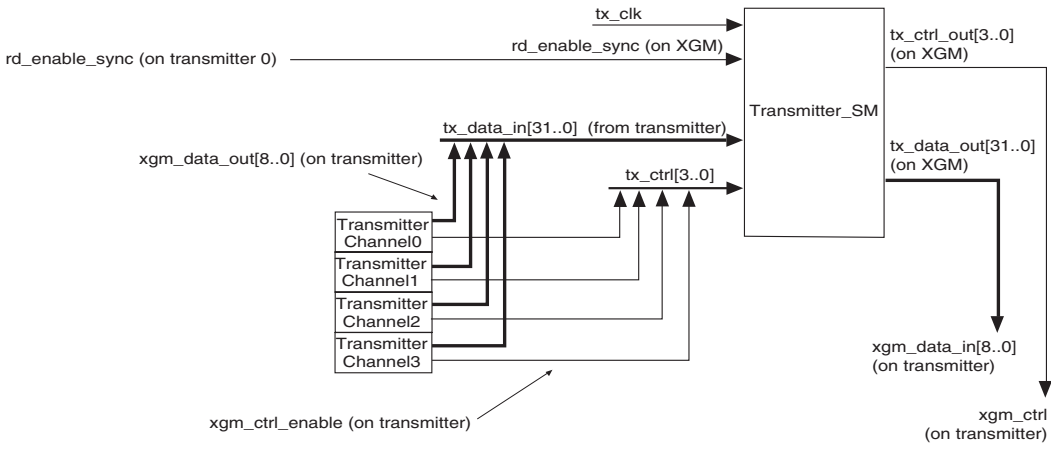
8b/10b Encoder (Transmitter Side)

The 8b/10b encoder system encodes an 8-bit data stream into a 10-bit code. When the 8b/10b system operates in XAUI mode, the data that is sent to the 8b/10b encoder is not the data that is sent from the Stratix GX device's core. It is first routed to a common transmit state machine, which controls the 8b/10b system.

Transmitter State Machine

The transmitter state machine (see [Figure 12](#)) translates XGMII characters into 8b/10b characters. There is one transmitter state machine per gigabit transceiver block, and the state machine replaces all idle symbols with `||A||`, `||K||`, and `||R||`. The state machine also determines whether the data should be passed to the PMA, or whether idle/sequence control characters (including randomizing `||A||`, `||K||`, and `||R||`) should be sent out. The 32-bit `txdatain` data bus is organized as a series of 8-bit data coming out of the transmitter that would have been sent to the 8b/10b encoder of each channel. There are four channels with 8 bits of data each. One 8-bit data per transmitter channel is sent.

Figure 12. Transmit State Machine



Transmitter SERDES

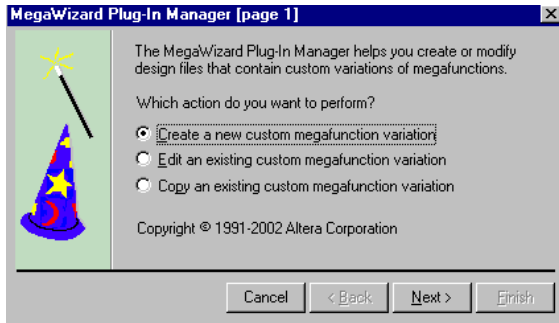
The transmitter SERDES is the final block in the transmitter, and it operates in 10-bit ($J = 10$) words. The transmitter SERDES block receives parallel data, serializes it, and then sends the serialized data to the I/O buffer.

Using the Quartus II Software to Implement XAUI

The Quartus II software allows you to easily and quickly instantiate megafunctions using the MegaWizard® Plug-In Manager. Use the following steps to implement a megafunction.

