10Gb Ethernet XGXS IP Core

User's Guide

April 2004
ipug15_02

Introduction

Lattice's 10GbE XGXS core provides an ideal solution that meets the need of today's LAN/WAN applications. The 10GbE XGXS core provides a solution for bridging between 10 Gigabit Media Independent Interface (XGMII) and 10 Gigabit Attachment Unit Interface (XAUI) devices. This core allows designers to focus on the application rather than the XGXS core, resulting in faster time to market.

Lattice's 10GbE XGXS core is a fully synchronous core developed in conjunction with the Lattice ORCA® ORT82G5 FPSC to provide a full solution. For more information on these and other Lattice products, refer to the Lattice web site at www.latticesemi.com.

This user's guide explains the functionality of the 10GbE XGXS core and how it can be implemented to provide a full XGMII-XAUI bridging solution. It also explains how to achieve the maximum level of performance.

The 10GbE XGXS core comes with the documentation and the files listed below:

- Data sheet
- Lattice gate level netlist
- RTL simulation model for evaluation
- Core instantiation template

Features

- Complete 10Gb Ethernet Extended Sublayer (XGXS) solution based on the ORCA ORT82G5 0.6-3.7Gbit/s 8b/10b Backplane Interface FPSC, enabling flexible10GbE LAN/WAN application solutions.
- IP targeted to the ORT82G5 programmable array section implements functionality conforming to IEEE 802.3ae, including:
  - 10GbE Media Independent Interface (XGMII).
  - Slip buffers for clock domain transfer to/from the XGMII interface.
  - Complete translation between XGMII and XAUI PCS layers, including 8b/10b encoding and decoding of Idle, Start, Terminate, Error and Sequence code groups and sequences, and randomized Idle generation in the XAUI transmit direction.
  - 64-bit data/8-bit control packet generator/checker on the XGMII side that supports standard compliant CRPAT and CJPAT generation and checking for XAUI interoperability testing.
  - Standard compliant MDIO/MDC interface.
  - Automatic initialization and synchronization of the embedded core.
  - Interface with the high-speed SERDES block embedded in the ORT82G5 that implements a standard XAUI.
- XAUI functionality supported by the embedded portion of the ORT82G5, including
  - Eight channels of 3.125Gbits/s serializer/deserializer with 8b10b encoding/decoding (four SERDES channels are used in this application).
  - XAUI compliant lane-by-lane synchronization.
  - Lane deskew functionality.
  - Microprocessor interface programmable via the Series 4 system bus.
- IP provided in encrypted netlist.
- ModelSim simulation models and test benches available for free evaluation.

General Description

The 10 Gigabit Ethernet Extended Sublayer (XGXS) Intellectual Property (IP) Core enables the creation of system solutions for 10 Gigabit Ethernet (10GbE) applications as defined by IEEE 802.3ae. This IP Core targets the programmable array section of the ORCA ORT82G5 FPSC and provides a bridging function between 10 Gigabit Media Independent Interface (XGMII) and 10 Gigabit Attachment Unit Interface (XAUI) devices.

The ORT82G5 is a high-speed transceiver with an aggregate bandwidth of up to 29.6Gbits/s that is targeted towards users in need of high-speed backplane and chip-to-chip interfaces using Ethernet and Fibre-Channel based protocols. The ORT82G5 has eight channels of integrated 0.6-3.7Gbits/s SERDES channels that can be used as 2x10Gbits/s XAUI interfaces.

XAUI is a high-speed interconnect that offers reduced pin count and is specified to drive up to 20 inches of PCB trace on standard FR-4 material. Each XAUI interface comprises four self-timed 8b/10b encoded serial lanes each operating at 3.125Gbits/s and thus is capable of transferring data at an aggregate rate of 10Gbits/s.

XGMII is a 156MHz Double Data Rate (DDR), parallel, short-reach interconnect interface (typically less than two inches). It supports interfacing to 10Gbits/s Ethernet Media Access Control (MAC) and PHY devices.

In this design, the XGXS core is implemented in the FPGA portion of the device. A packet generator/checker and MDIO interface are also implemented in the FPGA logic.

The XGXS IP core is provided with implementation scripts, test benches, and documentation to allow customers to integrate the functions for 10GbE LAN/WAN applications.

XGXS Application Overview

The location of the XGXS in the 10GbE protocol stack is shown in Figure 1. A simplified block diagram of the ORT82G5 XGXS solution is shown in Figure 2. The ORT82G5 with the XGXS IP implemented in the programmable logic array provides the bridging capability to extend a standard 36-bit DDR XGMII across a XAUI-compatible backplane.

Figure 1. XGXS Location in 10GbE Protocol Stack

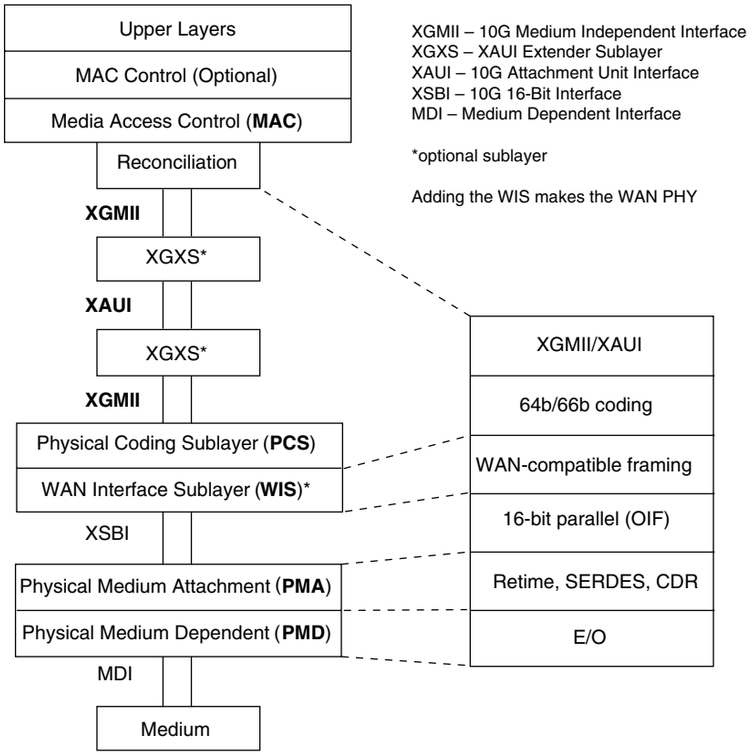
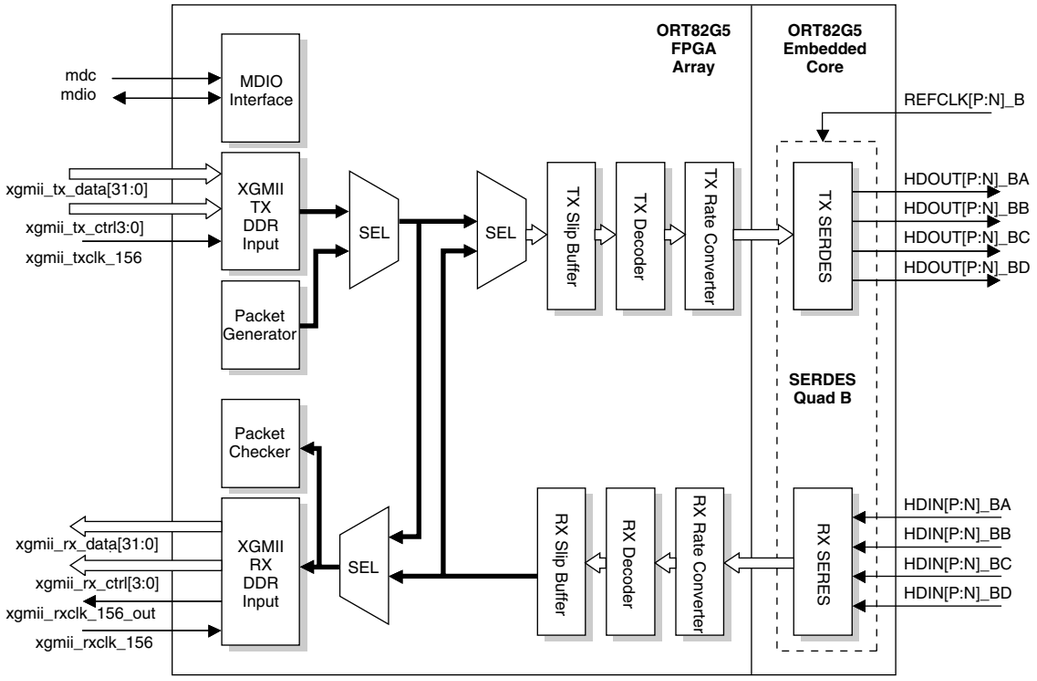


Figure 2. XGXS Solution Simplified Block Diagram



Functional Description

The XGXS receive path, shown in Figure 3, is the data path from the XAUI to the XGMII interface. In the receive direction, 8b/10b encoded data received at the XAUI SERDES interface is demultiplexed and passed to decoder logic, where it is translated and mapped to the XGMII data format. The output of the encoder is then passed through a slip buffer that compensates for XAUI and XGMII timing differences and then to the XGMII 36-bit (32-bit data and four control bits) 156MHz DDR external interface.

The receive direction data translations are shown in Figure 4. The ORT82G5 embedded core aligns the data from the four SERDES implementing the XAUI lanes and passes the aligned 8b/10b encoded data to the XGXS core implemented in the programmable array. The 8b/10b symbols on each XAUI lane are converted to XGMII format and passed to the corresponding XGMII lane.

The XGMII comprises four lanes, labeled [0:3], and one clock in both transmit and receive directions. Each lane includes eight data signals and one control signal. Double Data Rate (DDR) transmission is utilized, with the data and control signals sampled on both the rising and falling edges of a 156.25MHz (nom) clock for an effective data transfer rate of 2.5Gbit/s.

Figure 3. XGXS Receive Path

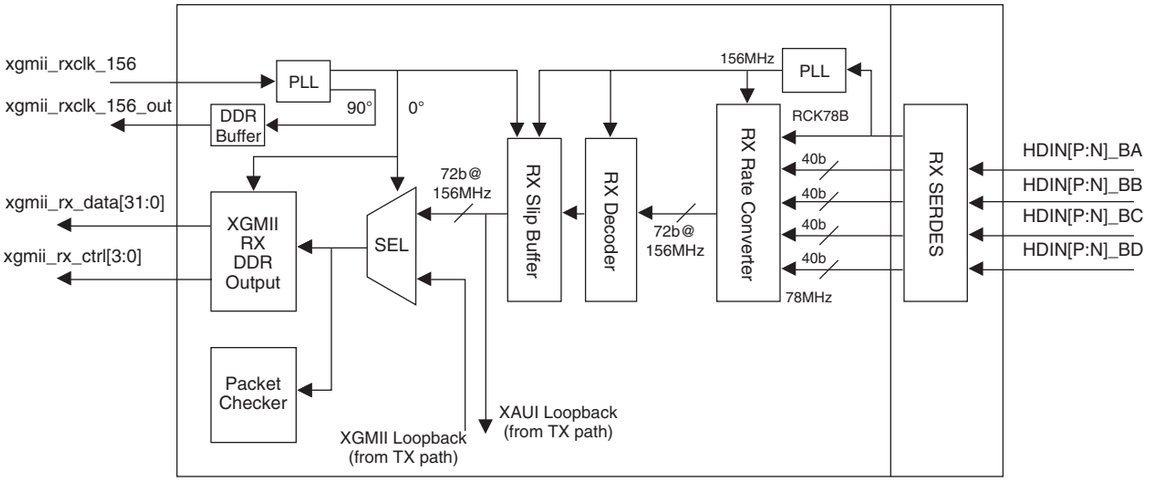
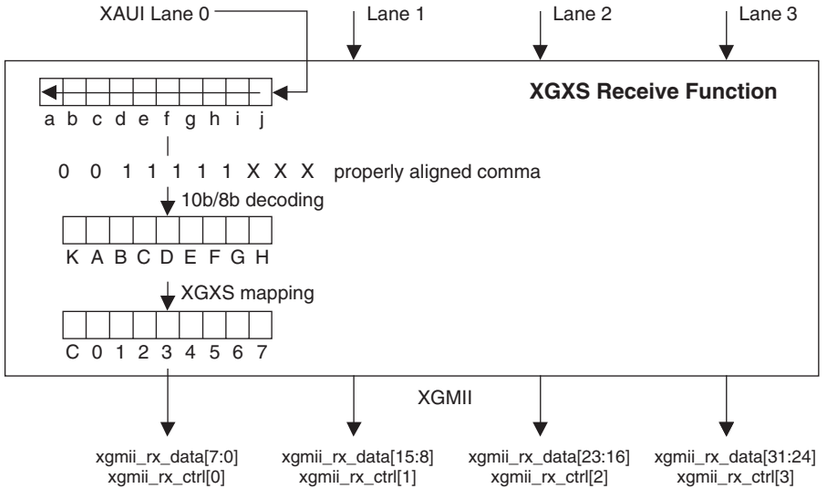


Figure 4. XGXS Receive Direction Data Translations



The transmit path, shown in Figure 5, is the data path from XGMII to XAUI. In the transmit direction, the 36-bit DDR data and control received at the XGMII are converted to single-edge timing and passed through a slip buffer that compensates for XAUI and XGMII timing differences. The XGMII data and control are then passed to the TX encoder, where they are translated and mapped to the 8b/10b XAUI transmission code and then passed to the SERDES interface.

