LatticeSCM XAUI to SPI4.2

July 2008 Reference Design RD1033

Introduction

The XAUI to SPI4.2 (X2S4) Bridge reference design is a cost-effective system solution for bridging SPI4.2 based network processors and 10G/10G+ Ethernet switching devices. On the XAUI side, the X2S4 optionally supports the HiGig™ protocol from Broadcom® and an over-clocked mode of operation supports the Broadcom HiGig+™ protocol with throughput rates of up to 12Gbps at the link layer. This solution leverages existing proven IP cores that include the LatticeSCM™ SPI4.2 hard IP block and the Ten-Gigabit Plus Ethernet Media Access Controller soft IP core as major building blocks of the design. This solution also takes advantage of LatticeSCM built-in features such as System Bus for microprocessor access and PCS SERDES high-speed serial I/O. These IP components are fixed entities for all bridging applications and are delivered in a IP protected format (encrypted). They can be altered in terms of behavior only so far as the existing supported programming options for those IP cores allow. The Bridge Interface Logic (BIL) that glues all of the IP together forming the bridge is, however, expected to be customized for each user application realizing that each has its own unique methods for achieving similar end results. This part of the design, as well as a comprehensive test-bench, is delivered in unprotected rtl format.

The X2S4 reference design package includes a full development level infrastructure starting from relevant documentation to “ready-to-load” bitstream including everything in between such as completed directories loaded with source rtl, IP, test-bench source, rtl simulation scripts and results, gate simulation scripts and results, ispLEVER® place and route, etc. Users will find the comprehensive simulation environment (control and data planes) especially useful in supporting verification of customizations they wish to make to the reference design.

As indicated previously, the X2S4 Bridge reference design incorporates the LatticeSCM SPI4.2 MACO hard IP core and the 10G+ Ethernet MAC (10G+MAC) IP core. The reference design package includes pre-generated versions of these cores to implement a fully functional solution. An IP license for the LatticeSCM SPI4.2 hard IP, available free of charge, is required to fully evaluate the complete reference design. An IP license for the 10G+MAC is required to use the reference design in a final design. Additional information on the licensing requirements is provided in the IP Licensing section of this document.

The X2S4 Bridge was developed specifically for the LatticeSCM series FPGA family with particular focus on the LFSCM15 device matched with a small-sized 256-pin package. It targets two specific network processors (EZChip and Intel) and Ethernet switch manufacturers (Broadcom and Marvell) and as such draws heavily upon their architectures in establishing bridge functionality.

Features

- SPI4.2 IP Core Features:
  - Fully compliant with the OIF System Packet Interface Level 4 Phase 2 Revision 1 (SPI4.2.1) interface standard.
  - Support for up to 256 independent channels (single channel for this X2S4 Bridge).
  - 156 to 500MHz DDR Dynamic timing mode operations.
  - Requires only 865 slices for a full 256-channel Dynamic mode core.
  - Parity error checking/generation on all receive and transmit control and data words (DIP4) and status (DIP2) interfaces. Also includes SPI4.2 protocol checking and error reporting.
  - Parity error force capabilities on data (independent controls: control word and data) and status interfaces.
  - Various run-time user controls: Individual receiver/transmitter resets, De-skew only reset, Adaptive Input Logic only reset, Force idles (transmitter), Enable/Disable Packing (transmitter)
  - Complete run-time programmability of all internal FIFO thresholds for efficient management of SPI4.2 line in terms of L (max) and packing.
  - Supports minimum transmit burst sizes in increments of 16 bytes from 16 bytes up to 1008 bytes for optimized Network Processor applications.

  - Fully configurable 512-location calendar RAM for Rx and Tx directions and associated 256-location status RAMs.
  - 8K-byte shared buffer space for both transmit and receive directions.

- 10G+MAC IP Core Features:
  - Compliant to IEEE 802.3-2005 standard
  - Optional Broadcom HiGig and HiGig+ format support.
  - 64-bit wide internal data path operating at 156.25-187 MHz supporting up to 12Gbps of throughput.
  - XAUI interface to the PHY layer (using PCS/SERDES external to the core)
  - Optional Multicast address filtering
  - Transmit and Receive statistics vector
  - Programmable Inter Frame Gap (including reduced HiGig IFG)
  - Supports:
    - Full duplex operation
    - Flow control using PAUSE frames (detect and act upon in receive direction, pause frame creation on command in transmit direction)
    - Optional Pause Frame drop (egress)
    - Automatic padding of short frames
    - Optional FCS generation during transmission
    - Optional FCS stripping during reception
    - Jumbo frames up to 16k
    - Interframe Stretch Mode during transmission
  - 2K transmit and receive buffers
  - Complete statistics support (reduced set for X2S4 Bridging)

- System Bus Features:
  - Full LatticeSCM built-in hard block System Bus resources operating in User Master mode. Two external interface styles (“External UMI” and JTAG) are provided as part of the reference design. An optional PowerPC style external interface (not available in the 256 fpBGA package) is not part of the standard package but can be easily added by the user).
  - SPI4.2 IP core supported by System Bus and Serial Management Interface (SMI) for in-circuit controllability and observability without using general logic resources.
  - PCS SERDES supported by the SMI for in-circuit controllability and observability without using general logic resources.
  - Provides full read/write access to all programmable registers for the SPI4.2 IP core, 10G+MAC IP core, and Bridge.

- Bridge Features:
  - Limited MAC and PCS statistics collection (MAC supports complete statistics data profile - collection addition/subtractions can easily be customized by the user).
  - Fully maskable interrupt structure and composite interrupt I/O signal pin for error notification and reporting.
  - User-configurable Shared Ingress and Egress FIFOs size (default = 32K byte each).
  - Store & Forward architecture used in the Ingress direction. Store & Forward or Cut-Thru is user selectable in the Egress direction.
  - Several options for error handling including error drop, packet filtering / correction (missing sop or eop), and mark and pass.
  - Graceful overflow drop dropping only the packet that causes an overflow.
  - Translation between Ethernet Pause based flow control and SPI4.2 Status based flow control
  - Two types of flow control (two-tiered, pass thru) offered.
  - Full bandwidth (10G/12G) Bridging

Scope

This document focuses on the Bridge Interface Logic (BIL) and control infrastructure of the Bridge leaving details of the major IP blocks to their respective stand-alone documents. These documents are listed in the Reference Information section below.

Reference Information

IP core documentation is available now on the Lattice Semiconductor website at www.latticesemi.com.

Information for the following system components are contained in the following documents:

- IPUG44 - LatticeSCM SPI4.2 MACO™ Core User's Guide
- IPUG39 - Ten-Gigabit Ethernet Plus Media Access Controller (10G+MAC) User's Guide
- TN1085 - Lattice SC/MPI System Bus
- DS1005 - LatticeSC/M Family flexiPCS™ Data Sheet
- RD1034 - Lattice XAUI To SPI4.2 Simulation Plan
- RD1035 - Lattice XAUI To SPI4.2 Hardware Evaluation Report

System-Level Block Diagram

Figures 1 and 2 show system level diagrams of the typical line card to Ethernet switch applications for which the X2S4 Bridge is intended. In these cases, the X2S4 design provides a bridging function between an SPI4.2 based Network Processor (NP) and Ethernet-based switching device allowing for efficient flow of data packets in either direction across both interface types.

The implementation chosen is based on the premise that direct connection between the NPU and the switch devices would be the ideal connection method if it were possible. It is with this thought that the X2S4 bridge design is constructed to appear as transparent as is possible in its action of exchanging data between the SPI4.2 and Ethernet endpoints.

Figure 1. 10GE Line Card Example

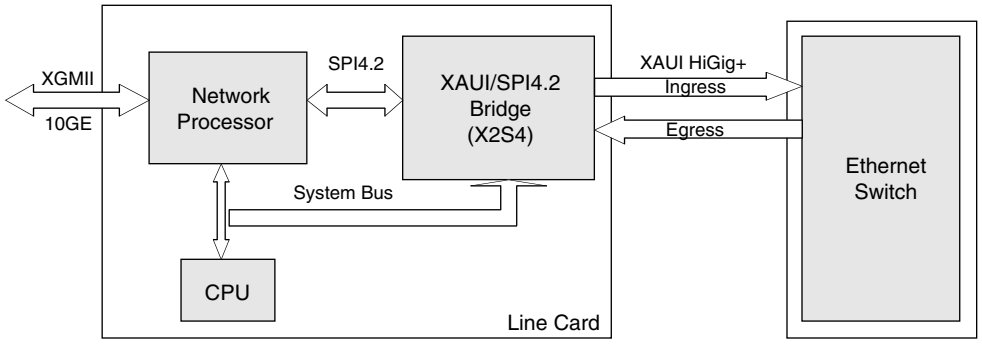
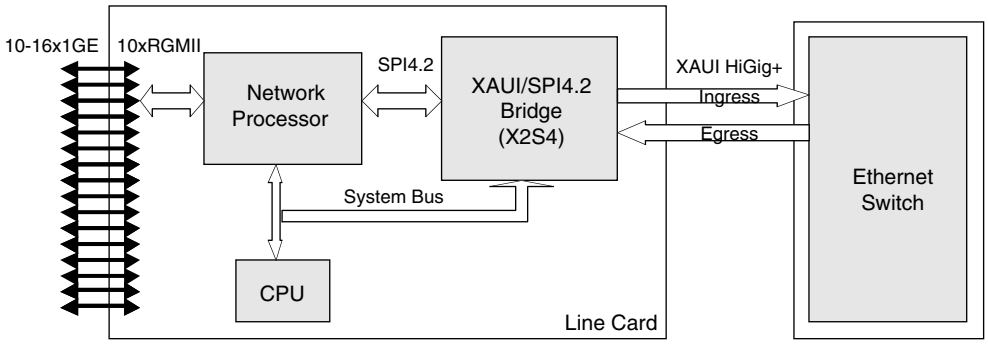


Figure 2. 10 - 16x 1GE Line Card Example



X2S4 Bridge Overview and Background Information

Inherent, and in some ways limiting, characteristics of the XAUI 10/12G Ethernet interface dictate many design aspects of this version of the X2S4 Bridge, ranging from channelization, to buffer configurations, and to flow control that extend across the bridge to the operational mode of the SPI4.2 interface. In particular, the absence of a multi-channel packet inter-leave capability in the Ethernet protocol means that the SPI4.2 bus does not need to inter-leave (i.e., it can transmit full packets, like Ethernet, instead of fragments from multiple packets on multiple channels). Because packets are transferred in whole, there is no need for a multi-channel buffer configuration to re-constitute full packets. Instead, a shared buffer arrangement works fine. The absence of a multi-channel flow control capability in Ethernet means that the SPI4.2 bus will also operate in a single channel context as it relates to flow control. This means that, like Ethernet Pause (X_On/X_Off), when the SPI4.2 bus is flow controlled, data transmission stops completely; not on a per-channel basis.

Flow control at the individual MPHY port level shown in Figure 2 is addressed outside of the bridge through alternative line-card-to-line-card communication. It is possible, based on user needs, to operate the SPI4.2 data interface in multi-channel mode for the purposes of channel identification (but still operating non-interleaved). The SPI4.2 channel number can then be translated and used in support of VLAN tagging for example as a possible customer based customization - this is not currently part of the design.

Lastly, line card to line-card flow control is supported in the HiGig2 version that would allow translation of SPI4.2 status channel number to flow control messaging. This may be addressed in a future version of the Bridge.

Flow Control

Two Tiered Approach

Flow control is viewed in the context of the full 10Gbps pipe as it relates to the exchange of data across the bridge in either direction. There are actually two independent “end-to-end” flow control segments, the first between NPU and Bridge, and the second between Bridge and Ethernet device. A two-tiered flow control structure is used where it is possible to flow control only one segment without flow controlling the other due to the internal Bridge FIFOs. For this configuration, the second segment would be flow controlled only if the internal Bridge FIFO has reached high water on the first segment. Within this approach, the X2S4 Bridge offers an option to either pass flow control information across the bridge immediately as it is encountered, or wait until the programmable FIFO threshold is reached. In either case, the X2S4 Bridge stops data transmission at the Bridge immediately when flow control is encountered. In terms of system analysis and setting of FIFO threshold values, the X2S4 Bridge will therefore need to be able to absorb all committed data from the transmitting end from the point where it passes flow control information to the transmitting end.

Translation

The standard Ethernet based Pause (X_On, X_Off) flow control scheme is indirectly translated into SPI4.2 based status and vice versa via the X2S4 internal Bridge FIFOs high and low water marks. Simple two state X_On (starved)/X_Off (satisfied) signaling is used for both the Ethernet and SPI4.2 interfaces. SPI4.2 Credit accounting through MaxBurst1 and MaxBurst2 parameters is not supported.

In the Ingress direction, as the MAC responds to Pause frames it receives by inhibiting Ingress transmit data flow, the X2S4 Ingress FIFO fill level builds resulting in SPI4.2 Status flow control. The MAC IP core provides Pause Quanta timing and clearing via X_On as well as a composite Pause signal revealing the Pause state of the MAC transmitter to the Bridge Interface Logic. In addition, Pause flow control frames received via the MAC can be optionally dropped inside the MAC so that they are not passed to the NPU. The X2S4 Bridge also offers a path into the SPI4.2 Status channel from the local Microprocessor interface.

In the Egress direction, when the SPI4.2 status channel is active (Satisfied), it inhibits the movement of data from the internal Egress FIFO to the SPI4.2 transmit FIFO. As the fill level builds inside the Egress FIFO, a programmable high-water mark is reached and results in the creation of a Pause control frame loaded with a Pause Quanta value from a locally programmed register. The 10Gbps MAC IP core contains the Pause Frame Generator but is controlled through an Egress Status Controller (ESC) outside the IP core. Extra long periods of flow control will result in the creation and transmission of multiple Pause control frames in order to maintain the flow control state.

Bridge Initiated Flow Control

The SPI4.2 interface is expected to have, in most cases, considerable over-speed relative to the Ethernet link. An SPI4.2 interface operating with a 400-500MHz DDR clock will yield between ~12.8Gbps and 16Gbps of bandwidth. The Ethernet link operating over an over-clocked XAUI+ interface is expected to yield at most, 12Gbps of bandwidth. This being the case, the X2S4 Bridge is not able get rid of data fast enough in the Ingress direction and requires periodic flow control during sustained periods of NPU data transmission. Bridge initiated flow control is based on the Ingress FIFO fill level. Thresholds for all FIFOs are completely programmable in order to smooth out data transmission.

In the Egress direction, the normal segmentation capabilities of the SPI4.2 protocol address the over-speed issue. In this direction, once the Bridge Interface Logic begins loading a packet into the SPI4.2 transmit FIFO, the SPI4.2 over-speed will quickly over-run the data source and appropriately segment the SPI4.2 line when it runs out of data (see SPI4.2 Burst Mode described in the LatticeSCM SPI4.2 IP Core User's Guide).

QoS Considerations

When considering transmitter functionality at both the NPU and Fabric ends of the bridge, the data has already been shaped in terms of QoS priority and so there was no compelling reason to implement any type of additional QoS scheduling capability inside the bridge.

Packet Drop

It is believed that all error'ed packets entering the network through line cards will be dropped by the NPU and therefore any error'ed packets detected in the bridge (Ethernet CRC, SPI4 DIP4, and Abort) are device-to-device errors on the same board/system. For this topology, error'ed packets are expected to occur at extremely low rates due to bad hardware. Bad packets detected in the bridge can, however, be optionally dropped or they can be marked and passed as they occur so they can be dropped at the next device node in the data path. Both the SPI4 and Ethernet protocols allow for marking all transmitted packets as bad and is useful in cut-thru modes since drop is not possible.

CRC Generation / Checking

Several options exist for CRC generating and checking inside the MAC in support of the XAUI interface. See the 10G+MAC User's Guide for details.

