VITESSE

VSC7326
Datasheet

Schaumburg™ – 12-Port Tri-Speed MAC with SPI-4.2 Host Interface

FEATURES

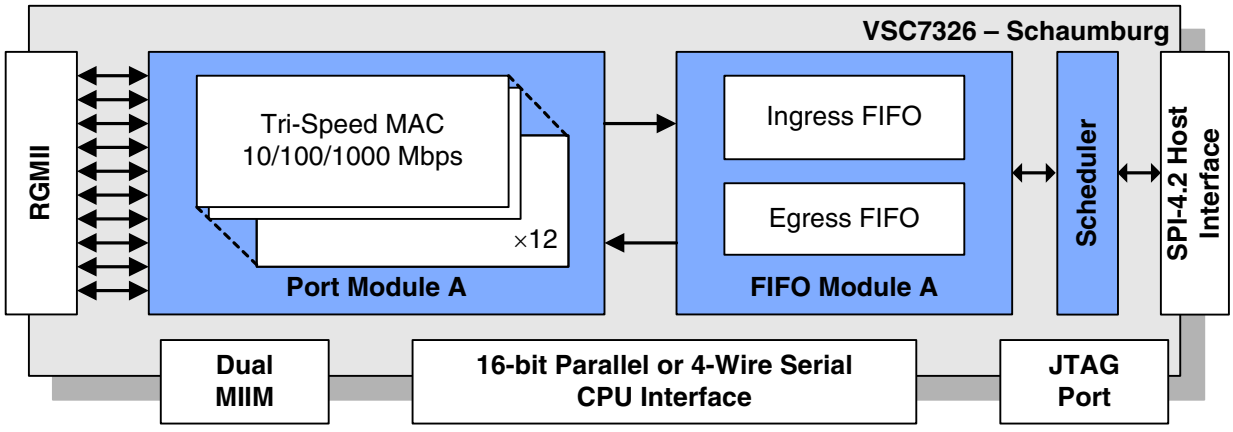
- 12 × 10/100/1000 Mbps RGMII interfaces
- SPI-4.2 host interface with dynamic deskew
- 4032 kilobit ingress buffer, 1152 kilobit egress buffer
- 12-port wire-speed operation
- Store-and-forward and cut-through FIFO operation
- Burst-interleaved or frame-interleaved SPI-4.2 mode
- Full-duplex pause frame flow control, half-duplex backpressure
- Protection of Layer 2 and Layer 3 control frames
- Frame filtering based on Layer 2–4 fields
- Traffic shaping and policing
- Full RMON 1 statistics group
- Applicable IEEE Std 802.3 and SNMP statistics
- Parallel and serial CPU interface
- 10 kB Jumbo Frame support

APPLICATIONS

- Modular switches and routers
- Ethernet-over-SONET access
- Network processor-based network appliances

GENERAL DESCRIPTION

The VSC7326 Schaumburg device is an advanced Ethernet MAC device. For host systems with a standard SPI-4 Phase 2 interface, the VSC7326 provides access to 12 tri-speed (10/100/1000 Mbps) RGMII Ethernet ports.



CORPORATE HEADQUARTERS
Vitesse Semiconductor Corporation
741 Calle Plano
Camarillo, CA 93012
Tel: 1-800-VITESSE • FAX:1-(805) 987-5896

For application support, latest technical literature, and locations of sales offices,
please visit our web site at
www.vitesse.com

PRINTED IN THE U.S.A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

REVISION HISTORY

This section describes the changes that were implemented in this document. The changes are listed by revision, starting with the most current publication.

Revision 4.1

Revision 4.1 of this datasheet was published on December 1, 2005. The following is a summary of the changes implemented in the datasheet:

- In the Tri-Speed MAC Clock/Reset Setup (DEV_SETUP) register, information was added about using RST loopback when running full-frame mode. For more information, see Table 73, page 125.
- In the Master Scheduler Configuration (MSCH) register, the maximum frame size supported when running full-frame mode was added to the description for the BURSTINTLV bit. For more information, see Table 271, page 219.
- In the DC Specifications for LVDS Drivers table, the minimum value for the output offset voltage (V_{OS}) was changed from 1125 mV to 1050 mV. For more information, see Table 282, page 226.
- In the AC Specifications for MII Management table, the values for both MDIO setup and MDIO hold on write were updated. For more information, see Table 289, page 232.
- Several areas concerning SPI-4.2 jitter were updated. In the SPI-4.2 Transmit Data Channel Interface Timing (Dynamic Alignment) table, the maximum value for peak-peak jitter at receiver relative to TDClk ($t_{jr(D)}$)was changed from 0.40 UI to 0.20 UI, and a cross-reference to the SPI-4.2 Jitter Considerations section was added. In the SPI-4.2 Receive Data Channel Interface Timing (Device Output) table, conditions for the RD and RCtrl phase jitter p-p ($t_{j(R)}$) were broken down into two separate conditions. One condition is the PLL1 differential clock with a maximum value of 0.12 UI, and the other is the PLL1 single-ended clock with a maximum value of 0.24 UI. The SPI-4.2 Jitter Considerations section was updated. For more information, see Table 293, page 237, Table 295, page 239, and "SPI-4.2 Jitter Considerations," page 293.

