REQUIREMENTS AND FUNCTIONAL SPECIFICATION

GBit Ethernet Chip V2

RFS Document: **A25092N0001**

Revision: 1.6

Brent Carlson, October 15, 2010

National Research Council Canada
Herzberg Institute of Astrophysics
Dominion Radio Astrophysical Observatory

P.O. Box 248, 717 White Lake Rd
Penticton, B.C., Canada
V2A 6K3

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

LTA – Long-Term Accumulator.

FPGA – Field Programmable Gate Array.

DDR – Double Data Rate. The FPGA that will be used for this chip contains DDR inputs, that are special chip functions to sample input data on the rising and falling edges of a clock.

MCB – Monitor and Control Bus. The on-circuit-board synchronous read/write bus for configuring and monitoring the operation of chips.

LVTTL – Low-Voltage TTL. A chip-to-chip voltage level standard.

CRC – Cyclic Redundancy Check. This is a code calculated on a serial bit stream using a "generator polynomial". Calculation at some other point in the system on the same bit-stream, should yield the same results otherwise a bit error has been introduced into the stream. The longer the generator polynomial, the more capable the calculation is of detecting errors. For Ethernet, a CRC-32 code is used, guaranteeing nearly 100% error detection.

PCML – Pseudo Current Mode Logic. Drivers and levels for very high speed signals.

SFP – Small Form-factor Pluggable. This is a small module that plugs into the Baseline Board, and provides the physical layer interface (PHY) from the FPGA GigE lines to the cable, which can be copper or fiber.

UDP/IP – User Datagram Protocol, over Internet Protocol. This is the protocol used on the Gbit/sec Ethernet output. It is a connectionless protocol that does not guarantee that packets are delivered, however it has significant performance and simplicity features that make it the protocol of choice for this application.

1000BASE-X auto-negotiation – The process whereby the Ethernet link transmission abilities are negotiated. If 1000BASE-X fiber is used, then it is auto-negotiation with the far end. If 1000BASE-T copper is used, then it is auto-negotiation with the SFP copper module, and the SFP copper module handles 1000BASE-T auto-negotiation with the far end transparently to the FPGA.

XAUI – For some 10 Gbit Ethernet implementations (including this one), this is a 4 wire pair, 3.125 Gbps each, which contains 4 byte lanes of data travelling from the FPGA to the transponder (XPAK) module. XAUI stands for "10 Gbit Attachment Unit Interface".

XGMII – This is the "10 Gbit Media Independent Interface". For this implementation, it is the interface in the FGPA between the user logic, and the hard-block transceiver with XAUI output. XGMII is classically, 4 byte lanes at 312.5 MHz. For the implementation in the Altera Stratix GX, it is 8 bytes lanes at 156.25 MHz.

1 Revision History

Revision	Date	Changes/Notes	Author
DRAFT	January 16, 2008	**V2**: A complete re-design and simplification of the chip, support for phased data packets, and support for 10 GigE: -remove all transmit IP enable registers. -always performs 1 msec round-robin, non-blocking scheduling. -Destination MAC for LTA frames is now set in the LTA. -LTA and LTA interface now runs at 133 MHz. -add support for transmission of phased array packets from RXP chips. -the DUMMY register address is 0xFE, to differentiate it for software from the original design. Address 0xFF is now a non-functional address. -Add registers for "turn-key" 10 GigE operation.	B. Carlson
1.0	June 26, 2009	First release with full design complete. Minor modifications to be consistent with design.	B. Carlson
1.1	Sept 9, 2009	Minor fixes and clarifications. Change some default register values. Add the LCCDR test register. Add details on RXP-to-GigE transmission protocol.	B. Carlson
1.2	Dec 16, 2009	Note in section 5.4.3.7 that only one receive channel per RXP is allowed. Section 5.4.3.20, allow frame capture on SFP2 or SFP1. Add "CC_SPEED" register, section 5.4.3.25. Add section 5.4.5, GigE V2 GUI concept	B. Carlson

1.3	Feb. 9, 2010	Add appendices with pin and package notes	B. Carlson
1.4	Mar. 2, 2010	Remove the PLL-RX and PLL-TX bits from the MCSR. Add the LTABSRR test register at address 0x45. Change the source IP default to be non-zero. Change the default IFDR to 20 (usec).	B. Carlson
1.41	Apr. 13, 2010	Add PAUSE frame handling capability on SFP1, and the PFCount register. Update description of the XOEN bit of the SXTCSR register. No S/W changes strictly required. Add the 'BVer' bit to the MCSR to allow for correct SFP1 status indication for V2.1 and V2.2 boards. **S/W changes to this bit should be implemented for best SFP1 status indication**.	B. Carlson
1.5	Apr. 30, 2010	Add the RAN bit to the MCSR register to allow for randomization of packet transmission times on SFP1. If the RAN bit in the MCSR is set, the IFDR register is in units of 4 usec, allowing for over 1 millisecond of inter-frame delay.	B. Carlson
1.51	May 5, 2010	When in 10 G mode, the IFDR register is in units of 1 usec; when in 1 G mode, the IFDR register is in units of 4 usec.	B. Carlson
1.52	May 25, 2010	MCSR-RAN bit randomization is now uniformly distributed over range of 0 to 2xIFDR delay.	B. Carlson
1.6	October 15, 2010	Add RXP_SMAC-0:5 and RXP_IPADRSRC-0:3 to allow VDIF frames, normally destined for SFP2, to have separates source MAC and IP addresses. Note these registers ONLY available in FPGA binaries dated ≥ October 15, 2010.	B. Carlson

2 Introduction

This document describes detailed requirements, designs, and CPU control/status registers for the Gbit Ethernet Chip FPGA V2 ("Version 2"). It is named "V2" because there are substantial changes and simplification to this chip design warranting a clear designation.

There is one of these FPGAs on every Baseline Board and it is responsible for gathering correlated data from LTA Controllers and phased data from RXP FPGAs, and transmitting them on the Gbit Ethernet serial output to the Correlator Back End and, for phased-array data, ultimately a VLBI data recorder.

V2 is vastly simplified compared to the original design, eliminating many of the features of the original design that were not found to be useful. V2 supports full-speed operation of the LTA at 133 MHz to enable full recirculation performance requirements (200 μsec dumps on all CCCs' lag frames). The design supports routing of phased data packets from RXP chips through to SFP module output, and contains the complete register set to support future 10 GigE operation.

The RFS specification for communication with the LTA Controller can be found in [1], and background information on the system design can be found in [2].

3 Context

A single Gbit Ethernet chip resides on the Baseline Board. The chip is used to gather data from the LTA Controllers, and transmit them on port SFP1 using UDP/IP to an external switch on standard Gbit/sec Ethernet for routing to a destination Correlator Back End (CBE) CPU. Phased data packets from Upper and/or Lower RXP FPGAs are transmitted on SFP2, and alternatively may be routed to SFP1. The Gbit Ethernet chip is implemented in an Altera Stratix GX-25 FPGA that has special functions to allow for Gbit Ethernet transmission. A simplified diagram of the critical connections of this chip to other devices and systems is shown in Figure 3-1 below.

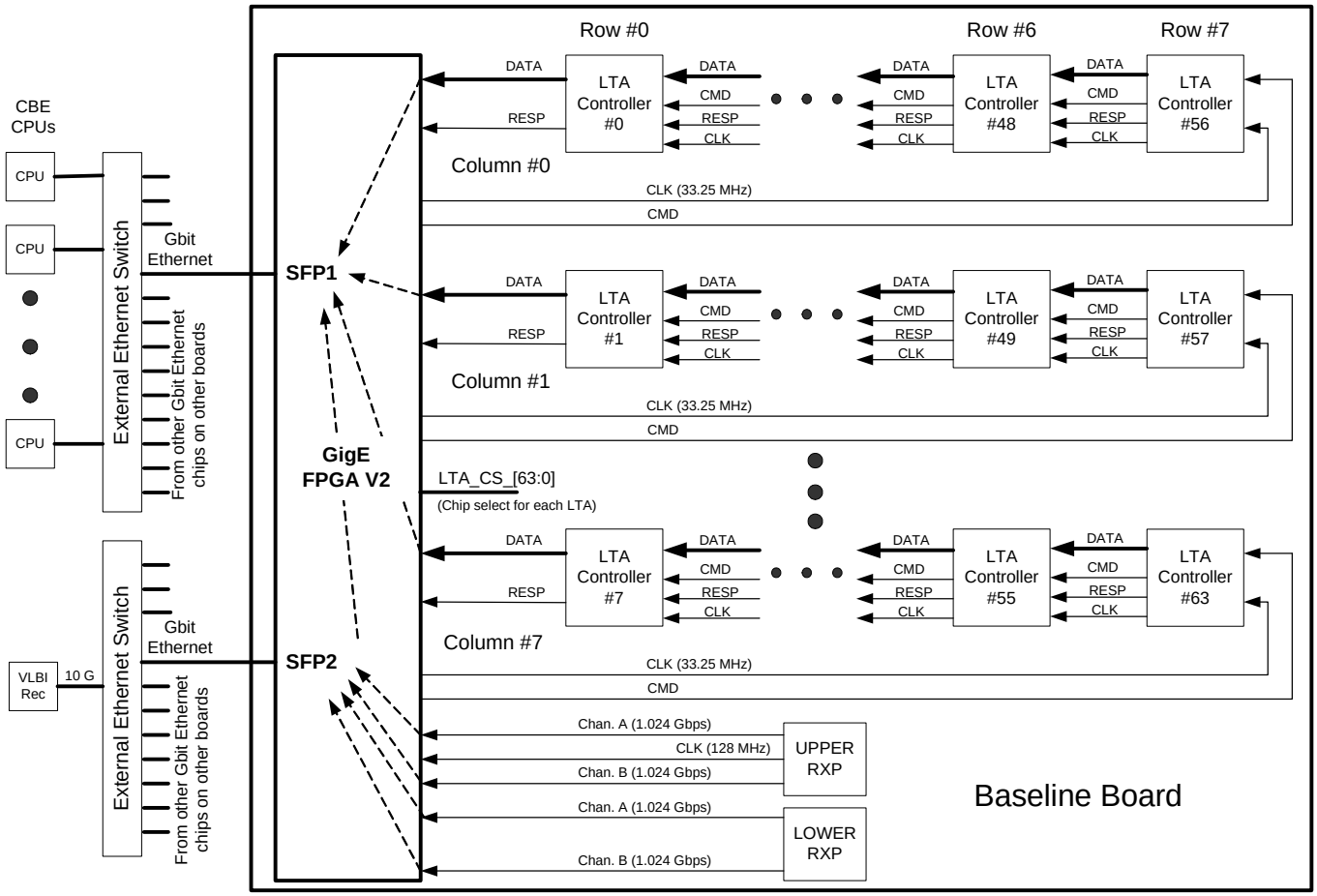


Figure 3-1 Simplified block diagram of the Gbit Ethernet chip's external world connections. The chip controls the routing of lag frames from the 64 LTA Controllers on the Baseline Board, and phased data packets from Upper and Lower RXP chips to the CBE and VLBI data recorder using UDP/IP on Gbit/sec Ethernet.

Not shown is the fact that the FPGA and board connections allow for an upgrade path to 10 Gbps Ethernet on XPAK. In this case, all packets are routed on the 10 Gbps connection.

4 Overview

A simplified block diagram of the Gbit Ethernet chip is shown in Figure 4-1.

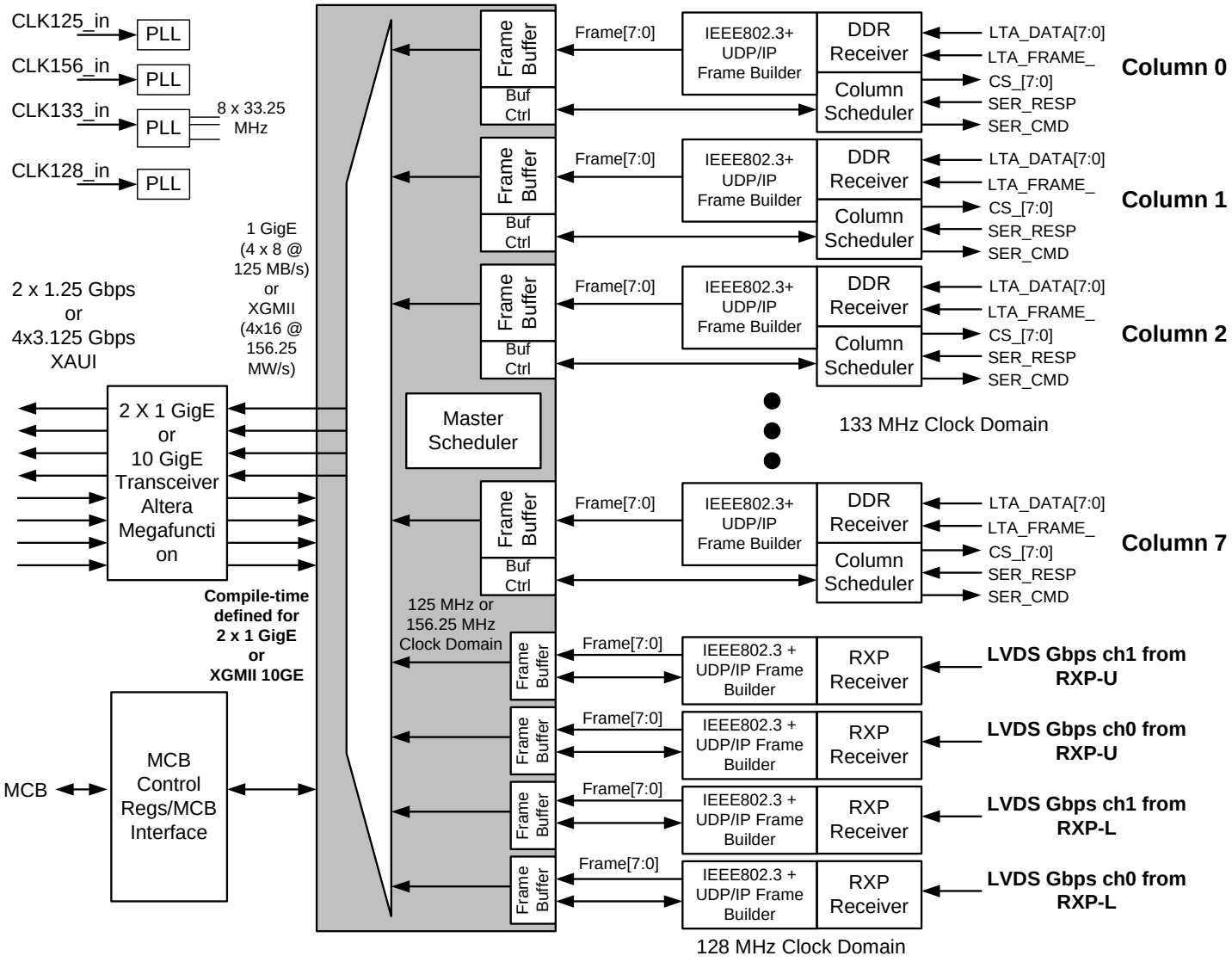


Figure 4-1 Simplified block diagram of the Gbit Ethernet chip. The chip talks to 8 columns of LTA Controllers simultaneously, and performs transfers of data to internal frame buffers. The Master Scheduler decides column's frame is to be transmitted in a round-robin fashion. The Ethernet Frame Builders generate Ethernet frames, complete with CRC-32, and the final serial output is produced with specialized FPGA functions. Provision is made in the design for a straightforward upgrade path to 10 G Ethernet.

The chip consists of several separate but interconnected blocks. A brief description of the operation of each of these blocks is as follows:

- **DDR Receiver**. This block contains a specialized FPGA function that samples the input on both the rising and falling edges of the internal 133 MHz clock, and

saves both samples in a frame buffer. The frame buffer which contains an error-free frame (by virtue of the LTA frame checksum), is used for further processing.

- **Column Scheduler**. This block visits each of the LTAs in the particular column in a round-robin fashion, looking for and downloading data when available.

- **IEEE 802.3+UDP/IP Frame Builder**. Under control of the Column Scheduler, this block builds the captured LTA frame into an IEEE 802.3 UDP/IP frame, stuffs it into an output Frame Buffer, readied for transmission by the Master Scheduler.

- **Master Scheduler**. This block, in a round-robin fashion, transmits Ethernet frames in Frame Buffers on SFP1 and/or SFP2 sourcing from LTAs or RXP FPGAs.

- **RXP Receiver.** This block receives phased data frames from an RXP FPGA, and saves them in an internal frame buffer. These frames are built into IEEE 802.3 UDP/IP frames and transmitted on SFP2, or, if selected, also transmitted on SFP1.

It is important to note that the Gbit Ethernet chip does not contain Ethernet receiver functions, and that it transmits UDP/IP frames only. Thus, there is no buffering, stack, or retransmission.

4.1 LTA Controller Interface Signals

There are 8 columns of 8 LTA Controllers; all LTAs within a column are daisy-chained and together, and share a final data transmission path to the GigE FPGA. **All of the signals on this interface are SSTL2-C1**.

- LTA_CS_pad_[63:0] – These are active-low chip selects that select one LTA Controller in each column at a time (with columns acting independently of each other) for active communication. These are connected in row order (e.g. LTA_CS_[0] is connected to the Row(Y)#0,Column(X)#0 LTA, LTA_CS_[1] is connected to the Row#0,Column#1 LTA etc.)