Store & Forward - Cut Thru

In the Ingress direction, full packets must be stored before beginning Ethernet transmission due to the burst capability of the SPI4 interface. Store & Forward architecture is therefore used in the Ingress direction and is not user-selectable without rtl modification of the Bridge logic. In the Egress direction, either mode can be selected via a parameter based on user needs. Cut-thru architectures are preferred in many cases due to low latency. Store & Forward is sometimes preferred because packet drop can be supported.

Packet Formats - HiGig Header Overlay

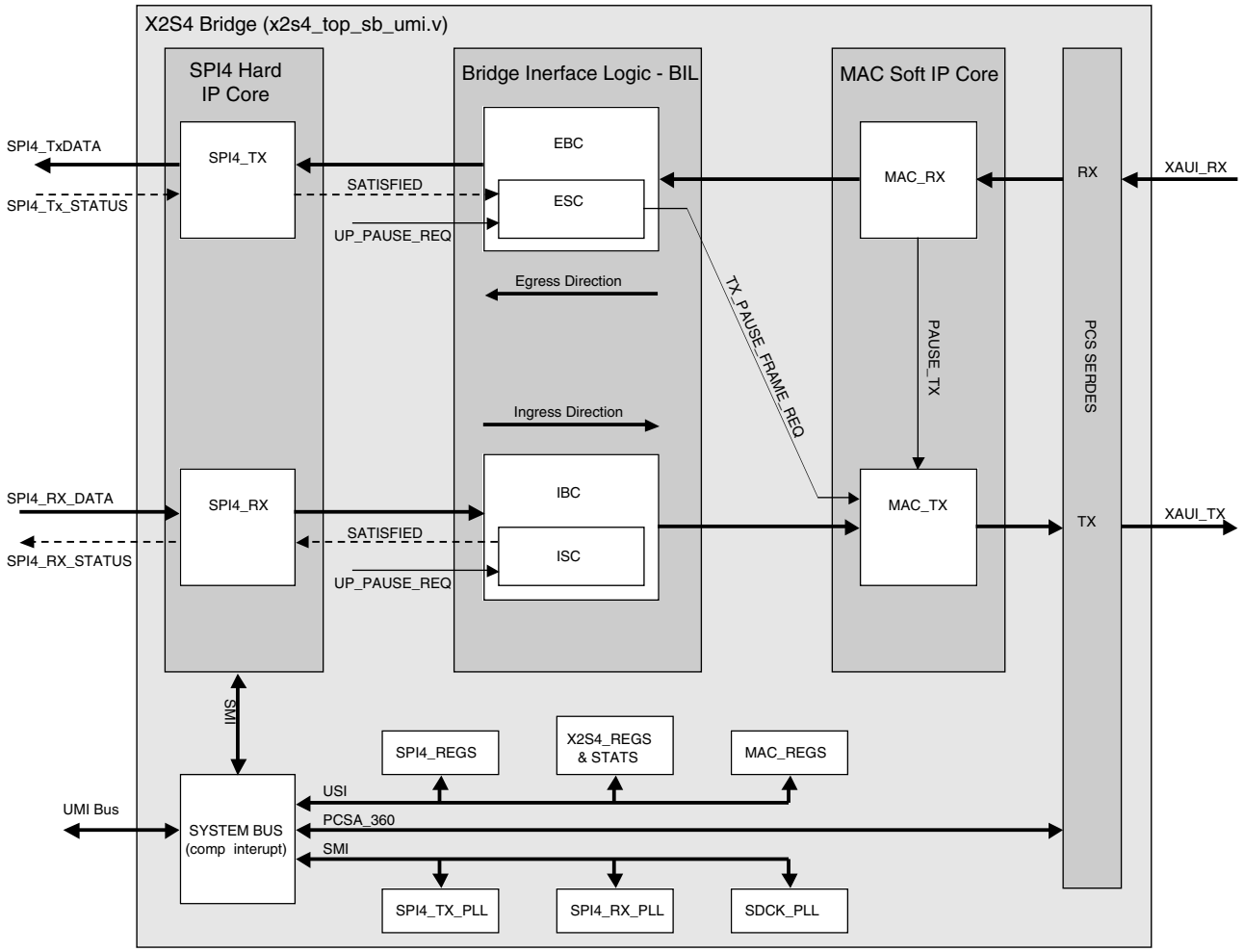
All packets received on the SPI4.2 interface from the NPU must also be in standard Ethernet or HiGig format depending upon bridge mode selected and CRC programming options selected in the MAC (see section - "Packet/Frame Formats").

In this version of the Bridge, HiGig header overlays are supported only on control frames (pause). These frames originate inside the MAC since the MAC is responsible for generating pause frames. Future versions of the Bridge may support insertion/deletion of the HiGig header on data frames but the initial version does not. Customers are free to add this capability to the Bridge given that the area of the circuit most likely to add the feature is not security protected.

X2S4 Bridge Operational Description

Figure 3 provides a block diagram of the X2S4 Bridge. Refer to this figure as well as Figure 4 (Bridge Interface Logic Block Diagram) in the next section while reading the Operational Overview.

Figure 3. X2S4 Bridge Block Diagram



Data Flow - Ingress Direction

In the ingress direction, data packets are transferred from the Network Processor (NP) to the Ethernet Switch (ES). When the NPU has a complete packet ready for transmission to the ES, it interrogates the status of the SPI4.2 interface via its input Status Channel. For this application, a single status channel is employed to convey status for the entire 10G+ SPI4.2 data interface and only two (Starved and Satisfied) of the three state signaling possibilities (Hungry, Starved, Satisfied) supported by SPI4.2 standard are used. An Ingress Status Controller (ISC) determines the final status state delivered to the NPU and it takes into consideration two sources of flow control information in its determination: 1) the Ingress Bridge FIFO Almost Full Flag, 2) Microprocessor requested flow control through local register programming. Assuming none of these sources are currently requesting flow control, the status received at the NPU end will reflect the Starved state and the NPU will begin transmission of the packet to the X2S4 bridge.

As the packet enters the bridge and begins to fill the SPI4.2 receive FIFOs, the bridge logic responds very quickly and starts moving the packet into the Ingress Bridge FIFO (IBF) keeping the SPI4.2 FIFOs very shallow throughout the transfer. Shortly after the start of moving the packet into the IBF, the Transmit Empty ('tx_aem') flag is removed causing the MAC to start reading data, assuming it is not currently in the Pause state, under control of the Ingress MAC Controller (IMC) and transferring it from the IBF into the MAC. As the data is read, it is transferred to the Bus Conversion block where the 128b bus is converted to a 64b bus for presentation to the 10G+MAC. Once the EOP is written into the IBF, the Ingress MAC Controller (IMC) checks the state of the ISC to ensure that the bridge is not currently flow controlled and if not, asserts the 'tx_data_avail' signal informing the MAC that it may begin transmis-

sion. The MAC will continue to read from the IBF until the Transmit Empty ('tx_aem') signal is again asserted or its internal FIFO fills to a pre-determined level transmitting the entire packet. After the packet is completely transmitted, the transmit Status Vector is available for collection by the Statistics collection logic.

The MAC contains only a small 4K-byte internal buffer that is kept very shallow under normal traffic conditions. This FIFO will build however under both microprocessor and Pause based flow control conditions. For microprocessor-initiated flow control, the IMC does not assert the data available signal ('tx_data_avail') and so the MAC does not transmit. The MAC is however allowed to move data from the IBF to the MAC internal FIFO until a predetermined fill level is reached. For Pause uninitiated flow control inside the MAC, the transmitter is not allowed to send data but it can load its internal FIFO.

For sustained periods of data transmission from the NPU to ES, the IBF will build quickly given that there must be at least one full packet in the IBF before it is allowed to be transmitted and because of the over-speed of the SPI4.2 interface relative to the XAUI interface. User programmable Almost Full and Almost Empty thresholds associated with the IBF are used to control the point at which flow control begins and then ends for this condition. Assuming that the user application supports Jumbo frames and has a desire to keep storage in the bridge to a minimum, the Almost Full threshold could be set to somewhere between 11 and 21Kbytes which are the 1/3, 2/3 points of a 32K Byte IBF. Combined total storage of all three FIFOs is 32K(IBF) + 8K(SPI4) + 4K(MAC) = 44Kbytes. This should be more than enough to absorb in-flight and committed data.

Data Flow - Egress Direction

In the egress direction, data packets are transferred from the Ethernet Switch (ES) to the Network Processor (NP). Once a valid start of frame is detected in the MAC, the destination address field of the incoming frame is analyzed along with a variety of IP programmable options having to do with specifying the types of packets that may be passed across the bridge. These options can be configured on an application-by-application basis and do not have material impact on the behavior of the X2S4 Bridge. If the beginning part of the packet meets the address filtering criteria in the MAC, data transfer begins unconditionally and almost immediately towards the Bridge Interface Logic (BIL). The MAC does contain a 4K-byte Egress buffer but only ~64 bytes are actually used for address filtering allowing the MAC to drop packets inside the IP core when necessary. This FIFO never builds.

As packet data moves into the BIL, it is first converted from 64-bit format to 128-bit format in preparation for later delivery to the SPI4.2 IP core. Data is then written into the EBF unconditionally as it is received. As the EBF fill level builds, the Egress Read FIFO control logic will first interrogate the status of the SPI4.2 interface via its input Status Channel. Like the Ingress direction, a single status channel is employed to convey status for the entire 10G+ SPI4.2 data interface and only two (Starved and Satisfied) of the three state signaling possibilities (Hungry, Starved, Satisfied) are supported. An Egress Status Controller (ESC) determines the final status state reported to the FIFO read control logic and takes into consideration two sources of flow control information in its determination: status received from the NPU, and Microprocessor requested flow control through local register programming. Assuming none of these sources are currently requesting flow control, the status received by the FIFO read control logic will reflect the Starved state and data will quickly start to move into the SPI4.2 transmit FIFO committing it to the SPI4.2 line (EOP must exist for Store & Forward mode). Note that due to the 64b to 128b bus conversion, writes to the EBF will occur every other cycle at 156-187MHz (over-clock) whereas the EBF can be read every 150MHz clock cycle as the Egress MAC Controller (EMC) moves data into the SPI4.2 IP core. This over-speed results in the EBF running very shallow during the absence of flow control. Under these conditions, packet data will be written into the SPI4.2 transmit FIFO for a clock cycle or two, stop and then restart continuously but will be smoothed out once it hits the SPI4.2 line using the Burst Mode capability of the SPI4.2 IP core. Burst sizes can be set to from 1 to 63 Credits. Setting this value to 8 for example guarantees that there will be no bursts smaller than 8 * 16 bytes = 128 bytes transmitted on the SPI4.2 line without an EOP. Building of the SPI4.2 transmit FIFO will occur only up to the point where the minimum burst size has been received. After a packet is completely transmitted, the receive Status Vector is available for collection by the Statistics collection logic.

If either the NPU, via the SPI4.2 status interface, or local microprocessor has requested flow control, a Satisfied state is reported to the FIFO read control logic and the movement of data from the EBF to the SPI4.2 transmit FIFO stops. Inside the SPI4.2 core, the transmitter will completely empty any data in the input FIFO and appropriately

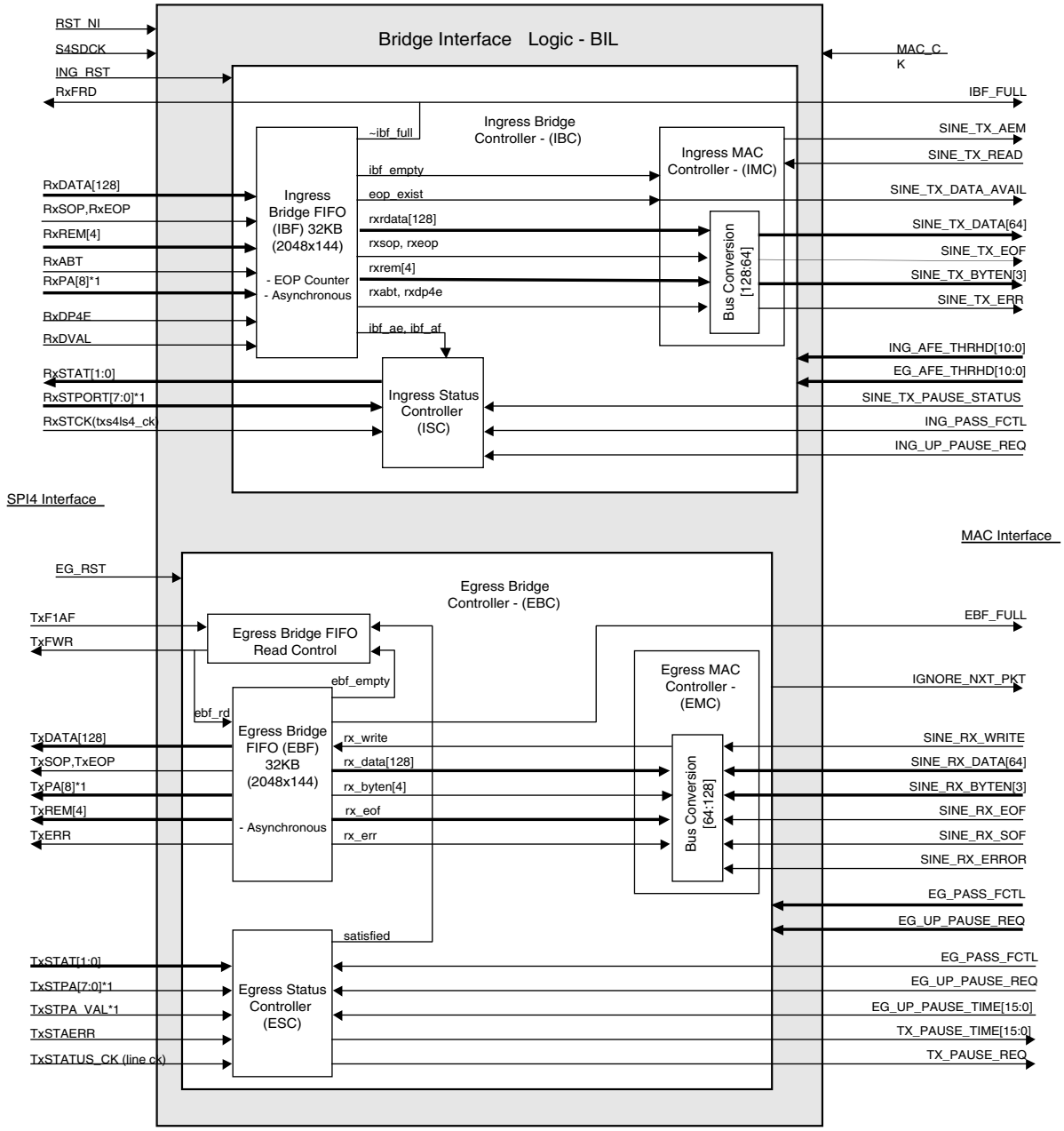
segment the line mid-packet and resume the packet when the flow control state is lifted and data again enters the FIFO (once in the SPI4.2 FIFO its committed). If Burst-Mode is enabled, the FIFO will only be emptied up to the point where there is less than the programmed burst size left in the FIFO. Unlike the Ingress direction where all three FIFOs (SPI4.2, Bridge, and MAC) sum to provide larger over-all storage during flow control, in the Egress direction, only the EBF is capable of providing this function. Unfortunately, the Egress direction FIFO storage is the most critical due to the fact that flow control is in-band rather than out-of-band like the SPI4.2 interface. The worst case scenario is when the EBF hits the high-water mark and makes a request of the MAC to generate a flow control frame to relieve the traffic condition but the MAC has just started sending a Jumbo frame. In this case Pause frame transmission must wait. When the Pause frame finally arrives at the switch end, it too may have just started sending a Jumbo frame.

In addition to flow controlling locally, the ESC also causes the MAC to generate and transmit a Pause control frame with a locally programmed Pause Quanta value over the Ethernet interface. The ESC starts a counter in order to time the flow control state. If the original source of flow control is still requesting the flow control state near the end of the timed value, the ESC directs the MAC to send another Pause frame in order to maintain the flow control state. When the source of flow control removes the flow controlled state, the ESC directs the MAC to send another Pause frame (X_On) removing the flow control.