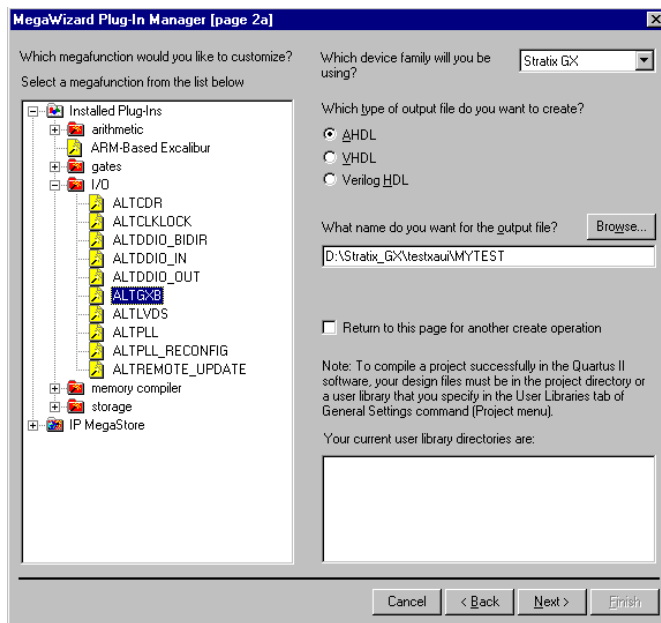
1. To launch the MegaWizard Plug-In Manager, choose **MegaWizard Plug-In Manager** (Tools menu) in the Quartus II software.
2. Select **Create A New Custom Megafunction Variation** and click **Next**. See [Figure 13](#).

Figure 13. MegaWizard Plug-In Manager



-
3. Select the Stratix GX device.
 4. Open the I/O megafunctions folder.
 5. Choose a ALTGXB megafunction. See [Figure 14](#).

Figure 14. Select a ALTGXB I/O Megafunction



6. Select an output file type and enter the desired name of the megafunction. You can choose AHDL (.tdf), VHDL (.vhd), or Verilog VHDL (.v) as the output file type. Along with these HDL files, the MegaWizard Plug-In Manager creates an include file (.inc), a VHDL component declaration file (.cmp), and a block symbol file (.bsf).

The following section describes the available configuration options for the ALTGXB megafunction.

The ALTGXB wizard provides customizable parameters for device family, protocol type, operation mode type, number of channels, and the width of the channel. Figures 15 through 19 show the ALTGXB wizard.

Figure 15. Page One of the ALTGXB MegaWizard Plug-In Manager

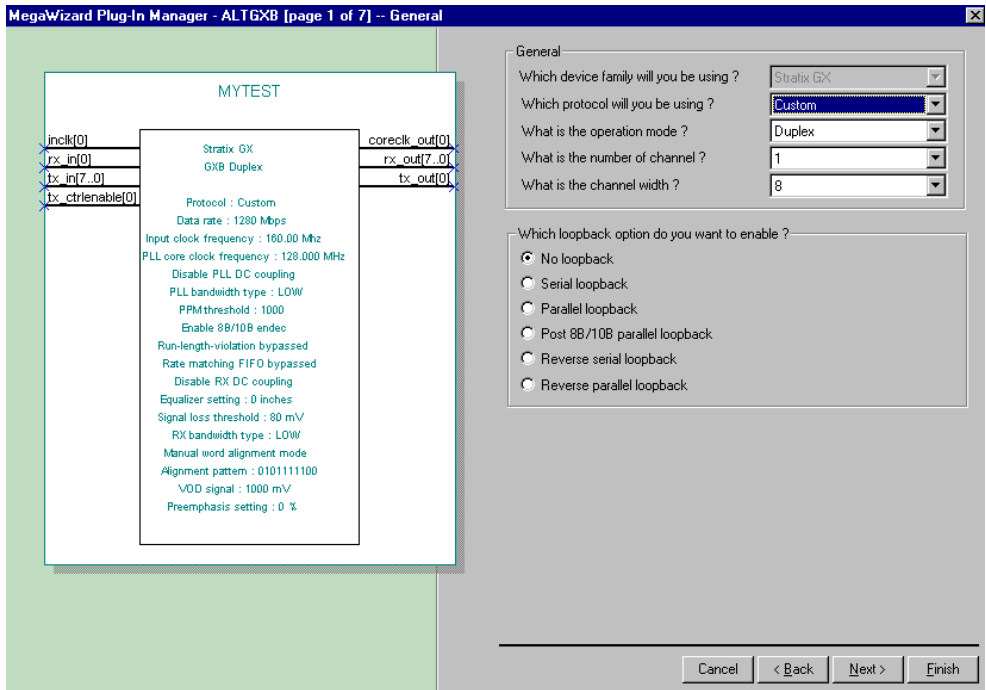


Figure 16 shows page two of the ALTGXB wizard, where you can select optional inputs, miscellaneous settings, and self-test mode.