- LTA_SER_CMD_pad[7:0] – These are serial command lines, one for each column of LTA Controllers. In each column, only the LTA Controller with an active chip select (LTA_CS_pad_) will respond to signaling on this line with LTA_SER_RESP_pad. Details of the commands that can be transmitted on this line can be found in [1]. This signal operates at 133 Mbits/sec.

- LTA_SER_RESP_pad[7:0] – These are serial response lines, one for each column. The lines contain responses to commands sent on LTA_SER_CMD_pad, and only one LTA in each column is actively responding at a time. Details of these responses can be found in [1]. This signal operates at 133 Mbits/sec.

- LTA_DATA0_pad[7:0], LTA_DATA1_pad[7:0], …, LTA_DATA7_pad[7:0] – These 8, byte-wide lanes, one set for each column, contain frame data from the selected LTA Controller only if the associated LTA_FRAME_pad_ is low. Data transmission will only occur after a transmit request is issued on LTA_SER_CMD_pad. The format of this data frame is described in detail in [1].

- LTA_FRAME_pad[7:0] – These lines, one for each column, when low indicates that there is valid data on the associated LTA_DATAn_pad[7:0]. When it goes high, it indicates that frame transmission is complete. Refer to the functional timing diagram in [1].

- LTA_SER_CMD_RET_pad[7:0] – Returned ("boomeranged") LTA_SER_CMD_pad signal from the LTA columns. These signals are not used in the FPGA.

4.2 MCB Interface Signals

All signals on this interface are 2.5V LVTTL.

- MCB_ADDR_pad[7:0] – 8 address lines for selecting internal registers for microprocessor configuration of the chip.

- MCB_DATA_pad[7:0] – 8 bi-directional data lines for reading and writing from/to the chip.

- MCB_CS_pad_ – Active-low chip select for microprocessor access.

- MCB_RDWR_pad_ – Controls whether data is read from or written to the chip.

- MCB_CLK_pad – 33 MHz clock for synchronous read/write access on the MCB interface.

4.3 Gbit Ethernet Interface Signals

These are **1.5 V PCML** I/Os that use specific pins on the chip for this purpose. There are 4 pairs, and only the first 2 pairs—for SFP1 and SFP2—are active in 2x1 GigE mode. When in 10 G mode, all 4 pairs are active, and running at 3.125 Gbps each.

- GBIT_TX_pad[3:0] – 4 transmit pairs, routing to SFP1-SPF4 (notes that SFP3 and SFP4 are not installed on the board, instead the XPAK module cage is; if SFP3 and SFP4 are required, the XPAK cage and connector can be removed as all circuitry is in place to enable use of SFP3 and SFP4.) and the XPAK connector. Pair [0] is for SFP1. Pair [1] is for SFP2. If XPAK is used (10 G mode), all four pairs are used for the XAUI interface.

- GBIT_RX_pad[3:0] – 4 receiver pairs routing from SFP1-SFP4, and the XPAK connector. Pair associations are identical to the above transmit pairs.

4.4 Clocks

- CLK125_in_pad – This is a single 125 MHz clock input from a local crystal oscillator. This clock is what all 1 GigE transmit functions on the chip are synchronized to. This clock is unused in 10 G mode, except for maintaining the correct frequency on the JTAG_CCRESET_TCK_pad output. **This clock input is 2.5V LVTTL**.

- CLK133_in_pad – This is a single 133 MHz clock input from a local crystal oscillator. This clock is divided down to 33.25 MHz, with one clock provided to each column of LTAs so they can operate at their maximum rate of 133 MHz. **This clock input is 2.5V LVTTL.**

- CLK156_in_pad – This is a single 156.25 MHz clock input from a local crystal oscillator. This clock is used only in 10 G Ethernet mode. **This clock input is LVDS.**

- CLK128_in_pad – This is a 128 MHz clock from the Upper RXP FPGA, and is synchronized to RXP 1.024 Gbps transmit data. **This clock input is LVDS.**

- CLK3125_out_pad[7:0] – 8 x 33.25 MHz clocks out[1]– These 8 clocks are ¼ the rate of the 133 MHz clocks, and are used to provide the columns of LTA Controllers with clocks for deriving their own 133 MHz clocks. These clocks are edge-aligned to the internal 133 MHz clock so that the LTA Controllers can properly synchronize to the LTA_SER_CMD_pad lines. **These clock outputs are SSTL2-C1**.

4.5 RXP Interface Signals

All signals on this interface are LVDS, DC-coupled.

- fromPHASING_UPPER_pad[1:0] – 2 independent LVDS lines running at 1.024 Gbps each, sourcing from the Upper RXP, phase synchronous with the CLK128_in_pad clock. Due to memory restrictions in the FPGA, only line [1] is actively received by the GigE FPGA.

- fromPHASING_LOWER_pad[1:0] -- 2 independent LVDS lines running at 1.024 Gbps each, sourcing from the Lower RXP, phase synchronous with the CLK128_in_pad clock. Due to memory restrictions in the FPGA, only line [1] is actively received by the GigE FPGA.

- toPHASING_UPPER_XON_pad – Unused output to the Upper RXP, originally intended for flow control.

[1] Naming reflects previous frequency derived from 125 MHz.

- toPHASING_LOWER_XON_pad – Unused output to the Lower RXP, originally intended for flow control.

4.6 Miscellaneous Signals

- RESET_pad_ – Active low, chip asynchronous reset. **2.5V LVTTL level**.

- TP0_pad[0:3] – Four output test pins for connecting to internal logic for testing purposes. All these outputs are low. **These signals are 2.5V LVTTL**.

- Fiber module control signals – These signals[2] are for the monitor and control of up to 4, 1 Gbps SFP fiber modules, or 1, 10 Gbps "XPAK" fiber module. These signals include:

    - SFP_*pad – 1 Gbps SFP fiber module monitor/control lines. In this design, these lines are assigned as follows;

        - SFP_RATE_SEL_pad = 0.

        - SFP_TX_DISABLE_pad = 0, to always enable the transmitter.

        - SFP_RX_LOS_pad[3:0] – These inputs [1:0] are used to determine the state of the front-panel SFP2 [1] and SFP1 [0] LEDs. **Note that in V2.1 Baseline Boards, SFP1 LOS (SFP_RX_LOS_pad[0]) is not connected.**

        - All other SFP_* outputs or I/Os are set high impedance.

    - XPAK_*pad – 10 Gbps XPAK fiber module monitor/control lines. In this design these lines are assigned as follows:

        - XPAK_MOD_DET_pad – Unused in the FPGA.

        - XPAK_LASI_pad – Set high impedance.

        - XPAK_MDIO_pad – Set high impedance.

        - XPAK_RST_pad – Actively resets the XPAK module using logic internal to the FPGA, driven by 156 MHz PLL lock status. 53 msec after PLL lock, this output goes low, otherwise high-z.

        - XPAK_TX_ON_pad – Actively turns on the XPAK transponder using logic internal to the FPGA, driven by 156 MHz PLL lock

[2] In V2.1 modules (S/Ns 2003-2008 inclusive), some of these signals are not available. Refer to ECO-A25300N0029-25080 form more information.

status. 52 µsec after PLL lock, this output goes low, otherwise high-z.

  - XPAK_MDC_pad – Set high impedance.
  - XPAK_PRTAD_pad[4:0] – Set high impedance.

- JTAG_CCRESET_TCK_pad – Nominally 0.947 MHz, but changed slightly using the CC_SPEED register. Frequencies going from lowest speed to highest are 0.893, 0.919, 0.947, 0.977, 1.008 MHz. This is used as the Accel regulator reference frequency, and used to keep the correlator chips from going into internal test mode (i.e. by keeping the correlator chip TCK toggling, and TMS input high).

- JTAG_CCRESET_TMS_pad – This output toggles a couple of times[3] and is then held high. This output is transmitted to all correlator chips' TMS input on the board to keep correlator chips from going into internal test mode.

- SFP1_LED_pad, SFP2_LED_pad, SFP3_LED_pad, SFP4_LED_pad – Drive the 4 front-panel LEDs associated with the SFP or XPAK module. In 1 G mode, SFP1_LED indicates if the SFP1 module is detecting a good signal on its input; similarly for SFP2_LED. SFP3_LED and SFP4_LED are unused. If 10 G mode, then each LED indicates that status of a XAUI receiver lane, indicating the health of the XPAK to FPGA connections.

[3] To overcome a glitch suppression circuit in the buffers on the board that distribute this signal.

5 Requirements

The following is a list of Gbit Ethernet chip requirements.

5.1 Functional Requirements

1. The FPGA must execute automatic round-robin scheduling, non-blocking, with a 1 millisecond timeslice. This operation requires no software support or intervention.

2. At all times, once a bank from an LTA Controller is selected for unloading data from, the bank will be unloaded until it is empty before proceeding to the alternate bank. This action facilitates pulsar phase binning, and burst mode dumping.

3. The destination MAC and destination IP address are transmitted by the LTA in front of the LTA frame. Thus, the first 6 bytes of the transmission is the destination MAC, the next 4 bytes are the destination IP, finally followed by the LTA frame.

4. The chip need only support 4-byte IP addressing (IPv4).

5. It must be possible to set all of the parameters in the IP and UDP headers described in section 5.4.2, as required by the description. The only portion that is automatically set is the destination MAC, destination IP (both transmitted from the LTA), and any on-the-fly-generated fields such as checksums.

6. It must be possible to set the 16-bit source port and 16-bit destination port for the UDP header.

7. It must be possible for the microprocessor to set a dead time between frames of between 0 and 255 usec. This is known as the inter-frame delay.

8. It must be possible for the chip to detect and respond to "XON/XOFF" receive packets for flow control. FURTHER INVESTIGATION REQUIRED TO DETERMINE EXACTLY HOW TO DO THIS.

9. The chip must implement the 1000BASE-X PCS Coding Sublayer (PCS) and Physical Medium Attachment Sublayer (PMA), according to clause 36 of IEEE 802.3-2005. Auto-negotiation support is not required, and the chip will work only with SFP modules that do not use/require auto-negotiation. All of this operation is transparent to the microprocessor.

10. Selection of valid frame data from the LTA Controllers (i.e. selection of which clock edge-sampled data to use) shall be transparent to the microprocessor, and require no microprocessor control. The microprocessor will only be informed via

status registers when both frame buffers report receiver errors (detected with checksums).

11. The chip must maintain the following statistics:

    a. SFP1 port frame count.

    b. SFP2 port frame count.

    c. RXP status: sync status, sync errs, frame detect, frame error.

    d. LTA column error transmission detect.

    e. Toggle status of every LTA column interface signal: LTA_FRAME_, LTA_SER_RESP, LTA_DATA[7:0].

12. Must be able to perform a global enable/disable for acquiring frames from LTA Controllers.

13. Must implement at “discard frames” function where data from LTAs is still flowing, but frames go into the bit bucket.

5.2 Performance Requirements

1. The chip has 3 clock domains: 125 MHz (156.25 MHz) for GigE (10 GigE) transceiver functions; 133 MHz for LTA interface; 128 MHz for RXP receiver functions.

2. The chip provides 8, 33.25 MHz clock signals, one to each column of LTA Controllers. The LTA Controllers use this clock to derive their own internal 133 MHz clocks. The phase of the 33.25 MHz clocks relative to the Gbit Ethernet chip's internal 133 MHz clock must be stable and edge aligned. This is to allow the LTA Controllers to generate an internal clock that properly samples the LTA_SER_CMD input on its rising edge.

3. The synchronous MCB interface shall be capable of operating with a clock that is neither frequency nor phase synchronous with the 125, 156.25, 133, or 128 MHz clocks. The chip will support an MCB interface clock with a maximum rate of 33 MHz.

4. The power dissipation of this device is not of great concern since there is only one such device on the Baseline Board. An additional heat sink can be used if necessary.

5.3 Environmental Requirements

1. The Gbit Ethernet FPGA will be surface-mounted on the Baseline Board motherboard.

2. All I/Os to this device are as defined in section 4. There is a mixture of 2.5 V LVTTL, SSTL2-C1, PCML, and LVDS.

3. The differential PCML output of the chip is to drive an SFP module or XPAK module (10 G).

4. SSTL2-C1 connections to/from LTA Controllers will be terminated externally to the chip.

5. The chip requires 1.5 V, 2.5 V, and 3.3 V supplies.

5.4 Interface Requirements

5.4.1 LTA Controller Interface Requirements

There are eight separate LTA Controller interfaces on this chip. Each one is identical as shown in Figure 4-1. Refer to the LTA Controller RFS document [1], section 5.4.2 for a detailed description of this interface. Note that the phase of all input signals relative to the internal 133 MHz clock is irrelevant (but must be stable). The phase of the LTA_SER_CMD output must be aligned with the internal 133 MHz clock and routed on the board so that the LTA Controller properly samples the data with its own internal 133 MHz clock derived from the provided 33.25 MHz clock.

5.4.2 Gbit Ethernet Chip Interface Requirements

The Gbit output signal contains UDP/IP frames, 8B/10B encoded with NRZ encoding on PCML running at 1.25 Gbps. Between frames, PCS/PMA encoding according to clause 36 of IEEE 802.3-2005 is used. The information content of the Gbit Ethernet frame according to the IEEE 802.3 standard and is according to the following diagram:

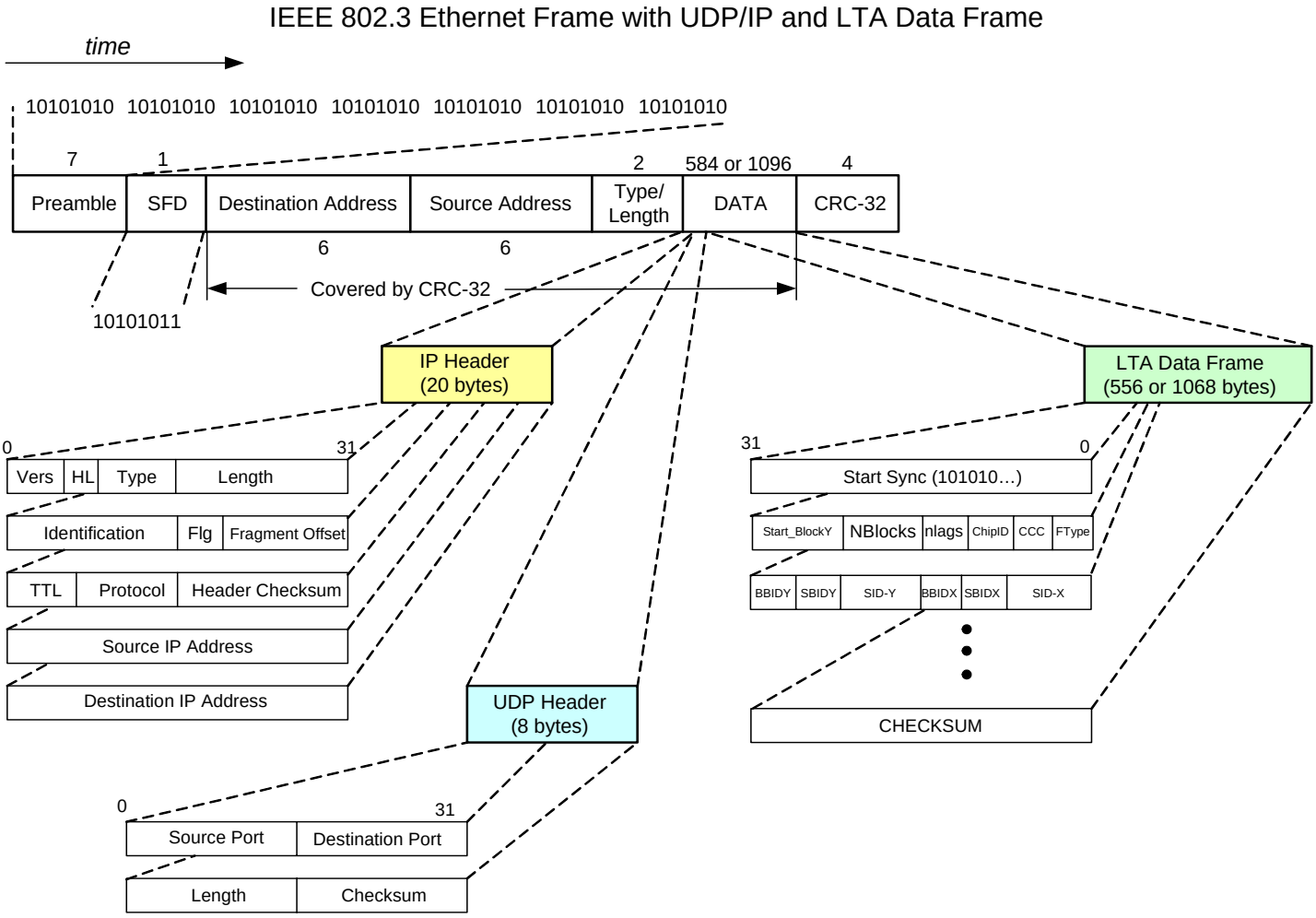


Figure 5-1 Gbit/sec Ethernet frame format including the UDP/IP headers, and the LTA Data Frame. Since 64 or 128 lag frames can be transmitted, the DATA portion of the Ethernet frame can be either 584 or 1096 bytes long. This satisfies the minimum frame length requirements of Gbit Ethernet. Note that this diagram does not include the effect of 8B/10B encoding.

Data bytes (a.k.a. "octets") within 32-bit words are transmitted Most Significant Byte first, and within a byte, are transmitted on the wire Least Significant Bit first. This is a requirement of the Internet Protocol standard [3]. Thus, bytes in the frame are transmitted as read left to right in the above figure, and show up in memory in the destination computer in *ascending* memory order. Note that the bit numbering in the LTA Data Frame is opposite to that in the IP and UDP headers, because it is defined as

such in the LTA Controller output frame format [1]. Nevertheless, byte ordering is as defined above.