Revision 4.0

Revision 4.0 of this datasheet was published on July 29, 2005. The following is a summary of the changes implemented in the datasheet:

- The SPI_RDClk frequency was increased from 400 MHz to 407 MHz.
- Reference to Layer 4 was removed from the pseudo code showing control frame classification logic and from the information about the control frame protection feature.
- The block size for the fifo_size parameter configuration was corrected.Information about the amount of internal memory was updated.
- Information was moved from the overview for some receive and transmit statistics counters to their respective registers.
- The BIST running time was increased from 100 ms to 200 ms, and the initialization pseudocode was updated.
- The delay of the internal serial link was changed to 1.00 µs.
- The section about the serial interface was modified.

- The diagram for the MIIM protocol was split into a diagram for the Write PHY and a diagram for the Read PHY.
- The values for all recovery bits in the Device 1 G Debug register were updated.
- The RSVO bit was added to the LOCAL_STATUS register.
- The value for V_I was changed from 3.3 V to 2.5 V in the Input/Output section for I_{IH} in the DC Specifications for Miscellaneous table.
- The value for t_{rec} was changed from 60 ns to 100 µs in the AC Specifications for nReset. In the footnote for t_{rec}, the FIFO RAM clearing time was changed from 110 µs to 210 µs.
- The specifications for the jitter measurement method were removed.
- For the SPI-4.2 transmit data channel interface timing (static alignment), the data and control setup time was changed from 170 ps to 225 ps and the data and control hold time was changed from 210 ps to 155 ps, respectively. For the SPI-4.2 transmit status channel interface timing, the TSClk frequency was updated: a minimum value was added, and the maximum value was changed from 100 MHz to 102 MHz. The figure for the SPI-4.2 transmit status channel test circuit was updated. For the SPI-4.2 receive status channel interface timing, information about the conditions for attaining minimum values was added to the paramters for the status setup and hold times.
- The worst-case timing for the width of nCS high ($t_{W(SH)}$) was changed to 1.00 µs.
- Both the diagram for a full posted read access and the PI nDRdy timing specifications were updated.
- Values for the maximum operating current were added.
- An additional configuration option was added to the table for typical current consumption.
- A list of all reserved pins was added.
- Information about the possible need for a differential clock source for the PLL1_Clk was added.
- The Design Considerations section was added.

Revision 2.0

Revision 2.0 of this datasheet was published on February 10, 2005. This was the first publication of the document.

1 FUNCTIONAL DESCRIPTIONS

1.1 Overview

The Schaumburg device features 12 tri-speed Ethernet MACs compliant with IEEE Std 802.3-2002. The following is a list of features.

- RGMII interface supporting 10/100/1000 Mbps operation
- Full-duplex operation, (10/100 Mbps in half-duplex mode)
- Pause frame flow control in full-duplex mode, symmetric or asymmetric
- Backpressure by collisions in half-duplex mode
- Jumbo frame support (10 kilobyte)
- Dual MII Management interface for PHY control

Extensive per-port statistics provide full support of:

- RMON 1 statistics group (RFC 2819)
- SNMP Ethernet MIB (IEEE Std 802.3-2002 Annex 3st0A counters)
- SNMP Interfaces group MIB (RFC 1213 and 1573)
- SNMP Ethernet-like group MIB (RFC 1643)

Large FIFO buffers provide shared on-chip memory.

- Total ingress buffer size is 4032 kilobits, corresponding to 42 kilobyte average per port.
- Total egress buffer size is 1152 kilobits, corresponding to 12 kilobyte average per port.
- Buffer size is configurable per port; 12 ports share one memory module per direction.
- Store-and-forward or cut-through operation.
- Advanced frame generation, replay, and capture features for test or debug purposes.
- Cut-through threshold for read-out control.
- High and low watermark used for entering and exiting flow control state.
- Priority threshold for protection of Layer 2 and Layer 3 control frames during congestion (ingress).
- Frame drop filter using Layer 2, Layer 3, and Layer 4 fields (ingress).

Policing and shaping functions are available through the FIFOs on a per-port basis.

- Policing of ingress data traffic
- Shaping of egress data traffic
- Common policer for aggregate ingress traffic control

The SPI-4.2 host interface provides connectivity to a network processor.