Bridge Interface Logic (BIL) Block Diagram

Figure 4 shows a block diagram view of the Bridge Interface Logic (BIL) used to bridge SPI4.2 and Ethernet functions together.

Figure 4. Bridge Interface Logic (BIL) Block Diagram



Packet Error Handling

There is a packet error-checking/correcting module on the input side of the bridge buffers that controls writing of packet data into the buffer. This module is always operating in the circuit and behaves differently based on whether Drop is enabled.

DROP Inactive

Depending upon the type of error and where it is detected, different handling takes place when Drop is inactive. Errors that are received with the payload such as SPI4 abort and DIP4 or MAC CRC errors are simply passed through the buffer and are used to mark the packet as bad before it leaves the device so that the far-end can properly dispose of it. Other errors detected inside the bridge, such as missing SOP and missing EOP, are handled as follows:

- **Missing EOP:** SOP followed by SOP. Insert an EOP and mark the packet as bad. It is best to get rid of packet fragments from the system as soon as possible.
- **Missing SOP:** EOP followed by EOP. Drop all data up to the next valid SOP.
- **FIFO Full/Overflow:** Drop only the packet, up to the reception of the EOP, that caused the FIFO to fill and then start writing the next packet into the buffer. If it results in full, drop it also and repeat forever. Note that the FIFO is not allowed to actually overflow. Note also that this overflow drop feature is present regardless of control signal drop.

Drop Active
When drop is active, all error'ed packets are dropped before the buffer as follows:

- **Missing EOP:** SOP followed by SOP. Drop SOP encapsulated data and begin writing data from second SOP into the buffer. If another SOP is received, drop it also. Repeat forever.
- **Missing SOP:** EOP followed by EOP. Drop all data up to the next valid SOP.
- **SPI4 Abort (Ingress):** Drop packet.
- **SPI4 DIP4 (Ingress) and MAC CRC (Egress) Errors:** Drop packet.

XAUI Overview

Standard XAUI supports a 10Gbps client side payload over a 12.5Gbps SERDES interface after the payload is 8b10b encoded requiring 25% over-speed. A group of four SERDES each operating at 3.125Gbps is used to provide the interface. This translates into a 156.25MHz client side clock over a 64-bit wide data bus.

Some Ethernet switch manufacturers offer an “over-clocked” XAUI mode that affords a full 12Gbps to the client side. This mode is also supported by the X2S4 reference design. In the over-clocked mode, the SERDES operate at 3.75Gbps providing ~20% over-speed (15Gbps). For this mode of operation, the client side interface operates at ~187MHz over a 64-bit interface in order to support the higher rate.

HiGig and HiGig+ - Broadcom

The X2S4 Bridge optionally supports Broadcom’s proprietary method of Ethernet switch stacking. Contact your local Lattice sales office for additional information.

HyperG.Stack - Marvell

The X2S4 Bridge supports Marvell’s method of Ethernet switch stacking where XAUI ports can be run in 10 or 12G modes. The over-clocked mode used at Marvell does not include a proprietary protocol - standard Ethernet frames are used.

External UMI Bus

This section describes the User Master Interface (UMI) implementation strategy for an external microprocessor interface. It pays particular attention to the topic of Big/Little Endian as it relates to the address and data buses.

Big Endian Data Bus Format

The X2S4 reference design utilizes many of the LatticeSC embedded System Bus capabilities to support external read/write access of various registers and memory inside the bridge. Some of the endpoints are FPGA hard blocks such as the system bus block itself, the PCS SERDES block and the MACO blocks while others are soft FPGA based register modules associated with the SPI4.2, 10G+MAC, and Bridge level IP. Because the system bus and PCS SERDES blocks are designed (both silicon and register documentation) to support a Big Endian data interface format and can not be changed, the other soft read/write end-points also conform to the Big Endian format in order to maintain consistency.

Therefore, the external UMI data bus format requires that the lsb of a given read/write item be carried over the wire/pin labeled D7. For example, the first byte of the LatticeSCM device ID register located in the System Bus hard

block is documented to contain a value of 0xEA. In the PCS block, Quad Interface Register (QIR) 0x16 is defined to carry the lower bits of the comma character 0x7c. In big endian format, the data for these accesses are carried as follows across the external UMI data bus.

Table 1. Big Endian

	Pin/Wire D7	Pin/Wire D6	Pin/Wire D5	Pin/Wire D4	Pin/Wire D3	Pin/Wire D2	Pin/Wire D1	Pin/Wire D0	Numeric Value
	LSB							MSB	
System Bus DEVICE ID 0x00000	0	1	0	1	0	1	1	1	0xEA
QIR Defined Comma Char 0x36016	0	0	1	1	1	1	1	0	0x7C

Simulation Models - UMI Driver/Monitor

The current jtag_drv (driver including display) / orcastra.v (jtag/umi convert) combination flips the data bus before receiving/presenting to/from the Big Endian based UMI interface converting it into a traditional Little Endian format. Therefore when writing a given register in simulation, the value presented to the simulation driver must be in Little Endian format. For example, if one wanted to enable User Slave Interrupt (USI_IRQ, bit6) via the System Bus Interrupt Enable Register (IER) using the simulation driver provided, a write would need to be performed to UMI address 0x00012 according to the format conveyed in Table 2. This action results in the format conveyed in Table 3 at the external UMI interface boundary because the driver flips the bus.

Table 2. Simulation Driver - Little Endian View Setting Of USI_IRQ Bit 6

	Pin/Wire D7	Pin/Wire D6	Pin/Wire D5	Pin/Wire D4	Pin/Wire D3	Pin/Wire D2	Pin/Wire D1	Pin/Wire D0	Numeric Value
	LSB							MSB	
System Bus Int Enable 0x00012	0	0	0	0	0	0	1	0	0x2

Table 3. UMI Interface Setting Of USI_IRQ Bit 6

	Pin/Wire D7	Pin/Wire D6	Pin/Wire D5	Pin/Wire D4	Pin/Wire D3	Pin/Wire D2	Pin/Wire D1	Pin/Wire D0	Numeric Value
	LSB							MSB	
System Bus Int Enable 0x00012	0	1	0	0	0	0	0	0	0x2

Soft Registers

For FPGA soft registers, the approach is the same as for the hard block registers. They are defined in the register map conforming to Big Endian format rules and this is also the view at the external UMI interface of the device. The simulation driver view is Little Endian and so the user must make the conversion. Figure 6 shows a write operation to the Bridge Control register. In order to set b7, a value of 0x1 must be used inside the driver and a value 0f 0x80 appears at the UMI interface.

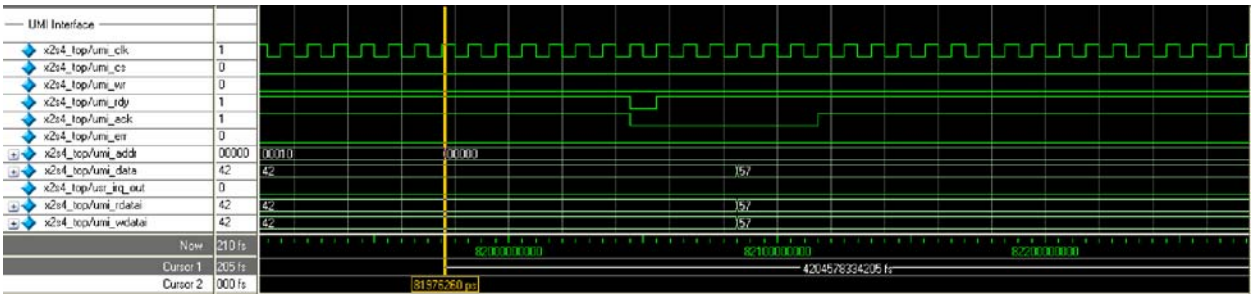
Little Endian Address Bus Format

The 18-bit UMI address bus operates in Little Endian format for all targets and accesses.

Simulation Timing Diagrams

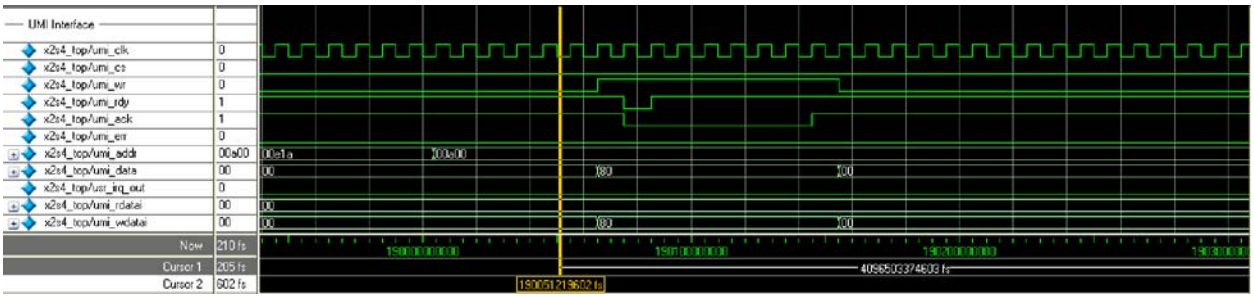
UMI External Interface view.

Figure 5. Read System Bus Device ID register - 0xEA



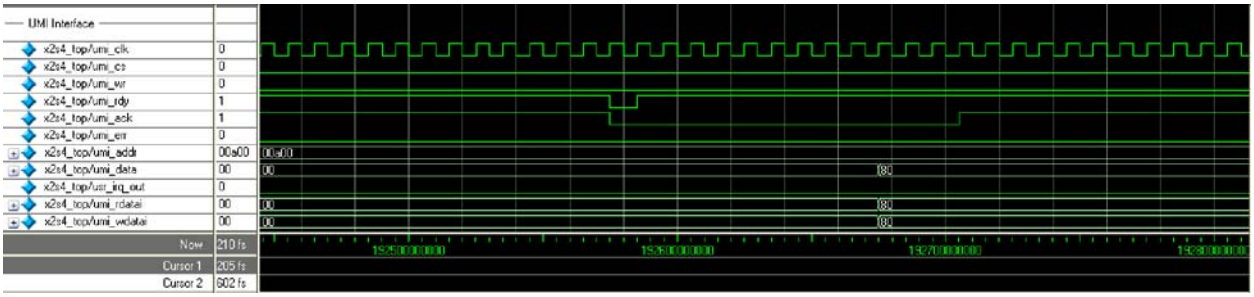
Write Bridge Control Register - 0x01

Figure 6. Write Bridge Control Register



Read Bridge Control Register - 0x01

Figure 7. Read Bridge Control Register



X2S4 Reference Design Registers

Table 4. X2S4 Register Map Summary

UMI_ADDR[17:0]	Size Bytes	Description
0x00000 - 0x0003f	64	System Bus internal registers
0x00410 - 0x0041f	16	SPI4 Transmit PLL
0x00420 - 0x0042f	16	SPI4 System Data Clock (SDCK) PLL
0x00430 - 0x0043f	16	SPI4 Receive PLL
0x00440 - 0x0044f	16	SPI4 Hard Block registers
0x00800 - 0x008ff	16	SPI4 Soft registers
0x00900 - 0x009ff	256	Unused
0x00a00 - 0x00bff	512	Bridge registers
0x00e00 - 0x00fff	2048	MAC registers

X2S4 Register Descriptions

System Bus Internal Registers

See Technical Note TN1085 - LatticeSC MPI/System Bus document for details regarding use and access of System Bus internal registers. Most of the System Bus internal registers do not require explicit initialization but two of the registers (Interrupt Cause Register - ICR, and Interrupt Enable Register - IER) are used in the interrupt scheme of the design and must be initialized if an external interrupt is expected.

For this X2S4 Bridging application, bits 1 (PCS_IRQ) and 6 (USI_IRQ) in the interrupt enable Register should be set and the remaining bits cleared. This is the highest level of interrupt consolidation in the design and results in I/O pin usr_irq_out being active if either of the two interrupts associated with these enable bits are active (see section “Interrupt Structure”).

PLL Registers

For descriptions of PLL internal registers see LatticeSC sysCLOCK PLL/DLL User’s Guide - TN1098.

SPI4.2 Hard Block Registers

For descriptions of SPI4.2 hard block internal registers see LatticeSC SPI4.2 Core MACO User’s Guide - IPUG44.

SPI4.2 Soft Registers

Table 5. SPI4.2 Soft Register Map Summary

Register Description	Mnemonic	I/O Address	Reset Value
SPI4 Rx Control	S4RX_CTL	0x800	0x0C
SPI4 Tx Control	S4TX_CTL	0x801	0x0C
SPI4 Status Threshold	S4STAT_THRSH	0x802	0x62
SPI4 Rx Calendar Repetition	S4RXCAL_M	0x803	0x04
SPI4 Rx Calendar Length	S4RXCAL_LEN	0x804	0x01
SPI4 Tx Calendar Repetition	S4TXCAL_M	0x805	0x04
SPI4 Tx Calendar Length	S4TXCAL_LEN	0x806	0x01
SPI4 Error Enable Register	S4EENREG	0x807-0x809	0x00
SPI4 Error Register	S4ERRREG	0x80a-0x80c	0x00

SPI4.2 Register Descriptions

Table 6. SPI4.2 Rx Control Registers

S4RX_CTL				Address: 0x800
Bit	**Name**	**Type**	**Default**	**Description**
0	S4RXRST	R/W	0	Receive Reset - Active high level sensitive reset resetting the entire SPI4 receiver.
1	S4RXRST_AIL	R/W	0	Receive Reset AIL - Active high edge sensitive reset of the Adaptive Input Logic functions in the input DDR gear-box.
2	S4RXRST_DSKW	R/W	0	Receive Reset Deskew - Active high level sensitive reset of the Deskew block.
3	S4RXFDIP2E	R/W	0	Receive Force DIP2 Error - Active high DIP2 force error control.
4	S4RXLOOP	R/W	0	Receive Loop - Unused.
5	S4RXINTSTC	R/W	0	Receive Internal Status Control - Unused.
6	S4RXRUN_AIL	R/W	1	Receive Run AIL - Active High control allowing AIL to track data edges.
7	S4RXSTEN	R/W	1	Receive Status Enable - Active high enable for the Status Channel logic.

Table 7. SPI4.2 Tx Control Registers

S4TXCTL				Address: 0x801
Bit	**Name**	**Type**	**Default**	**Description**
0	S4TXRST	R/W	0	Transmit Reset - Active high level sensitive reset resetting the entire SPI4 Transmit.
1	S4TXFIDLE	R/W	0	Transmit Force Idles - Active high control forcing the Tx to send SPI4 Idles.
2:3	S4TXFDIP4E	R/W	0H	Transmit Force DIP4 Error - Active high control forcing the Tx to send DIP4 errors. Bit 2 causes all control words to be sent with bad parity and bit 3 causes only control words ending data segments to be generated with bad parity.
4	S4TXEN_PACK	R/W	1	Transmit Enable Packing - Active high control allowing the Tx to pack the SPI4 line.
5	S4TXSTEN	R/W	1	Transmit Status Enable - Active high enable for the Status Channel logic.
6	S4TXINTSTC	R/W	0	Transmit Internal Status Control - Unused.
7	Unused	-	-	-

Table 8. SPI4.2 Status Threshold Registers

S4STAT_THRSH				Address: 0x802
Bit	**Name**	**Type**	**Default**	**Description**
0:2	S4TXNUMDIP2	R/W	011B	Transmit Number Of DIP2 - Specifies the number of correct DIP2 code words that are required before declaring alignment.
3	Unused	-	-	-
4:6	S4TXNUMDIP2E	R/W	010B	Transmit Number Of DIP2 Errors - Specifies the number of incorrect DIP2 code words that are required before declaring out of alignment.
7	Unused	-	-	-

Table 9. SPI4.2 Rx Calendar Repetition Register

S4RXCAL_M				Address: 0x803
Bit	**Name**	**Type**	**Default**	**Description**
0:7	S4RXCAL_M	R/W	04H	Receive Calendar Repetition - Specifies the number of times that the calendar sequence is repeated before the DIP2 code word and framing are inserted.