A UDP/IP frame format is used for RXP phased data, except that the payload is a frame of sampled data according to the "VDIF" standard. This FPGA does not decode or determine the RXP phased data payload, but rather encapsulates it in the UDP/IP frame, as shown in Figure 5-1 (except that a VDIF frame is used instead of an LTA frame).

A description of the each element of the frame in Figure 5-1 is as follows:

5.4.2.1 Ethernet Frame Header

Preamble – The preamble is 7 bytes of a 01010101 pattern. This preamble provides maximum 1's density on the line and is transmitted at the beginning of the frame. There must be 7 preamble bytes at the beginning of each frame.

SFD – Start Frame Delimiter. This is a 11010101 pattern that indicates that the frame is about to start. Essentially, this is a preamble with a start bit, similar to what is used in other parts of the correlator system.

Destination Address – This is the 6-byte destination MAC/hardware address. This field is sent by the LTA FPGA, in addition to the destination IP, to this chip. For phased RXP data, this field is configurable by the microprocessor (section 5.4.3.11).

Source Address – This is the 6-byte source MAC/hardware address. This field is configurable by the microprocessor (section 5.4.3.10).

Type/Length – Two-byte type or length of the DATA portion of the frame. This is always set to 0x0800, meaning IP v4 frame.

5.4.2.2 IP Header (20 bytes)

This header conforms to the Internet Protocol standard [3]. Refer to this standard for more detailed information.

Vers – 4-bit IP version number. For IP version 4, this is always set to"4" (0100).

HL – 4-bit Header Length. This is the number of 32-bit words in the IP header and is fixed at 5.

Type – The 8-bit Type of Service to be used with this IP datagram. This field is configurable by the microprocessor (section 5.4.3.14).

Length – The total length of the IP datagram in bytes (octets), including the IP header, the UDP header, and the LTA data frame. This can be one of 584, for a 64-lag frame or 1096, for a 128-lag frame. This value is set automatically in hardware. For phased VDIF packets from the RXP FPGA, this is set on the fly.

Identification – A 16-bit value assigned by the IP sender to help reassemble fragmented packets. This Identification gives some idea of the source of the packet as follows:

Bit 15 – 0=LTA source; 1=RXP source.

Bit [14:12] – LTA: LTA column; RXP: RXP channel (B14=0) [0=Lower-B; 1=Lower-A; 2=Upper-B; 3=Upper-A.

Bit[11:0] – Incremented on each frame.

Flg – A 3-bit flag used to control fragmentation. This field is set to 010 in the design.

Fragment Offset – A 13-bit offset into a non-fragmented datagram used to reassemble a datagram that has become fragmented. This field is set to 0 in the design.

TTL – The 8-bit maximum Time To Live the datagram is allowed to live as it travels through the network. This field is set to all 1s in the design.

Protocol – An 8-bit number that identifies the higher-level protocol that the data portion of the datagram belongs to. This field is set to 17 (decimal) by the design, indicating the UDP protocol [3].

Header Checksum – A 16-bit checksum of the header. This is calculated as the 16-bit 1's complement of the 1's complement sum of all 16-bit words in the header. For the purposes of computing the checksum, the value of the checksum field is 0.

Source IP Address – The 32-bit source IP address, taken from the microprocessor configuration (section 5.4.3.12).

Destination IP Address – The 32-bit destination IP address of the frame, taken from the data transmitted by the associated LTA Controller. For phased RXP VDIF data, this is configurable by the microprocessor (section 5.4.3.13)

5.4.2.3 UDP Header (8 bytes)

This header conforms to the UDP protocol standard [4].

Source Port – The 16-bit port number assigned on the sending computer to the application program that passed this message to UDP for transmission. Configurable by the microprocessor (section 5.4.3.15), **and set by default to 12001 decimal** for LTA data.

Destination Port – The 16-bit port number assigned on the receiving computer to the destination application program for this message. Configurable by the microprocessor (section 5.4.3.16), **and set by default to 12000 decimal** for LTA data.

Length – Number of bytes in this datagram including the UDP header and the LTA data frame. For LTA frames, set by the design to 564 for 64-lag frames and 1076 for 128-lag frames. For phased RXP VDIF frames, calculated on the fly.

Checksum – Set by the design to all 0s, indicating that this field is unused [4].

5.4.2.4 LTA Data Frame

The part of the Ethernet frame that contains the data from the LTA Controller. It could be either an LTA Data Frame (64-lags or 128 lags), or a "speed" data frame (that is really a raw correlator chip data frame). Both frame types are described in detail in the LTA Controller RFS document [1]. It must be noted that if a 64-lag LTA Data Frame is transmitted, there are 139, 32-bit words in the frame instead of the normal 267, 32-bit words.

For phased RXP VDIF frame format, refer to the RXP FPGA RFS [7].

5.4.3 MCB (Microprocessor) Interface Requirements

The MCB (Monitor & Control Bus) interface allows a microprocessor to write into the Gbit Ethernet chip to configure it, and to read from it to verify configuration information and obtain status information. Physical interface timing requirements are shown in Figure 5-2.

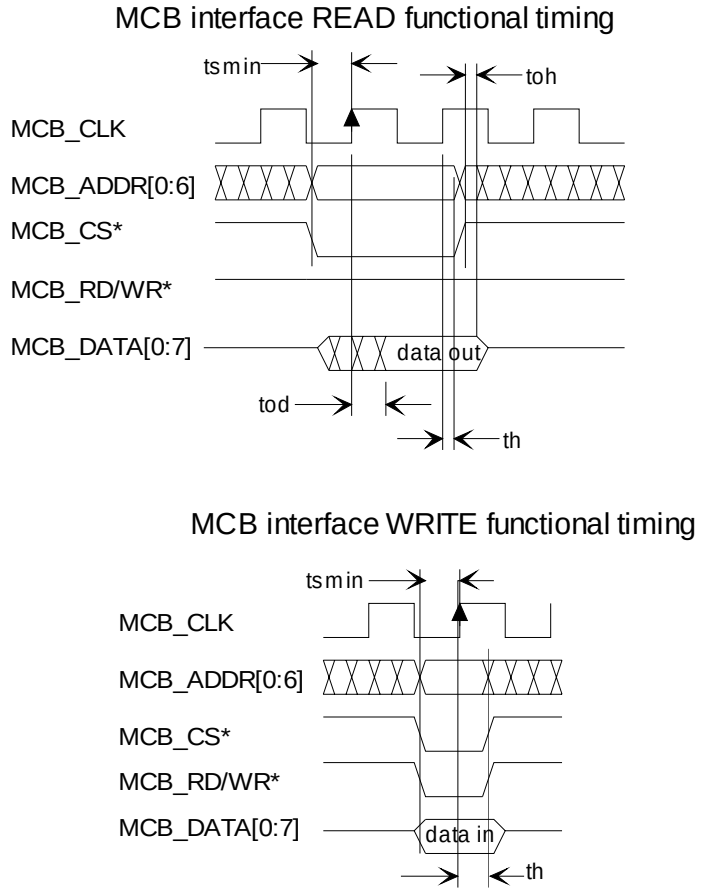


Figure 5-2 MCB interface functional timing requirements. READs require one clock cycle setup before data is ready—however, successive reads take one clock cycle each. WRITEs require one clock cycle.

The chip contains a number of configuration and status registers that can be accessed via the MCB. These registers are defined in the following sub-sections.

5.4.3.1 Master Control/Status Register (MCSR); R/W Address=0x00

This register sets global chip configuration, and detects global status information. A complete description of the register bits is as follows:

MCSR (R/W) Addr=0x00 (Reset=0x30)

7	6	5	4	3	2	1	0
RXP EN	PLL RXP	BVer	RAN	PLL 156	PLL 133	PLL 125	EN

EN (R/W) – Enable data flow from LTA Controllers. If reset (0), then data flow from LTA Controllers is disabled. If set (1), then data flow from the LTA Controllers is enabled. Ethernet frame control registers should not be modified unless this is reset (0), otherwise garbage frames may be produced. This bit's state may be changed at any time, and is not interlinked with frame transmission controllers in the FPGA. Changing this bit does not produce truncated frames.

PLL125 (R/W) – Writing a 1 to this bit enables the 125 MHz PLL; writing a 0 disables it. Reading the bit indicates 125 MHz PLL lock status (1=locked, 0=not locked). Only used if 1 GigE configuration.

PLL133 (R/W) – Writing a 1 to this bit enables the 133 MHz PLL; writing a 0 disables it. Reading the bit indicates 133 MHz PLL lock status (1=locked, 0=not locked).

PLL156 (R/W) – Writing a 1 to this bit enables the 156.25 MHz PLL; writing a 0 disables it. Reading the bit indicates 156.25 MHz PLL lock status (1=locked, 0=not locked). Only used if 10 GigE configuration.

BVer (R/W) – Baseline Board version. This allows the CPU to set the board version ('0'=V2.1; '1'=V2.2). (All Baseline Boards with S/Ns <=2008 are V2.1, and all boards with S/N >=2009 are version 2.2). This is to allow for correct operation of the RXD0 bit of the SXTCSR register. In V2.1 boards, the SFP1 LOS signal is not connected, and so if BVer='0', the RXD0 bit is not affected by SFP1 LOS (meaning that the bit won't reflect if a cable is connected to an active far end, only the status of receive data from the SFP module). In V2.2 boards, the SFP1 LOS signal is connected, and so the RXD0 bit reflects both status of connection to a far end device, and status of receive data from the SFP. Default is '1' (V2.2 board).

RAN (R/W) – This bit, if set (1), *uniformly* randomizes the inter-frame delay between packets transmitted on SFP1, over the range of 0 to 2xIFDR delay. Also, if this bit is set (1), the IFDR register (in 1 G mode) is in units of 4 usec, allowing for over 1 msec of

inter-frame delay. If this bit is reset (0), no time randomization occurs, and the IFDR register is always in units of 1 usec. Default is set (1).

PLLRXP (R/W) – Writing a 1 to this bit enables the 128 MHz PLL; writing a 0 disables it. Reading the bit indicates 128 MHz PLL lock status (1=locked, 0=not locked).

Note that all PLL lock status bits (with the exception of PLL156 if not configured for 10 GigE) must be high before pulling the EN bit high to enable data transfer.

RXPEN (R/W) – Enable data flow from the RXP chips to GigE output, either to SFP1 or SFP2, depending on the setting of the SXTCSR register. Individual RXP data channels can be enabled or disabled with the RXEN bit of the RXPCSR-0:3 registers; however, for any data flow from any RXP chips, this bit must be set.

5.4.3.2 FPGA Version Status Register (FVSR); R/W Address=0x01

This register contains the "discard frames" (DF) function bit, and provides status information about the FPGA version and external Gbit Ethernet fiber connections as follows:

FVSR (R/W) Addr=0x01 (Reset=0x02)

7	6	5	4	3	2	1	0
DF	10G	v5	v4	v3	v2	v1	v0

v[0:5] (R) – The current version of the FPGA. This "V2" version of the GigE FPGA is greater than or equal to 000010b (version 2).

10G (R) – This bit indicates if the FPGA bitstream is 2 x 1 GigE or 10 GigE. If reset(0), the FPGA is 2 x 1 GigE using two RFP modules. If set (1), the FPGA bitstream is 10 GigE, using 1 XPAK transceiver module.

DF (R/W) – Discard Frames. If set (1), then no Ethernet frames are transmitted on the GigE SFP ports, but data still continues to flow from the LTAs. In this case, Ethernet frames just go into the bit bucket, but data transmission from LTAs continues to operate as normal. If reset (0), then Ethernet data transmission functions normally. Note that this functionality is only available in Baseline Boards V2.1 and higher; for V2.0 boards (S/Ns 2001 and 2002), this bit is ignored, and will not stick if set (unless the EN bit of the MCSR is low).

5.4.3.3 Inter-Frame Delay Register (IFDR); R/W Address=0x02

This is an 8-bit register that sets the time, in units of 4 microseconds (**if the RAN bit of the MCSR is set (1); if the RAN bit is reset (0) it is in units of 1 microsecond**), between transmitted frames on SFP1 (or on XPAK if in 10 G mode). ***If in 10 G mode (the 10 G bit of the FVSR is set (1)), then this is always in units of 1 microsecond***.

This functionality is provided to allow the Gbit Ethernet output frame rate to be throttled and randomized to more evenly distribute network traffic. Note that this is the time *between* the end of one frame, and the start of the next frame, not the frame transmission period. A 64-lag frame takes ~4.9 μsec to transmit and a 128-lag frame takes ~9 μsec to transmit (on 1 GigE), and so the actual desired frame transmission rate calculation must take these times into account.

If performance is not being pressed, this should be set to a non-zero value, and the RAN bit of the MCSR should be set (1), to avoid incurring large peak packet rates in the network, which could result in dropped frames. In this case, the IFDR is the nominal inter-frame delay. Refer to the description of the RAN bit in the MCSR register for more information on randomization.

IFDR (R/W) Addr=0x02 (Reset=0x14)

Important Notes:

If RXP data frames are being routed to SFP1 (1 GigE implementation), this should be set to a low value to avoid lost frames, especially at high frame rates, as there is minimal RXP data frame buffering capability in the device. A reasonably precise calculation of the maximum value of the IFDR register can be made, knowing the correlator output frame rate, LTA frame sizes, and VDIF packet sizes. To avoid losing RXP (VDIF) data frames, the PRI bit of the SXTCSR register should be set (1), to establish VDIF packets as the priority in this case.

This register has no effect on RXP (VDIF) packets routed to SFP2.

5.4.3.4 SFP/XPAK Transceiver Control/Status Register (SXTCSR); R/W Address=0x03

This register is used to obtain status of GigE receivers (indicating if a line is up or down), to enable or disable serial loopback mode for testing, and to control where RXP (phased) data frames are transmitted, and with what priority.

SXTCSR (R/W) Addr=0x03 (Reset=0x80)

7	6	5	4	3	2	1	0
PRI	RXPD	XOEN	SLE	RXD3	RXD2	RXD1	RXD0

RXD[0:3] (R) – Rx Detect bits for each of the lanes of receive data from either SFP modules, or the 10 GigE XPAK module. If set (1), then the receive data line is active, receiving proper codes, and the SFP module reports signal detected (meaning a cable is connected to an active far end), except for V2.1 Baseline Boards on RXD0 (refer to the 'BVer' bit of the MCSR for more explanation). If reset (0), then the receive data line is not active or is not receiving proper codes. Cleared on read. Normally in the 2 x 1 GigE configuration, RXD2 and RXD3 are low, and RXD0 indicates SFP1 receiver status, while RXD1 indicates SFP2 receiver status. In 10 GigE mode, RXD[3:0] indicates status of each of the four XAUI lanes from the XPAK module and the 10 GigE receiver.

SLE (R/W) – Serial Loopback Enable. If set (1), then the transmitters are looped back to the receivers in the transceiver hard block on all channels. Data is still being transmitted, however the receive data path now contains transmit data, rather than receive data. Normally this is only ever enabled for testing.

XOEN (R/W) – XON/XOFF enable. If set (1), transmission flow control of LTA data frames on SFP1 is throttled by PAUSE frames received by the SFP1 receiver. If reset (0), PAUSE frames are ignored and transmission is free running, throttled only by the output rate from LTAs on the board and the value of the IFDR register. Flow control is not enabled (or even possible) on SFP2, or if RXP packets are directed to SFP1, or in 10 G XPAK mode.

RXPD (R/W) – RXP data destination. This bit controls the destination of phased-array frames from the RXP FPGAs when in 2x1 GigE SFP mode. If reset (0), phased-array RXP MkVC frames are transmitted on SFP module 2, and would then normally go to the MkVC data recorder. If set (1), phased-array RXP frames are transmitted on SFP module 1, and would go to the CBE for data capture, along with LTA data frames. If in 10 GigE mode, this bit is ignored, and all frames are always transmitted on the 10 GigE link.

PRI (R/W) – If the RXPD bit is set (1), then this bit determines the priority of phased data packets from RXP chips transmitted on SFP1. If this bit is reset (0), then LTA

frames have priority, which may result in RXP phased-data packets being dropped, if the RXP packet rate and/or packet size is high. If this bit is set (1), then RXP phased frames have priority, which may result in LTA frame buffers overflowing, and lost correlation coefficients. If in 10 GigE mode, the RXPD bit is ignored, and this bit sets transmission priority for traffic on the single 10 GigE link. Default is set (1). Normally, to reduce the chance that RXP packets are lost, this bit must be set.

5.4.3.5 GigE TX Frame Count Register SFP1 (GTXFCR1); (R) Addr=0x04, 0x05

This 16-bit register provides a count of the number of LTA frames transmitted on SFP module 1, when in GigE mode, or transmitted on 10 GigE, when in 10 GigE mode, since the last time this register was read. i.e. This counts the number of LTA frames transmitted.

Address 0x04 is the least significant count, and address 0x05 is the most significant count. These registers are cleared when address 0x05 is read.

5.4.3.6 GigE TX Frame Count Register SFP2 (GTXFCR2); (R) Addr=0x06, 0x07

This 16-bit register provides a count of the number of RXP/VDIF frames transmitted on SFP module 2, or SFP1 if RXP/VDIF packets routed there when in GigE mode; when in 10 GigE mode (10G bit of the FVSR register set), this register counts the number of RXP/VDIF frames transmitted on the 10G output. i.e. this counts the number of RXP/VDIF frames transmitted.