The transmit direction data translations are shown in Figure 6. Data and control from each of the four XGMII lanes are translated and mapped to the corresponding XAUI lanes. The transmit encoder includes the transmit idle generation state machine that generates a random sequence of /A/, /K/ and /R/ code groups as specified in IEEE 802.3ae.

Figure 5. XGXS Transmit Path

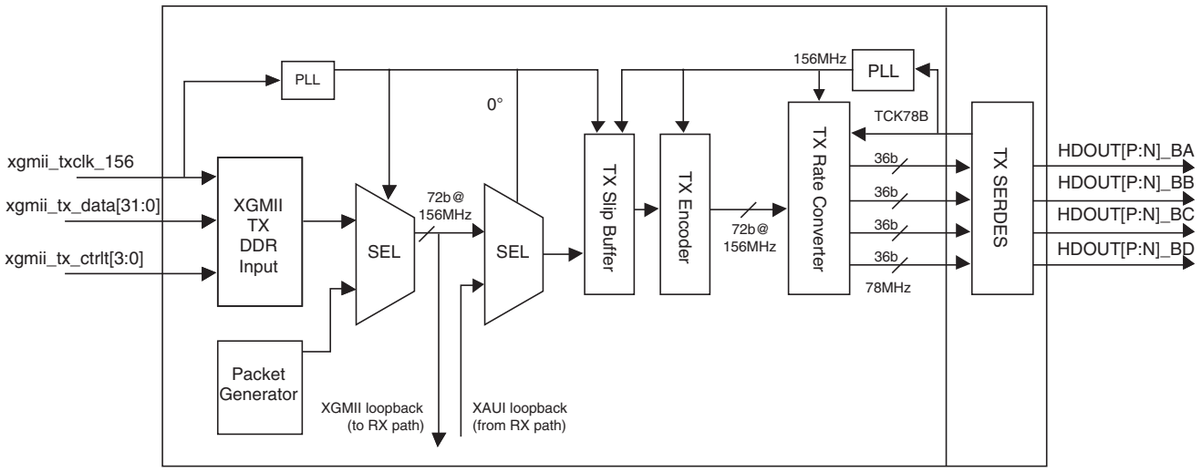
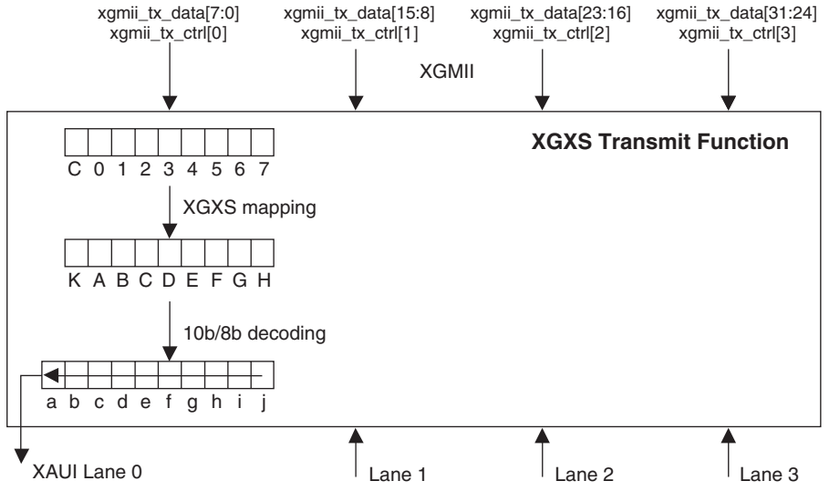


Figure 6. XGXS Transmit Direction Data Translations



XGMII and Slip Buffers

The 10Gigabit Media Independent Interface (XGMII) supported by the XGXS solution conforms to Clause 46 of IEEE 802.3ae.The XGMII is composed of independent transmit and receive paths. Each direction uses 32 data signals, four control signals and a clock. The 32 data signals in each direction are organized into four lanes of eight signals each. Each lane is associated with a control signal as shown in Table 1.

Table 1. XGMII Transmit and Receive Lane Associations[1]

Tx Data (xgmii_tx_data) Rx Data (xgmii_rx_data)	Tx Control (xgmii_tx_ctrl) Rx Control (xgmii_rx_ctrl)	XAUI Lane
[7:0]	[0]	0
[15:8]	[1]	1
[23:16]	[2]	2
[31:24]	[3]	3

1. The XGMII TX signals are XGXS inputs to the transmit path (XGMII to XAUI). The XGMII RX signals are XGXS outputs of the receive path (XAUI to XGMII).

The XGMII supports Double Data Rate (DDR) transmission, i.e. the data and control input signals are sampled on both the rising and falling edges of the corresponding clock. The XGXS XGMII input (tx) data is sampled based on an input clock typically sourced from the MAC or PHY device running at 156.25MHz, 1/64th of the 10Gb data rate. The XGXS XGMII output (rx) data is referenced to a forwarded clock that is phase locked to a 156.25MHz (typical) input reference.

The control signal for each lane is de-asserted when a data octet is being sent on the corresponding lane and asserted when a control character is being sent. Supported control octet encodings are shown in Table 2. All data and control signals are passed directly to/from the 8b/10b encoding/decoding blocks with no further processing by the XGMII block. Note that the packet Start control word is only valid on lane 0.

Table 2. XGMII Control Encoding

Control	Data	Description
0	0x00 - 0xFF	Normal data transmission
1	0x00 - 0x06	Reserved
1	0x07	Idle
1	0x08 - 0x9B	Reserved
1	0x9C	Sequence (only valid on lane 0)
1	0x9D - 0xFA	Reserved
1	0xFB	Start (only valid on lane 0)
1	0xFC	Reserved
1	0xFD	Terminate
1	0xFE	Error
1	0xFF	Reserved

The XGMII blocks incorporate slip buffers that accommodate small differences between XGMII and XAUI timing by inserting or deleting idle characters. The slip buffer is implemented as a 256 x 72 FIFO. There are four flags out of the FIFO: full, empty, partially full and partially empty. The partially empty flag is used as the watermark to start reading from the FIFO. If the difference between write and read pointers falls below the partially empty watermark and the entire packet has been transmitted, idle characters are inserted until the partially full watermark is reached. No idle is inserted during data transmission.

XAUI-to-XGMII Translation (Receive Interface)

A block diagram of the XGXS receive data path was shown previously in Figure 3. The XGXS solution utilizes ORT82G5 SERDES Quad B. The receive interface converts the incoming XAUI stream into XGMII-compatible signals. At the ORT82G5 embedded core interface, the XGXS receive block receives 40 bits of data at 78MHz (32 bits of data, four bits of control and four unused bits) from each XAUI lane. Data from the embedded core are first passed to the RX rate converter block where the 144 bits of data and control received at a 78MHz rate are converted to 72 bits of data and control clocked at 156MHz.

The data from the RX rate converter is passed to the RX decoder. The RX decoder block converts the XAUI code to the XGMII code. Table 3 shows the 8b/10b code points. Table 4 shows the code mapping between the two domains in the receive direction. XAUI /A/, /R/, /K/ characters are translated into XGMII Idle (/I/) characters.

Data from the RX decoder block is written to the RX slip buffer. As mentioned previously, the slip buffers are required to compensate for differences in the write and read clocks derived from the XAUI and XGMII reference clocks, respectively. Clock compensation is achieved by deleting (not writing) idle cells into the buffer when the "almost full" threshold is reached and by inserting (writing) additional idle cells into the buffer when the "almost empty" threshold is reached. All idle insertion/deletion occurs during the Inter-Packet Gap (IPG) between data frames.

Table 3. XAUI 8b/10b Code Points

Symbol	Name	Function	Code-Group
/A/	Align	Lane Alignment (XGMII Idle)	K28.3
/K/	Sync	Code-Group Alignment (XGMII Idle)	K28.5
/R/	Skip	Clock Tolerance Compensation (XGMII Idle)	K28.0
/S/	Start	Start of Packet Delimiter (in Lane 0 only)	K27.7
/T/	Terminate	End of Packet Delimiter	K29.7
/E/	Error	Error Propagation	K30.7
/Q/	Sequence	Link Status Message Indicator	K28.4
/d/	Data	Information Bytes	Dxx.x

Table 4. XAUI 8b/10b to XGMII Code Mapping

8b/10b Data from Embedded Core [7:0]	XGMII Data [7:0]	XGMII Control [3:0]
Dxx.x	0x00-0xFF (Data)	0
K28.5 (0xBC)	0x07 (Idle)	1
K28.3 (0x7C)	0x07 (Idle)	1
K28.0 (0x1C)	0x07 (Idle)	1
K27.7 (0xFB)	0xFB (Start)	1
K29.7 (0xFD)	0xFD (Terminate)	1
K30.7 (0xFE)	0xFE (Error)	1
K28.4 (0x9C)	0x9C (Ordered Set)	1

XGMII-to-XAUI Translation (Transmit Interface)

A block diagram of the XGXS receive data path was shown previously in Figure 4. The TX interface converts the incoming XGMII data into XAUI-compatible characters. 36-bit XGMII DDR input data and control signals are initially converted to a 72-bit bus based on a single edge 156MHz clock. The data and control read are then passed into a TX slip buffer identical to the one used for the RX interface.

After the slip buffer, the XGMII formatted transmit data and control are input to the TX encoder that converts the XGMII characters into 8b/10b format as shown in Table 4. The idle generation state machine in the TX encoder converts XGMII /I/ characters to a random sequence of XAUI /A/, /K/ and /R/ characters as specified in IEEE 802.3ae. XGMII idles are mapped to a random sequence of code groups to reduce radiated emissions. The /A/ code groups support XAUI lane alignment and have a guaranteed minimum spacing of 16 code-groups. The /R/ code groups are used for clock compensation. The /K/ code groups contain the 8b/10b comma sequence.

Table 5. XGMII to XAUI Code Mapping

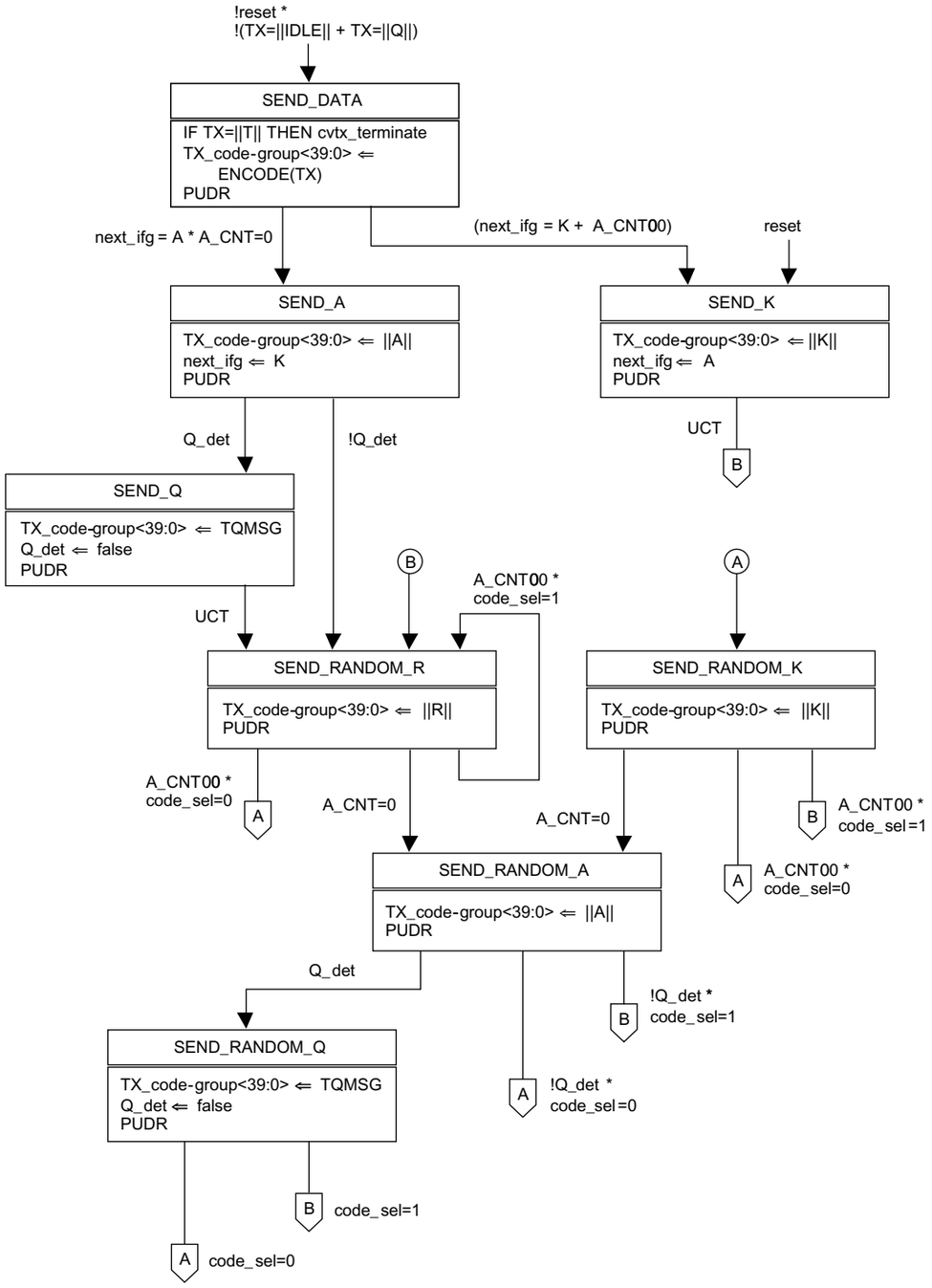
	XGMII	XAUI
Idle	/I/ = 0x07	Randomized /A/, /R/, /K/ Sequence /A/ = K28.3 = 0x7C /R/ = K28.0 = 0x1C /K/ = K28.5 = 0xBC (Comma)
Start	/S/ = 0xFB	/S/ = 0xFB
Error	/E/ = 0xFE	/E/ = 0xFE
Terminate	/T/ = 0xFD	/T/ = 0xFD
Ordered Set	/Q/ = 0x9C	/Q/ = 0x9C
Data	Control = 0	Control = 0

The random /A/, /R/, /K/ sequence is generated as specified in section 48.2.5.2.1 of IEEE 802.3ae and shown in the state machine diagram in Figure 7. In addition to idle generation, the state machine also forwards sequences of ||Q|| ordered sets used for link status reporting. These sets have the XGMII sequence control character on lane 0 followed by three data characters in XGMII lanes 1 through 3. Sequence ordered-sets are only sent following an ||A|| ordered set.