Table 10. SPI4.2 Rx Calendar Length Register

S4RXCAL_LEN				Address: 0x804
Bit	**Name**	**Type**	**Default**	**Description**
0:7	S4RXCAL_LEN	R/W	01H	Receive Calendar Length - Specifies the length of calendar sequence. This should be left at 1.

Table 11. SPI4.2 Tx Calendar Repetition Register

S4TXCAL_M				Address: 0x805
Bit	**Name**	**Type**	**Default**	**Description**
0:7	S4TXCAL_M	R/W	04H	Transmit Calendar Repetition - Specifies the number of times that the calendar sequence is repeated before the DIP2 code word and framing are inserted.

Table 12. SPI4.2 Calendar Length Register

S4TXCAL_LEN				Address: 0x806
Bit	**Name**	**Type**	**Default**	**Description**
0:7	S4TXCAL_LEN	R/W	01H	Transmit Calendar Length - Specifies the length of calendar sequence. This should be left at 1.

Table 13. SPI4.2 Error Enable Register

S4EENREG					Base Address: 0x807
Address Offset	**Bit**	**Name**	**Type**	**Default**	**Description**
0	0:7	S4EENREG_0	R/W	0H	SPI4 Error Enable - Bit for bit active high interrupt enables for the SPI4 error register.
1	0:7	S4EENREG_1	R/W	0H	
2	0:5	S4EENREG_2	R/W	0H	
2	6:7	Unused	-	-	-

Table 14. SPI4.2 Error Registers

S4ERRREG					Base Address: 0x80a
Address Offset	**Bits**	**Name**	**Type**	**Default**	**Description**
0	0	S4RXF1_PARERR	COR	0	Receive FIFO1 Parity Error - Unused.
0	1	S4RXF1FERR	COR	0	Enable Receive FIFO1 Full Error - Active high error indicating that RxFIFO1 has reached the full state (this should never occur).
0	2	S4RXDSKWD	COR	0	Receive Deskewed - Active high error indicating that the deskew block has lost, or has not achieved a deskewed state.
0	3	S4RXLT40_TRERR	COR	0	Rx Less Than 40 Training Pattern Error - Active high error indicating that the Deskew block received less than 40 repetitions of the Training pattern during a time when it had previously not been deskewed.
0	4	S4RXLT10_TRERR	COR	0	Receive Less Than 10 Training Pattern Error - Active high error indicating that the Deskew block received less than 10 repetitions of the Training pattern during a time when it was in the deskewed state.
0	5	S4RXAERR	COR	0	Receive Alignment Error - Active high error indicating that the receiver is out of alignment.
0	6	S4RXD4ERR	COR	0	Receive RXDIP4 Error - Active high error indicating that the receiver has detected a DIP4 error.
0	7	S4RXAIL_LOCK	COR	0	Receive AIL Lock Error - Active high error indicating that the AIL logic in the gear-boxes (IDDRX2A) have not locked on to a safe clock/data relationship.
1	0	S4TXSDP2ERR	COR	0	Transmit Status DIP2 Error - Active high error indicating the transmit status block has detected a DIP2 error on status received on the status channel.
1	1	S4TXSTAERR	COR	0	Transmit Status Alignment Error - Active high error indicating that the transmit status block is out-of-alignment.
1	2	S4TXF2_PARERR	COR	0	Transmit FIFO2 Parity Error - Unused.
1	3	S4TXBURST_ERR	COR	0	Transmit Burst Error - Active high error indicating transmitter has segmented a burst different than the burst size set through TxBLEN[5:0]. Ensures correct programming of TxF1AEBTHRSH[] which must be set to a value 2 greater than TxBLEN[5:0].

Table 14. SPI4.2 Error Registers (Continued)

S4ERRREG					Base Address: 0x80a
Address Offset	Bits	Name	Type	Default	Description
1	4	S4TXREM_ERR	COR	0	Transmit Remainder Error - Active high error indicating that TxFIFO1 was written with a remainder of other than 4'b1111 during a cycle when 'TXEOP' was not active; a clear error condition.
1	5	S4TXF2FE	COR	0	Transmit FIFO2 Full Error - Active high error indicating that Tx FIFO2 has hit the full error state.
1	6	S4TXF2E	COR	0	Transmit FIFO2 Empty Error - Active high error indicating that Tx FIFO2 has hit the empty error state (empty is also an error condition for this FIFO)
1	7	S4TXF1FE	COR	0	Transmit FIFO1 Full Error - Active high error indicating that Tx FIFO1 has hit the full error state.
2	0	S4RXF2FE	COR	0	Receive FIFO2 Full Error - Active high error signal indicating that receive FIFO2 has reached the Full error state.
2	1:5	S4RS4ERR	COR	0H	Receive SPI4 Protocol Errors - Active high error signals indicating that the receiver has seen a sequence of data and control words that are in violation of the SPI4 protocol. RS4ERR[4] - A valid write to RxFIFO2 where less than 16 bytes are marked valid without an EOP active. SPI4 burst aligned and "or"ed into the 'RXDP4E' error lane through RxFIFO2 and is activated to mark the segment as bad. RS4ERR[3] - Data preceded by an idle. Error flag is raised. RS4ERR[2] - Reserved Control Word. Error flag is raised. RS4ERR[1] - Any EOP preceded by an idle. RS4ERR[0] - Any control word preceded by a payload control word.
2	6:7	Unused	-	-	-

Bridge Registers

Table 15. Bridge Register Map Summary

Register Description	Mnemonic	I/O Address	Reset Value
Bridge Control Register 1	BRCREG	0xa00	0x02
Bridge Ingress Almost Empty Threshold Register	BRINGAETHRSH_L	0xa01	0xA0
Bridge Ingress Almost Full Threshold Register	BRINGAFTHRSH_L	0xa02	0x80
Bridge Ingress Almost Empty/Full Threshold Register	BRINGAEFTHRSH_H	0xa03	0x54
Bridge Egress Almost Empty Threshold Register	BREGAETHRSH_L	0xa04	0xA0
Bridge Egress Almost Full Threshold Register	BREGAFTHRSH_L	0xa05	0x80
Bridge Egress Almost Empty/Full Threshold Register	BREGAEFTHRSH_H	0xa06	0x54
Bridge Error Enable Register	BREENREG	0xa07	0x00
Bridge Error Register	BRERRREG	0xa08	0x00
Bridge Control Register 2	BRCREG	0xa09	0xC0
Bridge Statistics Counter 0 (pause packets received)	BRSTATCTR0	0xb00-0xb07	0x00
Bridge Statistics Counter 1 (packets received)	BRSTATCTR1	0xb08-0xb0f	0x00
Bridge Statistics Counter 2 (pause packets transmitted)	BRSTATCTR2	0xb10-0xb17	0x00
Bridge Statistics Counter 3 (packets transmitted)	BRSTATCTR3	0xb18-0xb1f	0x00
Bridge Statistics Counter 4 (mac Tx paused)	BRSTATCTR4	0xb20-0xb27	0x00

Register Description	Mnemonic	I/O Address	Reset Value
Bridge Statistics Counter 5 (spi4rx satisfied)	BRSTATCTR5	0xb28-0xb2f	0x00
Bridge Statistics Counter 6 (spi4tx satisfied)	BRSTATCTR6	0xb30-0xb37	0x00

Bridge Register Descriptions

Table 16. Bridge Control Register 1

BCREG				Address: 0xa00
Bits	**Name**	**Type**	**Default**	**Description**
0	BRINGRST	R/W	0	Ingress FIFO Reset - Active high level sensitive reset resetting the Ingress FIFO.
1	BREGRST	R/W	0	Egress FIFO Reset - Active high level sensitive reset resetting the Egress FIFO.
2	BRING_UP_PAUSE_REQ	R/W	0	Ingress Microprocessor Pause Request - Active high level sensitive control forcing a pause for the ingress direction data flow (spi4 status output).
3	BREG_UP_PAUSE_REQ	R/W	0	Egress Microprocessor Pause Request - Active high level sensitive control forcing a pause for the egress direction data flow (generates mac ctl pause frame).
4	BRING_PASS_FCTL	R/W	0	Ingress Pass Flow Control - Active high signal that causes ingress direction flow control to be passed directly across the Bridge without waiting for a specific ingress bridge FIFO fill level.
5	BREG_PASS_FCTL	R/W	0	Egress Pass Flow Control - Active high signal that causes egress direction flow control to be passed directly across the Bridge without waiting for a specific egress bridge FIFO fill level.
6	BRING_DROP	R/W	1	Ingress Drop - Active high control signal that causes ingress packets to be dropped when they are found to be received in error (spi4 dip4 and abort or internally detected missing sop or eop.
7	BREG_DROP	R/W	0	Egress Drop - Active high control signal that causes egress packets to be dropped when they are found to be received in error (mac CRC error or internally detected missing sop or eop.

Table 17. Bridge Ingress Almost Empty Threshold Register Low Byte

BRINGAETHRSH_L				Address: 0xa01
Bits	**Name**	**Type**	**Default**	**Description**
0:7	BRINGAETHRSH_L	R/W	A0H	Ingress Almost Empty Threshold Register Low Byte

Table 18. Bridge Ingress Almost Full Threshold Register Low Byte

BRINGAFTHRSH_L				Address: 0xa02
Bits	**Name**	**Type**	**Default**	**Description**
0:7	BRINGAFTHRSH_L	R/W	80H	Ingress Almost Full Threshold Register Low Byte

Table 19. Bridge Ingress Almost Empty/Full Threshold Register High Bits

BRINGAEFTHRSH_H				Address: 0xa03
Bits	**Name**	**Type**	**Default**	**Description**
0:2	BRINGAETHRSH_H	R/W	010B	Ingress Almost Empty Threshold Register High Bits 10:8
3:5	BRINGAFTHRSH_H	R/W	101B	Ingress Almost Full Threshold Register High Bits 10:8
6:7	Unused	-	-	-

Table 20. Bridge Egress Almost Empty Threshold Register Low Byte

BREGAETHRSH_L				Address: 0xa04
Bits	**Name**	**Type**	**Default**	**Description**
0:7	BREGAETHRSH_L	R/W	A0H	Egress Almost Empty Threshold Register Low Byte

Table 21. Bridge Egress Almost Full Threshold Register Low Byte

BREGAFTHRSH_L				Address: 0xa05
Bits	**Name**	**Type**	**Default**	**Description**
0:7	BREGAFTHRSH_L	R/W	80H	Egress Almost Full Threshold Register Low Byte

Table 22. Bridge Egress Almost Empty/Full Threshold Register High Bits

BREGAEFTHRSH_H				Address: 0xa06
Bits	**Name**	**Type**	**Default**	**Description**
0:2	BREGAETHRSH_H	R/W	010B	Egress Almost Empty Threshold Register High Bits (10:8)
3:5	BREGAFTHRSH_H	R/W	101B	Egress Almost Full Threshold Register High Bits (10:8)
6:7	Unused	-	-	-

Table 23. Bridge Error Enable Register

BREENREG				Address: 0xa07
Bits	**Name**	**Type**	**Default**	**Description**
0	BRIBFE_EN	R/W	0	Ingress Buffer Full Error Enable - Active high interrupt enable.
1	BREBFE_EN	R/W	0	Egress Buffer Full Error Enable - Active high interrupt enable.
2	BRING_MSOP_EN	R/W	0	Ingress Missing SOP Error - Active high interrupt enable.
3	BRING_MEOP_EN	R/W	0	Ingress Missing EOP Error - Active high interrupt enable.
4	BRING_PKT_DROPPED_EN	R/W	0	Ingress Packet Dropped - Active high interrupt enable.
5	BREG_MSOP_EN	R/W	0	Egress Missing SOP Error - Active high interrupt enable.
6	BREG_MEOP_EN	R/W	0	Egress Missing EOP Error - Active high interrupt enable.
7	BREG_PKT_DROPPED_EN	R/W	0	Egress Packet Dropped - Active high interrupt enable.

Table 24. Bridge Error Register

BRERREG				Address: 0xa08
Bits	**Name**	**Type**	**Default**	**Description**
0	BRIBFE	COR	0	Ingress Buffer Full Error - Active high error indicating that the Ingress Buffer has reached the full level.
1	BREBFE	COR	0	Egress Buffer Full Error - Active high error indicating that the Egress Buffer has reached the full level.
2	BRING_MSOP	COR	0	Ingress Missing SOP Error - Active high error indicating that a packet was received without an SOP. All data is dropped up to the next valid SOP.
3	BRING_MEOP	COR	0	Ingress Missing EOP Error - Active high error indicating that a packet was received without an EOP. If DROP is enabled, the packet is dropped. Otherwise, an EOP is inserted and the packet is marked as bad.
4	BRING_PKT_DROPPED	COR	0	Ingress Packet Dropped - Active high error indicating that a packet has been dropped.

BRERREG				Address: 0xa08
Bits	**Name**	**Type**	**Default**	**Description**
5	BREG_MSOP	COR	0	Egress Missing SOP Error - Active high error indicating that a packet was received without an SOP. All data is dropped up to the next valid SOP.
6	BREG_MEOP	COR	0	Egress Missing EOP Error - Active high error indicating that a packet was received without an EOP. If DROP is enabled, the packet is dropped otherwise an EOP is inserted and the packet is marked as bad.
7	BREG_PKT_DROPPED	COR	0	Egress Packet Dropped - Active high error indicating that a packet has been dropped.

Table 25. Bridge Control Register

BCREG2				Address: 0xa09
Bits	**Name**	**Type**	**Default**	**Description**
0	BRING_EN_FCTL	R/W	1	Ingress Enable Flow Control - Active high control signal that enables the Ingress Buffer Controller to flow control the SPI4 input interface.
1	BREGRST	R/W	1	Egress Enable Flow Control - Active high control signal that enables the Ingress Buffer Controller to flow control the MAC input interface.
2:7	Unused	-	-	-

Table 26. Bridge Statistics Counter 0

BRSTATCTR0					Base Address: 0xb00
Address Offset	**Bits**	**Name**	**Type**	**Default**	**Description**
0	0:7	BRSTATCTR0_0	R	0H	Statistics Counter 0 - A 48b rollover counter that counts the number of Pause packets received. A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR0_1	R	0H	
2	0:7	BRSTATCTR0_2	R	0H	
3	0:7	BRSTATCTR0_3	R	0H	
4	0:7	BRSTATCTR0_4	R	0H	
5	0:7	BRSTATCTR0_5	R	0H	

Table 27. Bridge Statistics Counter 1

BRSTATCTR1					Base Address: 0xb08
Address Offset	**Bits**	**Name**	**Type**	**Default**	**Description**
0	0:7	BRSTATCTR1_0	R	0	Statistics Counter 1 - A 48b rollover counter that counts the number of packets received excluding pause packets. A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR1_1	R	0	
2	0:7	BRSTATCTR1_2	R	0	
3	0:7	BRSTATCTR1_3	R	0	
4	0:7	BRSTATCTR1_4	R	0	
5	0:7	BRSTATCTR1_5	R	0	

Table 28. Bridge Statistics Counter 2

BRSTATCTR2					Base Address: 0xb10
Address Offset	Bits	Name	Type	Default	Description
0	0:7	BRSTATCTR2_0	R	0	Statistics Counter 2 - A 48b rollover counter that counts the number of pause packets transmitted. A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR2_1	R	0	
2	0:7	BRSTATCTR2_2	R	0	
3	0:7	BRSTATCTR2_3	R	0	
4	0:7	BRSTATCTR2_4	R	0	
5	0:7	BRSTATCTR2_5	R	0	

Table 29. Bridge Statistics Counter 3

BRSTATCTR3					Base Address: 0xb18
Address Offset	Bits	Name	Type	Default	Description
0	0:7	BRSTATCTR3_0	R	0	Statistics Counter 3 - A 48b rollover counter that counts the number of packets transmitted excluding pause packets. A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR3_1	R	0	
2	0:7	BRSTATCTR3_2	R	0	
3	0:7	BRSTATCTR3_3	R	0	
4	0:7	BRSTATCTR3_4	R	0	
5	0:7	BRSTATCTR3_5	R	0	