Address 0x06 is the least significant count, and address 0x07 is the most significant count. These registers are cleared when address 0x07 is read.

5.4.3.7 RXP Receiver Control/Status Register-0:3 (RXPCSR-0:3); (R/W) Addr=0x08, 0x09, 0x0A, 0x0B

These are control/status registers for each of the 1.024 Gbps receive data channels from the RXP FPGAs. Register assignment is according to the following table:

Register (Address)	RXP Receive Data Channel
RXPCSR-0 (0x08)	UPPER, channel A
RXPCSR-1 (0x09)	UPPER, channel B **UNUSED**
RXPCSR-2 (0x0A)	LOWER, channel A
RXPCSR-3 (0x0B)	LOWER, channel B **UNUSED**

Table 5-1 RXPCSR0-3 register assignments to RXP data channels.

NOTE **UNUSED** CHANNELS; THERE IS NOT ENOUGH MEMORY FOR THESE IN THE FPGA, AND SO THEY ARE NOT ACTIVE AND CAN BE IGNORED.

RXPCSR-0:3 (R/W) Addr=0x08-0x0B (Reset=0x00)

7	6	5	4	3	2	1	0
N/C	SE	FOR	FE	FD	SERR	SS	RXEN

RXEN (R/W) – Receiver ENable. If reset (0), the particular receive data channel is not enabled. If set (1), the data channel is enabled. At any time, cycling this off (0) and then on (1), causes the receiver to re-sync. Re-sync will only happen if an idle code set “52h”, “ABh” is detected (Figure 5-3).

SS (R) – Sync Status. If reset (0), the receiver is not synchronized. If set (1), the receiver is synchronized. Data from the RXP FPGA will not be received until this bit is set. Once this bit is set (receiver is synchronized), automatic H/W resync won’t happen and it is up to S/W to toggle RXEN if too many Frame Errors are detected.

SERR (R) – Sync Error. If reset (0), no synchronization error has been detected since this register was last read. If set (1), then at least one sync error has been detected. Cleared on read.

FD (R) – Frame Detect. If reset (0), then no frames have been detected in this data channel since the last time it was read. If set (1), then at least one frame has been detected. Cleared on read.

FE (R) – Frame Error. If reset (0), then no frame error has been detected since the last time it was read. If set (1), then at least one frame error has been detected. Cleared on read.

FOR (R) – Frame OverRun. If reset (0), then no frame overruns have been detected since the last time it was read. If set (1), then one or more frames from the RXP have been discarded because the receive frame buffer is full. Cleare on read.

SE (R) – Size Error. If set (1) it indicates that a frame from the RXP chip was detected which exceeds the maximum frame size of 8192 bytes. Clear on read.

5.4.3.8 RXP Packet Delay Register-{ UA, UB, LA, LB} (RXP_PDR); (R/W) Address=0x0C (UA), 0x0D (UB), 0x0E (LA), 0x0F (LB)

Each of these 8-bit registers sets a delay, in µsec, from the start of the incoming packet from the RXP FPGA, to the start of the outgoing packet on 1 GigE (or 10 GigE). This allows temporal network packet density to be controlled to some degree. UA=Upper RXP Channel A; UB=Upper RXP Channel B; LA=Lower RXP Channel A; LB=Lower RXP Channel B.

The maximum delay should not exceed one packet duration, if running near the maximum packet rate available. For example, if transmitting on 1 GigE, and an 8k packet with 3-bit samples (i.e. ~77% capacity), the packet transmission time is ~8192 x 1 byte every 125 MHz clock cycle~=65.5 usec. In this case this maximum delay shouldn't be set larger than ~65. If running at a lower capacity, more delay is available, although generally, the delay should never exceed the packet transmit duration and should never be such that the next packet from the RXP is arriving before the current packet is transmitted.

Note: When RXP packets are routed to SFP1, it is best to set this to a low or zero value to allow the scheduler maximum capability in routing packets, so no packet loss results.

5.4.3.9 LTA Column Rx Error Status Register (LTACRxESR); R Address=0x10

This read-only 8-bit register is used for determining the error status of data transmission from each column of LTAs. If a bit in this register is set (1), then there was an LTA column frame reception error from the associated LTA column since the last time this register was read. If a bit is reset (0), then there was no frame reception error. Cleared on read.

Note that knowledge of whether frames are being transferred from one or more columns of LTAs, can be determined by reading the toggle status registers LITSASR and LITSR. This register reports only errors.

5.4.3.10 Ethernet Source Address Registers (SMAC-0:5); R/W Address=0x11, 0x12, …, 0x16

These 6 registers set the Source Address ("MAC" address) that is stuffed into the LTA Ethernet Frame Header for all outgoing LTA frames (SFP1 module OR XPAK). SMAC-0 is at address 0x11 etc. Default: EA:EA:EA:EA:EA:EA.

5.4.3.11 RXP (Phased) Data Ethernet Destination Address Registers (RXP_DMAC-0:5); R/W Addresses=0x17, 0x18, …, 0x1C

These 6 registers set the Destination Address ("MAC" address) that is stuffed into the Ethernet Frame Header for all phased data frames from both RXP chips. RXP_DMAC-0 is at address 0x17 etc. Default: 00:00:00:00:00:00.

Note: In the GigE FPGA V2 design, the DMAC for LTA frames sources from the LTA FPGAs, and is not set in this FPGA anymore.

5.4.3.12 Internet Source IP Address Registers (IPADRSRC-0:3); R/W Address=0x20, 0x21, 0x22, 0x23

These four 8-bit registers are used to set the source IP address in the outgoing datagram for all LTA frame packets. IPADRSRC-0 is Byte 0 (least significant byte) of this address, and IPADRSRC-3 is Byte 3 (most significant byte) of this address. Default on chip reset is 192.139.21.78 (C0.8B.15.4E).

5.4.3.13 RXP (Phased) Data Internet Destination IP Address Registers (RXP_IPADRDEST-0:3); R/W Address=0x24, 0x25, 0x26, 0x27

These four 8-bit registers are used to set the destination IP addresses in the outgoing datagram for all RXP/phased frame packets. RXP_IPADRDEST-0 is Byte 0 (least significant byte) of this address, and RXP_IPADRDEST-3 is Byte 3 (most significant byte) of this address. Default on chip reset is 0.0.0.0.

5.4.3.14 Internet IP Type of Service Register (IPTOS); R/W Address=0x30

This 8-bit register is used to set the 8-bit "Type of Service" parameter to be used with all outgoing IP datagrams on both SFP ports, including RXP/phased data packets. Refer to the "Type" definition in section 5.4.2.2. This register will only be allowed to change if the MCSR EN bit is low. Default on chip reset is 0x00.

5.4.3.15 Internet UDP Source Port Registers (IUSP-0:1); R/W Address=0x31, 0x32

These two 8-bit registers are used to set the UDP source port in the UDP header (Figure 5-1) **for LTA frame data**. Set in hardware by default to **12001** decimal (0x2EE1 hex). The Least Significant byte is at address 0x31.

5.4.3.16 Internet UDP Destination Port Registers (IUDP-0:1); R/W Address=0x33, 0x34

These two 8-bit registers are used to set the UDP destination port in the UDP header (Figure 5-1) **for LTA frame data**. Set in hardware by default to **12000** decimal (0x2EE0 hex). The Least Significant byte is at address 0x33.

5.4.3.17 Internet UDP Source Port Registers for RXP (Phased) Data (RXP_IUSP-0:1); R/W Address=0x35, 0x36

These two 8-bit registers are used to set the UDP source port in the UDP header (Figure 5-1) **for phased data packets from RXPs**. Set in hardware by default to **12003** decimal (0x2EE3 hex). The Least Significant byte is at address 0x35.

5.4.3.18 Internet UDP Destination Port Registers (RXP_IUDP-0:1); R/W Address=0x37, 0x38

These two 8-bit registers are used to set the UDP destination port in the UDP header (Figure 5-1) **for phased data packets from RXPs**. Set in hardware by default to **12002** decimal (0x2EE2 hex). The Least Significant byte is at address 0x37.

5.4.3.19 GigE Receiver CAPture Register (GRCAP); R/W Address=0x40

This register, in conjunction with the GRCAP_CTRL register is used to capture Gig Ethernet received data for SFP module 1, (or, if 10 GigE, from the 10 GigE transceiver) and is mainly useful for debugging purposes. If serial loop-back in the SXTCSR register is enabled, it captures transmitted data, via loopback data path.

Writing any dummy value to this register initiates data capture. Data capture is complete when the CAP_RDY bit in the GRCAP_CTRL register is set (1).

After capture is complete, successive reads of this register read the 2048 successive data bytes that have been captured. If more than 2048 reads are performed, data wrap-around occurs.

5.4.3.20 GigE Receiver CAPture Control Register (GRCAP_CTRL); R/W Address=0x41 Reset=0x00

This register is used to set the GRCAP capture mode, and to indicate when capture is complete.

ECAP (R/W) (Bit 0) – This bit controls the capture mode. If reset (0) then capture occurs immediately with no alignment to any frame words. If set (1) then capture occurs at the start of an Ethernet frame. Since the capture buffer is 2048 bytes, it is more than adequate for capturing an entire frame generated by this chip (i.e. in serial loop-back mode).

CAP_RDY (R) (Bit 1) – This bit indicates data capture status. If set (1), then capture is complete and captured data can be read from the GRCAP register. If reset (0), then capture is not complete, or no data capture was initiated. This bit is cleared on read.

CAP_SEL (R/W) (Bit 2) – This bit selects whether receive or transmit data is captured. If reset (0), receive data is captured. If set (1), transmit data is captured.

SFP_SEL (R/W) (Bit 3) – This bit selects whether data is captured from SFP1 (bit 3=0), or from SFP2 (bit 3 =1). Ignored if 10G.

5.4.3.21 LTA Interface Toggle Status Address Select Register (LITSASR) R/W; Addr=0x42

This read/write register is used to select which toggle status register is accessed when reading the LITSR register at address 0x43. Writing to this register sets the address for a subsequent read of the LITSR register. Bits 7:5 are always read as 0. This register and the companion LITSR register provide valuable diagnostic information if the GigE FPGA is reporting checksum errors from one or more LTA columns. These registers can help to pinpoint the data paths that are failing.

Bits 4:0 – Sets the address select according to the following table:

LITSASR [4:0]	Description (LITSR read register contents)
0x00 – 0x07	Column 0 to 7 handshake status
0x10 – 0x17	Column 0 to 7 data status

Table 5-2 LITSASR addressing table for selecting data read via the LITSR register.

5.4.3.22 LTA Interface Toggle Status Register (LITSR) R; Addr=0x43

This read-only register provides toggle status information for handshake lines and data paths to LTA columns as described above. All toggle status bits are cleared on read.

If **handshake lines** are read (LITSASR = 0x00 to 0x07), then the bit assignments are as follows:

Bit 0 – LTA_FRAME_ toggle status. Set (1) if the LTA_FRAME_ input has been toggling since the last time this register (with the same LITSASR address setting) was read. Reset (0) if the LTA_FRAME_ input has not been toggling.

Bit 1 – LTA_SER_RESP toggle status. Set (1) if the LTA_SER_RESP input has been toggling; reset (0) if it has not been toggling.

Bit 2,3 – Unused, should always be 0.

Bits 4, 5 – Current level status for the LTA_FRAME_, and LTA_SER_RESP signals respectively. These bits are useful for only low-level debug access and do not require any specific software.

Bit 6,7 – Unused, should always be 0.

If **data lines** are read (LITSASR = 0x10 to 0x17), then all 8 bits of the register provide toggle status information for the 8-bit data paths coming from each column of LTA FPGAs.

Note: refer to the LTA RFS [1] for information on how data and control line toggle status information can be determined for LTA-to-LTA connections.

5.4.3.23 LTA Column Clockedge Detect Register (LCCDR) R; Addr=0x44

Each bit in this register represents the FPGA-determined clock edge for a particular LTA column used for clocking LTA data into the FGPA. This is a test read-only register, and no particular software support is required.

5.4.3.24 LTA Bank Select Read Register (LTABSRR) R; Addr=0x45

This is a test-only register. Each bit is the currently selected bank for a particular column, row0, that the chip is querying the LTA for data.

5.4.3.25 Correlator Chip Speed Register (CC_SPEED) R/W; Addr=0x50 Reset=0x02

This is nominally a test-only register that allows some control of the speed of the correlator chips on the board by controlling their nominal operating voltage.

Normally this register does not need to be touched, however if, over time, some correlator chip on the board is exhibiting behaviour that indicates it might not be fast enough, this register could be used to tweak the speed of all chips on the board up slightly by increasing their voltage. Conversely, power savings might be possible if the speed (voltage) is cranked down to the point where the chips still function, although as shown below power savings are hardly worth the effort.

Speed (voltage) settings are according to the following table:

CC_SPEED value	Speed Effect	Voltage Effect	Power Effect
0	Slowest	0.96 V	-12%
1	Slower	0.986 V	-6.5%
2	Normal/default	1.02 V	0%
3	Faster	1.05 V	+6%
4	Fastest	1.08 V	+13%
All others	Normal/default	1.02 V	0%

Note that the speed/voltage effect applies to all chips on the board. Nominally, at full speed with Gaussian noise, the correlator chip core power dissipation is ~1.88 W. The slowest speed would reduce this to ~1.65 W for a total board power savings of ~15 W; the fastest speed would increase this to ~2.12 W for a total board power increase of ~15 W (~2 kW for the entire system).

5.4.3.26 PAUSE Frame Count Register (PFCount) R; Addr=0x52

This 8-bit read-only register is a count of the number of PAUSE frames detected on the SFP1 receiver since the last time it was read. PAUSE frame counting is independent of whether flow control (set by the XOEN bit of the SXTCSR (0x03) register) is turned on or off, and is only for SFP1. This register is cleared on read.

5.4.3.27 RXP Ethernet Source Address Registers (RXP_SMAC-0:5); R/W Address=0x53, 0x54, …, 0x58

These 6 registers set the RXP Source Address ("MAC" address) that is stuffed into the VDIF Ethernet Frame Header for all outgoing VDIF frames whether destined for SFP2, SFP1, or the 10G XPAK. RXP_SMAC-0 is at address 0x53 etc. Default: ED:ED:ED:ED:ED:ED.

5.4.3.28 RXP Internet Source IP Address Registers (RXP_IPADRSRC-0:3); R/W Address=0x59, 0x5A, 0x5B, 0x5C

These four 8-bit registers are used to set the source IP address in the outgoing datagram for all RXP/VDIF frame packets, whether the output port is SFP2, SFP1, or 10G XPAK. RXP_IPADRSRC-0 is Byte 0 (least significant byte) of this address, and RXP_IPADRSRC-3 is Byte 3 (most significant byte) of this address. Default on chip reset is 192.139.21.80 (C0.8B.15.50).

5.4.3.29 DUMMY Register (DUMMY) R/W; Addr=0xFE

This register provides software with a read/write register and does not connect to, or affect any function in the chip. The address change of this register, along with the contents of the FVSR, indicates to S/W that it is the V2 design. (Note that the old DUMMY register at address 0xFF always reads 0xFF.)

5.4.3.30 Summary of Config/Status Registers and Addresses

Table 5-3 is a summary of all LTA Controller configuration and status registers and their associated addresses.

Table 5-3 Summary of Gbit Ethernet chip config/status registers

Register (page)	R/W?	Address	Description
MCSR (26)	R/W	0x00	Master Control/Status Register
FVSR (28)	R	0x01	FPGA Version Status Register
IFDR (29)	R/W	0x02	Inter-Frame Delay Register
SXTCSR (30)	R/W	0x03	SFP/XPAK Transceiver Control/Status Reg.
GTXFCR1 (32)	R	0x04-0x05	SFP1 Frame Count Register
GTXFCR2 (32)	R	0x06-0x07	SFP2 Frame Count Register
RXPCSR-0:3 (33)	R/W	0x08-0x0B	RXP Receiver Control/Status Reg.
RXP_PDR-UA…(35)	R/W	0x0C-0x0F	RXP Packet Delay Registers.
LTACRxESR (36)	R	0x10	LTA Column Rx Err Status Reg.
SMAC-0:5 (37)	R/W	0x11-0x16	LTA frame Ethernet Source Address Registers
RXP_DMAC-0:5 (37)	R/W	0x17-0x1C	Ethernet Destination Address Registers for RXP/phased data packets.
IPADRSRC-0:3 (38)	R/W	0x20-0x23	LTA frame Internet Source IP Address Registers.
RXP_IPADRDEST-0:3 (38)	R/W	0x24-0x27	Internet Destination IP Address Registers for RXP/phased data packets.
IPTOS (39)	R/W	0x30	Internet IP Type of Service Register
IUSP-0:1 (39)	RW	0x31-0x32	Internet UDP Source Port Register (LTA frames)
IUDP-0:1 (39)	R/W	0x33-0x34	Internet UDP Destination Port Registers (LTA frames)
RXP_IUSP-0:1 (39)	R/W	0x35-0x36	Internet UDP Source Port Registers (RXP/phased frames)
RXP_IUDP-0:1 (39)	R/W	0x37-0x38	Internet UDP Destination Port Registers (RXP/phased frames)
GRCAP (40)	R/W	0x40	GigE Receiver Capture Register (SFP1)
CRCAP_CTRL (40)	R/W	0x41	GigE Receiver Capture Control Register
LITSASR (41)	R/W	0x42	LTA Interface Toggle Status Address Select Register
LITSR (42)	R	0x43	LTA Interface Toggle Status Register
LCCDR (42)	R	0x44	LTA Column Clockedge Detect Register
LTABSRR (42)	R	0x45	LTA Bin Select Read Register
CC_SPEED (42)	R/W	0x50	Correlator Chip Speed Register
PFCount (43)	R	0x52	SFP1 PAUSE frame counter

RXP_SMAC-0:5 (43)	R/W	0x53-0x58	RXP/VDIF frame Ethernet Source Address Registers
IPADRSRC-0:3 (43)	R/W	0x59-0x5C	RXP/VDIF frame Internet Source IP Address Registers.
DUMMY (44)	R/W	0xFE	Dummy Read/Write Register.