- Compliant with OIF System Packet Interface Level 4, Phase 2 Implementation Agreement (OIF-SPI4-02.0)
- Ingress and egress data frequency range up to 407 MHz DDR
- Up to 12.8 Gbps raw data bandwidth in each direction
- Dynamic deskew
- Ingress full-stop flow control channel

The Schaumburg can be configured and controlled through the serial and parallel CPU interfaces.

- Serial, four-wire interface, operating at up to 24 MHz
- Parallel, 16-bit, 37-wire interface, asynchronous
- Intel microprocessor or Motorola/IBM Power PC mode
- Separate or shared Address/Data bus mode
- Configuration, status, and statistics register access
- MII Management reads and writes for PHY control
- GPIO reads and writes for miscellaneous external control and monitoring

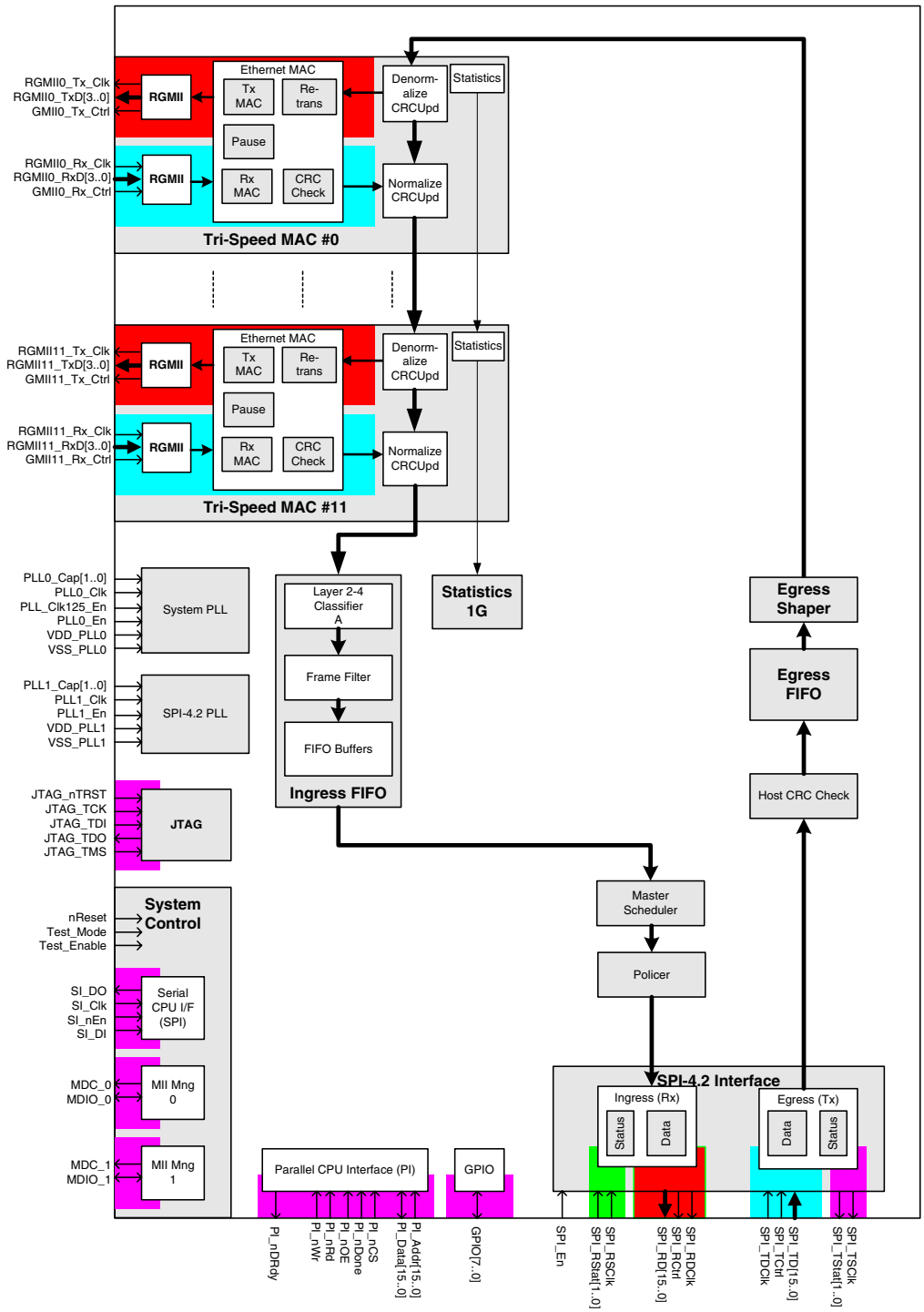


Figure 1. Detailed Block Diagram

A typical application for the Schaumburg is a 12-port Ethernet line card for a modular switch or router, as shown in Figure 2. The RGMII ports on the Schaumburg connect to copper PHYs or SerDes and laser drivers. The SPI-4.2 interface connects to a host processor, typically an ASIC or a network processor (NP), which, in turn, has a backplane interface to, for example, a switch fabric. The line card is controlled by a CPU residing on-board or by a management module connected directly to the Schaumburg and the NP.