Figure 16. Page Two of the ALTGXB MegaWizard Plug-In Manager

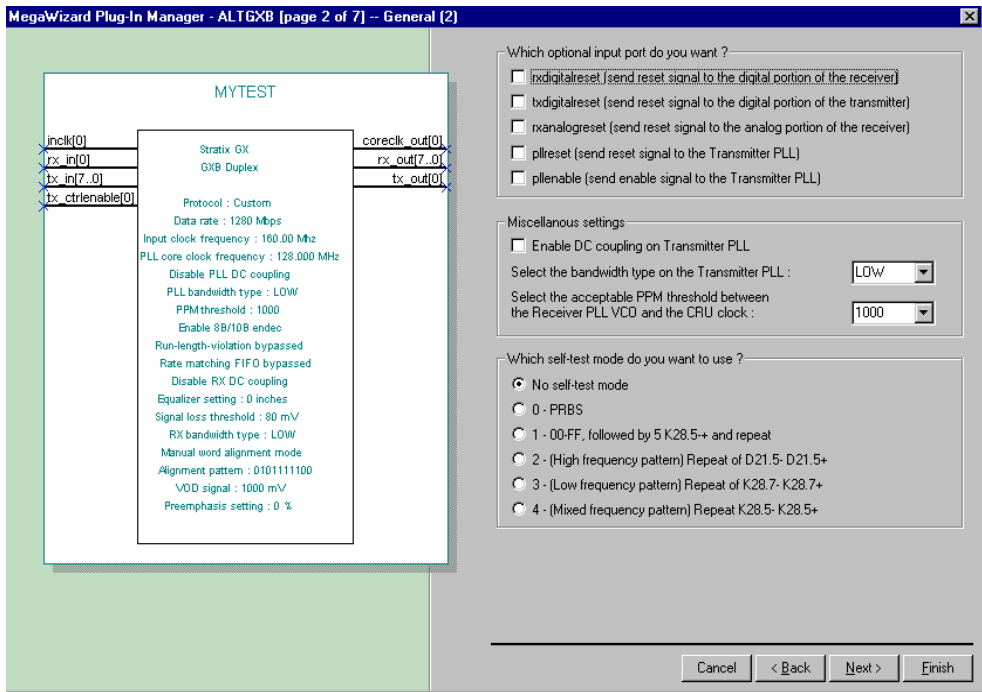


Figure 17 shows page three of the ALTGXB wizard, where receiver functionality is presented. Also, you can select equalizer control settings and DC coupling.

Figure 17. Page Three of the ALTGXB MegaWizard Plug-In Manager

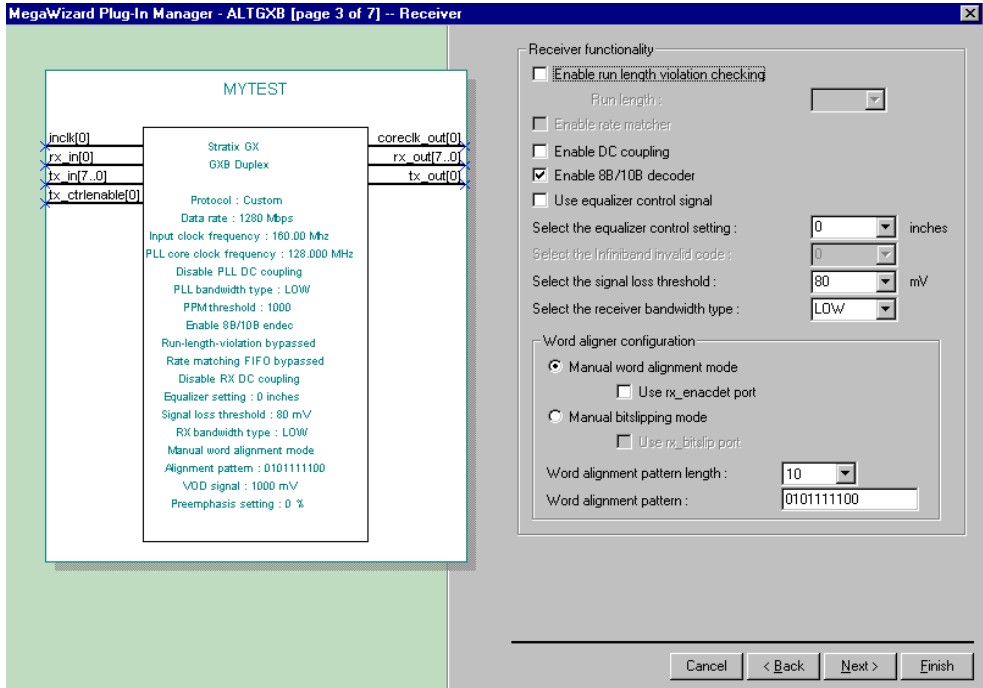


Figure 18 shows page four of the ALTGXB wizard, where data rate and clock frequency are presented.