Table 30. Bridge Statistics Counter 4

BRSTATCTR4					Base Address: 0xb20
Address Offset	Bits	Name	Type	Default	Description
0	0:7	BRSTATCTR4_0	R	0	Statistics Counter 4 - A 48b rollover counter that counts the amount of time that the MAC transmitter is in the pause state. Count values are measured in pause quanta increments (512 bit times). A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR4_1	R	0	
2	0:7	BRSTATCTR4_2	R	0	
3	0:7	BRSTATCTR4_3	R	0	
4	0:7	BRSTATCTR4_4	R	0	
5	0:7	BRSTATCTR4_5	R	0	

Table 31. Bridge Statistics Counter 5

BRSTATCTR5					Base Address: 0xb28
Address Offset	Bits	Name	Type	Default	Description
0	0:7	BRSTATCTR5_0	R	0	Statistics Counter 5 - A 48b rollover counter that counts the amount of time that the SPI4 receiver is in the satisfied state sending “satisfied” to the far-end. Count values are measured in pause quanta increments (512 bit times). A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR5_1	R	0	
2	0:7	BRSTATCTR5_2	R	0	
3	0:7	BRSTATCTR5_3	R	0	
4	0:7	BRSTATCTR5_4	R	0	
5	0:7	BRSTATCTR5_5	R	0	

Table 32. Bridge Statistics Counter 6

BRSTATCTR6					Base Address: 0xb30
Address Offset	Bits	Name	Type	Default	Description
0	0:7	BRSTATCTR6_0	R	0	Statistics Counter 6 - A 48b rollover counter that counts the amount of time that the SPI4 transmitter is in the satisfied state having received “satisfied” from the far-end. Count values are measured in pause quanta increments (512 bit times). A read of the base address latches the entire 48b counter for subsequent reading of the remaining bytes of the counter.
1	0:7	BRSTATCTR6_1	R	0	
2	0:7	BRSTATCTR6_2	R	0	
3	0:7	BRSTATCTR6_3	R	0	
4	0:7	BRSTATCTR6_4	R	0	
5	0:7	BRSTATCTR6_5	R	0	

MAC Registers

Table 33. MAC Register Map Summary

Register Description	Mnemonic	I/O Address	Reset Value
Mac Mode	MCMODE	0xe00	0xC0
Mac Transmit Configuration	MCTX_CONFIG	0xe01	0x80
Mac Receive Configuration	MCRX_CONFIG	0xe02	0x32
Mac Inter Packet Gap	MCIPG_VAL	0xe03	0x00
Mac Maximum Frame Length	MCMAX_FRM_LEN	0xe04-0xe5	0xEE, 0x05
Mac MAC_Address	MCMAC_ADDR	0xe06-0xe0b	0x00
Mac VLAN TAG	MCVLAN_TAG	0xe0c-0xe0d	0x00
Mac Pause Opcode	MCPAUSE_OP	0xe0e-0xe0f	0x01, 0x00
Mac Pause Time	MCPAUSE_TMR	0xe10-0xe11	0xFF, 0xFF
Mac Link Status	MCLNK_STS	0xe12	0xF8
Mac Receiver Statistics Interrupt	MCRXSTATVEC_INT	0xe13-0xe14	0x00
Mac Transmitter Statistics Interrupt	MCTXSTATVEC_INT	0xe15-0xe16	0x00
Mac Receiver Statistics Enable	MCRXSTATVEC_INT_EN	0xe17-0xe18	0x00
Mac Transmitter Statistics Enable	MCTXSTATVEC_INT_EN	0xe19-0xe1a	0x00

MAC Register Descriptions

Table 34. MAC Mode Register

MCMODE				Address: 0xe00
Bits	Name	Type	Default	Description
0	MCTX_EN	R/W	1	Tx MAC Enable - When set, the Tx MAC is enabled to transmit frames.
1	MCRX_EN	R/W	1	Rx MAC Enable - When set, the Rx MAC is enabled to receive frames.
2:7	Unused	-	-	-

Table 35. MAC Transmit Configuration Register

MCTX_CONFIG				Address: 0xe01
Bits	**Name**	**Type**	**Default**	**Description**
0	MCTX_PAUSE_EN	R/W	1	Enable Pause Transmission - When set, the Tx MAC responds to the flow control logic's request to generate PAUSE frames.
1	MCGEN_FCS	R/W	0	Generate FCS - When set, the Tx MAC generates the FCS field for the frames being transmitted.
2	MCIFG_STRETCH	R/W	0	IFG Stretch Mode - When set, the Tx MAC operates in the IFG stretch mode, to match the data rates of OC-192. The IPG required to match OC192 is added to the value specified in IPG_VAL.
3	MCDISABLE_PAD	R/W	0	Diable Padding - When set, the Tx MAC will not add padding to packets smaller than 76 bytes for HiGig or smaller than 64 bytes for standard XAUI. This feature is only available on the HiGig version of the MAC.
4:7	Unused	-	-	-

Table 36. MAC Receive Configuration Register

MCRX_CONFIG				Address: 0xe02
Bits	**Name**	**Type**	**Default**	**Description**
0	MCRX_DISCARD_FCS	R/W	0	Discard FCS - When set, the Rx MAC discards the FCS and PAD fields (if any) from the frame before transferring it to the client receive FIFO.
1	MCRX_RCV_BC	R/W	0	Receive Broadcast Frames - When set, the Rx MAC receives broadcast frames.
2	MCRX_PRMS	R/W	1	Promiscuous Mode - When set, the Rx MAC receives any frame. All address filtering rules that have been set will be abandoned
3	MCRX_PAUSE_EN	R/W	1	Rx Pause Enable - When set, the Rx MAC will trigger the Tx MAC to pause when a pause frame is received.
4	MCRX_RCV_ALL_MC	R/W	0	Receive All Multicast Frames - When set, all multicast frames will be received irrespective of their address.
5	MCRX_RCV_SHORT	R/W	0	Receive Short Frames - When set, all frames shorter than 64 bytes will also be received.
6	MCRX_PAUSE_DROP	R/W	1	Receive Pause Drop - When set, pause frames received will be dropped.
7	Unused	-	-	-

Table 37. MAC Transmit Inter Packet Gap Register

MCTX_IPG				Address: 0xe03
Bits	**Name**	**Type**	**Default**	**Description**
0:4	MCIPG_VAL	R/W	0H	IPG Value. Used by the Tx MAC to determine the number of idle symbols that needs to be inserted between frames. A value of 0 (HiGig mode) results in, on average, 8 idles inserted. A value of 1 (standard Ethernet) results in 12 Idles and so on. In HiGig mode, the average number of idles inserted is 8 when this field is set to zero. In standard Ethernet mode, the average is 12.
5:7	Unused	-	-	-

Table 38. MAC Maximum Frame Length Register

MCTX_IPG					Address: 0xe03
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCMAX_FRM_LEN 0	R/W	EEH	Maximum Frame Length - Used only for statistical purposes, all frames longer than the value given here will be marked as long. This value will in no manner affect the frame's reception.
1	0:7	MCMAX_FRM_LEN 1	R/W	05H	

Table 39. MAC Address Register

MCMAC_ADDR					Base Address: 0xe06
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCMAC_ADDR_0	R/W	0H	MAC Address - The MAC Address Registers contain the Ethernet address of the port. The MAC Address Register 0 and 1 has the bytes that are transmitted first and the MAC Address Register 4 and 5 has the two bytes that are transmitted last. Byte 1 is transmitted first, then Byte 0, then Byte 3, then Byte 2, followed by Byte 5 and then 4. Note that the MAC address is stored in the registers in Hexadecimal form, so for example to set the MAC Address to: AC-DE-48-00-00-80 would require writing 0xAC (octet 0) to address 0x1, 0xDE (octet 1) to address 0x0, 0x48 (octet 2) to address 0x03, 0x00 (octet 3) to address 0x02, 0x00 (octet 4) to address 0x05, and 0x80 (octet 5) to address 0x04. Note Octet 0 is transmitted first and octet 5 is transmitted last. Also used to source transmit pause frame address.
1	0:7	MCMAC_ADDR_1	R/W	0H	
2	0:7	MCMAC_ADDR_2	R/W	0H	
3	0:7	MCMAC_ADDR_3	R/W	0H	
4	0:7	MCMAC_ADDR_4	R/W	0H	
5	0:7	MCMAC_ADDR_5	R/W	0H	

Table 40. MAC VLAN Tag Register

MCVLAN_TAG					Base Address: 0xe0c
Address Offset	Bits	Name	Type	Default	Description
0	0:2	MCVLAN_TAG_UPF	RO	0H	User Priority Field - This user priority field (UPF) ranges from 0 to 7, with 7 being the highest priority.
0	3	MCVLAN_TAG_CFI	RO	0	Canonical Format Indicator - The CFI bit is used to indicate that all MAC addresses present in the MAC data field are in canonical format. This field is interpreted differently depending on whether it is an ethernet-encoded tag header or a SNAP-encoded tag header. In SNAP-encoded TPID the field indicates the presence or absence of the canonical format of addresses. In ethernet-encoded TPID, it indicates the presence of the Source-Routing Information (RIF) field after the length field. The RIF field indicates routing on ethernet frames.
0	4:7	MCVLAN_TAG_ID_0	RO	0H	Virtual LAN Tag Id - 12b VLAN tag id field.
1	0:7	MCVLAN_TAG_ID_1	RO	0H	

Table 41. MAC Pause Opcode Register

MCPAUSE_OP					Base Address: 0xe0e
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCPAUSE_OP_0	R/W	01H	Pause Opcode - The Opcode of a control frame that defines a valid pause frame. This value will be compared against the opcode field of an incoming control frame to determine whether it is a valid pause frame. Bit 0 of register 0 corresponds to the bit that was transmitted first and bit 7 of register 1 corresponds to the bit that was transmitted last.
1	0:7	MCPAUSE_OP_1	R/W	00H	

Table 42. MAC Pause Time Register

MCPAUSE_TMR					Base Address: 0xe10
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCPAUSE_TMR_0	R/W	FFH	Pause Time - This register contains the pause time value for a flow control packet sourced by the 10gmac transmitter.
1	0:7	MCPAUSE_TMR_1	R/W	FFH	

Table 43. MAC Link Status Register

MCLNK_STS				Address: 0xe12
Bits	**Name**	**Type**	**Default**	**Description**
0	MCTX_IDLE	RO	1	Transmit MAC Idle - When set, indicates that the Tx MAC is inactive.
1	MCRX_IDLE	RO	1	Receive MAC Idle - When set, indicates that the Rx MAC is inactive.
2	MCLNK_FLT	RO	1	Link Fault - Indicates that Local fault symbols were received on the link
3	MCRMT_FLT	RO	1	Remote Fault - Indicates that Remote fault symbols were received on the link
4	MCLNK_OK	RO	1	Link Ok - Indicates that no fault symbols were received on the link
5:7	Unused	-	-	-

Table 44. MAC Rx Status Vector Interrupt

MCRXSTATVEC_INT					Base Address: 0xe13
Address Offset	**Bits**	**Name**	**Type**	**Default**	**Description**
0	0	MCMCVLAN	COR	0	Virtual LAN Frame- Vlan frame received.
0	1	MCPAUSE	COR	0	Pause Frame - Pause frame received.
0	2	MCCONTROL	COR	0	Control Frame - Control frame received.
0	3	MCUNSPRT_OPCODE	COR	0	Unsupported Opcode - Frame received with unsupported opcode.
0	4	MCBROADCAST	COR	0	Broadcast Frame - Broadcast frame received.
0	5	MCMULTICAST	COR	0	Multicast Frame - Multicast frame received.
0	6	MCLNGTH_ERR	COR	0	Length Error - Frame length error as specified in length field of frame.
0	7	MCCEPSEEN	COR	0	Carrier Event - Carrier even previously seen.
1	0	MCPRMBL_SHRINK	COR	0	Preamble Shrink - Preamble Shrink.
1	1	MCIPGVIOL	COR	0	Inter Packet Gap Violation - IPG violation.
1	2	MCLONG	COR	0	Long Packet - Packet received that is longer than specified in MAX_FRM_LEN register.
1	3	MCCRC_ERR	COR	0	CRC Error - Frame received with CRC error.
1	4	MCOK	COR	0	OK - Frame received ok - no errors.
1	5	MCFIFO_FULL	COR	0	FIFO Full - Rx FIFO full.
1	6:7	Unused	-	-	-

Table 45. MAC Tx Status Vector Interrupt

MCTXSTATVEC_INT					Base Address: 0xe15
Address Offset	Bits	Name	Type	Default	Description
0	0	MCTXERR	COR	0	Transmit Error - Error detected during frame transmission.
0	1	MCUNDERRUN	COR	0	FIFO Underrun - Transmit FIFO under run.
0	2	MCCRC_ERR	COR	0	CRC Error - CRC error detected on frame received from client side in mode when FCS insertion is disabled.
0	3	MCLNGTH_ERR	COR	0	Length Error - Length check error detected on frame received from client side during transmission based on Length Field received.
0	4	MCLONG	COR	0	Long Packet - Packet transmitted that is longer than specified in MAX_FRM_LEN register.
0	5	MCMULTICAST	COR	0	Multicast Frame - Multicast frame transmitted.
0	6	MCBROADCAST	COR	0	Broadcast Frame - Broadcast frame transmitted.
0	7	MCUNICAST	COR	0	Unicast Frame - Unicast frame transmitted.
1	0	MCJUMBO	COR	0	Jumbo Frame - Jumbo frame (>1500 Bytes) transmitted.
1	1	MCPAUSE	COR	0	Pause Frame - Pause frame transmitted.
1	2	MCVLAN	COR	0	Virtual LAN Frame- VLAN frame transmitted.
1	3:7	Unused	-	-	-

Table 46. MAC Rx Status Vector Interrupt Enable

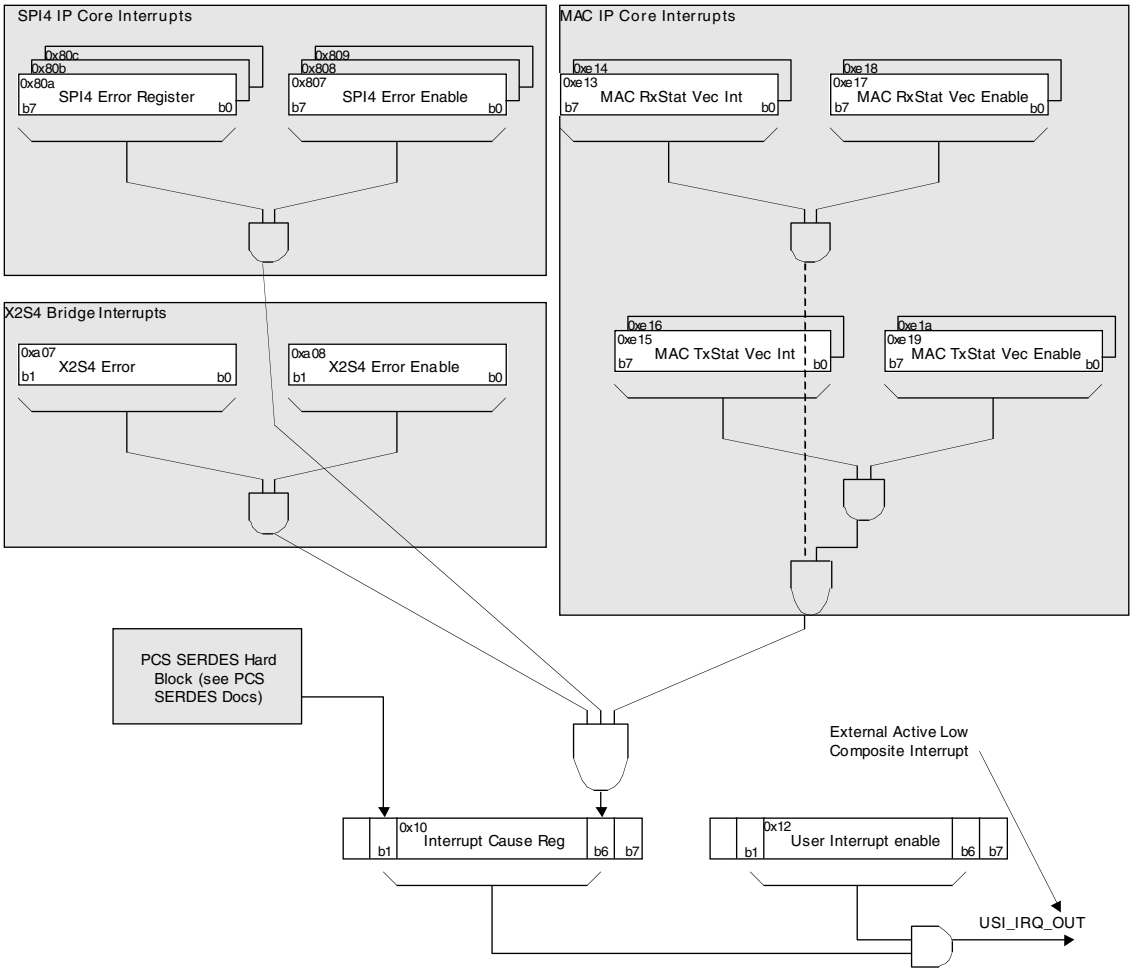
MCRXSTATVEC_INT_EN					Base Address: 0xe17
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCRXSTATVEC_INT_EN_0	R/W	0H	Receive Status Vector Interrupt Enable - Bit for bit active high interrupt enables for the Receive Status Vector Interrupt registers.
1	0:5	MCRXSTATVEC_INT_EN_1	R/W	0H	
1	6:7	Unused	-	-	