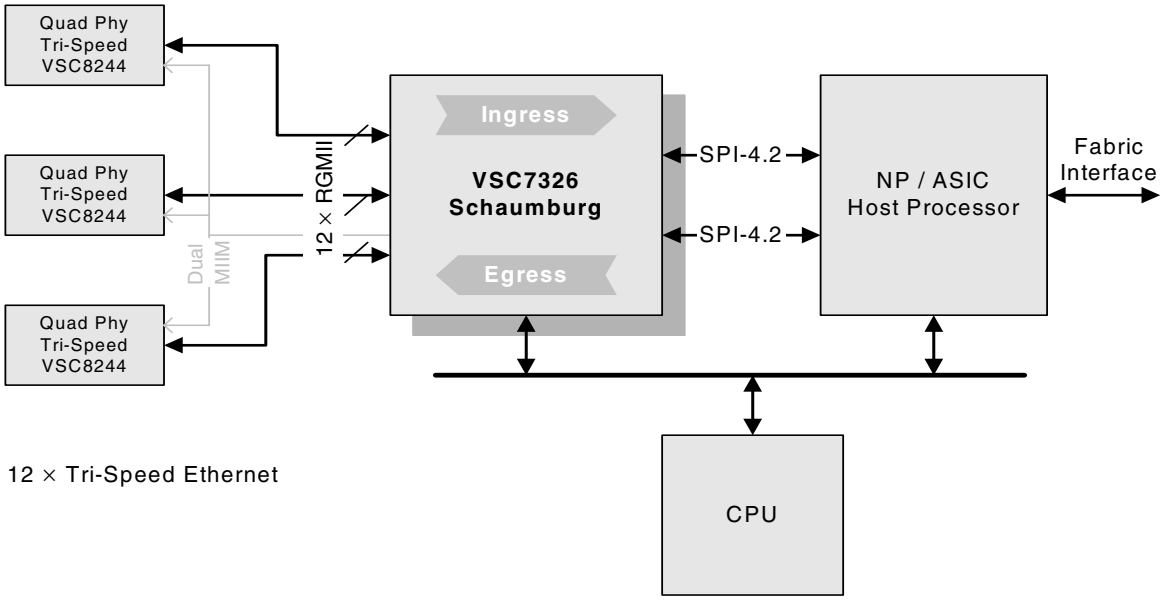


Figure 2. Application Example – 12-Port Tri-Speed Ethernet Line Card

The following sections describe the functions of the Schaumburg as they relate to configuration parameters. For information about individual parameters, see "Configuration Parameter Values," page 65.

1.2 Tri-Speed Ethernet Media Access Controller (MAC)

The Schaumburg device features 12 independent tri-speed (10/100/1000 Mbps) Ethernet MACs compliant with IEEE Std 802.3-2002. The MAC supports the following features and modes:

- Full-duplex mode at 10/100/1000 Mbps
- Half-duplex mode at 10/100 Mbps
- Flow control by pause frames in full-duplex mode, symmetric and asymmetric
- Flow control by collisions in half-duplex mode
- Jumbo frame support (10 kB frames)
- Filtering of fragments (ingress)
- Filtering of oversized frames (ingress)
- Padding of undersized frames (egress)
- CRC calculation and error detection
- Control frame identification

1.2.1 Full-Duplex Operation

In the full-duplex operation, the ingress and egress data streams are fully independent.

In the egress (transmit) direction, the MAC performs standard Ethernet framing. The MAC adds the preamble to the frames and inserts idle symbols to form at least the minimum nominal inter-frame gap (IFG) between the frames.

In the ingress (receive) direction, the preamble, frame delimiters, and idle symbols are stripped before the frame is stored in the FIFO.

The ingress and egress directions can be independently enabled and disabled by configuring the mac_enable parameter. The port speed and duplex mode are selected by configuring the mac_mode parameter.

1.2.2 Half-Duplex Operation

Half-duplex operation is applicable to 10 Mbps and 100 Mbps operation only.

In half-duplex (HDX) mode, the MAC performs the same framing as in full-duplex mode. In addition, the MAC detects collisions on the shared media and backs off for a certain period of time, followed by a re-transmission attempt. As collisions are a natural phenomenon in HDX mode, fragments occur frequently in the ingress direction as part of normal operation, but they are silently discarded by the MAC.

The port speed and duplex mode are selected by configuring the mac_mode parameter.