5.4.4 RXP Interface Requirements – RXP-to-GigE FPGA packet transmission protocol

The RXP FPGAs (Upper and Lower) send packets of real-time phased-array data to the GigE FPGA for transmission on SFP2 (or, optionally, SFP1). The protocol is a "pseudo-packet" protocol as the conversion to full UDP/IP is performed in the GigE FPGA. Also, to ensure reliable transmission and ensure the receivers maintain phase lock independent of data packet content, the data packet is chopped, using the chopper circuit developed for [6] . Two instances of chopper circuits are used for each channel, one for the LSNibble, and one for MSNibble.

This pseudo-packet frame is shown in the following figure.

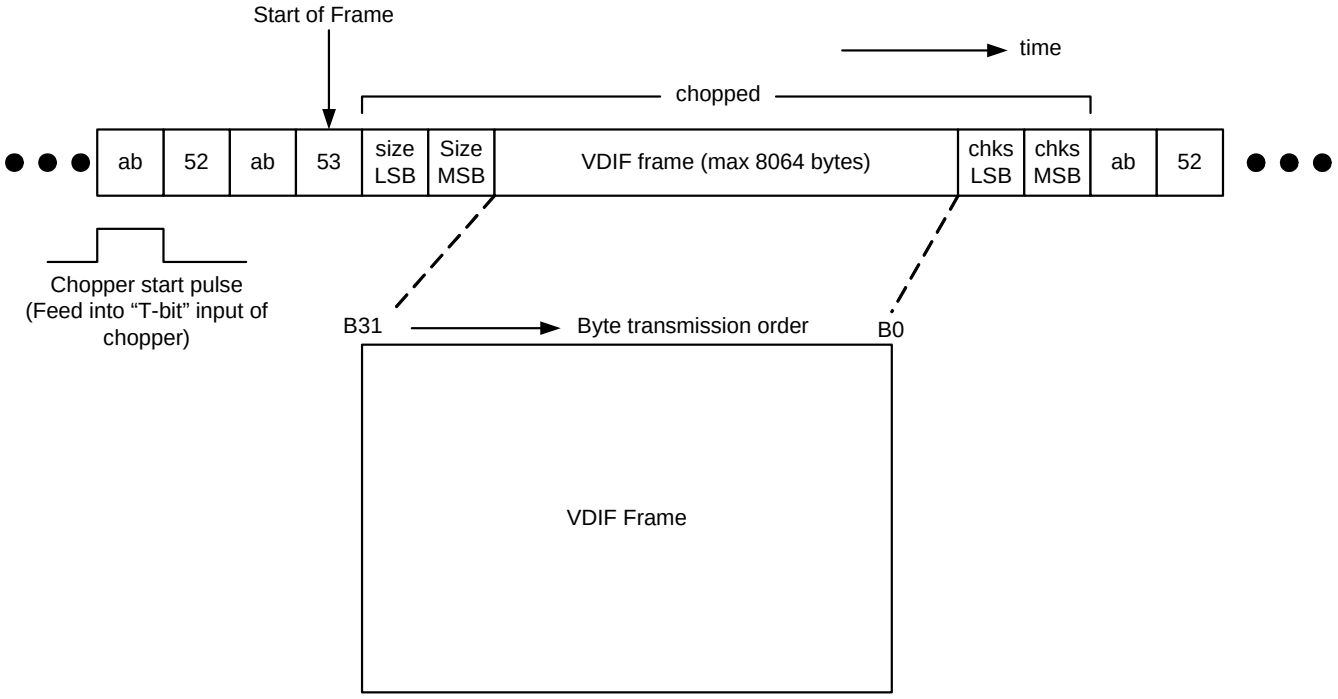


Figure 5-3 The pseudo-frame protocol for transmission of RXP-to-Gige VDIF frames.

The Upper RXP, channel [1] (the only active channel) has an LSN chopper seed of 0x82, and an MSN chopper seed of 0x89. The Lower RXP, channel [1] has an LSN chopper seed of 0x35, and an MSN chopper seed of 0x3C. Further details of VDIF frame contents are contained in the RXP RFS A25093N0000.

The VDIF frame is transmitted by the RXP in big endian format, and is translated to little endian format in the GigE FPGA before being sent out as a UDP/IP packet.

5.4.5 GigE V2 Graphical User Interface (GUI)

Figure 6-1 shows a simplified GUI concept of the GigE V2 design.

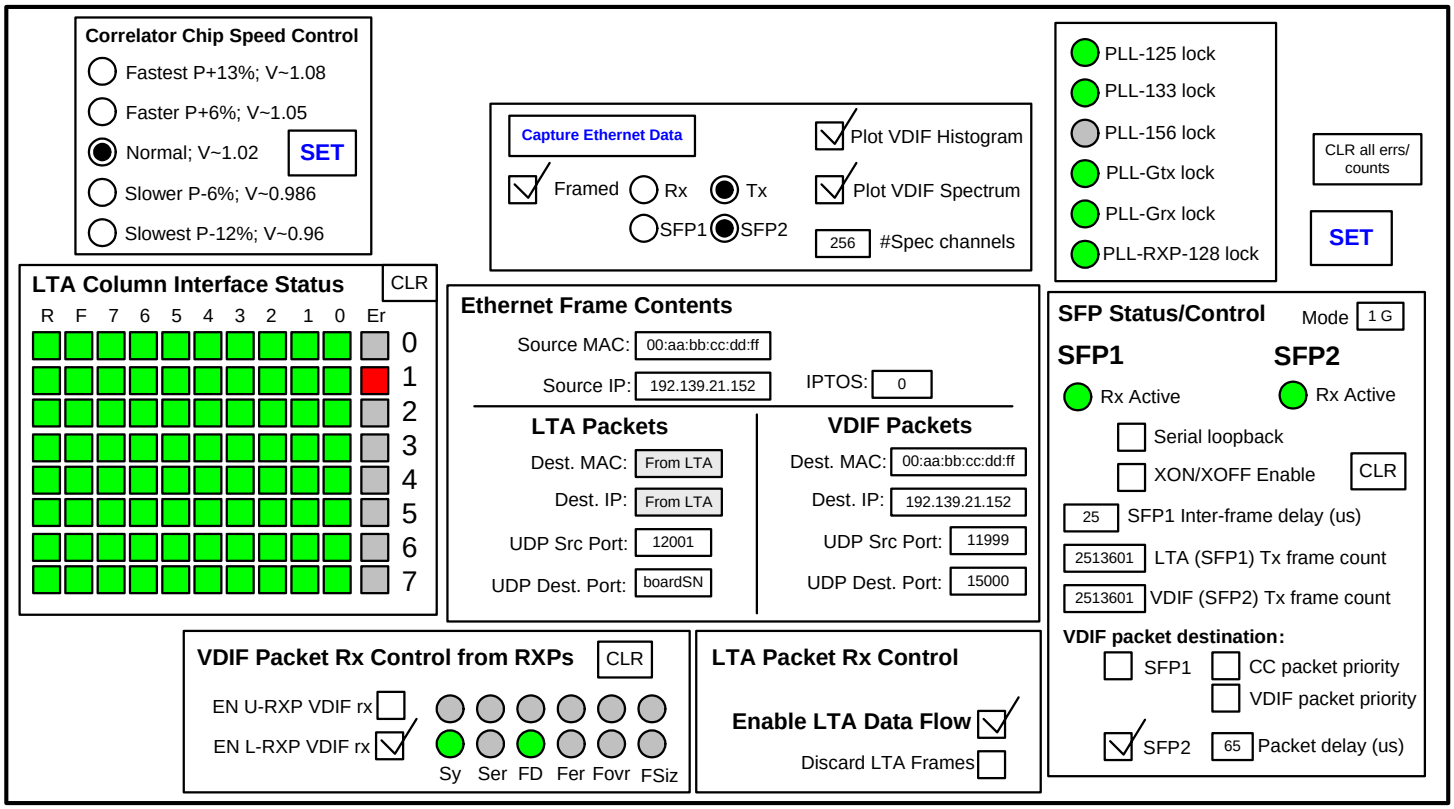


Figure 5-4 Simplified GigE V2 GUI concept.

It is left to the expert GUI developers to maintain “SET”, “CLR”, colouring, etc. consistency with other board and system GUIs.

5.4.6 Miscellaneous Interface Requirements

1. The chip must have JTAG test port capability.

2. The chip must be ISP (In-System Programmable), via an 8-bit (+control lines) programming interface, or, if preferred, a serial programming interface.

6 Functional Specifications

This section presents a plan for implementing the functionality of the Gbit Ethernet chip. Since an adequate description of the requirements and functionality of the chip is provided for in preceding sections, this section will touch on some key implementation details.

6.1 Ethernet CRC-32 Calculation

An important component of the Ethernet frame shown in Figure 5-1 is the CRC-32 "frame check sequence". A CRC is easily calculated on serial data using a shift register and exclusive-OR gates however, since the data in the FPGA that is accessible to the general logic in the chip is always in 8-bit parallel format, an algorithm must be used to generate the CRC code, while operating on 8-bit parallel data.

Fortunately, a search on the Internet has revealed a great deal of information on this topic, and source Verilog code to do the job. Refer to Xilinx application note XAPP209 V1.0 March 23, 2001 for more information.

6.2 1 Gbit Ethernet Output

The target chip that will be used for the implementation is the Stratix GX EP1SGX25CF672C5. This device contains 25,660 logic-elements and 4 Gigabit transceiver channels. All of the facilities required to take a byte-wide Ethernet frame data at 125 MHz and convert to a serial bit stream with the proper encoding are included as "megafunctions" in the chip. Thus, the user logic need only generate the PCS codes (Physical Coding Sub-Layer codes, which are present between the Ethernet frames), and the byte-wide Ethernet frame as defined in Figure 5-1, and megafunctions in the chip will do the rest.

A simplified description of inter-frame PCS coding (following Clause 36. of IEEE Std 802.3-2005) without auto-negotiation is as follows.

Applicable PCS codes: A sub-set of code group names (e.g. "K28.5" is a byte-code presented to the transmitter before 8B10B encoding) are as follows:

"**K28.5**"=0xBC, "**K27.7**"=0xFB, "**K29.7**"=0xFD, "**K23.7**"=0xF7, "**D0.0**"=0x00, "**D5.6**"=0xC5, "**D16.2**"=0x50

PCS "code sets": Higher-level code sets are also defined:

/I1/ (IDLE 1) = /K28.5/D5.6/ = 0xBC, 0xC5 (Correcting idle)
/I2/ (IDLE 2) = /K28.5/D16.2/ = 0xBC, 0x50 (Preserving idle)
/S/ (Start_of_Packet) = /K27.7/ = 0xFB
/T/ (End_of_Packet) = /K29.7/ = 0xFD

PCS code set usage:

Between Ethernet frames, the /I2/ code set is continuously active. Before the first Preamble byte (0x55) is the /S/ code, indicating Start_of_Packet. Then comes **exactly** 7 Preamble bytes, the SFD, etc., all the way to the CRC-32 word, which is the end of the packet. The PCS code /T/ is then inserted, followed by 1 or 2 K23.7 codes depending on whether there were an odd or even number of bytes in the Ethernet frame. Then comes a /I1/, followed by continuous /I2/ code sets until the next Ethernet packet starts as described.

In the Altera implementation, the transmit megafunction interface inside the FGPA is 1 byte wide, along with at "tx_ctrlenable" signal, which is high (1) if the byte is the first byte of a PCS code set, or low (0), if the byte is the second byte of a PCS code set, or an Ethernet frame byte (starting from the first preamble to the last CRC-32 byte). The interface clocks bytes 125 MHz, and all 8B/10B encoding is performed in the megafunction.

An example byte sequence presented to the transmit interface is as follows ("0x" hex indicators dropped for clarity):

BC 50 BC 50 BC 50 FB 55 55 55 … [last CRC byte] FD F7 F7 BC C5 BC 50 BC 50 …

6.3 10 Gbit Ethernet Output

From a user design perspective (i.e. the user-logic interface in the FPGA), the 10 Gbyte implementation is actually simpler than the 1 GigE implementation. The user-logic interface is a modified XGMII interface, consisting of 8 byte lanes, clocked at 156.25 MHz. There are only 3 special PCS codes, **/I/** (IDLE=0x07), **/S/** (START=0xFB), and **/T/** (TERMINATE=0xFD). Since this is a modified XGMII interface, some byte-lane multiplexing is used. Figure 6-1 nicely shows a transmit frame sequence, and the exact byte ordering on the 8-byte wide XGMII interface. Note that there are only 6 Preamble bytes, one of them being replaced by the /S/ code.

The Altera megafunction translates the 8-byte 156.25 MHz to 4 bit streams, each running at 3.125 Gbps c/w 8B/10B encoding (the XAUI interface). The 4 x 3.125 Gbps lines (4 transmit and 4 receive) then run to the nearby XPAK transponder module. As with the 1 GigE interface, there is an 8-line "tx_ctrlenable" input to the transmitter in the FPGA, 1 line for each byte to indicate PCS codes (if high) or user data (if low).

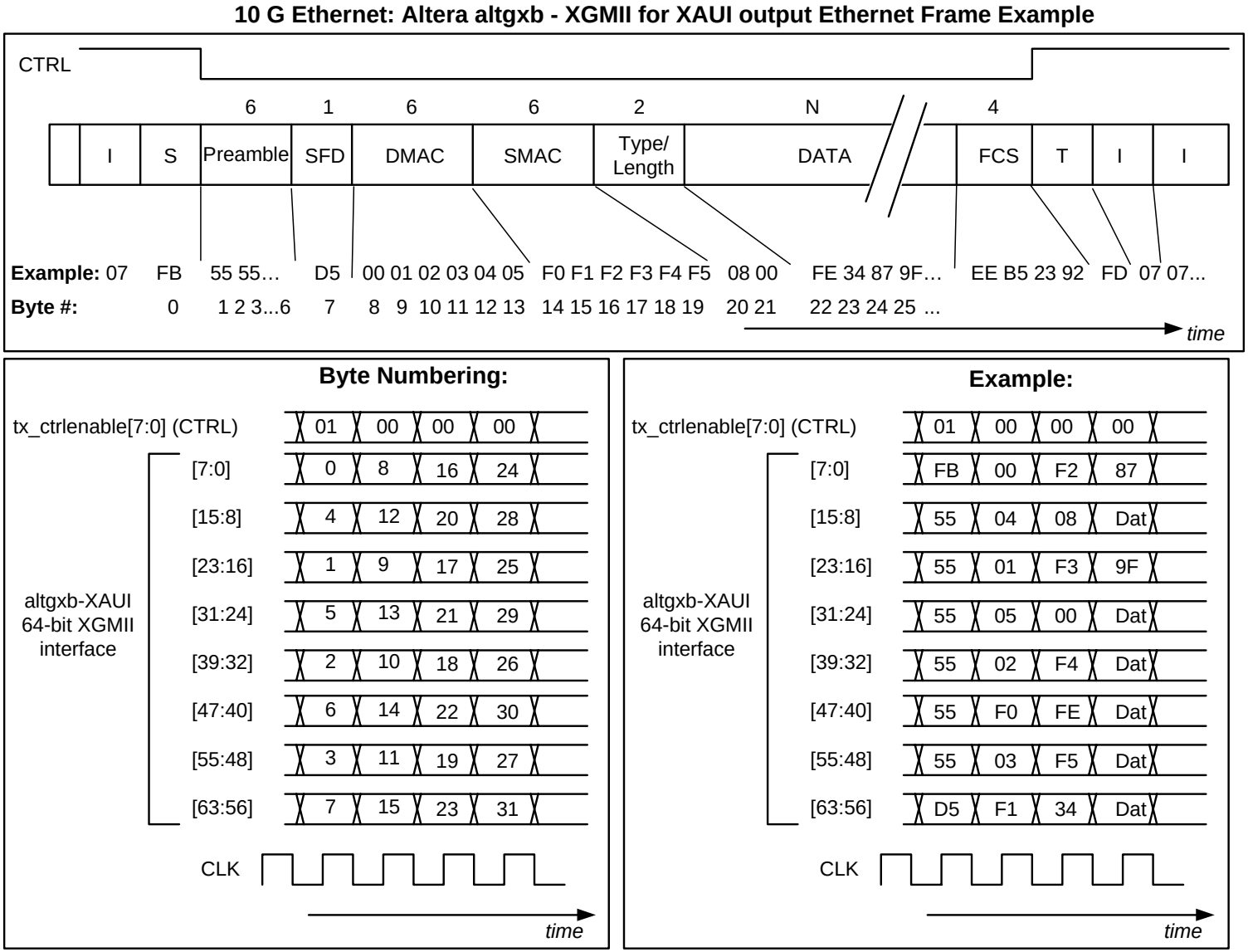


Figure 6-1 10 Gbit Ethernet modified XGMII interface byte lane assignments and PCS coding.

7 References

[1] Carlson, Brent, REQUIREMENTS AND FUNCTIONAL SPECIFICATION, LTA Controller FPGA, RFS Document: A25091N0000, Version DRAFT, October 15, 2004.