Table 47. MAC Tx Status Vector Interrupt Enable

MCTXSTATVEC_INT_EN					Base Address: 0xe19
Address Offset	Bits	Name	Type	Default	Description
0	0:7	MCTXSTATVEC_INT_EN_0	R/W	0H	Transmit Status Vector Interrupt Enable - Bit for bit active high interrupt enables for the Receive Status Vector Interrupt registers.
1	0:2	MCTXSTATVEC_INT_EN_1	R/W	0H	
1	3:7	Unused	-	-	

Interrupt Structure

Figure 8.



IP Core Configuration

This section provides information on the specific configuration used for each of the IP cores used in this design. The first part of each section lists and describes some of the more central aspects of the IP core configurations. The second part lists the contents of the entire “parameters” file for the IP core. This file shows all of the programmable entities of the IP core and their default values. Many of these parameters can be tailored for each user application.

SPI4.2 IP Core

Central Aspects

- Synchronization: An independent 100-125MHz SPI4.2 transmit timing reference and a frequency multiplication PLL (4x = 400-500MHz) are used to supply the SPI4.2 line rate DDR clock. This same timing reference also provides the timing reference to another PLL used to generate the System Data Clock (SDCK) operating with an output frequency of 130-150MHz. This clock is used for the interior user data interface side of the IP core providing for over-speed relative to the equivalent SPI4.2 line rate clock. A separate timing reference for SDCK can be brought into the device if necessary as a user customization.
- Status Mode: The core is operated in Transparent Status mode externally controlled via the Bridge Interface Logic (BIL). The Calendar Length (CAL_LEN) is set to one and the Calendar repetition rate (CAL_M_) set at 4 in

the default case. The CAL_LEN must be set to a value of one since the Calendar RAM EBR memory is eliminated from the design to improve routing but the CAL_M value can be set to anything based on each user's unique application (the receive and transmit calendar RAM outputs of the SPI4.2 are not used by the Bridge and therefore the Calendar RAM EBRs are minimized away during synthesis. The Calendar RAM connections are made between the SPI4.2 IP Core and the Bridge for future per-channel enhancements but inside the esc.v and isc.v modules they are not used in this initial version).

- Dynamic Timing: The SPI4.2 core operates in Dynamic Timing mode with Training enabled and the Deskew module equipped. Training, scheduling, etc. are set uniquely for each application.
- FIFO Thresholds: All SPI4.2 internal FIFO thresholds are "real-time" programmable via the SMI. Initial values are set to achieve as shallow as possible FIFOs to minimize data storage that must be absorbed during flow control. This will also minimize SPI4.2 Lmax.
- General Options: Burst Mode - NPUs typically like SPI4.2 burst segments to be larger than a single SPI4.2 credit (16 bytes) in order to efficiently move the data and utilize memory storage. To accommodate this, the Burst Mode can be used to ensure that blocks of at least N credits in size are transmitted without an EOP.
- Line Packing - In order to efficiently use the SPI4.2 line, packing is turned on but given the SPI4.2 line overspeed it is not required.

Parameters.v File

```
// Rx Top Level Defines For Static Inputs To S4 Core
`define RXNUMDIP4             5'd3       // Receive number of correct DIP4 before in-sync
`define RXNUMDIP4E            5'd7       // Receive number of incorrect DIP4 before out-sync
`define RXCAL_M               8'd4       // Receive calendar repetitions (Valid range 1 - 255)
`define RXCAL_LEN             10'd1      // Receive calendar length (Valid range 1 - 512)
`define RXINTSTC              1'b0       // Receive internal status control (act=1)
`define RX_STAT_FF_BEHAVIOR   1'b0       // Receive status FIFO flag behavior
`define RXSTREDGE                        // Status Channel active edge (defined=rising)
`define RXF1AETHRSH           9'd40      // Receive FIFO1 almost empty threshold
`define RXF1AFTHRSH           9'd400     // Receive FIFO1 almost full threshold
`define RXF2AETHRSH           9'd40      // Receive FIFO2 almost empty threshold
`define RXF2AFTHRSH           9'd80      // Receive FIFO2 almost full threshold
`define RXCALINITF                       // Receive calendar init file (not supported)
// `define RXSTLVDSBUF        // Receive status buffer type (defined=LVDS)
// Rx Top Level Defines for synthesis parameters
`define RXSTAT_MD                        // Receive status mode (RAM(default)/transparent(defined))
`define AIL                              // Defined = AIL, Not defined uses DLL/Static Delay Control
`define RXDCNTL               9'd0       // Non-AIL static mode data delay value
`define RXALNMD               1'b1       // Receive clk/data alignment (dynamic=1)
`define RXTRAIN_EN            1'b1       // Receive training enable (train req for in-sync = 1)
`define RXDESKEW_EN           1'b1       // Receive deskew enable (allow deskew = 1)

// Tx Top Level Defines For Static Inputs To M4 Core
`define TXNUMDIP2             3'd3       // Transmit number of correct DIP2 before in-sync
`define TXNUMDIP2E            3'd2       // Transmit number of incorrect DIP2 before out-sync
`define TXCAL_M               8'd4       // Transmit calendar repetitions (Valid range 1 - 255)
`define TXCAL_LEN             10'd1      // Transmit calendar length (Valid range 1 - 512)
`define TXINTSTC              1'b0       // Transmit internal status control (act=1)
// `define TXSTREDGE                     // Transmit status channel active edge (defined=rising)
`define TXBLEN                6'd4       // Transmit maximum single channel burst(16 byte units) (MaxBurst1)
`define TXEN_PACK             1'b1       // Transmit packing enable
`define TXMAXT                16'h8000   // Transmit maximum training interval (txmaxt=0 & txrep = 0 = no training)
`define TXREP                 8'd10      // Transmit training pattern repetitions
`define TXF1AEBTHRSH          7'd6       // Transmit FIFO1 almost empty Burst threshold
                                         // must be set at least 2 greater than TXBLEN
```

10G+MAC IP Core

Central Aspects

- Flow control: 1) The MAC is configured to "respond to flow control" in both directions. Inhibiting transmit data in the Ingress direction and generating pause frames for the Egress direction. 2) The MAC will optionally drop flow control frames.
- FCS: In the default state, the MAC is programmed to check and pass FCS in the receive direction and to check and pass FCS in the transmit direction. The transmit side can be optionally configured to append FCS and the receiver to strip FCS after checking it.

- IPG: For Hi-Gig mode operation the IPG value is set to 0 resulting in 8 bytes average. For standard Ethernet mode it is set to 1 resulting in 12 bytes of IPG average.
- Frame Length: Set for support of Jumbo Frames. Note that all packets are passed regardless of size. The frame length field is used to determine over-sized packets received.
- Address Filtering: There are a number of filtering options here that can be used to filter/drop packets that are not meant to cross the bridge. It is assumed that in this application that most of these capabilities will not be of much use but may be based on the user application without affecting bridging functions.
- Short Packets: Short packets are dropped inside the MAC.
- Pause OPCODE: Set to “00 01” as specified in the Ethernet standard for Pause. Programmable for custom applications.

Parameters.v File

```
// *** GUI Level Parameters ***
`define USER_NAME mac
`define XAUI_INTERFACE
// Mode
`define TX_EN                   1'b1
`define RX_EN                   1'b1
// Tx Config
`define TX_IPG_STRETCH          1'b0
`define TX_GEN_FCS              1'b0
`define TX_PAUSE_EN             1'b1
// Rx Config
`define RX_PAUSE_DROP           1'b1
`define RX_RECEIVE_SHORT        1'b1
`define RX_RECEIVE_ALL_MC       1'b0
`define RX_PAUSE_EN             1'b1
`define RX_PRMS                 1'b1
`define RX_RECEIVE_BC           1'b0
`define RX_DISCARD_FCS          1'b0
// TX IPG
`define TX_IPG_VAL              5'b00000

// RX MAX Frame Length
`define RX_MAX_FRM_LEN_L        8'hee
`define RX_MAX_FRM_LEN_H        8'h05
// MAC Address
`define RX_MAC_ADDR_0_L         16'h0000
`define RX_MAC_ADDR_0_H         16'h0000
`define RX_MAC_ADDR_1_L         16'h0000
`define RX_MAC_ADDR_1_H         16'h0000
`define RX_MAC_ADDR_2_L         16'h0000
`define RX_MAC_ADDR_2_H         16'h0000
// Pause Opcode
`define TX_PAUSE_OPCODE_L       8'h01
`define TX_PAUSE_OPCODE_H       8'h00
// Pause Time
`define TX_PAUSE_TIME_L         8'hff
`define TX_PAUSE_TIME_H         8'hff
// VERID
`define VERID                   8'h01
```

PCS SERDES

The PCS SERDES block comes packaged to support full compatibility from Serial I/O to the XGMII interface of the IEEE 802.3-2002 XAUI standard. We do not expect modification of this block in terms of functionality.

xaui_pcs.txt File

```
# This file is used by the simulation model as well as the ispLEVER bitstream
    # generation process to automatically initialize the PCS quad to the mode
    # selected in the IPexpress. This file is expected to be modified by the
    # end user to adjust the PCS quad to the final design requirements.
    # channel_0 is in “XAUI” mode
    # channel_1 is in “XAUI” mode
    # channel_2 is in “XAUI” mode
    # channel_3 is in “XAUI” mode

    ch0 13 03  # Powerup Channel
    ch0 00 01
    ch1 13 03  # Powerup Channel
```

```
    ch1 00 01
    ch2 13 03  # Powerup Channel
    ch2 00 01
    ch3 13 03  # Powerup Channel
    ch3 00 01
    quad 28 40  # Reference clock multiplier
    quad 29 01  # default
    # quad 02 00  # ref_pclk source is ch0
    quad 18 01  # XAUI Mode
    quad 14 7F  # Word Alignment Mask
    quad 15 03  # +ve K
    quad 16 7C  # -ve K
    quad 0D 97  # Watermark level on CTC
    quad 0E 00  # insertion/deletion control of CTC
    quad 12 1C  # XAUI /R/ pattern for CTC match
    quad 13 01
    quad 19 4C  # MCA x4 alignment
    quad 01 00  # MCA mclk select to ch0
    quad 04 0F  # MCA enable 4 channels
    quad 05 00  # MCA latency
    quad 06 05  # MCA depth
    # quad 07 FF  # MCA alignment mask
    # quad 08 7C  # MCA alignment character
    # quad 09 7C  # MCA alignment character
    # quad 0A 15  # MCA k control
    ch0 14 90  # 16% pre-emphasis
    ch1 14 90  # 16% pre-emphasis
    ch2 14 90  # 16% pre-emphasis
    ch3 14 90  # 16% pre-emphasis
    ch0 15 10  # +6dB equalization
    ch1 15 10  # +6dB equalization
    ch2 15 10  # +6dB equalization
    ch3 15 10  # +6dB equalization
    quad 30 04  # Set TX Sync Bit

# These lines must appear last in the autoconfig file.  These lines apply the correct
# reset sequence to the PCS block upon bitstream configuration
    quad 41 00  # de-assert serdes_rst
    quad 40 ff  # assert datapath reset for all channels
    quad 41 03  # assert MCA reset
    quad 41 00  # de-assert MCA reset
    quad 40 00  # de-assert datapath reset for all channels
```

Packet / Frame Formats

SPI4.2 Packet

Figure 9 shows the format of packets expected on the SPI4.2 internal user interface. CRC is optional.

Figure 9. SPI4.2 Packet Format

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15-xx
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Destination Address						Source Address						Length		Data

Ethernet Data Frame

Figure 10 shows the format of an Ethernet data frame.

Figure 10. Ethernet Data Frame

1	2	3	4	5	6	7	8	9	10	11	12	13	14
FB	55	55	55	55	55	55	D5	00	00	00	00	00	00
SOF	Preamble						SFD	Destination Address					

15	16	17	18	19	20	21	22	23 - 68	69	70	71	72
00	00	00	00	00	00	00	00	00	26	6B	AE	0A
Source Address						Length		Data/PAD	CRC			

Tagged Ethernet Data Frame

Figure 11 shows the format of a Tagged Ethernet data frame. Note that only the 12-bit VID field is used. The User Priority and CFI fields are ignored.

Figure 11. Tagged Ethernet Data Frame

1	2	3	4	5	6	7	8	9	10	11	12	13	14
FB	55	55	55	55	55	55	D5	00	00	00	00	00	00
SOF	Preamble						SFD	Destination Address					

15	16	17	18	19	20	21-24				25	26	27 - 68	69	70	71	72
00	00	00	00	00	00	81	00	00	00	00	00	00	26	6B	AE	0A
Source Address						VTAG				Length		Data/ PAD	CRC			

Ethernet Pause Frame

Figure 12 shows the format of an Ethernet Control Frame of the Pause type.

Figure 12. Ethernet Control Pause Frame

1	2	3	4	5	6	7	8	9	10	11	12	13	14
FB	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					

15	16	17	18	19	20	21	22	23	24	25-25	27-68
00	00	00	00	00	00	88	08	00	01	00	00
Source Address						Type		Opcode		PT1-PT2	Data/Pad

69	70	71	72
26	6B	AE	0A
CRC			

X2S4 I/O Signal Descriptions

Table 48. X2S4 I/O Signal Descriptions

Signal Name	Direction, Buffer Type	Description
Common/General I/O		
GRST_N	Input - LVCMOS25	SPI4 IP Core Global core reset (active low).
TXS4_REF	Input - LVCMOS25	Transmit SPI4 Reference Clock - Clock input used to drive the SPI4 line. This input is usually delivered at a rate 1/8 or 1/4 of the SPI4 line (50, 100MHz) and multiplied inside using a PLL to a 4x rate (400MHz) to drive the line. It is also used in the default configuration to source the internal SPI4 System Data Clock (S4SDCK) see below. The default configuration provided utilizes a 50MHz TXS4_REF. If a 100MHz TXS4_REF is desired, the CLKFB_DIV setting for both the spi4_tx_pll.v (8 to 4) and the spi4_sdck_pll.v (52 to 26) must be changed. Note that only one of the two transmit PLL types (spi4_tx_pll.v or spi4_tx_pll_loc_md.v) provided are used and depends on the device package chosen.