1.2.3 Ethernet Flow Control

In full-duplex mode, the MAC can flow control the ingress data stream by issuing pause frames, and flow control the egress data stream by reacting to received pause frames. Pause frame flow control can be enabled independently in ingress and egress directions, allowing asymmetric as well as symmetric operation by using the mac_flow_control parameter.

Enabling pause frame flow control in the Rx direction tells the MAC to react on received pause frames; that is, to stop transmitting frames for the time specified in the received pause frame. Reception of a pause frame with the value zero tells the MAC to end pausing and re-engage in frame transmission. Thus, Rx flow control controls the egress data traffic.

Enabling pause frame flow control in the Tx direction allows the MAC to generate pause frames when needed in order to force the peer to pause frame transmission. The pause frames are issued with a pause value configured with the mac_pause_time parameter. The pause frames are issued when the flow control state is entered, (when the HIGH_WM is crossed) and optionally periodically when in the flow control state. Also optionally, a pause frame with the pause value of zero can be issued when exiting the flow control state. The pause value is set by using the mac_pause_time parameter. Thus, Tx flow control controls the ingress data traffic.

In half-duplex mode (10/100 Mbps only), flow control is carried out by colliding ingress frames, thus forcing other hosts on the Ethernet segment to back off and re-transmit. Backpressure is applied as long as the MAC is in flow control state.

In either mode, the flow control state may be determined from the FIFO fill level using the configured watermarks, or by signalling from the host processor over the SPI-4.2 status channel. For more information, see "Flow Control," page 86.

1.2.4 Ingress Frame Filtering

The following describes the frame filtering capabilities in the Schaumburg device. Frame filtering applies to the ingress traffic only; that is, traffic from RGMII to SPI-4.2. The purpose is to reduce traffic on the SPI-4.2 interface by dropping frames.All the dropped frames are counted with a single per-port drop counter, which also counts frames dropped due to FIFO buffer overflow. If the FIFO buffer is configured for cut-through operation, and a frame transmission has been initiated on the SPI-4.2 interface, the frame is not dropped or counted, but instead, is transmitted with an EOP/Abort.

The Schaumburg also contains a protection feature for Layer 2 and 3 control frames. It can mark frames as protected, thereby giving them lower drop precedence. The frame filter and the frame protection mechanism do not cooperate. If both features are enabled, the frame filter drops frames regardless of whether they are marked as protected. For more information about the Layer 2 and 3 control frames, see "Control Frame Protection," page 41.

The frame filtering capabilities are:

- DMAC Value Filter—Drop non-matching frames
- DMAC Mask Filter—Drop non-matching frames
- DMAC Broadcast Filter—Drop matching frames
- EtherType Filter—Drop matching frames
- Pause Control Frames—Drop matching frames
- Other Control Frames—Drop matching frames
- Bad Ethernet Frames—Drop matching frames

1.2.4.1 DMAC Value Filter

The DMAC value filter is composed of two identical filters, making it possible to match two different DMAC values. Each filter is composed of a 43-bit global address , and a 5-bit port-specific address (least significant bits of DMAC address). The filters can be programmed to match either unicast or multicast addresses by setting the I/G bit (bit 40 of the 48-bit DMAC field) to either 0 (for unicast) or 1 (for multicast). Both DMAC value filters must use the same value for the I/G bit.

If the DMAC value filter is enabled, frames are dropped if they belong to the same group (unicast, multicast) as the filter and do not match one of the two filter addresses. The enabling or disabling of DMAC value filtering is configurable on a per-port basis.

Typical use for this filter is to match the port unicast DMAC address, suppressing all other unicast frames.

The DMAC value filter is controlled by registers ING_FFILT_UM_EN, ING_FFILT_VAL0, and ING_FFILT_VAL1. For more information about these registers, see Table 91, page 138, Table 93, page 140, and Table 94, page 140.

1.2.4.2 DMAC Mask Filter

The DMAC mask filter is composed of a 48-bit port-specific address and a 48-bit port specific mask. The filter can be programmed to match either unicast or multicast or both of these types of addresses. Clearing the I/G bit (bit 40 of the 48-bit mask) in the mask causes the filter to match both unicast and multicast DMAC addresses. Setting the I/G mask bit to 1 and the I/G bit (bit 40 of the 48-bit DMAC field) to either 0 (for unicast) or 1 (for multicast) causes the filter to match either unicast or multicast DMAC addresses.

If the DMAC mask filter is enabled, frames are dropped if they belong to the same group (unicast, multicast, both) as the address plus mask, and do not match the DMAC mask address at the masked bit positions. The enabling or disabling of DMAC mask frame filtering is configurable on a per-port basis.

Typical use for this filter is to enable it for a range of multicast DMAC addresses expected on its port.