[2] Carlson, Brent, Refined EVLA WIDAR Correlator Architecture, NRC-EVLA Memo# 014, October 2, 2001.

[3] RFC 791, INTERNET PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION, September 1981.

[4] RFC 768, User Datagram Protocol, August 1980.

[5] Carlson, Brent, INTERFACE CONTROL DOCUMENT: EVLA Correlator System Numbering Plan, ICD Document A25010N0002, Revision 1.0, November 9, 2004.

[6] Carlson, Brent, PROTOCOL SPECIFICATION, HM Gbps Cable Signaling Specification, Protocol Document: A25022N0041, Revision 1.41, March 18, 2009.

[7] Carlson, Brent, REQUIREMENTS AND FUNCTIONAL SPECIFICATION, Baseline Board RXP FPGA, RFS Document: A25093N0000, Revision 2.1a, February 1, 2010.

8 Appendix I – Chip Pinouts, Pin, and Package Notes

This section contains exhaustive chip pinout listings, generated by the FPGA compiler, and notes regarding pin connectivity and use on the board. The GigE FPGA is implemented in an Altera Stratix GX EP1SGX25CF672C5 FPGA.

Pin Name/Usage	Location	Dir.	I/O Standard	Voltage	I/O Bank
GND	A2	gnd			
VCCIO4	A3	power		2.5V	4
RESERVED_INPUT_WITH_WEAK_PULLUP	A4				4
RESERVED_INPUT_WITH_WEAK_PULLUP	A5				4
LTA_CS_pad_[4]	A6	output	SSTL-2 Class I		4
LTA_CS_pad_[53]	A7	output	SSTL-2 Class I		4
LTA_CS_pad_[31]	A8	output	SSTL-2 Class I		4
LTA_CS_pad_[41]	A9	output	SSTL-2 Class I		4
LTA_CS_pad_[24]	A10	output	SSTL-2 Class I		4
GND	A11	gnd			
VCCIO4	A12	power		2.5V	4
MCB_ADDR_pad[7]	A13	input	2.5 V		10
LTA_SER_CMD_pad[5]	A14	output	SSTL-2 Class I		3
VCCIO3	A15	power		2.5V	3
GND	A16	gnd			
toPHASING_LOWER_XON_pad	A17	output	2.5 V		3
RESERVED_INPUT_WITH_WEAK_PULLUP	A18				3
LTA_CS_pad_[23]	A19	output	SSTL-2 Class I		3
LTA_CS_pad_[6]	A20	output	SSTL-2 Class I		3
RESERVED_INPUT_WITH_WEAK_PULLUP	A21				3
RESERVED_INPUT_WITH_WEAK_PULLUP	A22				3
RESERVED_INPUT_WITH_WEAK_PULLUP	A23				3
VCCIO3	A24	power		2.5V	3
GND	A25	gnd			
GBIT_RX_pad[2]	AA1	input	1.5-V PCML		15
GBIT_RX_pad[2](n)	AA2	input	1.5-V PCML		15
GXB_GND	AA3				
GBIT_TX_pad[3]	AA4	output	1.5-V PCML		15
GBIT_TX_pad[3](n)	AA5	output	1.5-V PCML		15
GND	AA6	gnd			
LTA_DATA4_pad[0]	AA7	input	SSTL-2 Class I		7
LTA_DATA5_pad[0]	AA8	input	SSTL-2 Class I		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AA9				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AA10				7
LTA_DATA3_pad[2]	AA11	input	SSTL-2 Class I		7
LTA_DATA5_pad[2]	AA12	input	SSTL-2 Class I		7
LTA_SER_RESP_pad[6]	AA13	input	SSTL-2 Class I		11
SFP_MOD_DEF2_2_pad	AA14	bidir	2.5 V		12
RESERVED_INPUT_WITH_WEAK_PULLUP	AA15				8
LTA_CS_pad_[33]	AA16	output	SSTL-2 Class I		8
LTA_CS_pad_[1]	AA17	output	SSTL-2 Class I		8
LTA_CS_pad_[61]	AA18	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AA19				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AA20				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AA21				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AA22				1
GND+	AA23				1
GND+	AA24				1
GND+	AA25				1
GND+	AA26				1
GXB_GND	AB1				
GXB_GND	AB2				
GXB_GND	AB3				
GXB_GND	AB4				
GXB_GND	AB5				
VCCINT	AB6	power		1.5V	

LTA_DATA6_pad[6]	AB7	input	SSTL-2 Class I		7
LTA_DATA7_pad[6]	AB8	input	SSTL-2 Class I		7
LTA_DATA4_pad[6]	AB9	input	SSTL-2 Class I		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AB10				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AB11				7
MCB_CLK_pad	AB12	input	2.5 V		7
LTA_SER_RESP_pad[0]	AB13	input	SSTL-2 Class I		11
LTA_SER_RESP_pad[5]	AB14	input	SSTL-2 Class I		12
RESERVED_INPUT_WITH_WEAK_PULLUP	AB15				8
LTA_CS_pad_[18]	AB16	output	SSTL-2 Class I		8
LTA_CS_pad_[13]	AB17	output	SSTL-2 Class I		8
LTA_CS_pad_[29]	AB18	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AB19				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AB20				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AB21				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AB22				1
GND+	AB23				1
GND+	AB24				1
GND+	AB25				1
GND+	AB26				1
GBIT_RX_pad[3]	AC1	input	1.5-V PCML		15
GBIT_RX_pad[3](n)	AC2	input	1.5-V PCML		15
GND	AC3	gnd			
RESERVED_INPUT_WITH_WEAK_PULLUP	AC4				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AC5				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AC6				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AC7				7
LTA_DATA1_pad[6]	AC8	input	SSTL-2 Class I		7
LTA_DATA6_pad[0]	AC9	input	SSTL-2 Class I		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AC10				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AC11				7
LTA_DATA7_pad[2]	AC12	input	SSTL-2 Class I		7
LTA_SER_RESP_pad[2]	AC13	input	SSTL-2 Class I		11
SFP_RX_LOS_pad[3]	AC14	input	2.5 V		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AC15				8
LTA_CS_pad_[27]	AC16	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AC17				8
LTA_CS_pad_[44]	AC18	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AC19				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AC20				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AC21				1
RESERVED_INPUT_WITH_WEAK_PULLUP	AC22				1
GND+	AC23				1
GND+	AC24				1
GND+	AC25				1
GND+	AC26				1
GXB_GND	AD1				
GXB_GND	AD2				
GND	AD3	gnd			
XPAK_MDIO_pad	AD4	bidir	2.5 V		7
LTA_SER_CMD_RET_pad[6]	AD5	input	SSTL-2 Class I		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AD6				7
LTA_DATA3_pad[6]	AD7	input	SSTL-2 Class I		7
LTA_DATA2_pad[0]	AD8	input	SSTL-2 Class I		7
LTA_DATA1_pad[0]	AD9	input	SSTL-2 Class I		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AD10				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AD11				7
LTA_SER_RESP_pad[3]	AD12	input	SSTL-2 Class I		11
LTA_SER_RESP_pad[4]	AD13	input	SSTL-2 Class I		11
RESERVED_INPUT_WITH_WEAK_PULLUP	AD14				8
LTA_CS_pad_[32]	AD15	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AD16				8
LTA_CS_pad_[49]	AD17	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AD18				8
LTA_CS_pad_[28]	AD19	output	SSTL-2 Class I		8
LTA_SER_CMD_RET_pad[4]	AD20	input	SSTL-2 Class I		8
LTA_CS_pad_[14]	AD21	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AD22				8
GND+	AD23				1
GND+	AD24				1
GND+	AD25				1

VCCIO1	AD26	power		2.5V	1
GND	AE1	gnd			
SFP3_LED_pad	AE2	output	2.5 V		7
SFP4_LED_pad	AE3	output	2.5 V		7
RESERVED_INPUT_WITH_WEAK_PULLUP	AE4				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AE5				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AE6				7
RESERVED_INPUT_WITH_WEAK_PULLUP	AE7				7
LTA_DATA4_pad[4]	AE8	input	SSTL-2 Class I		7
LTA_DATA0_pad[0]	AE9	input	SSTL-2 Class I		7
LTA_DATA6_pad[4]	AE10	input	SSTL-2 Class I		7
LTA_DATA7_pad[4]	AE11	input	SSTL-2 Class I		7
LTA_SER_RESP_pad[1]	AE12	input	SSTL-2 Class I		11
LTA_SER_RESP_pad[7]	AE13	input	SSTL-2 Class I		12
SFP_MOD_DEF0_1_pad	AE14	input	2.5 V		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AE15				8
LTA_CS_pad_[16]	AE16	output	SSTL-2 Class I		8
LTA_SER_CMD_RET_pad[5]	AE17	input	SSTL-2 Class I		8
LTA_CS_pad_[12]	AE18	output	SSTL-2 Class I		8
LTA_CS_pad_[62]	AE19	output	SSTL-2 Class I		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AE20				8
RESERVED_INPUT_WITH_WEAK_PULLUP	AE21				8
RESERVED_INPUT_WITH_WEAK_PULLUP	AE22				8
GND+	AE23				1
GND+	AE24				1
SFP_TX_FAULT_pad[0]	AE25	input	2.5 V		1
GND	AE26	gnd			
GND	AF2	gnd			
VCCIO7	AF3	power		2.5V	7
RESERVED_INPUT_WITH_WEAK_PULLUP	AF4				7
LTA_DATA2_pad[6]	AF5	input	SSTL-2 Class I		7
XPAK_MOD_DET_pad	AF6	input	2.5 V		7
LTA_DATA5_pad[6]	AF7	input	SSTL-2 Class I		7
LTA_DATA1_pad[4]	AF8	input	SSTL-2 Class I		7
LTA_DATA1_pad[3]	AF9	input	SSTL-2 Class I		7
LTA_DATA5_pad[4]	AF10	input	SSTL-2 Class I		7
GND	AF11	gnd			
VCCIO7	AF12	power		2.5V	7
SFP_MOD_DEF2_1_pad	AF13	bidir	2.5 V		12
RESERVED_INPUT_WITH_WEAK_PULLUP	AF14				8
VCCIO8	AF15	power		2.5V	8
GND	AF16	gnd			
LTA_CS_pad_[46]	AF17	output	SSTL-2 Class I		8
LTA_CS_pad_[30]	AF18	output	SSTL-2 Class I		8
LTA_CS_pad_[60]	AF19	output	SSTL-2 Class I		8
SFP_RX_LOS_pad[1]	AF20	input	2.5 V		8
RESERVED_INPUT_WITH_WEAK_PULLUP	AF21				8
RESERVED_INPUT_WITH_WEAK_PULLUP	AF22				8
LTA_SER_CMD_RET_pad[1]	AF23	input	SSTL-2 Class I		8
VCCIO8	AF24	power		2.5V	8
GND	AF25	gnd			
GND	B1	gnd			
RESERVED_INPUT_WITH_WEAK_PULLUP	B2				4
TP0_pad[2]	B3	output	2.5 V		4
RESERVED_INPUT_WITH_WEAK_PULLUP	B4				4
SFP_TX_FAULT_pad[1]	B5	input	2.5 V		4
RESERVED_INPUT_WITH_WEAK_PULLUP	B6				4
LTA_CS_pad_[5]	B7	output	SSTL-2 Class I		4
LTA_CS_pad_[25]	B8	output	SSTL-2 Class I		4
LTA_CS_pad_[9]	B9	output	SSTL-2 Class I		4
RESERVED_INPUT_WITH_WEAK_PULLUP	B10				4
RESERVED_INPUT_WITH_WEAK_PULLUP	B11				4
LTA_SER_CMD_pad[6]	B12	output	SSTL-2 Class I		9
MCB_CS_pad_	B13	input	2.5 V		10
CLK125_in_pad	B14	input	2.5 V		3
MCB_DATA_pad[3]	B15	bidir	2.5 V		3
LTA_CS_pad_[19]	B16	output	SSTL-2 Class I		3
XPAK_RST_pad	B17	bidir	2.5 V		3
XPAK_PRTAD_pad[0]	B18	bidir	2.5 V		3
LTA_CS_pad_[7]	B19	output	SSTL-2 Class I		3
LTA_CS_pad_[54]	B20	output	SSTL-2 Class I		3

LTA_CS_pad_[39]	B21	output	SSTL-2 Class I		3	
XPAK_PRTAD_pad[4]	B22	bidir	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	B23				3	
RESERVED_INPUT_WITH_WEAK_PULLUP	B24				3	
GND+	B25				2	
GND	B26	gnd				
SFP1_LED_pad	C1	output	2.5 V		4	
SFP2_LED_pad	C2	output	2.5 V		4	
SFP_TX_FAULT_pad[2]	C3	input	2.5 V		4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C4				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C5				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C6				4	
LTA_CS_pad_[52]	C7	output	SSTL-2 Class I		4	
LTA_CS_pad_[40]	C8	output	SSTL-2 Class I		4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C9				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C10				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	C11				4	
MCB_ADDR_pad[3]	C12	input	2.5 V		9	
LTA_SER_CMD_pad[1]	C13	output	SSTL-2 Class I		9	
MCB_ADDR_pad[5]	C14	input	2.5 V		3	
LTA_CS_pad_[3]	C15	output	SSTL-2 Class I		3	
MCB_DATA_pad[6]	C16	bidir	2.5 V		3	
XPAK_PRTAD_pad[3]	C17	bidir	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	C18				3	
LTA_SER_CMD_RET_pad[2]	C19	input	SSTL-2 Class I		3	
LTA_CS_pad_[22]	C20	output	SSTL-2 Class I		3	
LTA_CS_pad_[55]	C21	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	C22				3	
GND+	C23				2	
GND+	C24				2	
GND+	C25				2	
VCCIO2	C26	power		3.3V	2	
GXB_GND	D1					
GXB_GND	D2					
GND	D3	gnd				
RESERVED_INPUT_WITH_WEAK_PULLUP	D4				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	D5				4	
SFP_RX_LOS_pad[0]	D6	input	2.5 V		4	
LTA_CS_pad_[21]	D7	output	SSTL-2 Class I		4	
LTA_CS_pad_[63]	D8	output	SSTL-2 Class I		4	
LTA_CS_pad_[57]	D9	output	SSTL-2 Class I		4	
toPHASING_UPPER_XON_pad	D10	output	2.5 V		4	
RESERVED_INPUT_WITH_WEAK_PULLUP	D11				4	
MCB_ADDR_pad[0]	D12	input	2.5 V		4	
LTA_SER_CMD_pad[4]	D13	output	SSTL-2 Class I		9	
MCB_ADDR_pad[6]	D14	input	2.5 V		3	
LTA_DATA0_pad[3]	D15	input	SSTL-2 Class I		3	
SFP_MOD_DEF0_3_pad	D16	input	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	D17				3	
XPAK_MDC_pad	D18	bidir	2.5 V		3	
SFP_MOD_DEF0_0_pad	D19	input	2.5 V		3	
LTA_CS_pad_[38]	D20	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	D21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	D22				2	
GND+	D23				2	
GND+	D24				2	
GND+	D25				2	
GND+	D26				2	
GND*	E1					
GND*	E2					
GXB_GND	E3					
RESERVED_INPUT_WITH_WEAK_PULLUP	E4				4	
XPAK_LASI_pad	E5	input	2.5 V		4	
LTA_CS_pad_[37]	E6	output	SSTL-2 Class I		4	
LTA_CS_pad_[36]	E7	output	SSTL-2 Class I		4	
RESERVED_INPUT_WITH_WEAK_PULLUP	E8				4	
RESERVED_INPUT_WITH_WEAK_PULLUP	E9				4	
CLK3125_out_pad[0]	E10	output	SSTL-2 Class I		4	
CLK3125_out_pad[4]	E11	output	SSTL-2 Class I		4	
MCB_ADDR_pad[1]	E12	input	2.5 V		4	
LTA_SER_CMD_pad[3]	E13	output	SSTL-2 Class I		9	

MCB_RDWR_pad_	E14	input	2.5 V		10	
MCB_DATA_pad[1]	E15	bidir	2.5 V		3	
MCB_DATA_pad[2]	E16	bidir	2.5 V		3	
LTA_SER_CMD_RET_pad[0]	E17	input	SSTL-2 Class I		3	
XPAK_TX_ON_pad	E18	bidir	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	E19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	E20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	E21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	E22				2	
GND+	E23				2	
GND+	E24				2	
GND+	E25				2	
GND+	E26				2	
GXB_GND	F1					
GXB_GND	F2					
GXB_GND	F3					
GXB_GND	F4					
GXB_GND	F5					
VCCINT	F6	power		1.5V		
LTA_CS_pad_[20]	F7	output	SSTL-2 Class I		4	
CLK3125_out_pad[1]	F8	output	SSTL-2 Class I		4	
TMS	F9	input			4	
TDI	F10	input			4	
TDO	F11	output			4	
MCB_ADDR_pad[2]	F12	input	2.5 V		4	
LTA_SER_CMD_pad[7]	F13	output	SSTL-2 Class I		9	
MCB_ADDR_pad[4]	F14	input	2.5 V		10	
LTA_CS_pad_[35]	F15	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	F16				3	
MCB_DATA_pad[0]	F17	bidir	2.5 V		3	
LTA_CS_pad_[51]	F18	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	F19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	F20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	F21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	F22				2	
GND+	F23				2	
GND+	F24				2	
GND+	F25				2	
GND+	F26				2	
GND*	G1					
GND*	G2					
GXB_GND	G3					
GND*	G4					
GND*	G5					
GND	G6	gnd				
LTA_CS_pad_[47]	G7	output	SSTL-2 Class I		4	
CLK3125_out_pad[3]	G8	output	SSTL-2 Class I		4	
SFP_MOD_DEF0_2_pad	G9	input	2.5 V		4	
LTA_CS_pad_[15]	G10	output	SSTL-2 Class I		4	
CLK3125_out_pad[6]	G11	output	SSTL-2 Class I		4	
RESET_pad_	G12	input	2.5 V		4	
GNDA_PLL5	G13	gnd				
TP0_pad[0]	G14	output	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	G15				3	
XPAK_PRTAD_pad[2]	G16	bidir	2.5 V		3	
MCB_DATA_pad[5]	G17	bidir	2.5 V		3	
JTAG_CCRESET_TMS_pad	G18	output	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	G19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	G20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	G21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	G22				2	
GND+	G23				2	
GND+	G24				2	
GND+	G25				2	
GND+	G26				2	
GXB_GND	H1					
GXB_GND	H2					
GXB_GND	H3					
GXB_GND	H4					
GXB_GND	H5					
VCCINT	H6	power		1.5V		