The random selection of /A/, /K/, and /R/ characters is based on the generation of uniformly distributed random integers derived from a PRBS. Minimum ||A|| code group spacing is determined by the integer value generated by the PRBS (A_cnt in Figure 7). ||K|| and ||R|| selection is driven by the value of the least significant bit of the generated integer value (code_sel in Figure 7). The idle generation state machine specified in IEEE 802.3ae and shown in Figure 7 transitions between states based on a 312MHz system clock. The TX encoder implemented in the XGXS IP runs at a system clock rate of 156MHz. Thus the XGXS state machine implementation performs the equivalent of two state transitions each clock cycle.

The final stage in the programmable core is the transmit demultiplexer, which converts the 72 bits of data and control clocked at 156MHz to a 144-bit wide bus clocked at 78MHz, which is then passed to the ORT82G5 embedded core. Data and control are distributed across the four XAUI lanes as discussed previously. The transmit data bus per lane is 36 bits wide at the embedded core boundary as compared to the receive data bus which is 40 bits wide.

Figure 7. XGXS Idle Transmit State Diagram



Packet Generator and Checker

The XGXS IP includes packet generator and checker capabilities supporting the CRPAT and CJPAT test patterns specified in IEEE 803.2ae. These capabilities may be utilized in place of the external XGMII interface and are provided to support system debugging and/or interoperability testing.

The generator/checker blocks have the following programmable features controllable via the MDIO interface:

- Enable test pattern transmission (versus XGMII data transmission).
- Select CRPAT or CJPAT.
- Transmit continuous packets or a specific number of packets.
- Program the number of packets to be sent in non-continuous mode.
- Program the number of additional idle bytes to be sent during the IPG.

16-bit registers are also implemented which provide counts of:

- Packets transmitted by the generator.
- Packets received by the checker.
- Errored packets received.

The numbers of packets transmitted and received are determined by counting packet Start control bytes. Errored packets are detected by checking the packet CRC. These registers are accessible via the MDIO.

Continuous Random Test Pattern (CRPAT)
The purpose of the CRPAT capability is to provide a data pattern that has broad spectral content and minimal peaking that can be used for the measurement of jitter. This pattern consists of a continuous stream of identical packets separated by a minimum IPG. Each packet within this pattern consists of eight bytes of PREAMBLE/SFD followed by 1488 data bytes followed by four octets of CRC and 12 octets of IPG. The specific pattern is as follows:

Start/Preamble/SFD:

FB 55 55 55 55 55 55 D5

Data bytes:

BE (repeat for 4 bytes)
D7 (repeat for 4 bytes)
23 (repeat for 4 bytes)
47 (repeat for 4 bytes)
6B (repeat for 4 bytes)
8F (repeat for 4 bytes)
B3 (repeat for 4 bytes)
14 (repeat for 4 bytes)
5E (repeat for 4 bytes)
FB (repeat for 4 bytes)
35 (repeat for 4 bytes)
59 (repeat for 4 bytes)

(repeat data byte pattern 31 times)

CRC:

F8 79 05 59

IPG:

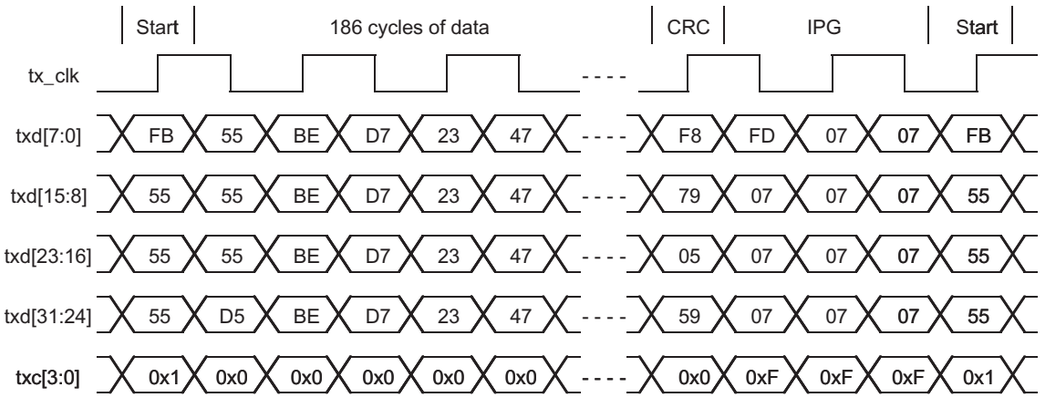
FD 07 07 07 07 07 07 07 07 07 07 07

The CRPAT packet sequence is generated by the XGMII packet generator and is mapped into appropriate XAUI code groups by the XGXS TX encoder block. The CRPAT frame is mapped to the XAUI lanes as shown in Table 6. The packet generator output may also be loop back to the XGMII RX interface. Figure 8 shows how CRPAT packets are mapped to the 32-bit XGMII data bus. Each byte lane corresponds to a XAUI lane as described in Table 1. All values shown in the figure are in hexadecimal.

Table 6. Lane Mapping of Frame Data

Lane 0	Lane 1	Lane 2	Lane 3
SOP (FB)	P1 (55)	P2 (55)	P3 (55)
P4 (55)	P5 (55)	P6 (55)	SFD (D5)
D1 (BE)	D2 (BE)	D3 (BE)	D4 (BE)
—	—	—	—
D1485 (59)	D1486 (59)	D1487 (59)	D1488 (59)
CRC1 (F8)	CRC2 (79)	CRC3 (05)	CRC4 (59)
EOP (FD)	Idle (07)	Idle (07)	Idle (07)
Idle (07)	Idle (07)	Idle (07)	Idle (07)
Idle (07)	Idle (07)	Idle (07)	Idle (07)

Figure 8. Packet Transmission at the XGMII Interface



Continuous Jitter Test Pattern (CJPAT)

The purpose of the CJPAT capability is to provide a data pattern that exposes a receiver's CDR to large instantaneous phase jumps. The pattern alternates repeating low transition density patterns with repeating high transition density patterns. The CJPAT pattern consists of a continuous stream of identical packets separated by a minimum IPG. Each packet consists of eight bytes of START/PREAMBLE/ SFD followed by 1504 data bytes followed by four CRC bytes followed by a minimum of 12 bytes of IPG. The specific pattern is as follows:

START/PREAMBLE/SFD (same as CRPAT)

Data bytes:

0B (lane 0)
7E (for 3 bytes - lanes 1, 2 and 3)
7E (for 524 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)

EB (for 4 bytes)
F4 (for 4 bytes)
AB (for 4 bytes)
B5 (for 160 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
7E (for 528 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
AB (for 4 bytes)
B5 (for 160 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)
EB (for 4 bytes)
F4 (for 4 bytes)

CRC:

BD 9F 1E AB

IPG: (same as CRPAT)

Loopback Configuration

The XGXS IP solution supports three loopback capabilities as shown in Figure 9. Two of the loopbacks are implemented in the IP resident in the programmable core and one is implemented in the ORT82G5 embedded core. All of these loopbacks are controllable via the MDIO interface.

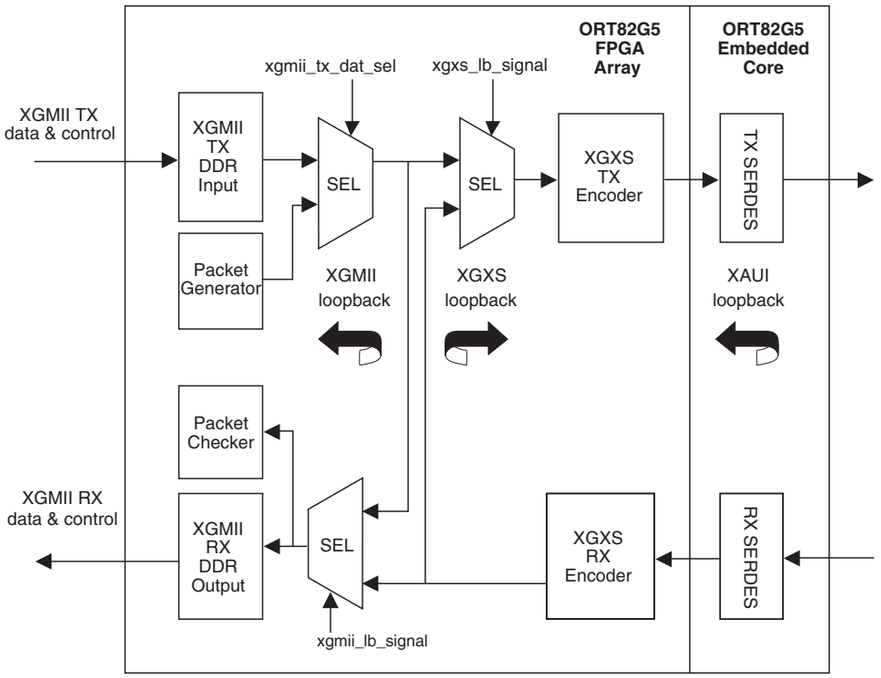
The XGXS loopback is implemented in the programmable core and loops back receive XAUI data to the transmit XAUI interface. The XGXS loopback only works when the XAUI transmit and receive clocks are synchronized.

The XGMII loopback is implemented in the programmable core and loops back receive XGMII TX data to the receive XGMII interface. The XGMII loopback only works when the XGMII transmit and receive clocks are synchronized.

Note that the XGMII and XAUI loopbacks can operate simultaneously.

The XAUI loopback is implemented in the ORT82G5 embedded core and loops back transmit XGMII data in the receive direction. Please refer to the ORT82G5 data sheet for additional information about this loopback capability.

Figure 9. XGXS Loopback Capabilities



Inject Error Capability

An active low input (`inj_err_n`) is provided that supports the ability to inject errors in the packet generator transmit path. One errored packet is generated each time `inj_err_n` is pulsed low. The error injected may occur anywhere within the packet, but will only occur within the data cycle phase, and not while special characters are being sent. If `inj_err_n` is active after the end of a packet, the XGXS waits until after the next start of packet to inject the error.

Automatic Configuration Capability

The XGXS IP supports the ability to automatically configure the ORT82G5 embedded core upon power up or hardware/software reset. This capability automatically sets all appropriate embedded registers to enable QUAD B Channel alignment in XAUI mode. Automatic configuration of the embedded core is enabled or disabled using the external pin "pwrup_init_en" (active high - automatic configuration enabled).

Automatic configuration performs the sequence of operations specified in the following list. Equivalent configuration may be performed via the MDIO or SYSBUS interface.

In the following list:

- mmb = write memory byte. The command is followed by a hex address value and a hex data value.
- dmb = read memory byte. A hex address value and the number of consecutive addresses to read follow the command.