The DMAC mask filter is controlled by registers ING_FFILT_UM_EN, ING_FFILT_MASK0, ING_FFILT_MASK1, and ING_FFILT_MASK2. For more information about these registers, see Table 91, page 138, Table 95, page 140, Table 96, page 141, and Table 97, page 141.

1.2.4.3 DMAC Broadcast Filter

Broadcast frames are defined as frames with DMAC equal to all-ones. If the broadcast filter is enabled, all broadcast frames are dropped.

The DMAC broadcast filter is controlled by registers ING_FFILT_UM_EN and ING_FFILT_BE_EN. For more information about the registers, see Table 91, page 138 and Table 92, page 139.

1.2.4.4 EtherType Filter

The EtherType filter is composed of two 16-bit global registers. If the EtherType frame filter is enabled, all frames where the Type/Length field match one of the two 16-bit register values, are dropped. Having two 16-bit global values makes it possible to filter two EtherType frames.

Typical use is to drop frames with Type/Length value = 0x8100 (VLAN tagged frames).

The EtherType filter is controlled by registers ING_FFILT_UM_EN, ING_FFILT_BE_EN, and ING_FFILT_ETYPE. For more information about the registers, see Table 91, page 138, Table 92, page 139, and Table 98, page 141.

1.2.4.5 Pause Control Frames

Pause control frames are defined as frames with Type/Length field = 0x8808, DMAC = reserved multicast address or the port MAC address, and Opcode (first two payload bytes) = 0x0001.

If the pause control frame filter is enabled (Default), all pause control frames are dropped.

The pause control filter is controlled by register field NORMALIZER::DROP_PAUSE. For more information about this register, see Table 71, page 122.

1.2.4.6 Other Control Frames

Other control frames are defined as frames with Type/Length field = 0x8808, DMAC = reserved multicast address or the port MAC address, and Opcode (first two payload bytes) not equal to 0x0001.

If the other control frame filter is enabled, all control frames not being pause control frames are dropped.

The other control frame filter is controlled by register field NORMALIZER::DROP_CTRL. For more information about this register, see Table 71, page 122.

1.2.4.7 Bad Ethernet Frames

Bad Ethernet frames are defined as frames with:

- Wrong CRC
- Length less than 64 bytes
- Length greater than maximum programmed allowed length

If bad Ethernet frame filtering is enabled (Default), all these frames are dropped.

The bad Ethernet frame filter is controlled by register fields DEV_DEBUG::KEEP_BAD, MAXLEN_CFG::MAX_LEN, and NORMALIZER::NO_CRC. For more information about these registers, see Table 75, page 127, Table 68, page 121, and Table 71, page 122.

1.2.5 Ingress CRC Error Detection

The MAC checks the CRC in the frame check sequence (FCS) field of ingress frames and discards frames with detected errors. In cut-through mode, however, the frame may be partially transferred over the SPI-4.2 interface before being terminated with an EOP Abort indication. CRC checking can be disabled by using the ingress_crc_mode parameter.

1.2.6 Egress CRC Generation

The MAC is capable of calculating and inserting a correct CRC in the FCS field of egress frames. The CRC can be added to or updated depending on the configuration of the egress_crc_mode parameter. This is shown in the upper half of Figure 3. CRC Update can be used when frames could potentially be modified by a host chip, but do not have the resources to correct the CRC. When either mode is enabled, egress frames that are less than 64 bytes are padded with 0x00 bytes to achieve the minimum allowed Ethernet length of 64 bytes as shown in the lower half of Figure 3.