Table 48. X2S4 I/O Signal Descriptions (Continued)

Signal Name	Direction, Buffer Type	Description
SDCK	Input - LVCMOS25	System Data Clock - Optional input that can be used to source the SPI4 System Data Clock (S4SDCK) directly or as a timing reference input to an internal PLL. If this input is not used, a S4SDCK clock is still required inside the Bridge and is usually “looped” up from the TXS4_REF timing reference input to a frequency that has ~30% over-speed relative to the 400MHz SPI4 line clock / 4 = 100MHz * 1.3 = 130MHz s4sdck). Selections take place at the top level rtl file. This input is not available in the default configuration provided but can be added as needed.
MONOUT[11:0]	Output - LVCMOS25	Monitor Output - General purpose monitor I/O for debug. These pins should be made probe accessible.
SPI4.2 Bus I/O		
RCLK_P	Input - LVDS	Receive SPI4 Line Clock (P) - Used to sample the RDAT_[P:N][15:0] data bus.
RCLK_N	Input - LVDS	Receive SPI4 Line Clock (N) - Used to sample the RDAT_[P:N][15:0] data bus.
RDAT_P[15:0]	Input - LVDS	Receive SPI4 Data (P) - SPI4 16 bit input DDR data bus.
RDAT_N[15:0]	Input - LVDS	Receive SPI4 Data (N) - SPI4 16 bit input DDR data bus.
RCTL_P	Input - LVDS	Receive SPI4 Control (P) - Control (=1) / Data (=0) indicator.
RCTL_N	Input - LVDS	Receive SPI4 Control (N) - Control (=0) / Data (=1) indicator
RX_STATUS[1:0]	Output - LVTTL33	Receive Status - Output status channel associated with input/receive data path.
RX_STATUS_CK	Output - LVTTL33	Receive Status Clock - Output status channel clock. Driven at the top-level via Primary Clock network.
TDCLK_P	Output- LVDS	Transmit SPI4 Line Clock (P) - Forward clock associated with the TDAT_[P:N][15:0] data bus.
TDCLK_N	Output- LVDS	Transmit SPI4 Line Clock (N) - Forward clock associated with the TDAT_[P:N][15:0] data bus.
TDAT_P[15:0]	Output - LVDS	Transmit SPI4 Data (P) - SPI4 16 bit output DDR data bus.
TDAT_N[15:0]	Output - LVDS	Transmit SPI4 Data (N) - SPI4 16 bit output DDR data bus.
TCTL_P	Output - LVDS	Transmit SPI4 Control (P) - Control (=1) / Data (=0) indicator.
TCTL_N	Output - LVDS	Transmit SPI4 Control (N) - Control (=0) / Data (=1) indicator
TX_STATUS[1:0]	Input - LVTTL33	Transmit Status - Input status channel associated with input/receive data path.
TX_STATUS_CK	Input - LVTTL33	Transmit Status Clock - Input status channel clock.
UMI Bus		
UMI_CLK	Input - LVCMOS25	User Master Interface Clock - Synchronous User Master mode clock.
UMI_CS	Input - LVCMOS25	User Master Interface Chip Select - Active low chip select
UMI_WR	Input - LVCMOS25	User Master Interface Write - Write/Read select (0=read, 1=write)
UMI_RDY_N	Input - LVCMOS25	User Master Interface Ready - Active low ready strobe indicating that address is valid for read and address and data for write cycles.
UMI_ADDR[17:0]	Input - LVCMOS25	User Master Interface Address - User master interface address (b17 is most significant bit, b0 is least).
UMI_ACK	Output - LVCMOS25	User Master Interface Acknowledge - Active high acknowledgment that data is available on reads and that another cycle may begin. On writes, acknowledgment that another cycle may begin.
UMI_DATA[7:0]	Biput - LVCMOS25	User Master Interface Data - User master bi-directional data bus
UMI_ERR	Output - LVCMOS25	User Master Interface Error - Active high error signal indicating that an out-of-range address was accessed.
USI_IRQ_OUT	Output - LVCMOS25	User Slave Interface Interrupt Request Output - Active low composite interrupt output for the entire Bridge.
PCS SERDES		
A_REFCLKP	Input - CML	Reference Clock Positive - SERDES reference clock pos rail (156.25MHz 10G operation, 187.5MHz 12G operation)

Table 48. X2S4 I/O Signal Descriptions (Continued)

Signal Name	Direction, Buffer Type	Description
A_REFCLKN	Input - CML	Reference Clock Negative - SERDES reference clock neg rail (156.25MHz 10G operation, 187.5MHz 12G operation)
HDINP[3:0]	Input - CML	High Speed Data Input Positive - High speed SERDES input pos rail.
HDINN[3:0]	Input - CML	High Speed Data Input Negative - High speed SERDES input neg rail.
HDOUTP[3:0]	Output - CML	High Speed Data Output Positive - High speed SERDES output pos rail.
HDOUTN[3:0]	Output - CML	High Speed Data Output Negative - High speed SERDES output neg rail.

Note: Total I/O = XX

X2S4 I/O Timing Specifications

Table 49. X2S4 Input Timing Specifications

Input	Edge	Reference Clock	Setup	Hold	Pu/Pd
GRST_N		ASYNC			Pu
RDAT_P[15:0]	+	RCLK_[P:N][1]	-	-	-
RDAT_N[15:0]	+	RCLK_[P:N][1]	-	-	-
RCTL_P	+	RCLK_[P:N][1]	-	-	-
RCTL_N	+	RCLK_[P:N][1]	-	-	-
TX_STATUS[1:0]	+/-	TX_STATUS_CK[2]	2	.5	-
UMI_CS	+	UMI_CLK	2	1	-
UMI_WR	+	UMI_CLK	2	1	-
UMI_RDY_N	+	UMI_CLK	2	1	-
UMI_ADDR[17:0]	+	UMI_CLK	2	1	-
UMI_DATA[7:0]	+	UMI_CLK	2	1	-
HDINP[3:0]		A_REFCLK[P:N][3]	-	-	-
HDINN[3:0]		A_REFCLK[P:N][3]	-	-	-

1. These inputs are sampled using the Adaptive Input Logic (AIL) feature. The internal AIL circuitry automatically aligns the clock/data relationship to ensure the necessary setup/hold window is maintained. AIL supports bit rates in excess of 500MHz DDR.
2. These inputs can be supported on either edge of the input clock through programmable options. The default is the rising edge.
3. These are high-speed SERDES inputs where automatic clock recovery is used in sampling. Clock signal A_REFCLK[P:N] is used simply as a timing reference to in internal PLL.

Table 50. X2S4 Output Timing Specification

Input	Edge	Reference Clock	Prop Delay Mix/Max		Cap Loading
MONOUT[11:0]	+	TXS4_REF[1]	0	20	40pf
RX_STATUS[1:0]	+	RX_STATUS_CK	-1	2	25pf
RX_STATUS_CK	+	TXS4_REF[2]	-	-	25pf
TDCLK_P	+	TXS4_REF[3]	-	-	10pf
TDCLK_N	+	TXS4_REF[3]	-	-	10pf
TDAT_P[15:0]	+	TDCLK_[P:N]	-.01	.024	10pf
TDAT_N[15:0]	+	TDCLK_[P:N]	-.01	.024	10pf
TCTL_P	+	TDCLK_[P:N]	-.01	.024	10pf
TCTL_N	+	TDCLK_[P:N]	-.01	.024	10pf
UMI_ACK	+	UMI_CLK	0	8.5	25pf

Table 50. X2S4 Output Timing Specification (Continued)

Input	Edge	Reference Clock	Prop Delay Mix/Max		Cap Loading
UMI_DATA[7:0]	+	UMI_CLK	0	8.5	25pf
UMI_ERR	+	UMI_CLK	0	8.5	25pf
USI_IRQ_OUT	+	UMI_CLK	0	8.5	25pf
HDOUTP[3:0]		A_REFCLK[P:N][4]	-	-	-
HDOUTN[3:0]		A_REFCLK[P:N][4]	-	-	-

1. These debug signals are clocked off chip with internal clock signal S4SDCK which is driven from an internal PLL receiving a timing reference from TXS4_REF. MONOUT[12,13, and 14] are outputs of the “asynchronous” type and do not have a clock reference. Actually, they are themselves clocks. See section for MONOUT[] debug assignments.
2. This output clock can be optionally inverted before leaving the device in order to support either edge sampling at the far-end of the SPI4.2 line. Actual delay (min/max) from TXS4_REF is irrelevant given that this clock output is used in a clock forwarded interface configuration.
3. Actual delay (min/max) from TXS4_REF is irrelevant given that this clock output is used in a clock forwarded interface configuration.
4. These are high-speed SERDES outputs. A_REFCLK[P:N] is used as a timing reference to an internal PLL which supplies the clock used to clock data off-chip. Since the far-end will be using clock recovery techniques, propagation delay (min/max) is not relevant here (see Lattice web site for more details on SERDES specifications).

Pins and Packages

Shown below are the pinouts for the X2S4 Bridge implemented in an SCM15 array and the FPBGA256 package. Pinouts for other device and package combinations will be made available on as needed basis upon request.

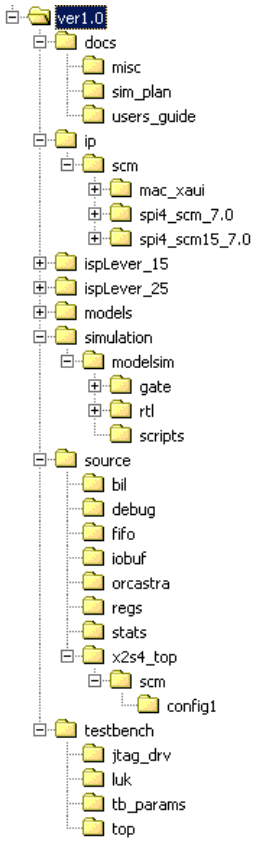
Port Name	Pin/Bank	Buffer Type	Site	Properties
A_REFCLKN	A_REFCLKN_L/0		A_REFCLKN_L	
A_REFCLKP	A_REFCLKP_L/0		A_REFCLKP_L	
DIFFR_2	G12/2	REF_RESISTOR	PR22D	
DIFFR_3	L13/3	REF_RESISTOR	PR35	
DIFFR_6	M2/6	REF_RESISTOR	PL35D	
DIFFR_7	G5/7	REF_RESISTOR	PL22D	
grst_n	F5/0		RESETN	
hdinn00	A_HDINN0_L/0		A_HDINN0_L	
hdinn01	A_HDINN1_L/0		A_HDINN1_L	
hdinn02	A_HDINN2_L/0		A_HDINN2_L	
hdinn03	A_HDINN3_L/0		A_HDINN3_L	
hdinp00	A_HDINP0_L/0		A_HDINP0_L	
hdinp01	A_HDINP1_L/0		A_HDINP1_L	
hdinp02	A_HDINP2_L/0		A_HDINP2_L	
hdinp03	A_HDINP3_L/0		A_HDINP3_L	
hdoutn00	A_HDOUTN0_L/0		A_HDOUTN0_L	
hdoutn01	A_HDOUTN1_L/0		A_HDOUTN1_L	
hdoutn02	A_HDOUTN2_L/0		A_HDOUTN2_L	
hdoutn03	A_HDOUTN3_L/0		A_HDOUTN3_L	
hdoutp00	A_HDOUTP0_L/0		A_HDOUTP0_L	
hdoutp01	A_HDOUTP1_L/0		A_HDOUTP1_L	
hdoutp02	A_HDOUTP2_L/0		A_HDOUTP2_L	
hdoutp03	A_HDOUTP3_L/0		A_HDOUTP3_L	
monout_0	F8/2	LVCMOS25_OUT	PR22B	DRIVE:8mA SLEW:SLOW PULL:UP
monout_1	G4/7	LVCMOS25_OUT	PL18D	DRIVE:8mA SLEW:SLOW PULL:UP
monout_10	L5/5	LVCMOS25_OUT	PB13C	DRIVE:8mA SLEW:SLOW PULL:UP
monout_11	M7/5	LVCMOS25_OUT	PB15D	DRIVE:8mA SLEW:SLOW PULL:UP
monout_2	H4/7	LVCMOS25_OUT	PL24C	DRIVE:8mA SLEW:SLOW PULL:UP
monout_3	H5/7	LVCMOS25_OUT	PL22C	DRIVE:8mA SLEW:SLOW PULL:UP
monout_4	H6/7	LVCMOS25_OUT	PL24D	DRIVE:8mA SLEW:SLOW PULL:UP
monout_5	J5/6	LVCMOS25_OUT	PL26C	DRIVE:8mA SLEW:SLOW PULL:UP
monout_6	J6/6	LVCMOS25_OUT	PL26D	DRIVE:8mA SLEW:SLOW PULL:UP
monout_7	K4/6	LVCMOS25_OUT	PL28D	DRIVE:8mA SLEW:SLOW PULL:UP
monout_8	L2/6	LVCMOS25_OUT	PL31C	DRIVE:8mA SLEW:SLOW PULL:UP
monout_9	L4/6	LVCMOS25_OUT	PL28C	DRIVE:8mA SLEW:SLOW PULL:UP
rctl_p	P16/4	LVDS_IN	PB44A	DIFFRES:120
rdat_p_0	R3/5	LVDS_IN	PB3A	DIFFRES:120
rdat_p_1	P5/5	LVDS_IN	PB8A	DIFFRES:120
rdat_p_10	T10/4	LVDS_IN	PB28A	DIFFRES:120
rdat_p_11	N9/4	LVDS_IN	PB31A	DIFFRES:120
rdat_p_12	T11/4	LVDS_IN	PB32A	DIFFRES:120
rdat_p_13	P11/4	LVDS_IN	PB35A	DIFFRES:120
rdat_p_14	T12/4	LVDS_IN	PB36A	DIFFRES:120
rdat_p_15	T13/4	LVDS_IN	PB37A	DIFFRES:120