Configuration Steps

1. mmb 30105 64 ; do a software reset
2. dmb 30105 1 ; wait for a while
3. dmb 30105 1 ; wait for a while
4. mmb 30105 44 ; remove software reset
5. mmb 30A00 00 ; set RX and TX clock selects to BA

6. mmb 30002 40 ; power down TX AA
7. mmb 30003 40 ; power down RX AA
8. mmb 30012 40 ; power down TX AB
9. mmb 30013 40 ; power down RX AB
10. mmb 30022 40 ; power down TX AC
11. mmb 30023 40 ; power down RX AC
12. mmb 30032 40 ; power down TX AD
13. mmb 30033 40 ; power down RX AD
14. mmb 30103 30 ; BA RX 1/1 rate, alarm ovrd, no lnk FSM,8b10br
15. mmb 30113 30 ; BB RX 1/1 rate, alarm ovrd, no lnk FSM,8b10br
16. mmb 30123 30 ; BC RX 1/1 rate, alarm ovrd, no lnk FSM,8b10br
17. mmb 30133 30 ; BD RX 1/1 rate, alarm ovrd, no lnk FSM,8b10br
18. mmb 30102 31 ; BA TX 1/1 rate, 8b10bT + full pre-emphasis
19. mmb 30112 31 ; BB TX 1/1 rate, 8b10bT + full pre-emphasis
20. mmb 30122 31 ; BC TX 1/1 rate, 8b10bT + full pre-emphasis
21. mmb 30132 31 ; BD TX 1/1 rate, 8b10bT + full pre-emphasis
22. mmb 30900 FF ; enable BYTSYNC; lock PLL to data
23. mmb 30933 01; enable characterization pins
24. mmb 30920 01 ; XAUI mode
25. mmb 30921 00 ; don't bypass chnl align. On B
26. mmb 30104 40 ; BA MASK ALARM, NO TESTEN
27. mmb 30114 40 ; BB MASK ALARM, NO TESTEN
28. mmb 30124 40 ; BC MASK ALARM, NO TESTEN
29. mmb 30134 40 ; BD MASK ALARM, NO TESTEN
30. mmb 30901 00 ; NO LOOPBACK, Allow WD align for BA BB BC BD
31. mmb 30910 0F ; Enable Ch. Align. for BA BB BC & BD
32. mmb 30911 55 ; FMPU SYNC MODE FOR BA BB BC BD (QUAD)
33. PROGRAM OTHER FPGA REGISTERS HERE IF NECESSARY
34. dmb 30904 1 ; read XAUI states (before resync)
35. dmb 30905 1 ; read word aligner status and ch248
36. dmb 30914 1 ; read OOS status
37. mmb 30105 64 ; Perform another software reset
38. dmb 30105 1 ; wait for a while before removing reset
39. dmb 30105 1 ; wait for a while before removing reset
40. mmb 30105 44 ; remove software reset
41. mmb 30910 00 ; perform word align
42. mmb 30910 0F ;
43. mmb 30920 01 ; perform resync
44. mmb 30920 03

Automatic SERDES Channel Alignment

The XGXS IP core supports automatic XAUI word and quad channel alignment resynchronization upon loss of sync at the XAUI interface. Automatic resynchronization is enabled/disabled by setting/clearing XGXS register bit 4.8002.15 via the MDIO interface (see the Functional Description section). The resynchronization procedure includes the following steps:

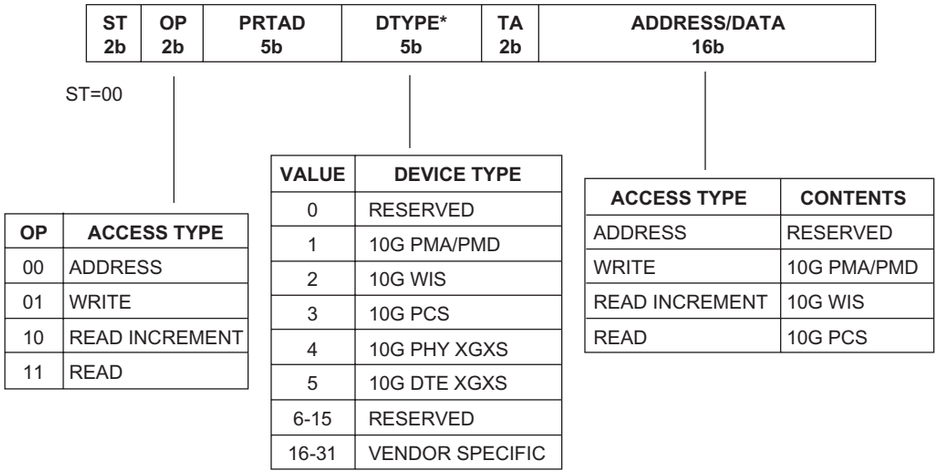
1. Perform a word DEMUX alignment on all channels by:
   Writing 0x0F to ORT82G5 embedded core register 30910.
   Writing 0xFF to ORT82G5 embedded core register 30910.
2. Do a four channel alignment resynchronization by:
   Writing 0x01 to ORT82G5 embedded core register 30920.
   Writing 0x03 to ORT82G5 embedded core register 30920.

Management Data Input/Output (MDIO) Interface

The MDIO interface provides access to the internal XGXS registers. The register access mechanism corresponds to Clause 45 of IEEE 802.3ae. The XGXS core provides access to XGXS registers 0x0000-0x0024 as specified in IEEE 802.3ae. Additional registers in the vendor-specific address space have been allocated for implementation-specific control/status functions.

The physical interface consists of two signals: MDIO to transfer data/address/control to and from the device, and MDC, a clock up to 2.5MHz sourced externally to provide the synchronization for MDIO. The fields of the MDIO transfer are shown in Figure 10.

Figure 10. Fields of MDIO Protocol



OP	ACCESS TYPE
00	ADDRESS
01	WRITE
10	READ INCREMENT
11	READ

VALUE	DEVICE TYPE
0	RESERVED
1	10G PMA/PMD
2	10G WIS
3	10G PCS
4	10G PHY XGXS
5	10G DTE XGXS
6-15	RESERVED
16-31	VENDOR SPECIFIC

ACCESS TYPE	CONTENTS
ADDRESS	RESERVED
WRITE	10G PMA/PMD
READ INCREMENT	10G WIS
READ	10G PCS

* If ST=01, this field is REGAD (register address).

Management Frame Structure

Each management data frame consists of 64 bits. The first 32 bits are preamble consisting of 32 contiguous 1s on the MDIO. Following the preamble is the start-of-frame field (ST) which is a 00 pattern. The next field is the operation code (OP) that is shown in Figure 10.

The next two fields are the port address (PRTAD) and device type (DTYPE). Since the physical layer function in 10GbE is partitioned into various logical (and possibly separate physical) blocks, two fields are used to access these blocks. The PRTAD provides the overall address to the PHY function. The first port address bit transmitted and received is the MSB of the address. The DTYPE field addresses the specific block within the physical layer function.

Device address zero is reserved to ensure that there is not a long sequence of zeros. If the ST field is 01 then the DTYPE field is replaced with REGAD (register address field of the original clause 22 specification). The XGXS core does not respond to any accesses with ST = 01.

The TA field (Turn Around) is a 2-bit turnaround time spacing between the device address field and the data field to avoid contention during a read transaction. The TA bits are treated as don't cares by the XGXS core.

During a write or address operation, the address/data field transports 16 bits of write data or register address depending on the access type. The register is automatically incremented after a read increment. The address/data field is 16 bits.

For an address cycle, this field contains the address of the register to be accessed on the next cycle. For read/write/increment cycles, the field contains the data for the register. The first bit of data transmitted and received in the address/data field is the MSB (bit 15). An example access is shown in Figure 11.

Figure 11. Indirect Address Example

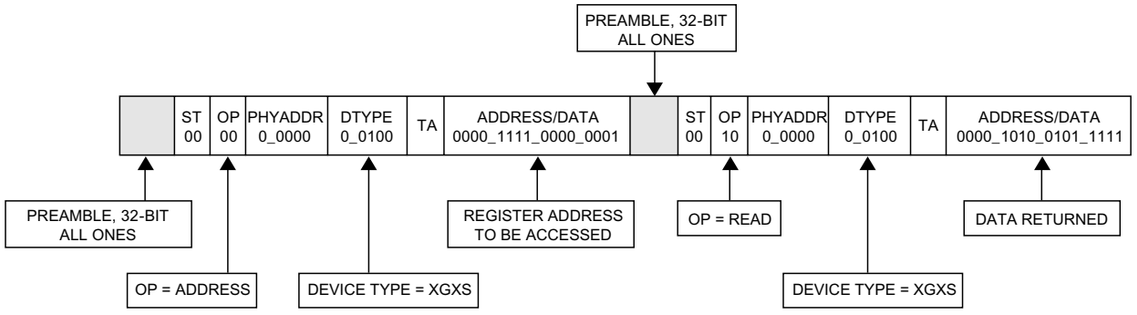


Table 8 shows PHY XGXS registers as described in IEEE Draft P802.3ae. The shaded areas are used to indicate register addresses that are specified in the draft but are not used in this implementation.

There are two vendor supported register ranges. The 4.8000h register range is used for accessing and programming the XGXS registers implemented in the programmable array of the ORT82G5 (XGXS and packet generator/checker registers). All corresponding registers are listed in Table 7. All ORT82G5 embedded core registers can be accessed through the 4.9xxxh registers shown in Table 8, where the address is directly mapped to the ORT82G5 embedded registers. Refer to the ORT82G5 data sheet for complete details on the ORT82G5 embedded core register memory map and register definitions.

Register Descriptions

Table 7. Register Map for XGXS IP (Device Address = 4)

Register Address	Register Name
0	PHY XGXS Control 1
1	PHY XGXS Status 1
2, 3	PHY XGXS Identifier
4	Reserved
5	PHY XGXS Status 2
6 - 23	Reserved
24	10G PHY XGXS Lane status
25 - 32767	Reserved
32768 - 65535	Vendor specific (TBD)

Table 8. XGXS Registers

Bit(s)	Name	Description	R/W	Reset Value
Control 1 Registers				
4.0.15	Reset	1 = PHY XA reset, 0 = Normal operation	R/W S/C	0
4.0.14	Loop Back	1 = Enable Loop back, 0 = Disable Loop back Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30801 [0,1,2,3] CH B = 30901 [0,1,2,3]	R/W	0
4.0.13	Speed Selection	Value always 0	R	0
4.0.12	Reserved	Value always 0	R	0

Table 8. XGXS Registers (Continued)

Bit(s)	Name	Description	R/W	Reset Value
4.0.11	Low Power	0 = Low Power Mode 1 = Normal operation This bit is connected to the ORT82G5 "PASB_PDN" signal.	R/W	1
4.0.[10:7]	Reserved	Value always 0	R	0
4.0.6	Speed Selection	Value always 0	R	0
4.0.[5:2]	Speed Selection	Value always 0	R	0
4.0.[1:0]	Reserved	Value always 0	R	0
Status 1 Registers				
4.1.[15:8]	Reserved	Value always 0	R	0
4.1.7	Fault (Not Supported)	0 = No Fault condition	R	0
4.1.[6:3]	Reserved	Value always 0	R	0
4.1.2	PHY XS TX link status	1 = Link is up, 0 = Link is down Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30814[5] CH B = 30914[5]	R	0
4.1.1	Low Power Ability	1 = Low Power Mode support	R	1
4.1.0	Reserved	Value always 0	R	0
XGXS Identifier Registers				
4.2:[15:0]	PHY XS Identifier	MSB = 0x0000	R	0
4.3:[15:0]	PHY XS Identifier	LSB = 0x0004	R	0004
XGXS Reserved Registers				
4.4.[15:1]	Reserved	Value always 0	R	0
4.4.0	10 G Capable	Value always 1	R	1
Status 1 Registers				
4.5.[15:6]	Reserved	Value always 0	R	0
4.5.5	DTE XS Present	Value always 0	R	0
4.5.4	PHY XS Present	Value always 1	R	1
4.5.3	PCS Present	Value always 0	R	0
4.5.2	WIS Present	Value always 0	R	0
4.5.1	PMD/PMA Present	Value always 0	R	0
4.5.0	Clause 22 regs present	Value always 0	R	0
XGXS Reserved Registers				
4.6.15	Vendor specific device present	Value always 0	R	0
4.6.[14:0]	Reserved	Value always 0	R	0
XGXS Reserved Registers				
4.8.[15:14]	Device present	10 = Device responding to this address	R	10
4.8.[13:12]	Reserved	Value always 0	R	0
4.8.11	Transmit Fault (Not Supported)	0 = No fault of tx path	R	0
4.8.10	Receive Fault (Not Supported)	0 = No fault of tx path	R	0
4.8.9:0	Reserved	Value always 0	R	0
4.15, 4.14	Package Identifier	Value always 0	R	0
XGXS Reserved Registers				
4.24.[15:13]	Reserved	Value always 0	R	0

Table 8. XGXS Registers (Continued)