Figure 18. Page Four of the ALTGXB MegaWizard Plug-In Manager

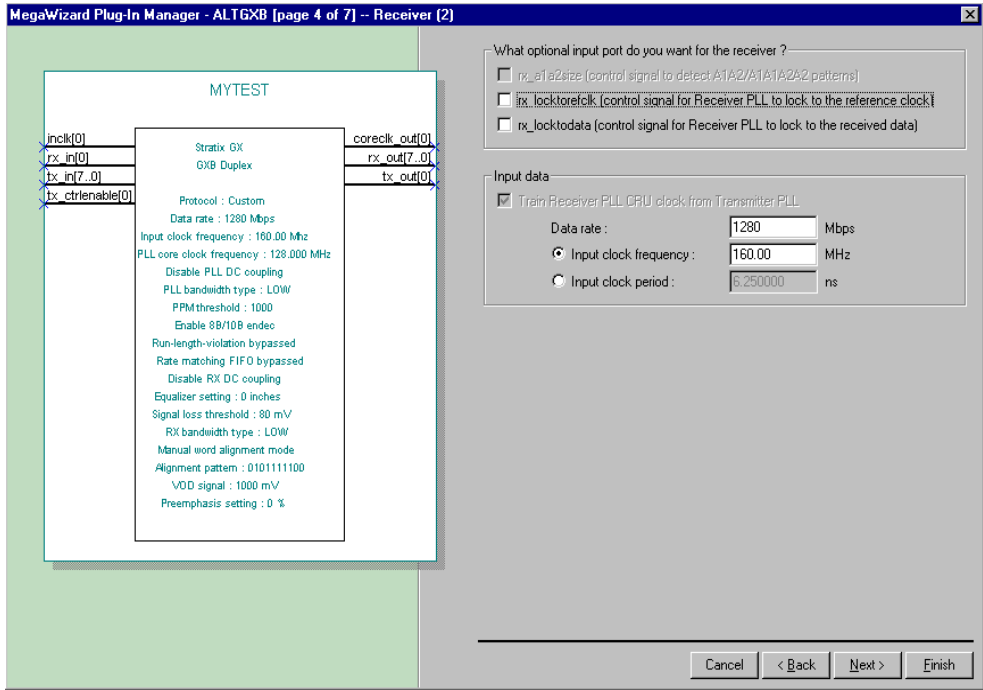


Figure 19 shows page five of the ALTGXB wizard, where you can select optional output ports.