RESERVED_INPUT_WITH_WEAK_PULLUP	H7					
RESERVED_INPUT_WITH_WEAK_PULLUP	H8				4	
TRST	H9	input			4	
~DATA0~ / RESERVED_INPUT	H10	input	2.5 V		4	
CLK3125_out_pad[2]	H11	output	SSTL-2 Class I		4	
VCCG_PLL5	H12	power		1.5V		
VCCA_PLL5	H13	power		1.5V		
SFP_RX_LOS_pad[2]	H14	input	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	H15				3	
CLK156_in_pad	H16	input	3.3-V LVTTL		3	
GND	H17					
XPAK_PRTAD_pad[1]	H18	bidir	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	H19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	H20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	H21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	H22				2	
GND+	H23				2	
GND+	H24				2	
GND+	H25				2	
GND+	H26				2	
GND*	J1					
GND*	J2					
GXB_GND	J3					
GND*	J4					
GND*	J5					
GND	J6	gnd				
LTA_CS_pad_[56]	J7	output	SSTL-2 Class I		4	
TCK	J8	input			4	
CLK3125_out_pad[7]	J9	output	SSTL-2 Class I		4	
TEMPDIODEn	J10					
TEMPDIODEp	J11					
GNDG_PLL5	J12	gnd				
VCC_PLL5_OUTA	J13	power		2.5V	9	
TP0_pad[1]	J14	output	2.5 V		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	J15				3	
VREF0B3	J16			1.25V	3	
VREF1B3	J17			1.25V	3	
VREF2B3	J18			1.25V	3	
RESERVED_INPUT_WITH_WEAK_PULLUP	J19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	J20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	J21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	J22				2	
GND+	J23				2	
GND+	J24				2	
GND+	J25				2	
GND+	J26				2	
GXB_GND	K1					
GXB_GND	K2					
GXB_GND	K3					
GXB_GND	K4					
GXB_GND	K5					
VCCINT	K6	power		1.5V		
RREFB15A	K7					
CLK3125_out_pad[5]	K8	output	SSTL-2 Class I		4	
GND	K9	gnd				
GND	K10	gnd				
GND	K11	gnd				
VCC_PLL5_OUTB	K12	power		2.5V	10	
DCLK	K13				3	
nSTATUS	K14				3	
CONF_DONE	K15				3	
LTA_SER_CMD_pad[2]	K16	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	K17				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	K18				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	K19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	K20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	K21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	K22				2	
fromPHASING_LOWER_pad[1]	K23	input	LVDS		2	
fromPHASING_LOWER_pad[1](n)	K24	input	LVDS		2	
fromPHASING_LOWER_pad[0]	K25	input	LVDS		2	

fromPHASING_LOWER_pad[0](n)	K26	input	LVDS		2	
GND*	L1					
GND*	L2					
GXB_GND	L3					
GND*	L4					
GND*	L5					
VCCT_B15	L6	power		1.5V	15	
VCCR_B15	L7	power		1.5V	15	
VCCP_B15	L8	power		1.5V	15	
VCCINT	L9	power		1.5V		
VCCINT	L10	power		1.5V		
VCCINT	L11	power		1.5V		
VCCIO4	L12	power		2.5V	4	
VCCIO3	L13	power		2.5V	3	
nCONFIG	L14				3	
MCB_DATA_pad[4]	L15	bidir	2.5 V		3	
MCB_DATA_pad[7]	L16	bidir	2.5 V		3	
GND	L17	gnd				
GND	L18	gnd				
RESERVED_INPUT_WITH_WEAK_PULLUP	L19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	L20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	L21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	L22				2	
fromPHASING_UPPER_pad[1]	L23	input	LVDS		2	
fromPHASING_UPPER_pad[1](n)	L24	input	LVDS		2	
GND+	L25				2	
GND	L26	gnd				
GXB_GND	M1					
GXB_GND	M2					
GXB_GND	M3					
GXB_GND	M4					
GXB_GND	M5					
VCCT_B15	M6	power		1.5V	15	
VCCR_B15	M7	power		1.5V	15	
VCCP_B15	M8	power		1.5V	15	
VCCINT	M9	power		1.5V		
VCCINT	M10	power		1.5V		
GND	M11	gnd				
GND	M12	gnd				
GND	M13	gnd				
VCCINT	M14	power		1.5V		
JTAG_CCRESET_TCK_pad	M15	output	2.5 V		3	
LTA_SER_CMD_pad[0]	M16	output	SSTL-2 Class I		3	
RESERVED_INPUT_WITH_WEAK_PULLUP	M17				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	M18				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	M19				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	M20				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	M21				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	M22				2	
fromPHASING_UPPER_pad[0]	M23	input	LVDS		2	
fromPHASING_UPPER_pad[0](n)	M24	input	LVDS		2	
GND+	M25				2	
VCCIO2	M26	power		3.3V	2	
GND*	N1					
GND*	N2					
GXB_GND	N3					
GND*	N4					
GND*	N5					
GND	N6	gnd				
VCCA_B15	N7	power		3.3V	15	
VCCG_B15	N8	power		1.5V	15	
VCCINT	N9	power		1.5V		
GND	N10	gnd				
VCCINT	N11	power		1.5V		
GND	N12	gnd				
VCCINT	N13	power		1.5V		
GND	N14	gnd				
VCCINT	N15	power		1.5V		
VCCINT	N16	power		1.5V		
GND	N17	gnd				
VCCIO2	N18	power		3.3V	2	

GNDG_PLL1	N19	gnd				
VCCA_PLL1	N20	power		1.5V		
VCCG_PLL1	N21	power		1.5V		
GNDA_PLL1	N22	gnd				
GND+	N23				2	
RESERVED_INPUT_WITH_WEAK_PULLUP	N24				2	
fromPHASING_CLOCK_pad	N25	input	LVDS		2	
fromPHASING_CLOCK_pad(n)	N26	input	LVDS		2	
GXB_GND	P1					
GXB_GND	P2					
GXB_GND	P3					
GXB_GND	P4					
GXB_GND	P5					
GND	P6	gnd				
VCCG_B15	P7	power		1.5V	15	
VCCA_B15	P8	power		3.3V	15	
nCE	P9				7	
VCCINT	P10	power		1.5V		
GND	P11	gnd				
VCCINT	P12	power		1.5V		
GND	P13	gnd				
VCCINT	P14	power		1.5V		
GND	P15	gnd				
VCCINT	P16	power		1.5V		
GND	P17	gnd				
VCCIO1	P18	power		2.5V	1	
GNDG_PLL2	P19	gnd				
VCCG_PLL2	P20	power		1.5V		
VCCA_PLL2	P21	power		1.5V		
GNDA_PLL2	P22	gnd				
LTA_DATA7_pad[3]	P23	input	SSTL-2 Class I		1	
LTA_DATA5_pad[3]	P24	input	SSTL-2 Class I		1	
LTA_DATA4_pad[3]	P25	input	SSTL-2 Class I		1	
LTA_DATA6_pad[3]	P26	input	SSTL-2 Class I		1	
GBIT_RX_pad[1]	R1	input	1.5-V PCML		15	
GBIT_RX_pad[1](n)	R2	input	1.5-V PCML		15	
GXB_GND	R3					
GBIT_TX_pad[1]	R4	output	1.5-V PCML		15	
GBIT_TX_pad[1](n)	R5	output	1.5-V PCML		15	
VCCT_B15	R6	power		1.5V	15	
VCCR_B15	R7	power		1.5V	15	
VCCP_B15	R8	power		1.5V	15	
VCCSEL	R9				7	
LTA_DATA6_pad[2]	R10	input	SSTL-2 Class I		7	
VCC_PLL6_OUTB	R11	power		2.5V	12	
GND	R12	gnd				
LTA_CS_pad_[10]	R13	output	SSTL-2 Class I		8	
GND	R14	gnd				
LTA_CS_pad_[58]	R15	output	SSTL-2 Class I		8	
VREF0B1	R16			1.25V	1	
LTA_DATA0_pad[1]	R17	input	SSTL-2 Class I		1	
LTA_DATA5_pad[1]	R18	input	SSTL-2 Class I		1	
LTA_DATA1_pad[1]	R19	input	SSTL-2 Class I		1	
LTA_DATA2_pad[1]	R20	input	SSTL-2 Class I		1	
LTA_DATA0_pad[5]	R21	input	SSTL-2 Class I		1	
LTA_DATA6_pad[5]	R22	input	SSTL-2 Class I		1	
LTA_DATA4_pad[5]	R23	input	SSTL-2 Class I		1	
LTA_DATA1_pad[5]	R24	input	SSTL-2 Class I		1	
LTA_DATA6_pad[1]	R25	input	SSTL-2 Class I		1	
VCCIO1	R26	power		2.5V	1	
GXB_GND	T1					
GXB_GND	T2					
GXB_GND	T3					
GXB_GND	T4					
GXB_GND	T5					
VCCT_B15	T6	power		1.5V	15	
VCCR_B15	T7	power		1.5V	15	
VCCP_B15	T8	power		1.5V	15	
nCEO	T9				7	
nIO_PULLUP	T10				7	
VCCA_PLL6	T11	power		1.5V		

VCCIO7	T12	power		2.5V	7	
MSEL1	T13				8	
VCCIO8	T14	power		2.5V	8	
LTA_CS_pad_[48]	T15	output	SSTL-2 Class I		8	
LTA_CS_pad_[17]	T16	output	SSTL-2 Class I		8	
VREF2B1	T17			1.25V	1	
VREF1B1	T18			1.25V	1	
LTA_DATA5_pad[5]	T19	input	SSTL-2 Class I		1	
LTA_DATA2_pad[5]	T20	input	SSTL-2 Class I		1	
LTA_FRAME_pad_[7]	T21	input	SSTL-2 Class I		1	
LTA_FRAME_pad_[6]	T22	input	SSTL-2 Class I		1	
LTA_FRAME_pad_[4]	T23	input	SSTL-2 Class I		1	
LTA_FRAME_pad_[5]	T24	input	SSTL-2 Class I		1	
LTA_DATA4_pad[1]	T25	input	SSTL-2 Class I		1	
GND	T26	gnd				
GBIT_RX_pad[0]	U1	input	1.5-V PCML		15	
GBIT_RX_pad[0](n)	U2	input	1.5-V PCML		15	
GXB_GND	U3					
GBIT_TX_pad[0]	U4	output	1.5-V PCML		15	
GBIT_TX_pad[0](n)	U5	output	1.5-V PCML		15	
GND	U6	gnd				
RREFB15	U7					
LTA_DATA3_pad[4]	U8	input	SSTL-2 Class I		7	
LTA_CS_pad_[8]	U9	output	SSTL-2 Class I		7	
PORSEL	U10				7	
LTA_DATA2_pad[4]	U11	input	SSTL-2 Class I		7	
GNDG_PLL6	U12	gnd				
MSEL0	U13				8	
LTA_CS_pad_[26]	U14	output	SSTL-2 Class I		8	
LTA_CS_pad_[11]	U15	output	SSTL-2 Class I		8	
LTA_CS_pad_[0]	U16	output	SSTL-2 Class I		8	
LTA_FRAME_pad_[3]	U17	input	SSTL-2 Class I		1	
LTA_FRAME_pad_[0]	U18	input	SSTL-2 Class I		1	
LTA_DATA2_pad[7]	U19	input	SSTL-2 Class I		1	
LTA_DATA0_pad[7]	U20	input	SSTL-2 Class I		1	
SFP_TX_DISABLE_pad	U21	output	2.5 V		1	
SFP_MOD_DEF1_pad	U22	bidir	2.5 V		1	
LTA_DATA3_pad[7]	U23	input	SSTL-2 Class I		1	
LTA_DATA1_pad[7]	U24	input	SSTL-2 Class I		1	
LTA_DATA7_pad[1]	U25	input	SSTL-2 Class I		1	
LTA_DATA3_pad[1]	U26	input	SSTL-2 Class I		1	
GXB_GND	V1					
GXB_GND	V2					
GXB_GND	V3					
GXB_GND	V4					
GXB_GND	V5					
VCCINT	V6	power		1.5V		
~GXB_RESERVED_IO~0	V7	output	3.3-V LVTTL		7	
LTA_DATA0_pad[2]	V8	input	SSTL-2 Class I		7	
VREF2B7	V9			1.25V	7	
VREF1B7	V10			1.25V	7	
VREF0B7	V11			1.25V	7	
VCCG_PLL6	V12	power		1.5V		
VCC_PLL6_OUTA	V13	power		2.5V	11	
MSEL2	V14				8	
VREF2B8	V15			1.25V	8	
LTA_CS_pad_[43]	V16	output	SSTL-2 Class I		8	
VREF1B8	V17			1.25V	8	
VREF0B8	V18			1.25V	8	
LTA_DATA4_pad[7]	V19	input	SSTL-2 Class I		1	
LTA_DATA7_pad[7]	V20	input	SSTL-2 Class I		1	
LTA_SER_CMD_RET_pad[3]	V21	input	SSTL-2 Class I		1	
SFP_RATE_SEL_pad	V22	output	2.5 V		1	
GND+	V23				1	
LTA_SER_CMD_RET_pad[7]	V24	input	SSTL-2 Class I		1	
LTA_DATA7_pad[5]	V25	input	SSTL-2 Class I		1	
LTA_DATA3_pad[5]	V26	input	SSTL-2 Class I		1	
GXB_VCC*	W1			1.5V	15	
GXB_GND*	W2				15	
GXB_GND	W3					
GBIT_TX_pad[2]	W4	output	1.5-V PCML		15	

```
GBIT_TX_pad[2](n)            : W5        : output : 1.5-V PCML        :         : 15        :
GND                          : W6        : gnd    :                   :         :           :
RESERVED_INPUT_WITH_WEAK_PULLUP : W7     :        :                   :         : 7         :
RESERVED_INPUT_WITH_WEAK_PULLUP : W8     :        :                   :         : 7         :
LTA_DATA3_pad[0]             : W9        : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA0_pad[4]             : W10       : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA1_pad[2]             : W11       : input  : SSTL-2 Class I    :         : 7         :
GNDA_PLL6                    : W12       : gnd    :                   :         :           :
RESERVED_INPUT_WITH_WEAK_PULLUP : W13    :        :                   :         : 8         :
RESERVED_INPUT_WITH_WEAK_PULLUP : W14    :        :                   :         : 8         :
PLL_ENA                      : W15       :        :                   :         : 8         :
LTA_CS_pad_[59]              : W16       : output : SSTL-2 Class I    :         : 8         :
RESERVED_INPUT_WITH_WEAK_PULLUP : W17    :        :                   :         : 8         :
LTA_CS_pad_[50]              : W18       : output : SSTL-2 Class I    :         : 8         :
SFP_MOD_DEF2_0_pad           : W19       : bidir  : 2.5 V             :         : 1         :
LTA_DATA2_pad[3]             : W20       : input  : SSTL-2 Class I    :         : 1         :
RESERVED_INPUT_WITH_WEAK_PULLUP : W21    :        :                   :         : 1         :
RESERVED_INPUT_WITH_WEAK_PULLUP : W22    :        :                   :         : 1         :
LTA_DATA5_pad[7]             : W23       : input  : SSTL-2 Class I    :         : 1         :
LTA_DATA6_pad[7]             : W24       : input  : SSTL-2 Class I    :         : 1         :
LTA_FRAME_pad_[2]            : W25       : input  : SSTL-2 Class I    :         : 1         :
LTA_FRAME_pad_[1]            : W26       : input  : SSTL-2 Class I    :         : 1         :
GXB_GND                      : Y1        :        :                   :         :           :
GXB_GND                      : Y2        :        :                   :         :           :
GXB_GND                      : Y3        :        :                   :         :           :
GXB_GND                      : Y4        :        :                   :         :           :
GXB_GND                      : Y5        :        :                   :         :           :
VCCINT                       : Y6        : power  :                   : 1.5V    :           :
LTA_DATA0_pad[6]             : Y7        : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA7_pad[0]             : Y8        : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA3_pad[3]             : Y9        : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA2_pad[2]             : Y10       : input  : SSTL-2 Class I    :         : 7         :
LTA_DATA4_pad[2]             : Y11       : input  : SSTL-2 Class I    :         : 7         :
CLK133_in_pad                : Y12       : input  : 2.5 V             :         : 7         :
RESERVED_INPUT_WITH_WEAK_PULLUP : Y13    :        :                   :         : 8         :
RESERVED_INPUT_WITH_WEAK_PULLUP : Y14    :        :                   :         : 8         :
LTA_CS_pad_[42]              : Y15       : output : SSTL-2 Class I    :         : 8         :
LTA_CS_pad_[2]               : Y16       : output : SSTL-2 Class I    :         : 8         :
LTA_CS_pad_[45]              : Y17       : output : SSTL-2 Class I    :         : 8         :
LTA_CS_pad_[34]              : Y18       : output : SSTL-2 Class I    :         : 8         :
TP0_pad[3]                   : Y19       : output : 2.5 V             :         : 1         :
RESERVED_INPUT_WITH_WEAK_PULLUP : Y20    :        :                   :         : 1         :
RESERVED_INPUT_WITH_WEAK_PULLUP : Y21    :        :                   :         : 1         :
SFP_MOD_DEF2_3_pad           : Y22       : bidir  : 2.5 V             :         : 1         :
SFP_TX_FAULT_pad[3]          : Y23       : input  : 2.5 V             :         : 1         :
GND+                         : Y24       :        :                   :         : 1         :
GND+                         : Y25       :        :                   :         : 1         :
GND+                         : Y26       :        :                   :         : 1         :
-------------------------------------------------------------------------------------------
 -- NC           : No Connect. This pin has no internal connection to the device.
 -- DNU          : Do Not Use. This pin MUST NOT be connected.
 -- VCCINT       : Dedicated power pin, which MUST be connected to VCC  (1.5V).
 -- VCCIO        : Dedicated power pin, which MUST be connected to VCC
 --                of its bank.
 --                                    Bank 1:          2.5V
 --                                    Bank 2:          3.3V
 --                                    Bank 3:          2.5V
 --                                    Bank 4:          2.5V
 --                                    Bank 7:          2.5V
 --                                    Bank 8:          2.5V
 --                                    Bank 9:          2.5V
 --                                    Bank 10: 2.5V
 --                                    Bank 11: 2.5V
 --                                    Bank 12: 2.5V
 -- GND          : Dedicated ground pin. Dedicated GND pins MUST be connected to GND.
 -- GND+         : Unused input pin. It can also be used to report unused dual-purpose pins.
 --                                    This pin should be connected to GND.
 -- GND*         : Unused  I/O  pin.
 -- RESERVED     : Unused I/O pin, which MUST be left unconnected.
 -- RESERVED_INPUT    : Pin is tri-stated and should be connected to the board.
 -- RESERVED_INPUT_WITH_WEAK_PULLUP    : Pin is tri-stated with internal weak pull-up resistor.
 -- RESERVED_INPUT_WITH_BUS_HOLD       : Pin is tri-stated with bus-hold circuitry.
 -- RESERVED_OUTPUT_DRIVEN_HIGH        : Pin is output driven high.
 -- GXB_GND*     : Unused GXB Receiver, Transmitter or dedicated reference clock pin. This pin
```

```
--                 can be left unconnected or connected to GXB_GND through a 10k Ohm resistor.
--                 Connecting this pin to GXB_GND through a 10k Ohm resistor may improve the
--                 device's immunity to noise.
-- GXB_VCC*      : Unused GXB Receiver, Transmitter or dedicated reference clock pin. This pin
--                 can be left unconnected or connected to GXB_VCC through a 10k Ohm resistor.
--                 Connecting this pin to GXB_VCC through a 10k Ohm resistor may improve the
--                 device's immunity to noise.
-- GXB_RESERVED_IO    : Pin is reserved to avoid coupling noise into transceivers and should be
--                      left unconnected on the board.
-------------------------------------------------------------------------------------------
```