Bit(s)	Name	Description	R/W	Reset Value
4.24.12	PHY XGXS Lane Alignment	1= TX lanes aligned 0 =TX lanes not aligned Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30814[5] CH B = 30914[5]	R	00
4.24.11	Pattern Testing ability	0 = Not able to generate pattern.	R	0
4.24.10	PHY XGXS has loop back capability	1 = Has loop back capability	R	1
4.24.[9:4]	Reserved	Value always 0	R	0
4.24.3	Lane 3 Sync	1 = Lane 3 synchronized 0 = Lane 3 not synchronized Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30804[6,7] CH B = 30904[6,7]	R	00
4.24.2	Lane 2 Sync	1 = Lane 2 synchronized 0 = Lane 2 not synchronized Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30804[4,5] CH B = 30904[4,5]	R	00
4.24.1	Lane 1 Sync	1 = Lane 1 synchronized 0 = Lane 1 not synchronized Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30804[3,2] CH B = 30904[3,2]	R	00
4.24.0	Lane 0 Sync	1 = Lane 0 synchronized 0 = Lane 0 not synchronized Writing to this bit sets the corresponding ORT82G5 register bits: CH A = 30804[0,1] CH B = 30904[0,1]	R	00
XGXS Reserved Registers				
4.25.15:3	Reserved	Value always 0	R	0
4.25.2	Receive test pattern enable	0 = Receive test pattern not enabled	R	0
4.25.1:0	Test pattern select	00	R	00
4.8000.15:0	XGXS TX PACKET COUNTER	Counts in XGXS the number of packets sent in the TX direction. Cleared on Read	R	0
4.8001.15:0	XGXS RX PACKET COUNTER	Counts in XGXS the number of packets received in the RX direction. Cleared on Read	R	0
4.8002.15	XGXS Automatic channel alignment resync.	Bit enables automatic resync by XGXS upon loss of sync. 1 = enable 0 = disable	R/W	1
4.8002.14	XGXS Loopback	Enables XGXS loopback 1 = enable 0 = disable	R/W	0
4.8002.13	XGMII Loopback	Enables XGMII loopback 1 = enable 0 = disable	R/W	0
4.8002.12:9	RESERVED	UNUSED	—	—
4.8002.8:0	RESERVED	UNUSED	—	—

Table 8. XGXS Registers (Continued)

Bit(s)	Name	Description	R/W	Reset Value
4.8003.15	XGMII TX DATA SELECT	Selects between the packet generator TX XGMII data and that of the IO DDR. 1 = packet generator 0 = IO DDR	R/W	1
4.8003.14	RUN PACKETS	Enables packet generator data transmission 1 = transmit DATA (CRPAT/CJPAT) 0 = transmit IDLE	R/W	0
4.8003.13	SELECT CRPAT OR CJPAT	Selects between CRPAT and CJPAT 1 = transmit CRPAT 0 = transmit CJPAT	R/W	0
4.8003.12	CONTINUOUS/FIXED PACKETS	The value determines whether packets are sent continuously or for a fixed number of times any time 4.8003.14 is set. The fixed number of times is determined by the value of 4.800C.15:0. 1 = continuous 0 = fixed	R/W	0
4.8003.11:8	RESERVED	UNUSED	—	—
4.8003.7:0	RESERVED	UNUSED	—	—
4.8004.15:8	RESERVED	UNUSED	—	—
4.8004.7:0	TX SLIP BUFFER FULL THRESHOLD	The Threshold for when The FIFO controller considers the TX slip buffer FIFO full.	R/W	80
4.8005.15:8	RESERVED	UNUSED	—	—
4.8005.7:0	TX SLIP BUFFER EMPTY THRESHOLD	The Threshold for when The FIFO controller considers the TX slip buffer FIFO empty.	R/W	70
4.8006.15:8	RESERVED	UNUSED	—	—
4.8006.7:0	RX SLIP BUFFER FULL THRESHOLD	The Threshold for when The FIFO controller considers the RX slip buffer FIFO full.	R/W	80
4.8007.15:8	RESERVED	UNUSED	—	—
4.8007.7:0	RX SLIP BUFFER EMPTY THRESHOLD	The Threshold for when The FIFO controller considers the RX slip buffer FIFO empty.	R/W	70
4.8008.15:8	RESERVED	UNUSED	—	—
4.8008.7:0	SERDES CHANNEL BA CODE VIOLATION COUNTER.	CH. BA code violations will increment the counter (cleared on read). This feature is only supported with V3 of ORT82G5	R	
4.8009.15:8	RESERVED	UNUSED	—	—
4.8009.7:0	SERDES CHANNEL BB CODE VIOLATION COUNTER.	CH. BB code violations will increment the counter (cleared on read). This feature is only supported with V3 of ORT82G5	R	
4.800A.15:8	RESERVED	UNUSED	—	—
4.800A.7:0	SERDES CHANNEL BC CODE VIOLATION COUNTER.	CH. BC code violations will increment the counter (cleared on read). This feature is only supported with V3 of ORT82G5	R	—
4.800B.15:8	RESERVED	UNUSED	—	—
4.800B.7:0	SERDES CHANNEL BD CODE VIOLATION COUNTER.	CH. BD code violations will increment the counter (cleared on read). This feature is only supported with V3 of ORT82G5	R	—
4.800C.15:0	NUMBER OF PACKETS TO SEND	Fixed number of times CRPAT/CJPAT packets are to be sent.	R/W	0
4.800D.15:0	PKT CHECKER RECEIVED ERROR COUNTER	Keeps count of the number of error-ed packets received. Cleared on Read	R	0

Table 8. XGXS Registers (Continued)

Bit(s)	Name	Description	R/W	Reset Value
4.8002.15:4	RESERVED	UNUSED	—	—
4.800E.3:0	ADDITIONAL IDLES TO GENERATOR IPG	ADDS X= 8(N+1) ADDITIONAL IDLE BYTES where N>=1 (N= 4.800E.3:0 decimal)	R/W	0

Table 9. XGXS Vendor Specific Registers 4.9xxxh

Bits	Name	802.3ae Description	R/W	ORT82G5 Register Bits	Reset Value
4.9xxx.15:8	Reserved	Value always 0	R	None	0
4.90xx.7:0	SERDES A registers	None	R/W	300xx regs	Spec
4.91xx.7:0	SERDES B registers	None	R/W	301xx regs	Spec
4.98xx.7:0	Channel A [A..D]	None	R/W	308xx regs	Spec
4.99xx.7:0	Channel B [A..D]	None	R/W	309xx regs	Spec
4.9Axx.7:0	Global registers	None	R/W	30Axx regs	Spec
4.A0x.15:8	Reserved	Value always 0	R	None	0
4.A0xx.7:0	System bus	None	R/W	000xx regs	Spec

I/O Signal Descriptions

Table 10. XGXS Solution I/O

Signal Name	Direction	Description
XGMII Signals		
xgmii_tx_data[31:0]	input	32-bit wide DDR XGMII input data.
xgmii_tx_ctrl[3:0]	input	Per-byte DDR XGMII control inputs.
xgmii_txclk_156	input	156MHz XGMII transmit (XGXS input) clock.
xgmii_rx_data[31:0]	output	32-bit wide DDR XGMII output data.
xgmii_rx_ctrl[3:0]	output	Per-byte DDR XGMII control outputs.
xgmii_rxclk_156	input	XGMII receive (XGXS output) reference clock.
xgmii_rxclk_156_out	output	Forwarded XGMII receive (XGXS output) clock.
XAUI Signals		
HDINN_BA	input	High-speed CML receive data input - SERDES quad B, channel A.
HDINP_BA	input	High-speed CML receive data input - SERDES quad B, channel A.
HDINN_BB	input	High-speed CML receive data input - SERDES quad B, channel B.
HDINP_BB	input	High-speed CML receive data input - SERDES quad B, channel B.
HDINN_BC	input	High-speed CML receive data input - SERDES quad B, channel C.
HDINP_BC	input	High-speed CML receive data input - SERDES quad B, channel C.
HDINN_BD	input	High-speed CML receive data input - SERDES quad B, channel D.
HDINP_BD	input	High-speed CML receive data input - SERDES quad B, channel D.
HDOUTN_BA	output	High-speed CML transmit data output - SERDES quad B, channel A.
HDOUTP_BA	output	High-speed CML transmit data output - SERDES quad B, channel A.
HDOUTN_BB	output	High-speed CML transmit data output - SERDES quad B, channel B.
HDOUTP_BB	output	High-speed CML transmit data output - SERDES quad B, channel B.
HDOUTN_BC	output	High-speed CML transmit data output - SERDES quad B, channel C.
HDOUTP_BC	output	High-speed CML transmit data output - SERDES quad B, channel C.
HDOUTN_BD	output	High-speed CML transmit data output - SERDES quad B, channel D.
HDOUTP_BD	output	High-speed CML transmit data output - SERDES quad B, channel D.
REFCLKP_B	input	SERDES Quad B reference clock.
REFCLKN_B	input	SERDES Quad B reference clock.
Please refer to the ORT82G5 Data Sheet for additional information on configuring the SERDES interface for specific applications.		
MDIO Interface Signals		
mdio	input/output	MDIO bi-directional data.
mdc	input	MDIO clock.
XGXS Soft IP Control and Status Signals		
reset_n	input	XGXS programmable core reset (active low).
pwrup_init_en	input	Enable automatic configuration of embedded core (active high - see Sec. 2.7 for details).
inj_err_n	input	Inject error (active low - see Sec. 2.6 for details).
ORT82G5 Embedded Core Control, Global I/O and FPGA Configuration I/O		
Please refer to the ORCA Series 4 FPGA Data Sheet and the ORT82G5 Data Sheet for information on the various configuration options.		

I/O Pin Assignments

Table 11 lists a verified XGMII I/O DDR pinout for the XGXS IP core using an ORT82G5 and the BM680 package. Table 12 lists the remaining FPGA related I/Os. Other pinout configurations may be specified as long as the I/O placement conforms to the constraints specified in Lattice technical note TN1037 *ORCA Series 4 Fast DDR Interface.*

Fixed embedded core pins are not listed. Please refer to the ORT82G5 data sheet for that information.

Table 11. XGMII Related I/Os

Signal	Pin	Direction	Buffer
xgmii_tx_ctrl_0	B26	input	HSTL1
xgmii_tx_ctrl_1	B25	input	HSTL1
xgmii_tx_ctrl_2	B27	input	HSTL1
xgmii_tx_ctrl_3	A27	input	HSTL1
xgmii_rx_ctl_0	AP21	output	HSTL1
xgmii_rx_ctl_1	AK16	output	HSTL1
xgmii_rx_ctl_2	AM18	output	HSTL1
xgmii_rx_ctl_3	AM20	output	HSTL1
xgmii_tx_data_0	C24	input	HSTL1
xgmii_tx_data_1	C22	input	HSTL1
xgmii_tx_data_10	A25	input	HSTL1
xgmii_tx_data_11	A24	input	HSTL1
xgmii_tx_data_12	B23	input	HSTL1
xgmii_tx_data_13	E17	input	HSTL1
xgmii_tx_data_14	E16	input	HSTL1
xgmii_tx_data_15	B22	input	HSTL1
xgmii_tx_data_16	C18	input	HSTL1
xgmii_tx_data_17	C19	input	HSTL1
xgmii_tx_data_18	A22	input	HSTL1
xgmii_tx_data_19	A21	input	HSTL1
xgmii_tx_data_2	C23	input	HSTL1
xgmii_tx_data_20	D17	input	HSTL1
xgmii_tx_data_21	D18	input	HSTL1
xgmii_tx_data_22	B20	input	HSTL1
xgmii_tx_data_23	B19	input	HSTL1
xgmii_tx_data_24	A20	input	HSTL1
xgmii_tx_data_25	A19	input	HSTL1
xgmii_tx_data_26	B18	input	HSTL1
xgmii_tx_data_27	C17	input	HSTL1
xgmii_tx_data_28	D16	input	HSTL1
xgmii_tx_data_29	A17	input	HSTL1
xgmii_tx_data_3	B24	input	HSTL1
xgmii_tx_data_30	B16	input	HSTL1
xgmii_tx_data_31	E15	input	HSTL1
xgmii_tx_data_4	D20	input	HSTL1

Table 11. XGMII Related I/Os (Continued)

Signal	Pin	Direction	Buffer
xgmii_tx_data_5	D19	input	HSTL1
xgmii_tx_data_6	E19	input	HSTL1
xgmii_tx_data_7	E18	input	HSTL1
xgmii_tx_data_9	C20	input	HSTL1
xgmii_rx_data_0	AL18	output	HSTL1
xgmii_rx_data_1	AN21	output	HSTL1
xgmii_rx_data_10	AN25	output	HSTL1
xgmii_rx_data_11	AL22	output	HSTL1
xgmii_rx_data_12	AL23	output	HSTL1
xgmii_rx_data_13	AN27	output	HSTL1
xgmii_rx_data_14	AM25	output	HSTL1
xgmii_rx_data_15	AP29	output	HSTL1
xgmii_rx_data_16	AN29	output	HSTL1
xgmii_rx_data_17	AN28	output	HSTL1
xgmii_rx_data_18	AM26	output	HSTL1
xgmii_rx_data_19	AK23	output	HSTL1
xgmii_rx_data_2	AM21	output	HSTL1
xgmii_rx_data_20	AL25	output	HSTL1
xgmii_rx_data_21	AP31	output	HSTL1
xgmii_rx_data_22	AK24	output	HSTL1
xgmii_rx_data_23	AM28	output	HSTL1
xgmii_rx_data_24	AN30	output	HSTL1
xgmii_rx_data_25	AL26	output	HSTL1
xgmii_rx_data_26	AL28	output	HSTL1
xgmii_rx_data_27	AN31	output	HSTL1
xgmii_rx_data_28	AK26	output	HSTL1
xgmii_rx_data_29	AM30	output	HSTL1
xgmii_rx_data_3	AN22	output	HSTL1
xgmii_rx_data_30	AL29	output	HSTL1
xgmii_rx_data_31	AK27	output	HSTL1
xgmii_rx_data_4	AK18	output	HSTL1
xgmii_rx_data_5	AN23	output	HSTL1
xgmii_rx_data_6	AP26	output	HSTL1
xgmii_rx_data_7	AK19	output	HSTL1
xgmii_rx_data_8	AL21	output	HSTL1
xgmii_rx_data_9	AM23	output	HSTL1
xgmii_rxclk_156	AN18	input	HSTL1
xgmii_rxclk_156_out	AN19	output	HSTL1
xgmii_txclk_156	A23	input	HSTL1

Table 12. Additional I/Os

Signal	Pin	Direction	Buffer
MDIO I/O			
mdio	AA4	input/output	LVCMOS2
mdc	U1	input	LVCMOS2
Control and Status Signals			
reset_n	W1	input	LVCMOS2
pwrup_init_en	Y1	input	LVCMOS2
inj_err_n[1]	F5	input	LVCMOS2
ORT82G5 Embedded Core Control, Global I/O and FPGA Configuration I/O			
Please refer to the ORCA Series 4 FPGA Data Sheet and the ORT82G5 Data Sheet for information on the various configuration options.			
Test Points[2]			
TP0	AC5	input	LVCMOS
TP1	AD1	input	LVCMOS
TP2	AF3	input/output	LVCMOS
TP3	AD3	output	LVCMOS
TP4	A3	output	LVCMOS

1. External pull-up required.
2. I.O that may be used to provide observability of internal signals for debugging purposes. It is recommended that these I/O be routed to accessible points on the PWB.