Figure 19. Page Five of the ALTGXB MegaWizard Plug-In Manager

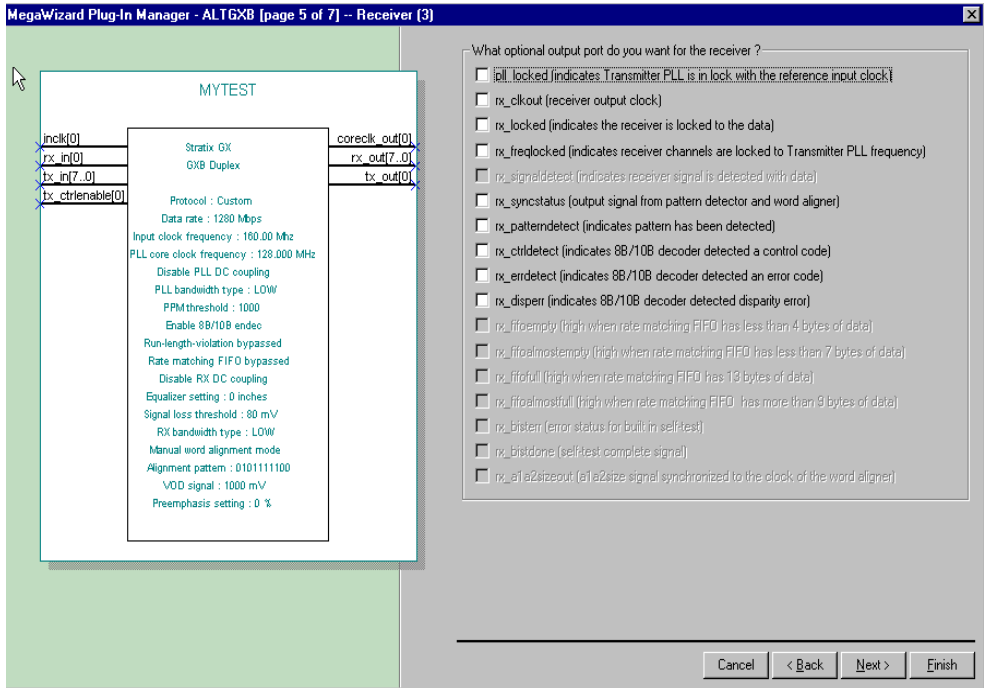


Figure 20 shows page six of the ALTGXB wizard, where you can select transmitter functionality and optional input ports.

Figure 20. Page Six of the ALTGXB MegaWizard Plug-In Manager

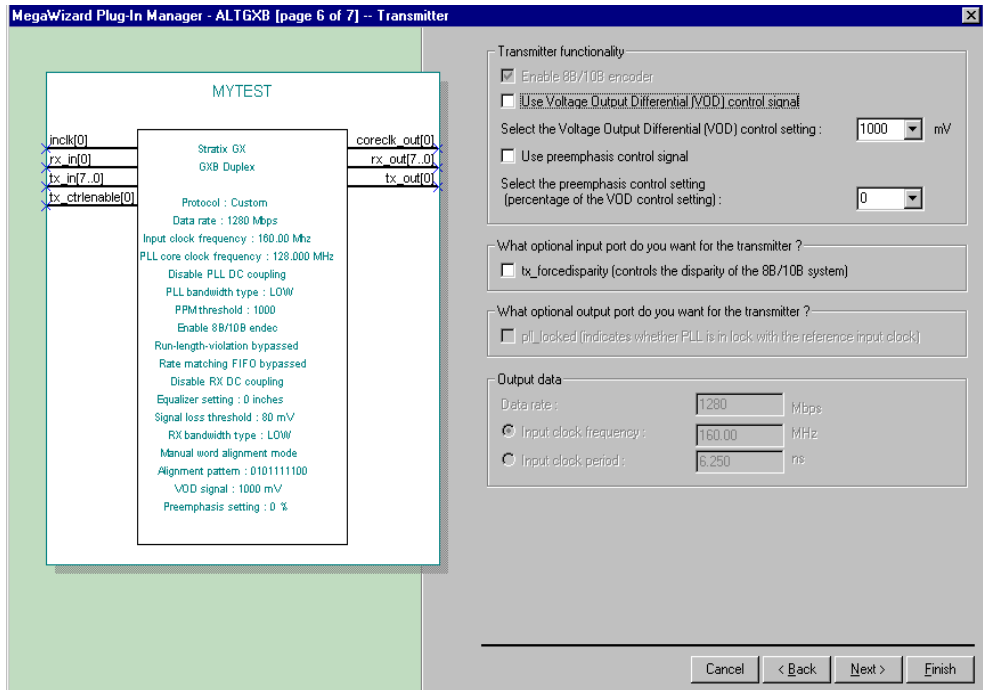
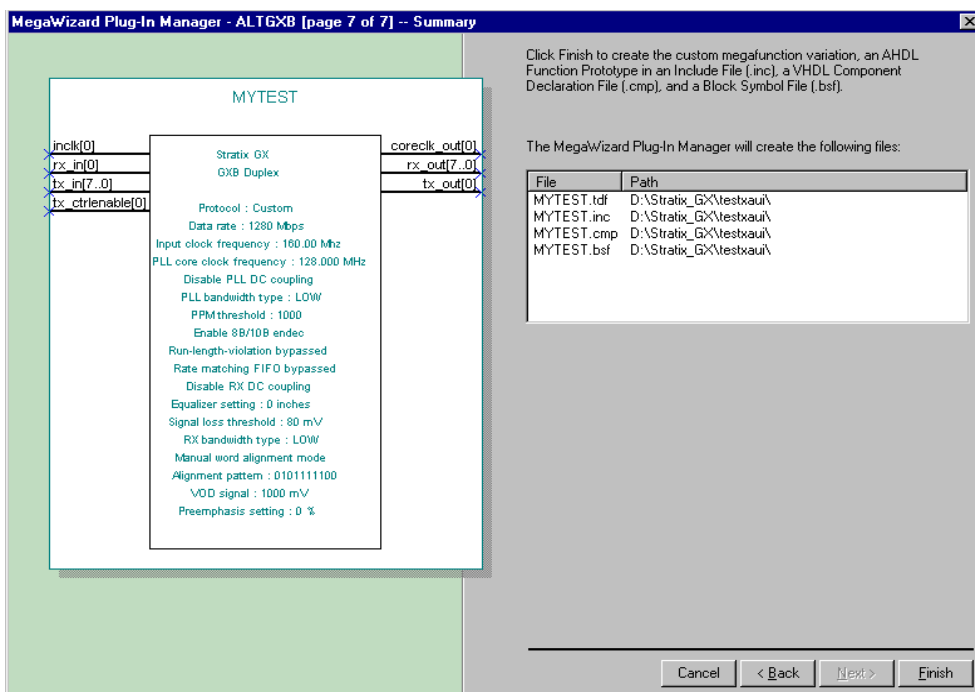