rdat_p_2	T4/5	LVDS_IN	PB9A	DIFFRES:120
rdat_p_3	R6/5	LVDS_IN	PB12A	DIFFRES:120
rdat_p_4	P6/5	LVDS_IN	PB15A	DIFFRES:120
rdat_p_5	R8/5	LVDS_IN	PB16A	DIFFRES:120
rdat_p_6	N7/5	LVDS_IN	PB17A	DIFFRES:120
rdat_p_7	R9/5	LVDS_IN	PB20A	DIFFRES:120
rdat_p_8	M8/5	LVDS_IN	PB21A	DIFFRES:120
rdat_p_9	P8/5	LVDS_IN	PB24A	DIFFRES:120
rdclk_p	N13/4	LVDS_IN	PB49A	DIFFRES:120
rstatus_0	M12/4	LVTTL33_OUT	PB43B	DRIVE:8mA SLEW:SLOW PULL:UP
rstatus_1	R14/4	LVTTL33_OUT	PB47C	DRIVE:8mA SLEW:SLOW PULL:UP
rstatus_ck	L12/4	LVTTL33_OUT	PB43A	DRIVE:8mA SLEW:SLOW PULL:UP
tctl_p	P1/6	LVDS_OUT	PL45A	DIFFCUR:3P5 REFC:INT
tdat_p_0	F2/7	LVDS_OUT	PL15A	DIFFCUR:3P5 REFC:INT
tdat_p_1	F3/7	LVDS_OUT	PL17A	DIFFCUR:3P5 REFC:INT
tdat_p_10	H14/2	LVDS_OUT	PR23A	DIFFCUR:3P5 REFC:INT
tdat_p_11	M15/3	LVDS_OUT	PR43A	DIFFCUR:3P5 REFC:INT
tdat_p_12	K1/6	LVDS_OUT	PL28A	DIFFCUR:3P5 REFC:INT
tdat_p_13	L3/6	LVDS_OUT	PL35A	DIFFCUR:3P5 REFC:INT
tdat_p_14	M1/6	LVDS_OUT	PL37A	DIFFCUR:3P5 REFC:INT
tdat_p_15	M5/6	LVDS_OUT	PL43A	DIFFCUR:3P5 REFC:INT
tdat_p_2	H3/7	LVDS_OUT	PL22A	DIFFCUR:3P5 REFC:INT
tdat_p_3	H1/7	LVDS_OUT	PL23A	DIFFCUR:3P5 REFC:INT
tdat_p_4	J2/7	LVDS_OUT	PL24A	DIFFCUR:3P5 REFC:INT
tdat_p_5	J4/6	LVDS_OUT	PL26A	DIFFCUR:3P5 REFC:INT
tdat_p_6	H16/3	LVDS_OUT	PR37A	DIFFCUR:3P5 REFC:INT
tdat_p_7	L15/3	LVDS_OUT	PR35A	DIFFCUR:3P5 REFC:INT
tdat_p_8	F16/3	LVDS_OUT	PR28A	DIFFCUR:3P5 REFC:INT
tdat_p_9	J14/3	LVDS_OUT	PR26A	DIFFCUR:3P5 REFC:INT
tdclk_p	F14/2	LVDS_OUT	PR17A	DIFFCUR:3P5 REFC:INT
tstatus_0	R16/4	LVTTL33_IN	PB40B	PULL:UP
tstatus_1	N12/4	LVTTL33_IN	PB39C	PULL:UP
tstatus_ck	T15/4	LVTTL33_IN	PB40A	PULL:UP
txs4_ref	E14/2	LVCMOS25_IN	PR15A	PULL:UP
umi_ack	C10/1	LVCMOS25_OUT	PT27A	DRIVE:16mA SLEW:SLOW PULL:UP
umi_addr_0	E7/1	LVCMOS25_IN	PT21C	PULL:UP
umi_addr_1	D16/2	LVCMOS25_IN	PR24C	PULL:UP
umi_addr_10	G16/2	LVCMOS25_IN	PR18D	PULL:UP
umi_addr_11	K12/3	LVCMOS25_IN	PR31C	PULL:UP
umi_addr_12	M16/3	LVCMOS25_IN	PR45B	PULL:UP
umi_addr_13	N5/5	LVCMOS25_IN	PB5D	PULL:UP
umi_addr_14	H12/2	LVCMOS25_IN	PR24A	PULL:UP
umi_addr_15	P2/6	LVCMOS25_IN	PL41D	PULL:UP
umi_addr_16	T2/5	LVCMOS25_IN	PB3D	PULL:UP
umi_addr_17	T3/5	LVCMOS25_IN	PB3C	PULL:UP
umi_addr_2	E15/2	LVCMOS25_IN	PR15B	PULL:UP
umi_addr_3	F9/2	LVCMOS25_IN	PR22A	PULL:UP
umi_addr_4	G13/2	LVCMOS25_IN	PR22C	PULL:UP
umi_addr_5	H11/2	LVCMOS25_IN	PR24B	PULL:UP
umi_addr_6	J11/3	LVCMOS25_IN	PR26D	PULL:UP
umi_addr_7	K13/3	LVCMOS25_IN	PR28C	PULL:UP
umi_addr_8	K16/3	LVCMOS25_IN	PR41D	PULL:UP
umi_addr_9	L16/3	LVCMOS25_IN	PR45A	PULL:UP
umi_clk	J12/3	LVCMOS25_IN	PR26C	PULL:UP
umi_cs	E12/1	LVCMOS25_IN	PT39B	PULL:UP
umi_data_0	A11/1	LVCMOS25_BIDI	PT33B	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_1	C12/1	LVCMOS25_BIDI	PT35A	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_2	A12/1	LVCMOS25_BIDI	PT35B	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_3	B12/1	LVCMOS25_BIDI	PT36C	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_4	A13/1	LVCMOS25_BIDI	PT36D	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_5	D10/1	LVCMOS25_BIDI	PT37A	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_6	C11/1	LVCMOS25_BIDI	PT37B	DRIVE:16mA SLEW:SLOW PULL:UP
umi_data_7	E10/1	LVCMOS25_BIDI	PT37C	DRIVE:16mA SLEW:SLOW PULL:UP
umi_err	B11/1	LVCMOS25_OUT	PT33A	DRIVE:16mA SLEW:SLOW PULL:UP
umi_rdy_n	E9/1	LVCMOS25_IN	PT32D	PULL:UP
umi_wr	A10/1	LVCMOS25_IN	PT27B	PULL:UP
usr_irq_out	E8/1	LVCMOS25_OUT	PT32B	DRIVE:16mA SLEW:SLOW PULL:UP

User Information

Figure 13 is a screen shot of the X2S4 directory structure. Note that the HiGig directories are not shown and will not be present unless arrangements are made with Lattice to provide support for the HiGig Ethernet over XAUI capability.

Figure 13. X2S4 Directory Structure



Simulation (rtl, gate, gate + timing)

See X2S4 Simulation Plan document for details.

Place and Route

Fully populated place and route directories are provided in four specific combinations.

- ispLever_15: Standard Ethernet mode XAUI operation. UMI control interface. SCM15-FPBGA256p package.
- ispLever_higig_15: HiGig mode XAUI operation. UMI control interface. SCM15-FPBGA256p package (Contact Lattice sales).
- ispLever_25: Standard Ethernet mode XAUI operation. JTAG control interface. All packages larger than the 256 fpBGA.
- ispLever_higig_25: HiGig mode XAUI operation. JTAG control interface. All packages larger than the fpBGA (Contact Lattice sales).

Other configurations can be created by the user if necessary (e.g., LFSCM-25 with a UMI interface). Supporting configuration information used to control various settings of the IP (SPI4.2, MAC) and Bridge can be found in one of three different “parameters.v” files (spi4_256ch_params.v, x2s4_params.v, ten_gbemac_defines.v) located inside the ispLEVER place and route directories. These files are copies of the files used during simulation that are located in \source\x2s4_top\scm\config1. Care must be taken to ensure that any change made to these files for simulation is then also made to the same file in the Place and Route directories. Descriptions of the individual parameters can be found in the corresponding User’s Guide for the IP blocks. These parameters can be altered by the user in a manner consistent with advisement from the user’s guide.

Trce Exceptions for Designs Using ispLEVER 7.0 Service Pack 2 or Earlier Software
The following note is from the readme.htm file for the SPI4 IP core located under the IP directory: "Trce, in its current form, cannot handle correctly the analysis of data transfer from the rxs4ls2_ck to rxs4ls4_ck clock domains for hold checking only. In circuit operation, every other rxs4ls2_ck edge is used to launch data into the rxs4ls4_ck domain and the edge selected is the one co-incident with the falling edge of the low speed clock. Trce incorrectly defaults to using the high-speed edge co-incident with the rising edge of the low-speed clock and therefore flags a hold problem since the high-speed clock domain produces the low-speed clock."

There is no actual hold violation and so these hold-time errors can be ignored. The preference file provided contains a "block net" preference on these transfers into the EBR blocks (BLOCK PATH FROM CLKNET "rxs4ls2_ck" TO CLKNET "rxs4ls4_ck" ;) that is commented out. For hold checks, these lines can be uncommented and errors will not be reported.

Warning: This BLOCK NET cannot be used for setup checking. Refer to this readme.htm file for other SPI4 IP Core for related issues and questions.

Configuration Information
It is important to understand which Verilog file, parameter, or variable must be changed in order to get the desired circuit. Here are some examples of possible changes and how to make the change.

- Switch from UMI to JTAG Control Interface: This change requires two different Verilog source files depending upon interface chosen. In ispLEVER, remove iobuf_umi.v and x2s4_top_sb_umi.v and replace them with iobuf.v and x2s4_top_sb.v.
- Change of the SPI4.2 Calendar length from the default value of 0x4 to 0x8: Simply edit the spi4_256ch_params.v file.

Other Important Information
ispLEVER 6.1sp2 Only: If running place and route for the LatticeSCM15 using ispLEVER 6.1 SP2, the user must set a system variable as shown below. Do not set the variable for ispLEVER 7.0 and later implementations.

```
PAR_EXCLUDE_WIRES=<path>\exclude.txt
```

where path is equal to the system path to the directory where place and route is run.

ispLEVER 7.1 Only: By default as received in the X2S4 package, the synthesis property "pipe-lining and retiming" is turned off (set to none). This property must be remain turned off in ispLEVER 7.1 X2S4 designs.

Resource Utilization

Table 51. Resource Utilization

No	XAUI Mode	Proc Bus Interface	Device / Package	Slice, LUT, Register, EBR Count	Tested RTL Sim, Gate Sim, Syn, Par, Static Timing	Tested Lab
1	HiGig/HiGig+	UMI	LFCSM15-256	5245, 6313, 6479, 50	Pass	N/A
2	Standard	UMI	LFCSM15-256	5295, 6421, 6542, 50	Pass	N/A
3	HiGig/HiGig+	JTAG	LFCSM25-900	5306, 6396, 6553, 50	Pass	Pass
4	Standard	JTAG	LFCSM25-900	5286, 6471, 6392, 50	Pass	Pass

1) Numbers above were obtained using ispLEVER 7.0

Power Requirements

Table 52. Power Requirements

No	XAUI Mode	XAUI Speed	Test Condition	1.2V A	2.5V A	Power
1	HiGig	10G	Idle XAUI Line	1.39	.380	2.618
			Full Line Rate Increment Byte	1.65	.395	2.967
			Full Line Rate Random Byte	1.75	.410	3.125
2	HiGig+	12G	Idle XAUI Line	1.45	.380	2.690
			Full Line Rate Increment Byte	1.70	.395	3.027
			Full Line Rate Random Byte	1.80	.410	3.185
3	Standard	10G	Idle XAUI Line	1.385	.365	2.575
			Full Line Rate Increment Byte	1.625	.39	2.925
			Full Line Rate Random Byte	1.725	.405	3.086

1) Numbers shown are measured values based on a given test condition.
2) 1.2V supplies both FPGA core and PLL logic, 2.5V supplies both VCCIO and VCCAUX (LVDS).
3) Numbers shown are for full 10/12Gbs sustained line rates (5 minutes). Actual numbers may be less than this for more typical traffic conditions).

IP Licensing

As mentioned previously, an IP license for the LatticeSCM SPI4.2 MACO IP core is required to fully evaluate the X2S4 Bridge reference design and an IP license for the 10GMAC+ IP core is required to use the X2S4 Bridge in a final design.

A MACO IP license is required to generate bitstreams for any designs containing MACO IP cores. All MACO IP is licensed free of charge, however a license key is required. Contact your local Lattice sales office to obtain a license key.

The 10G+MAC soft IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. With this capability and a MACO IP license it is possible to generate programming files that can be downloaded into a device and evaluate Bridge functionality in hardware. After four hours the Bridge will stop working and it will be necessary to reprogram the device to re-enable operation. Users can generate versions of the Bridge with no restrictions by obtaining a license for the 10GMAC+ IP core. The Ordering Part Number (OPN) for the 10GMAC+ IP core targeting LatticeSCM devices is ETHER-10G-SC-U3.

For HiGig/HiGig+ support, please contact your local Lattice sales office. The Ordering Part Number (OPN) for the HiGig+ MAC IP core targeting LatticeSC/M devices is HIG-MAC-SC-U1.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
August 2007	01.0	Initial release.
October 2007	01.1	Updated descriptions for TXS4_REF and SDCK in the X2S4 I/O Signal Descriptions table.
October 2007	01.2	Added Trce Exceptions text section.
April 2008	01.3	Updates corresponding to the release of ver1.2 of the X2S4 reference design.
June 2008	01.4	Updated Resource Utilization table.
July 2008	01.5	Added Appendix A.

Appendix A. XAUI to SPI4.2 Bridge Version 1.2 Release Notes

Introduction

Numerous modifications are included with version 1.2 of the XAUI to SPI4.2 Bridge:

- Packet Drop & Packet Checking/Correcting features
- Programmable user control to enable or disable flow-control for both ingress and egress directions
- For HiGig mode only, support for small packets (48 byte minimum) without padding
- Correction made in the statistics collection area
- Miscellaneous: Added Aldec simulation support, fixed SPI4 IP hold violations, changed name of primary I/O pin "UMI_RDY" to "UMI_RDY_N" .

This appendix lists the file additions and file changes to version 1.1 required to upgrade an existing design to the 1.2 version. This version of the XAUI to SPI4.2 Bridge Demo Design is being released with ispLEVER 7.1.

File Modifications

RTL New File Additions

- es4c.v: Egress SPI4 Controller – Controls proper egress buffer reading for either packet drop (Store & Forward) or non-drop (Cut-Thru) modes of operation.
- pkt_chk.v: Packet Check – Provides a packet check/correct function for data before being allowed to enter the ingress or egress buffers.

RTL Modifications

- x2s4_top_sb.v, x2s4_top_sb_umi.v – General top level edits to accommodate described feature additions. Changed name of primary I/O pin "UMI_RDY" to "UMI_RDY_N" to correctly reflect it's polarity relative to the internal UMI_RDY signal.
- bill.v, ibc.v, ebc.v – Structural RTL edits to pass new I/O signals through the hierarchy.
- imc.v – Remove dependency on the FIFO empty signal when in packet drop mode.
- isc.v, esc.v – Add new feature to disable flow control in both ingress and egress directions.
- fifo.v, fifo_ctrl.v, eop_ctr.v – Majority of the changes necessary to support packet drop.
- stat_ctr.v – The transmit packet counter (ctr-3) counted both data and pause packets and could under some circumstances miss a count. Data and pause packets are now reported separately for both directions.
- x2s4_regs.v – New register bits for control and error reporting over the packet drop and packet checking functions.
- ten_gbemac_regs.v – New control bit for disabling padding (HiGig only).
- Test-bench RTL Files
  - x2s4_tb_top.v – Miscellaneous modifications to support the new data formats described below under latspi4_df.v
  - latspi4_df.v – Added new data formats (DF11, 12, and 13) to test new features.

New IP

The SPI4.2 IP core has been updated and is required for both standard XAUI and HiGig modes. The HiGig MAC IP core has also been updated. If HiGig is used, it must also be updated. In order to update to the new versions of the IP cores:

- Replace the IP directories (note new names for SPI4.2),

- Copy the .ngo file located in the top level directory for each IP core to the ispLEVER place and route directory,
- When starting ispLEVER for the first time, delete rtl file spi4_scm[15]_70_bb.v (black-box) because the name has changed and load in the new black-box module from the IP directory (example: spi4_scm15_bb.v).

Simulation Libraries

Added pre-compiled simulation libraries for “SE” version of ModelSim (PC or Linux) support (/ver1.2/simulation/modelsim/lib/7.1/jtag_mti_work, pcsa_mti_work, sysbusa_mti_work). Built-in libraries are still used for the OEM version of ModelSim. Perl scripts are updated to take advantage of new libraries.

Users will no longer need to acquire simulation libraries their own. This is now automatic.

Aldec Simulation Support

Added a “/simulation/aldec” directory in support of the new OEM simulator. Simulation operation is the same as for ModelSim.

Perl Scripts

- run_x2s4.plx, run_x2s4_gate.plx – Standard Ethernet mode simulation run scripts edited to support the new features.
- run_x2s4_hg.plx, run_x2s4_hg_gate.plx – HiGig mode simulation run scripts edited to support the new features.
- Perl scripts updated to take advantage of new Modelsim “SE” libs above.
- Perl scripts updated to support Aldec simulation.

Place & Route

Preference File:

- Added new constraints to cover passing of wr_ptr from write clock domain into read domain in support of drop.
- Added new constraint made available in ispLEVER 7.1 to cover illegitimate hold violations (144) associated with the SPI4 IP core (MULTICYCLE FROM CLKNET "rxs4ls2_ck" TO CLKNET "rxs4ls4_ck" 1.000000 X ;)
- Added some general “BLOCK PATH” constraints for paths that can be ignored.

Docs

LatticeSCM XAUI to SPI4.2 Reference Design Guide - RD1033.