Table 13 shows voltage bank notations and Table 14 shows pin locations for HSTL1 reference voltages as assigned for the XGXS IP core.

Table 13. VDDIO for XGXS IP Core - By Bank

Bank	VDDIO
VDDIO0	2.5V
VDDIO1	1.5V
VDDIO5	1.5V
VDDIO6	2.5V
VDDIO7	2.5V

Table 14. V_{REF} for XGXS IP Core - By Bank/Group

Bank/Group	Pin	V_{REF}
1/2	D21	Vref_HSTL1
1/3	A26	Vref_HSTL1
1/4	B21	Vref_HSTL1
1/5	A18	Vref_HSTL1
1/6	A15	Vref_HSTL1
5/1	AL17	Vref_HSTL1

Proper isolation of the V_{REF} pins is critical for proper circuit performance. Please see Lattice technical note TN1036 *ORCA® Series 4 I/O User's Guide,* available on the Lattice web site at www.latticesemi.com, for recommendations on the proper use of these pins.

Additional pinout specification information may be found in the XGXS core ORCA PAD specification files included with this IP.

I/O Timing and Electrical Specifications

XGMII Specifications

Clause 46 of IEEE 802.3ae specifies HSTL1 I/O with a 1.5V output buffer supply voltage for all XGMII signals. The HSTL1 specifications comply with EIA/JEDEC Standard EIA/JESD8-6 using Class I output buffers with output impedance greater than 38Ω to ensure acceptable overshoot and undershoot performance in an unterminated interconnection. The parametric values for HSTL XGMII signals are given in Table 15. The HSTL termination scheme is shown in Figure 12. Timing requirements for chip-to-chip XGMII signals are shown in Figure 13.

Additional information on XGMII interface specifications may be found in IEEE 802.3ae. Additional information on implementing the XGMII DDR capabilities in the ORT82G5 may be found in Lattice technical notes TN1036 *ORCA Series 4 I/O User's Guide* and TN1037 *ORCA Series 4 Fast DDR Interface.*

Table 15. XGMII DC and AC Specifications

Parameter	Condition	Min.	Typ.	Max.	Units
VDDIO	—	1.4	1.5	1.8	V
VREF	—	0.68	0.75	0.9	V
VTT[1]	—	-	0.75	-	V
VIH	—	V_{REF} + 100mV	0.85	V_{DDIO} + 0.3	V
VIL	—	-0.3	0.65	V_{REF} - 100mV	V
VOH[2]	IOH > 8mA	1	1.1	—	V
VOL	IOL > -8mA	—	—	0.4	V

1. 50% V_{DDIO}
2. V_{DDIO} - 400mV

Figure 12. HSTL1 Circuit Topology

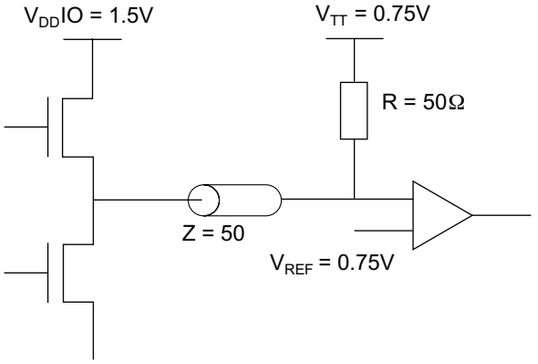
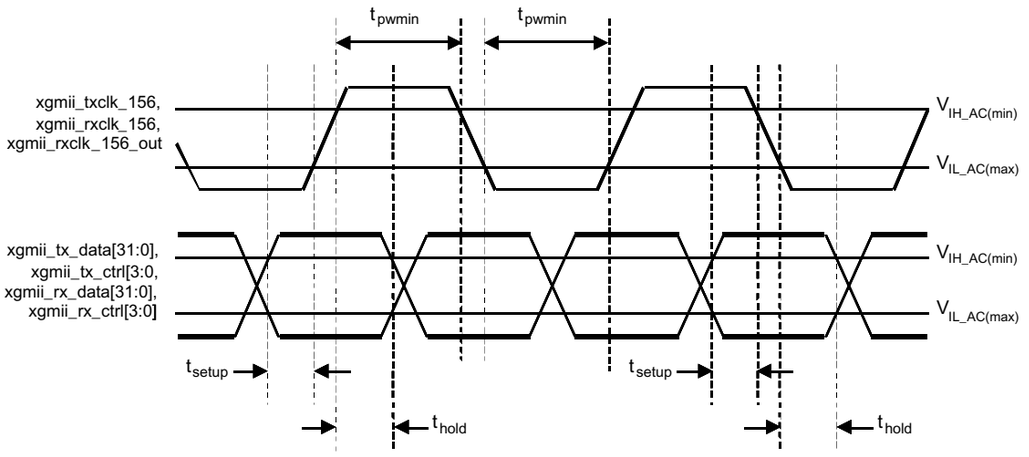


Figure 13. XGMII Timing Parameters



Symbol	Driver	Receiver	Units
t_{SETUP}	960	480	ps
t_{HOLD}	960	480	ps
t_{PWMIN}	2.5	—	ns

XAUI Specifications

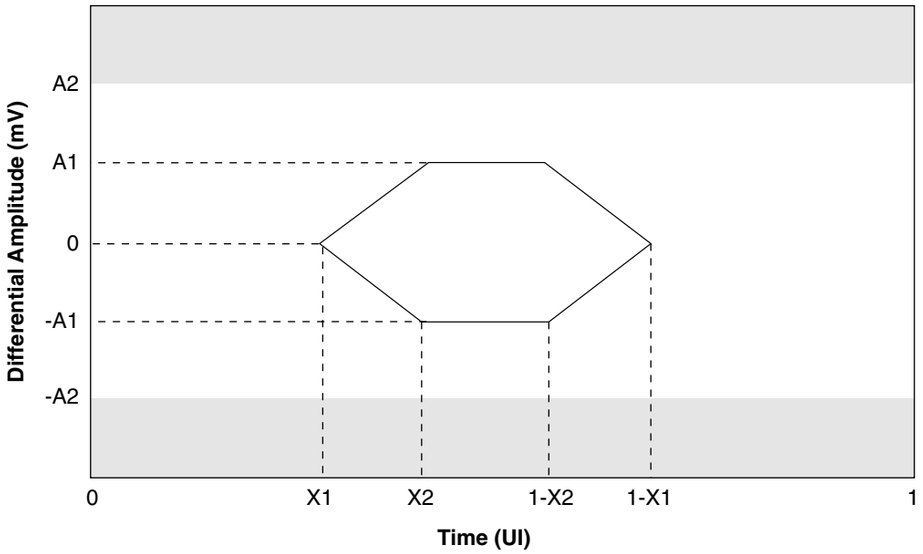
The electrical characteristics of the XGXS XAUI signals conform to Clause 47.3 of IEEE 802.3ae. The general XAUI driver requirements are given in Table 16. The XAUI driver template and receiver characteristics are given in Figure 14 and Table 17, respectively. The XAUI SERDES interface circuit topology is shown in Table 15. Complete specifications on the SERDES may be found in the ORT82G5 Data Sheet.

Table 16. XAUI Driver Characteristics

Parameter	Value	Units
Baud rate tolerance	3.125Gbd (100ppm	GBd ppm
Unit interval nominal	320	ps
Differential amplitude maximum	1600	mVp-p
Absolute output voltage limits Maximum Minimum	 2.3 -0.4	 V V
Differential output return loss minimum	See Equation 1[1]	dB
Output jitter Near-end maximums Total jitter Deterministic jitter Far-end maximums Total jitter Deterministic jitter	 ± 0.175 peak from the mean ± 0.085 peak from the mean ± 0.275 peak from the mean ± 0.185 peak from the mean	 UI UI UI UI

1. Equation 1: s_{11} = -10dB for 312.5MHz < Freq(f) < 625MHz, and
-10 + 10log (f/625)dB for 625MHz ≤ Freq(f) ≤ 3.125GHz

Figure 14. XAUI Driver Template



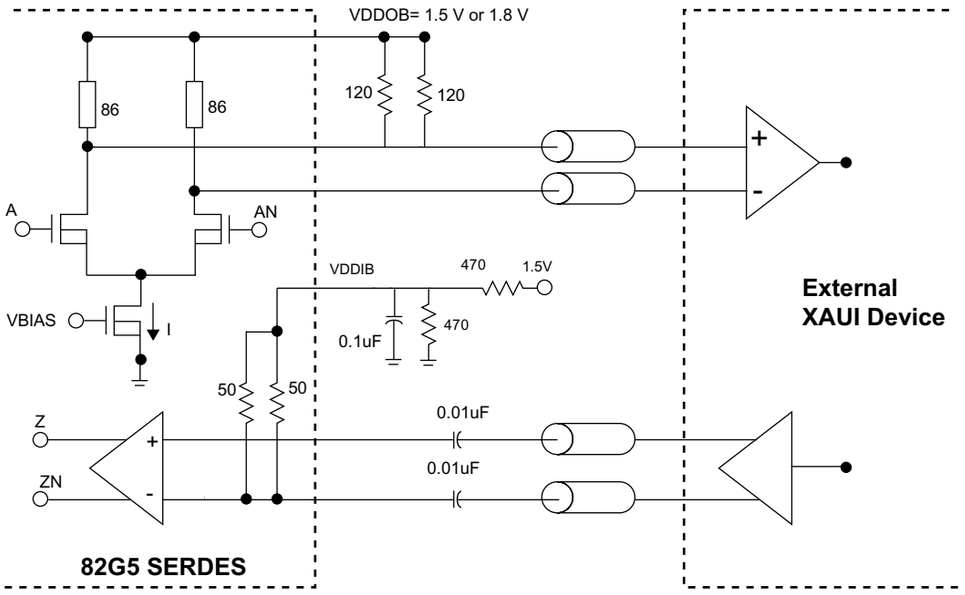
Symbol	Near-end Value	Far-end Value	Units
X1	0.175	0.275	UI
X2	0.390	0.400	UI
A1	400	100	mV
A2	800	800	mV

Table 17. XAUI Receive Characteristics

Parameter	Value	Units
Baud rate Tolerance	3.125 ±100ppm	GBd ppm
Unit interval (UI) nominal	320	ps
Receiver coupling	AC	—
Return loss[1] Differential Common mode	 10 6	 dB dB
Jitter amplitude tolerance[2]	0.65	UI_{p-p}

1. Relative to 100Ω differential and 25Ω common mode. See IEEE 802.3ae Sec. 47.3.4.5 for input impedance details.
2. See IEEE 802.3ae Sec. 47.3.4.6 for jitter tolerance details.

Figure 15. XAUI SERDES Interface Circuit Topology



MDIO Specifications

The electrical specifications of the MDIO signals conform to Clause 45.4 of IEEE 802.3ae.