Figure 21 shows page seven of the ALTGXB wizard, which is the summary page that shows a list of MegaWizard-generated files.

Figure 21. Page Seven of the ALTGX B MegaWizard Plug-In Manager



Example Applications Using Stratix GX Devices and XAUI

In the traditional Ethernet space, enterprise network administrators are continually pressured to satisfy users' unrelenting demand for more bandwidth. This bandwidth needs to be supported at the aggregation points in the campus network, which are typically the campus backbone, inter-campus links, and server farms. However, a much broader use of 10 Gigabit Ethernet includes areas such as data centers, MAN access and backbone, and WAN access.

Because of the wide variety of 10 Gigabit Ethernet applications, it must be functionally versatile. The 10 Gigabit Ethernet's ability to bolt on many types of network layer protocols and perform various types of packet processing (i.e., per application) proves its versatility, and is the reason for its success. At the same time, the 10 Gigabit Ethernet standard provides the building blocks required up to the network layer interface.

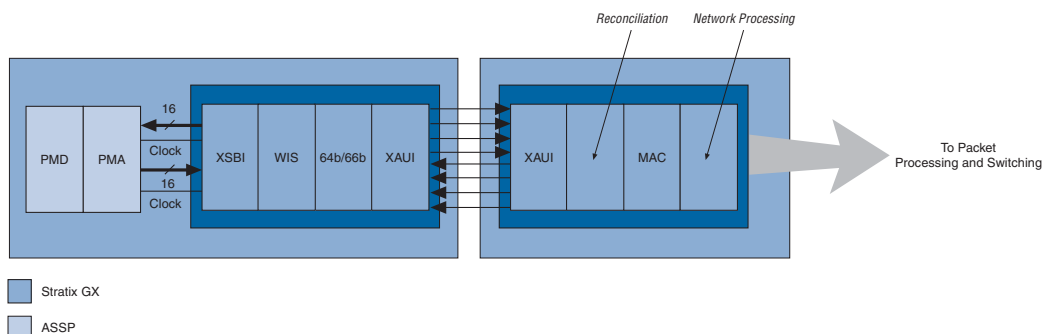


Design engineers sometimes find that they need other protocols in their system to bridge between components that have different interfaces. In these cases, bridging between one interface to XAUI is required. For more information on bridging between the popular SPI-4.2 interface and XAUI, see ["Bridging High-Speed Communications Protocols"](#) on page 28.

Implementing 10 Gigabit Ethernet

Figure 22 provides an overview of a 10 Gigabit Ethernet solution. As discussed in the “Fundamentals of 10 Gigabit Ethernet & XAUI” on page 1, the 10 Gigabit Ethernet architecture is divided into three sublayers (i.e., PMD, PMA, and PCS). For example, optical transceivers are PMD sublayers. The PMA converts the data between the PMD sublayer and the PCS sublayer. The PCS is made up of 64b/66b and interface functions from XSBI to XGMII. Also, for 10GBASE-W PHY implementations, a WIS layer is introduced to help translate SONET/SDH data to 10 Gigabit Ethernet data. The WIS contains OC-192 framing and scrambling functions. XAUI is implemented in the XSGS layers of the 10 Gigabit Ethernet standard. XAUI allows more separation in distance between the MAC and PHY layers, and it uses four lanes of 8b/10b coded data at a rate of 3.125 Gbps.