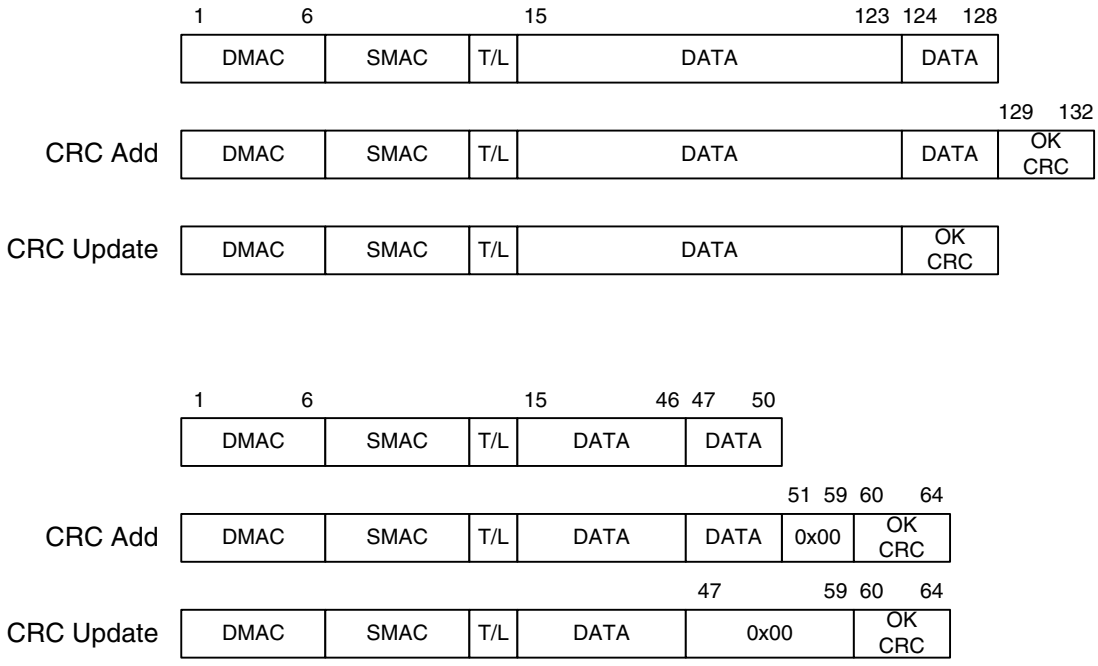


Figure 3. Result of CRC Add and CRC Update on a 128-Byte Frame and a 50-Byte Frame

1.3 FIFO

The FIFO module contains two separate memory instances; one for ingress traffic and one for egress traffic. Each memory instance is sub-divided into consecutive buffer areas, one per physical Ethernet port, allocated by configuration.

Each port buffer functions as a cyclic FIFO buffer with a number of configurable thresholds to control the input and output data flow.

If the FIFO is nearly full, it discards all frames that do not fit into the free space left in the FIFO. This means that small frames (more likely to fit in the free space before running full) are prioritized higher than large frames when the FIFO is nearly full. The number of frames dropped is counted in a separate counter for each FIFO

1.3.1 Buffer Allocation

The Schaumburg device provides more than 5 megabit of internal memory shared by 12 ports.

- Ingress memory: 4032 kilobit total, 42 kilobyte average per port
- Egress memory: 1152 kilobit total, 12 kilobyte average per port

The buffer memories are allocated to the individual ports by configuring the fifo_size parameter in blocks of 2048 byte. The individual FIFOs are internally defined by the EGR_TOP_BOTTOM::TOP and EGR_TOP_BOTTOM::BOTTOM values for each port. (For more information about this register, see Table 115, page 154.) The buffer occupies a contiguous piece of memory starting with block number BOTTOM and ending with block number TOP-1. The following figure illustrates the partitioning of a buffer memory instance.

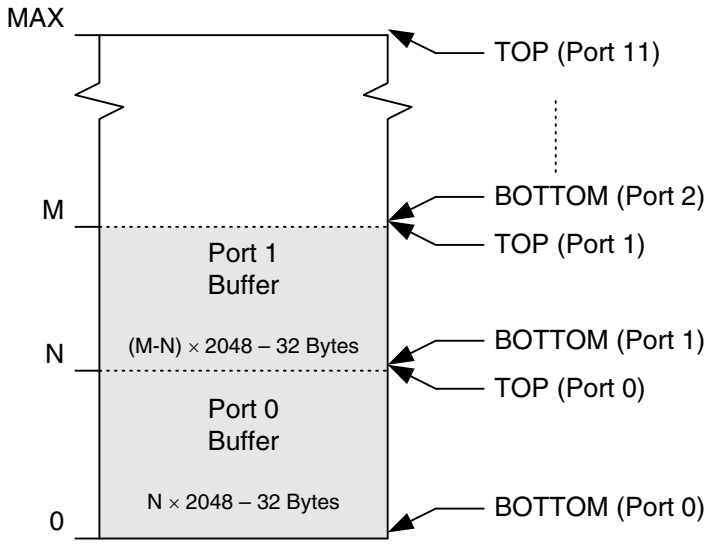


Figure 4. FIFO Memory Allocation

The available buffer size for a given port is calculated as (TOP – BOTTOM) × 2048 – 128 bytes.

1.3.2 Buffer Thresholds

Each port buffer has a number of configurable thresholds controlling the input and output data flow of the FIFO. Table 1 describes the configurable buffer thresholds.

Table 1. Configurable Buffer Thresholds

Threshold Name	Description	Fill Level
CT_THRHLD	Cut-through	The fill level at which the FIFO buffer holds enough data to start reading out a frame, even if the end-of-frame has not yet been stored.
LOW_WM	Low watermark	The fill level at which the FIFO exits the flow control state.
HIGH_WM	High watermark	The fill level at which the FIFO enters the flow control state.
CLASS_THRHLD	Control frame protection	The fill level at which the FIFO discards normal data frames to reserve space for protecting control frames.
CLASS_THRHLD	Frame size fair dropping	The fill level above which new frames may not be stored. The storing of a single frame is allowed to pass this threshold, to make frame size fair dropping possible.