Figure 16. MDIO Timing

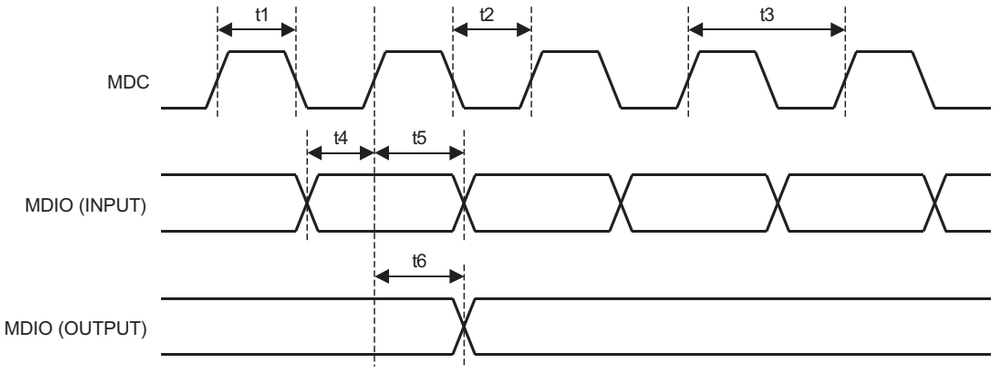


Table 18. MDIO Interface Timing

Symbol	Description	Min.	Max.	Units
t1	MDC high pulse width	200	—	ns
t2	MDC low pulse width	200	—	ns
t3	MDC period	400	—	ns
t4	MDIO(I) setup to MDC rising edge	10	—	ns
t5	MDIO(O) hold time from MDC rising edge	10	—	ns
t6	MDIO(O) valid from MDC rising edge	0	300	ns

Miscellaneous I/O Specifications

Table 19. Miscellaneous Inputs

Signal Name	DC Voltage Levels		Rise/Fall Time		Ref. Clock	Edge (+/-)	Setup Time Min. (ns)	Hold Time Min. (ns)	Internal Pullup
	V_{IL} Max.	V_{IH} Min.	Min. (ns)	Max. (ns)					
reset_n*	0.8	2.0	2	2	asynch	na	na	na	no
pwrup_init_en[1]	0.8	2.0	2	2	asynch	na	na	na	no
inj_err_n[1]	0.8	2.0	2	2	asynch	na	na	na	no

1. External pull-up required.

Clocking Strategies

The general XGXS clocking strategy is discussed in this section. As noted earlier, slip buffers are implemented in both the transmit and receive paths to accommodate for timing differences between the clock domain associated with the SERDES blocks and the clock domain for the XGMII interface.

XGMII Clocks

There are two clocks associated with the XGMII interface.

xgmii_rxclk_156

xgmii_rxclk_156 is a 156.25MHz reference clock used to clock all internal registers in the XGXS receive path. It is also used to generate the xgmii_rxclk_156_out output clock from the XGMII interface. The XGMII outputs, xgmii_rx_data[31:0] and xgmii_rx_crtl[3:0], are synchronous to xgmii_rxclk_156_out.

xgmii_txclk_156

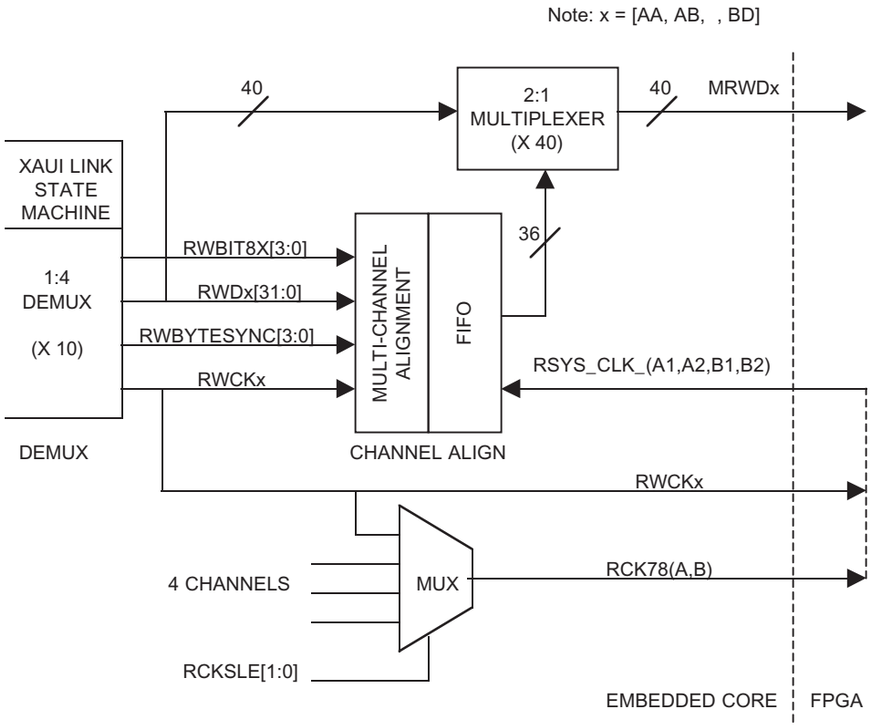
xgmii_txclk_156 is a 156.25MHz clock input for the XGMII interface. The XGMII inputs, xgmii_tx_data[31:0] and xgmii_tx_ctrl[3:0], are synchronous to this clock. A slip buffer compensates for the differences in the xgmii_txclk_156 clock domain and the internal XAUI-based 156.25MHz clock domain.

XGMII timing was shown previously.

Embedded Core Clocks

There are several clocks associated with the interface to the SERDES block. Only SERDES block B is currently used by the XGXS IP core. Figure 17 and Figure 18 show the transmit and receive interfaces between the FPSC FPGA core and the embedded core. Only the clock related signals are discussed here. For additional discussion, see the ORT82G5 Data Sheet.

Figure 17. Embedded Core <-> FPGA Receive Interface



REFCLKP_[A:B], REFCLKN_[A:B]

REFCLKP_[A:B] and REFCLKN_[A:B] are differential reference clocks provided to the SERDES block in the ORT82G5 device. There is a reference clock for each quad SERDES block in the design and it is used as the reference clock for both TX and RX paths. For the XAUI interface, these reference clocks are 156.25MHz.

RWCK[AA:BD]

RWCK[AA:BD] are the low-speed clocks from the embedded core to the FPGA across the core-FPGA interface. These clocks are derived from the recovered low-speed complementary clocks from the SERDES blocks. RWCK_BA belongs to Channel BA; RWCK_BB belongs to channel BB and so on. With a reference clock input of 156.25MHz and full rate mode, these clocks operate at 78MHz.

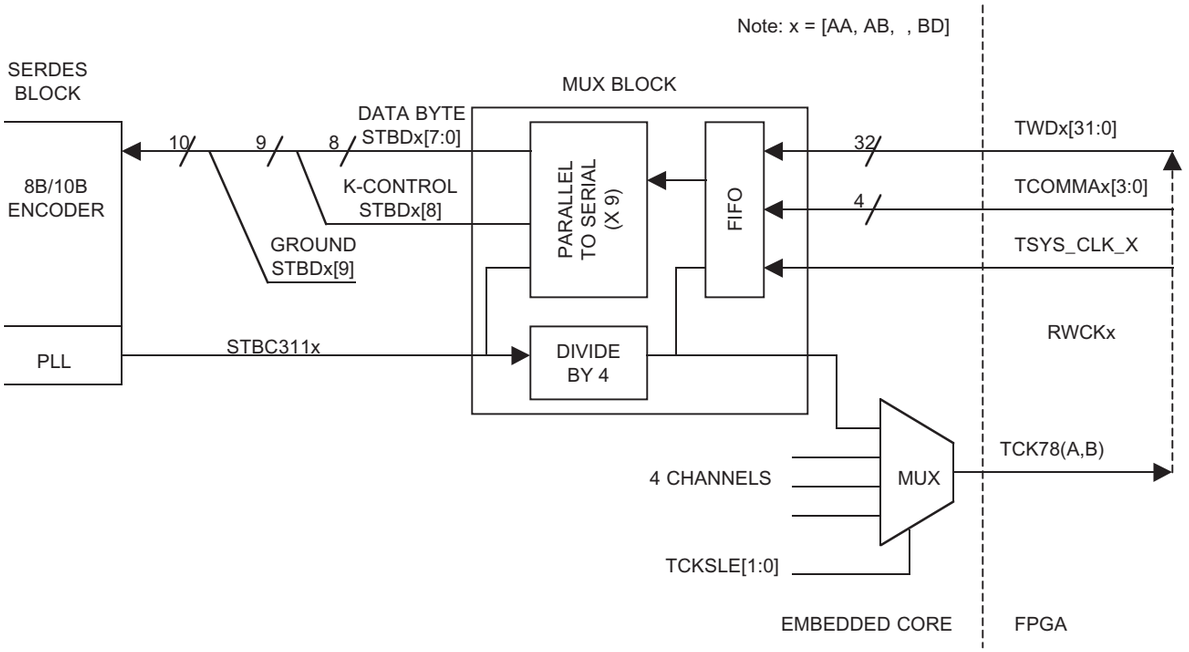
RCK78[A:B]

RCK78[A:B] are muxed outputs of RWCKA[A:D] and RWCKB[B:D] respectively. With a reference clock input of 156.25MHz, these clocks operate at 78MHz. For the XGXS IP core, the mux is set such that RCK78B is supplied by RWCKBA.

RSYS_CLK_[A:B][1:2]

RSYS_CLK_[A:B][1:2] are inputs to the SERDES quad block A and B respectively from the FPGA. These are used by each channel as the read clock to read data from the alignment FIFO within the embedded core. Clocks RSYS_CLK_A[1:2] are used by channels in the SERDES quad block A and RSYS_CLK_B[1:2] by channels in the SERDES quad block B. To guarantee that there is no overflow in the alignment FIFO, it is a requirement that the write and read clocks are synchronous. The recommended clocking strategy is to use RCK78B from the core and feed it to RSYS_CLK_B[1:2].

Figure 18. Embedded Core <-> Transmit Interface



TSYS_CLK[AA,...BD]

TSYS_CLK[AA,...BD] are inputs to the SERDES quad block A and B respectively from the FPGA. These clocks are used by each channel to control the timing of the Transmit Data Path. The recommended clocking strategy is to use TCK78B from the core and feed it to TSYS_CLK_BA, TSYS_CLK_BB, TSYS_CLK_BC and TSYS_CLK_BD.

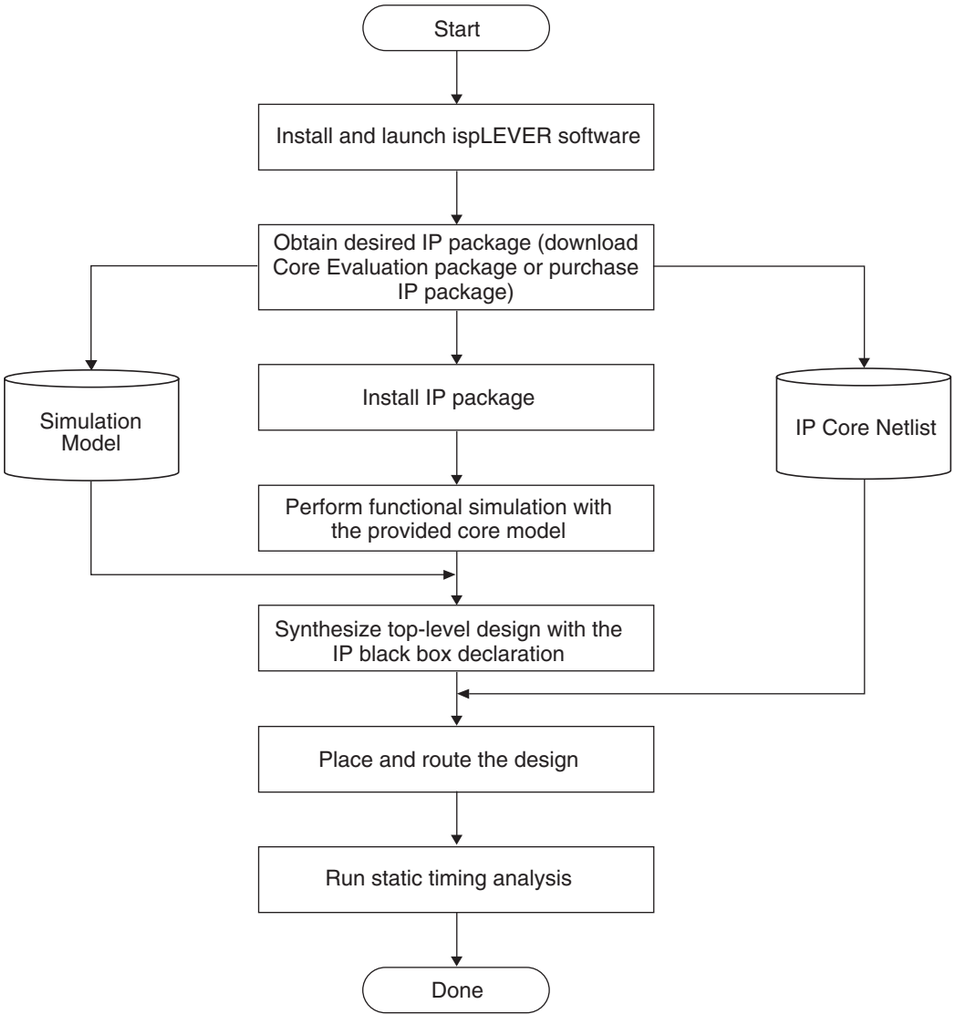
TCK78[A:B]

Each of TCK78[A:B] is a muxed output of one of the four clocks operating at up to 78MHz in the embedded core to the FPGA across the core-FPGA interface. There is one clock output per SERDES quad block. TCK78B is wired to TSYS_CLK_[BA-BD]. For the XGXS IP core, the TCK78B mux should be set such that TCK78B is supplied by the clock source for channel BA.