Figure 22. Implementing a 10 Gigabit Ethernet Subsystem



To implement integrated 10 Gigabit Ethernet solutions, an Altera Megafunction Partners Program (AMPPSM) partner—MorethanIP—offers a 10 Gigabit Ethernet MAC core that can be used with the Stratix GX device transceiver. For more information on this core, refer to the Intellectual Property section of the Altera web site at www.altera.com.

Bridging High-Speed Communications Protocols

Network-equipment designers require seamless communication among system devices. Two common problems that can adversely affect system communications are:

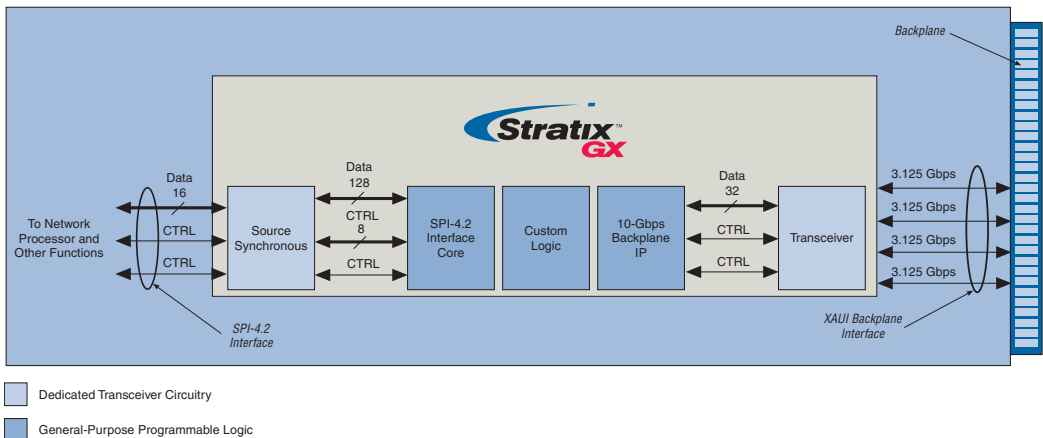
- Mismatched protocols between components
- Implementation of custom functions in the data path

There are various specialized protocols for differing types of data-transfer topologies, such as backplanes or chip-to-chip communication. Stratix GX devices were built to bridge various high-speed communications protocols and to fully accommodate value-added custom functions.

The use of high-speed serial links in backplane and chip-to-chip communication using is growing rapidly. The chip-to-chip source-synchronous interface SPI-4.2 (i.e., also known as SPI-4 Phase 2 and POS-PHY Level 4) is also popular and is being used as a template for emerging interfaces such as CSIX streaming.

Figure 23 illustrates how a Stratix GX device can bridge device communication—between XAUI and the SPI-4.2 interface—and still allow custom-logic implementation in the data path. In Figure 23, four serial links—running at 3.125 Gbps each—run over a backplane to a Stratix GX device. Stratix GX devices are ideal for this type of bridging application as they can support up to 20 serial links running at 3.125 Gbps. In addition, you can customize the Stratix GX device's programmable logic to implement traffic management, queue management, statistical metering, and control functions. You can then use Stratix GX device source-synchronous I/O blocks to implement the 16-bit LVDS SPI-4.2 interface with dynamic phase alignment (DPA).

Figure 23. Stratix GX Device in XAUI to SPI-4.2 Bridge Application



Summary

High-speed communications systems are becoming increasingly complex, and thus, communication between devices and over backplanes can be more problematic. However, because Stratix GX devices are designed to fully support XAU1—as well as other 10 Gigabit Ethernet interface protocols—system bottlenecks can be improved tremendously. For example, Stratix GX devices support up to 20 channels of 3.125 Gbps, deliver 1 Gbps source-synchronous channels with DPA, and offer a high-performance logic array to integrate advanced functionality in a single chip.

This application note demonstrates that Stratix GX devices enable full support of XAU1, which includes meeting electrical specifications and containing all the necessary components to implement the XGXS sublayer of the 10 Gigabit Ethernet.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com

Copyright © 2002 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, mask work rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