8.1 Notes on Activating the 10 G XPAK Interface

FPGA pinouts are identical for the 10 GB design, except that the GBIT_TX_pad[3:0] and GBIT_RX_pad[3:0] are 4 transmit and receive pairs that form a XAUI interface, with each pair running at 3.125 Gbps.

The following steps must be taken to activate 10 G capability on the board:

1. Remove 16 0 ohm resistors R205-R220 inclusive. These are located on the FRONT of the PCB, near the FPGA vias, within a silkscreen box labeled "REMOVE FOR 10 GE". Removal of these resistors disconnects GBIT_TX_pad[3:0] and GBIT_RX_pad[3:0] from the SFP modules, leaving connections to the 10 G XPAK connector intact (i.e. the XPAK connector connections are always in place—the XPAK connector is a short stub on these lines when SFP modules are used).

2. Remove the dust plugs, and install the XPAK transponder module in the XPAK cage. Example XPAK transponder modules are the Finisar V23833-G6005-A111, and the Merge Optics TRP10GVP2027. The SFP modules can be left in place or removed.

3. Configure the FPGA using the 10 G FPGA FPGA binary "bb_gbitV2_10g_MmmD-YY.rbf"

With 10 G capability, the aggregate output data rate of the entire board has increased by nearly a factor of 10, although maximum individual correlator chip output data rates and individual VDIF packet data rates are still governed by the bandwidth of connections from the GigE FPGA to the LTAs and the RXPs respectively.

Note that in 10 G mode, both LTA data frames and VDIF data frames are now output on the same interface. Refer to register descriptions in this RFS for exceptions regarding 10 G processing.

Note that the 10 G design has been tested thoroughly on V2.1 and V2.2 boards, but most boards have not been individually tested with XPAK modules. It is possible that some auxiliary circuitry associated with the XPAK module could have an undetected component or manufacturing defect.

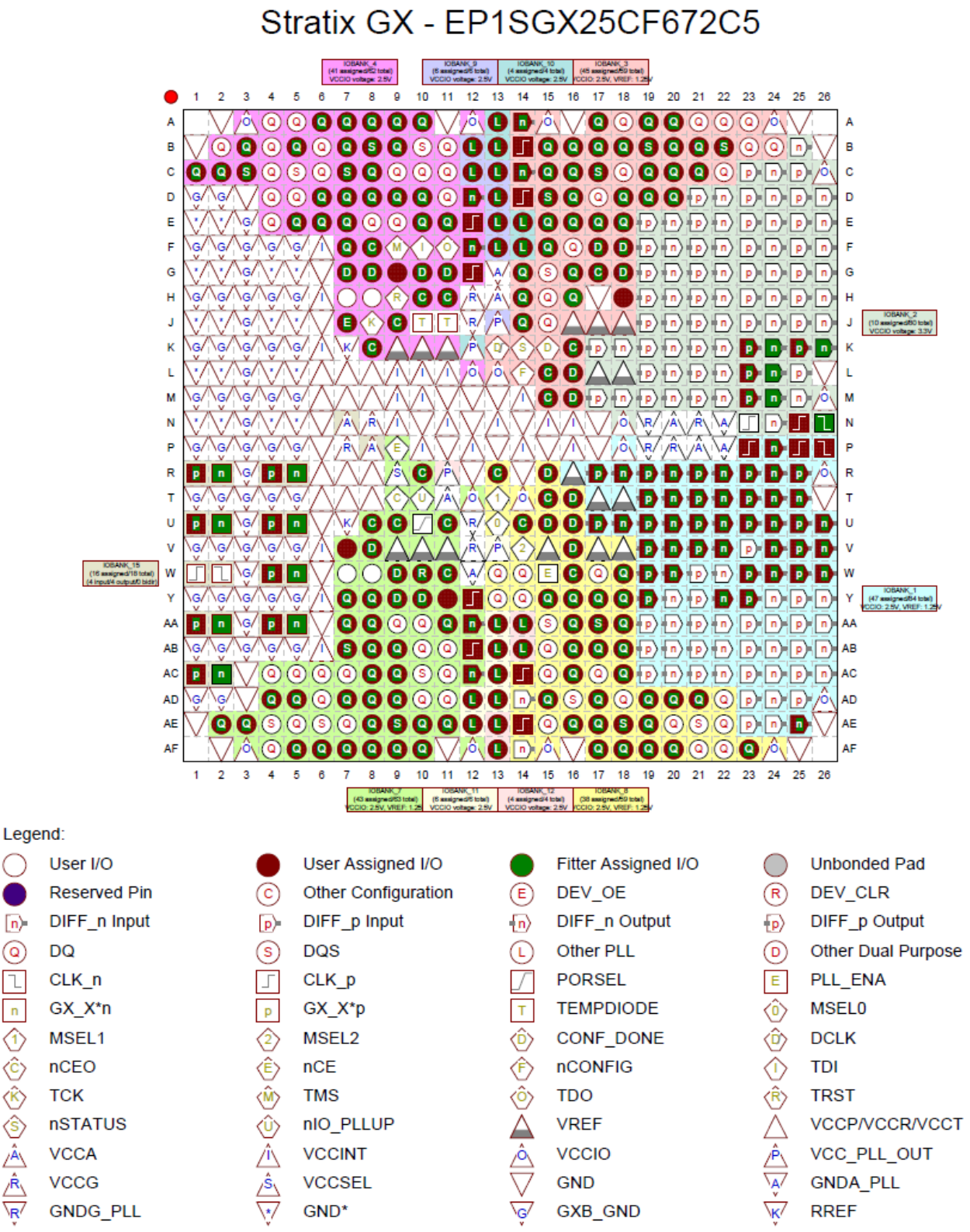
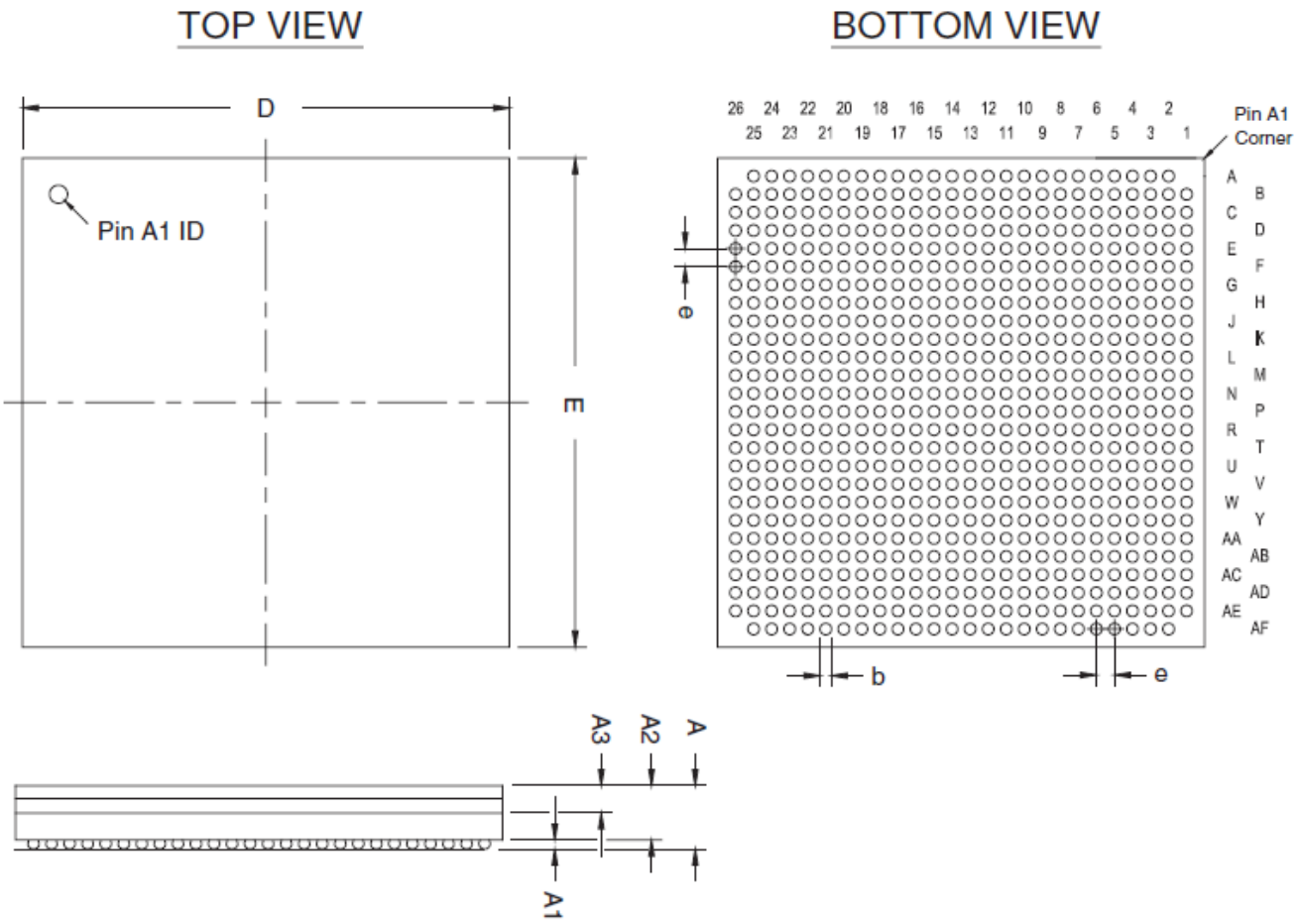


Figure 8-1 GigE FPGA Altera Quartus-II pin planner output. Top view of the chip.

8.2 Altera EP1SGX25CF672C5 Package Drawing



Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	27.00 BSC		
E	27.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 8-2 GigE FPGA F672 package outline and dimensions. Drawings and dimensions taken from the Altera Stratix-II package information data sheet.

8.3 Altera FPGA Programming Notes

The Baseline Board on which the GigE FPGA resides is set up for 1-bit Passive Serial programming via the PCMC FPGA. As there is only one FPGA in the programming path, generating of the raw binary file (.rbf) is straightforward, via the "Convert Programming Files" dialog in the Altera Quartus-II software, shown below:

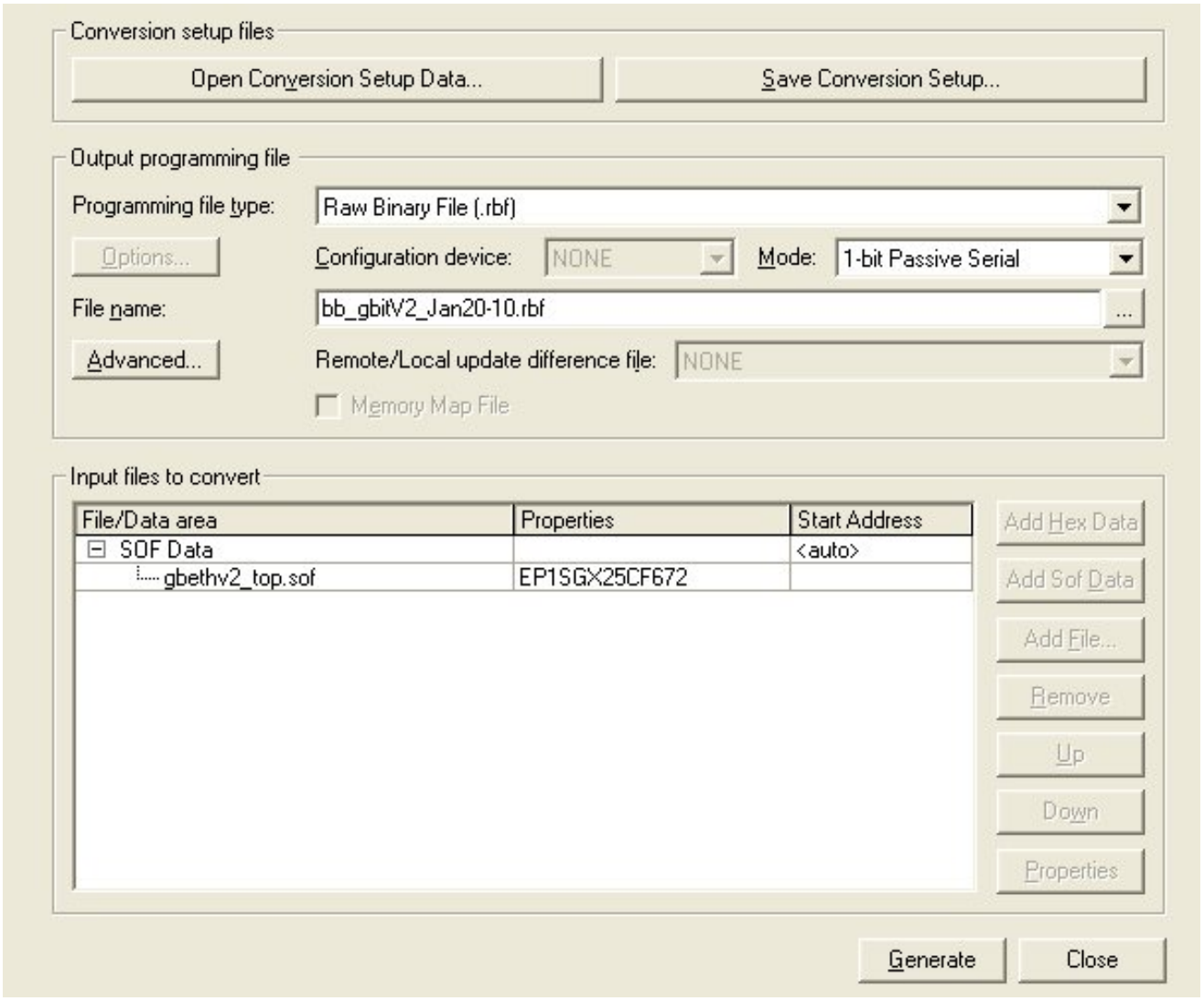


Figure 8-3 Altera Quartus-II "Convert Programming Files" screenshot showing settings necessary for generating the .rbf file for the GigE FPGA. The 10 G FPGA design is nearly identical, except that the output .rbf file is named differently (e.g. "bb_gbitV2_10g_Jan20-10.rbf") to differentiate it from the 1 GigE design.

9 Index

G

H

I

J

L

M

O

P

R

S

T