XGXS Core Design Flow

The XGXS IP Core can be implemented using various methods. Figure 19 illustrates the software flow model used when evaluating with the XGXS-XGMII core.

Figure 19. Lattice IP Core Evaluation Flow



Functional Simulation under ModelSim (PC Platform)

Once the XGXS core has been downloaded and unzipped to the designated directory, the core is ready for evaluation. The RTL simulation environment contains a testbench and a simple application that uses the XGXS design. The application instantiates the XGXS core, an ORCA ORT82G5 module and an ORCA SYSBUS module. The module name of the application is called "xgxs_top_xgmii". The testbench includes a basic XGMII driver, a SMI driver, and an instantiation of the "xgxs_top_xgmii" application. The SERDES outputs are looped externally back to the SERDES inputs in this testbench. The XGMII driver sends 25 normal CRPAT packets. The reception of packets is monitored internally by the pattern monitoring function and is queried using the SMI interface. The XGMII driver then sends 30 CRPAT packets of which 20 are errored. These are monitored and also queried through the SMI interface. The SMI interface is used to control the FPSC/FPGA application registers and to query results stored there and it reports to the ModelSim transcript.

A simulation script file is provided in the "eval\simulation" directory for RTL simulation. The script file run_eval.bat uses precompiled models provided with this package. The XGXS design and testbench models have been compiled into the work directory in directory "eval\simulation". The ORCA ORT82G5 and SYSBUS models have been provided in the directory "eval\lib\modelsim" as tar.gz archives (ort82g5_work.tar.gz and sysbus_work.tar.gz). These files should be unarchived into the directory "eval\lib\modelsim" creating two compiled work directories "eval\lib\modelsim\ort82g5_work and "eval\lib\modelsim\sysbus_work".

Simulation Procedures

1. Go to directory "eval\simulation".
2. Type run_eval.bat

For more information on the use of ModelSim, please refer to the *ModelSim User's Manual.* Note that the pre-compiled ORT82G5 simulation models provided in this IP evaluation package do not work with the OEM version of ModelSim embedded in the ispLEVER® software. The full, licensed version of ModelSim is required to run this simulation.

Core Implementation

Lattice's XGXS evaluation package includes a XGXS user application and scripts for synthesizing, mapping and routing the XGXS IP solution.

The XGXS evaluation package includes the following components:

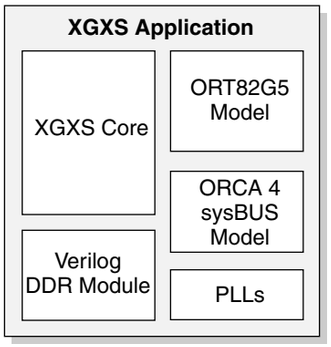
- Basic XGXS IP core, including SMI and packet generator/checker functions;
- Verilog module that instantiates the ORT82G5 component;
- Verilog module that instantiates the ORCA4 SYSBUS with User Master component, providing a Motorola Power PC interface to the core's register interface;
- Verilog module that instantiates the DDR interface components.

This evaluation package is illustrated in Figure 20. The following Verilog files are provided:

- xgxs_define.v for XGXS parameters (**Note: This file and all IP parameter files must not be modified in any way. If this file is modified, this IP core may not run at specification)**;
- xgbe_xgxs_o4_1_002.v for the XGXS core;
- xgmii_io_if.v for the DDR interface;
- ORT82G5_INTF.v for the ORT82G5 module;
- xgxs_sysbus.v for the SYSBUS module;
- xgxs_clk_tx.v for the Tx Clk PLL;
- xgxs_clk_mx2.v for FPGA/FPSC PLLs;
- ring_osc.nmc for a ring oscillator macro to drive um_clk;
- xgxs_top_xgmii.v for top-level module that ties all the application components together.

The XGXS Core is delivered as a gate-level netlist (xgbe_xgxs_o4_1_002.ngo). Note that this file and all IP parameter files must not be modified in any way. If this file is modified, this IP core may not run at specification. Users can compile the entire design shown in Figure 20 to realize a turnkey solution, or instantiate the XGXS Core as a block box together with any of the other blocks shown and/or their own designs, to realize a unique system-level project. Users may use xgxs_top_xgmii.v as a template for their own application.

Figure 20. XGXS Top-Level Application



Implementing a design in an ORT82G5 device requires Lattice ispLEVER software and an ORT82G5 FPSC Design Kit. For more information, contact your local Lattice sales representative or visit the Lattice web site at www.latticesemi.com.

Black Box Considerations

Since the core is delivered as a gate-level netlist, the synthesis software will not re-synthesize the internal nets of the core. For more information regarding Synplify's black box declaration, please refer to the Instantiating Black Boxes in the Verilog section of the *Synplify Reference Manual.*

Synthesis

The following sections provide procedures for synthesizing the XGXS IP solution with the Synplicity Synplify and Leonardo Spectrum synthesis tools, which are included in Lattice's ispLEVER software. These procedures generate an EDIF netlist containing the XGXS core as a black box.

Synthesis Using Synplicity Synplify

To synthesize the XGXS solution using Synplicity Synplify in one step, go to the directory "eval\synthesis\orca4\synplicity\config1" and enter "runsyn.bat". A top-level EDIF for the application will be produced. Users may use runsyn.bat as a guide and template if they are creating their own unique system-level project.

The following step-by-step procedure may also be executed. Note that results may vary from those obtained with the scripted run due to small differences in options.

1. Create a new working directory for synthesis.

2. Launch the Synplify synthesis tool.

3. Start a new project and add the specified files in the following order:
   eval\source\synplicity\orca4_synplify.v
   eval\source\xgxs_define.v
   eval\source\ring_osc.v
   eval\source\synplicity\xgxs_clk_mx2.v
   eval\source\synplicity\xgxs_clk_tx.v
   eval\source\synplicity\xgxs_sys_bus.v
   eval\source\synplicity\ORT82G5_INTF.v
   eval\source\synplicity\xgmii_io_if.v
   eval\source\xgbe_xgxs_o4_1_002.v
   eval\source\synplicity\xgxs_top_xgmii.v
   eval\synthesis\orca4\synplicity\config1\xgxs_top.sdc

4. In the Implementation Options select target device Lattice ORCA Series 4.

5. Specify an EDIF netlist filename and EDIF netlist output location in the Implementation Options. This top-level EDIF netlist will be used during Place and Route.

6. In the Implementation Options, set the following:
   Fanout guide: 1000
   Enable FSM Compiler
   Set the global frequency constraint to 160MHz.

7. Select run.

Synthesis Using LeonardoSpectrum
To synthesize the XGXS solution in LeonardoSpectrum in one step, go to the directory "eval\synthesis\orca4\exemplar\config1" and enter "runsyn.bat". A top-level EDIF for the application will be produced. Users may use runsyn.bat as a guide and template if they are creating their own unique system-level project.

The following step-by-step procedure may also be executed. Note that results may vary from those obtained with the scripted run due to small differences in options.

The step-by-step procedure provided below describes how to run synthesis using LeonardoSpectrum.

1. Create a new working directory for synthesis.

2. Launch the LeonardoSpectrum synthesis tool.

3. Start a new project and select Lattice device technology ORCA-4E.

4. Select Input tab, set the Working Directory path pointed to the source directory.

5. Open the specified files in the following order:
   eval\source\exemplar\orca4_leonardo.v
   eval\source\xgxs_define.v
   eval\source\ring_osc.v
   eval\source\exemplar\xgxs_clk_mx2.v
   eval\source\exemplar\xgxs_clk_tx.v
   eval\source\exemplar\xgxs_sys_bus.v
   eval\source\exemplar\ORT82G5_INTF.v
   eval\source\exemplar\xgmii_io_if.v
   eval\source\xgbe_xgxs_o4_1_002.v
   eval\source\exemplar\xgxs_top_xgmii.v

6. Select "xgxs_top.xgmii.v". Click the right mouse button and select "Make xgxs_top_xgmii.v Top of the Design" from the list.

7. In the Constraints tab, set Clock Frequency as 156MHz.

8. Set the synthesis directory, created in step 1, as the path where you would like to save the output netlist.

9. Specify an EDIF netlist filename for the output file. This top-level EDIF netlist will be used during Place and Route.

10. Click on the Advanced Flow Tabs icon.
    a. In the Technology tab, check the Manual GSR box and fill "reset_n" in the Signal entry box.
    b. In the Input Tab, specify "eval/source" directory in the Input File Search text box.

11. Select Run Flow

Place and Route

Once the EDIF netlist is generated, the next step is to map, place and route the design. To map, place and route the entire XGXS solution in one step, Go to the directory "par\orca4\config1" and type runpar.bat. This script will process the EDIF file to map, place and route the XGXS IP solution.

The step-by-step procedure provided below may also be followed. Note that results may vary from those obtained with the scripted run due to small differences in options.

Once the EDIF netlist is generated, import the EDIF into the Project Navigator. The ispLEVER software automatically detects the provided EDIF netlist of the instantiated IP core in the design. The step-by-step procedure provided below describes how to perform Place and Route in ispLEVER for an ORCA device:

1. Create a new working directory for Place and Route.
2. Start a new project, assign a project name and select the project type as EDIF.
3. Select the ORT82G5 target device, with -3 speed grade and BM680 package.
4. Copy the following files to the Place and Route working directory:
   a) eval\par\xgbe_xgxs_o4_1_002.ngo
   b) eval\par\xgxs_top_exemplar.prf if the EDIF was generated using LeonardoSpectrum or eval\par\xgxs_top_synplicity.prf if the EDIF was generated using Synplicity Synplify
   c) eval\source\ring_osc.nmc
   d) The top-level EDIF netlist generated from running synthesis
5. Rename the .prf file (in step 4) to match the project name. For example, if the project name is "demo", then the .prf file must be renamed to demo.prf. The preference file name must match that of the project name.
6. Import the EDIF netlist into the project.
7. In the ispLEVER Project Navigator, select Tools->Timing Checkpoint Options. The Timing Checkpoint Options window will pop-up. In both Checkpoint Options, select Continue.
8. In the ispLEVER Project Navigator, highlight Place & Route Design, with a right mouse click select Properties. Set the following Properties:

- Placement Iterations: 1
- Placement Save Best Run: 1
- Placement Iteration Start Point: 1 if the EDIF was generated using Synplify, or 1 if the EDIF was generated using LeonardoSpectrum
- Routing Resource Optimization: 1
- Routing Delay Reduction Passes: 6
- Routing Passes: 25
- Placement Effort Level: 5

All other options remain at their default values.

9. Select the Place & Route Trace Report in the project navigator to execute Place and Route and generate a timing report for ORCA.
10. Highlight Place & Route TRACE Report, with a right mouse click and select Force One Level. A new timing report is generated.

Note that timing results may change under different versions of synthesis tools or releases of ispLEVER. If this is the case, multiple placement iterations would need to be run to achieve zero timing errors. Multiple placement iterations are run by increasing the "Placement Iterations" value.

Reference Information

The XGXS IP core solution is compliant with IEEE 802.3ae except where specifically noted. A complete description of XGXS functionality is given in the specification document.

Additional information on implementing this solution is contained in the following documents:

- ORCA ORT42G5 and ORT82G5 Data Sheet
- ORCA Series 4 FPGAs Data Sheet

These documents are available on the Lattice Semiconductor web site at www.latticesemi.com.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Lattice ***manufacturer:***

Other Similar products are found below :

RAPPID-560XBSW RAPPID-567XFSW DG-ACC-NET-CD SRP004001-01 SW006021-1NH SW163052 SYSWINEV21 Core429-SA SW500006-HPA CWP-BASIC-FL W128E13 CWP-PRO-FL SYSMACSE210L SYSMACSE203L AD-CCES-NODE-1 NT-ZJCAT1-EV4 CWA-BASIC-FL RAPPID-567XKSW CWA-STANDARD-R SW89CN0-ZCC CWA-LS-DVLPR-NL VDSP-21XX-PCFLOAT RAPPID-563XMSW IPS-EMBEDDED SWR-DRD-L-01 SDAWIR-4532-01 SYSMAC-SE201L MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR LIB-PL-PC-N-1YR-DISKID SYSMACSE2XXL LS1043A-SWSP-PRM 1120270005 1120270006 MIKROBASIC PRO FOR FT90X (USB DONGLE) MIKROC PRO FOR AVR (USB DONGLE LICENSE) MIKROC PRO FOR FT90X (USB DONGLE) MIKROBASIC PRO FOR AVR (USB DONGLE LICEN MIKROBASIC PRO FOR FT90X MIKROC PRO FOR DSPIC30/33 (USB DONGLE LI MIKROC PRO FOR FT90X MIKROC PRO FOR PIC32 (USB DONGLE LICENSE 52202-588 MIKROPASCAL PRO FOR ARM (USB DONGLE LICE MIKROPASCAL PRO FOR FT90X MIKROPASCAL PRO FOR FT90X (USB DONGLE) MIKROPASCAL PRO FOR PIC32 (USB DONGLE LI SW006021-2H