The buffer thresholds are counted in words of 32 bytes, relative to the start of the cyclic FIFO. In other words, the thresholds are counted from the current read pointer, not the fixed bottom address.

1.3.3 Cut-Through and Store-and-Forward Mode

The store-and-forward and cut-through modes are two classical MAC operating concepts. In the cut-through mode, frame forwarding may begin before the complete frame has been stored. Store-and-forward mode permits the erroneous frames to be discarded, but at the cost of increased latency and jitter.

The store-and-forward mode is selected by using the fifo_read_mode parameter. In this mode, data is scheduled out of the FIFO only if an end-of-frame indication is present in the FIFO.

The cut-through mode is also configured by using the fifo_read_mode parameter. When either the FIFO fill level reaches the cut-through threshold, or a complete frame is stored in the FIFO, data forwarding is initiated and continues until the end-of-frame threshold is reached. Even when the cut-through mode is enabled, frames smaller than the cut-through threshold experience a store-and-forward operation.

Normally, the egress FIFO should be operated in the store-and-forward mode to avoid problems with FIFO underrun if the SPI-4.2 Host is not sending consistently enough to sustain 1 GbE.

1.3.4 High and Low Watermarks

Each per-port FIFO uses both a High Watermark (HIGH_WM) and a Low Watermark (LOW_WM) for determining the flow control state. For an Ethernet application, flow control is used in ingress direction for pause frame generation on the Ethernet ports. In the egress direction, flow control is used to backpressure on the SPI-4.2 interface.

The FIFO buffer asserts flow control when the fill level exceeds the HIGH_WM and deasserts flow control when the fill level falls below the LOW_WM. This mechanism reduces the risk of asserting too many pause frames on the Ethernet ports when the fill level fluctuates around the watermarks. The HIGH_WM must be greater than or equal to the LOW_WM.

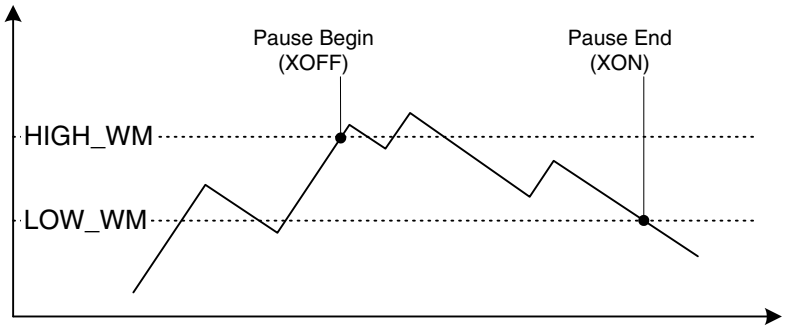


Figure 5. FIFO Fill Levels and Watermarks

The watermark settings determine flow control performance. To obtain the best performance, the following rules must be observed:

- The HIGH_WM must be set high enough to avoid activating flow control under normal uncongested operating conditions.
- The HIGH_WM must be set low enough to avoid buffer overflow after activating flow control.
- The LOW_WM must be set high enough to avoid buffer underflow after de-activating flow control.
- The LOW_WM and HIGH_WM must be set with adequate distance to avoid excess flow control signalling due to normal fluctuations around the watermarks.

The actual watermark settings depend on several parameters including frame size and link speed. A set of recommended values for HIGH_WM and LOW_WM are listed in the following table. For more information about how flow control functions, see "Flow Control," page 86. For information about how to calculate watermark settings, see "Watermark Calculations," page 106.

Table 2. Recommended HIGH_WM and LOW_WM Settings

	Burst-Interleaved		Frame-Interleaved		
MTU = Maximum frame size	10000	1522	10000	1522	4000
Maximum cable length	2 km	2 km	2 km	2 km	2 km
Maximum number of ports	12	12	6	12	12
Ingress					
HIGH_WM	0x279	0x279	0x6CF	0x279	0x279
LOW_WM	0x259	0x259	0x2C0	0x259	0x259
Egress					
HIGH_WM	0x142	0x142	0x171	0x06C	0x0BA
LOW_WM	0x142	0x142	0x171	0x06C	0x0BA

1.3.5 Control Frame Protection

The ingress FIFO provides a mechanism for protecting incoming control frames during congestion, which is enabled with ING_TEST::CLASS_EN. This mechanism reserves memory space from the CLASS_THRHLD threshold to the top of the FIFO. When the fill level rises above the CLASS_THRHLD threshold, the FIFO starts discarding normal frames and stores only protected frames, as illustrated in the following figure. This effectively reduces the FIFO size for normal frames by the amount of protected space specified by the fifo_protection_size parameter. For more information about this parameter, see Table 26, page 69.

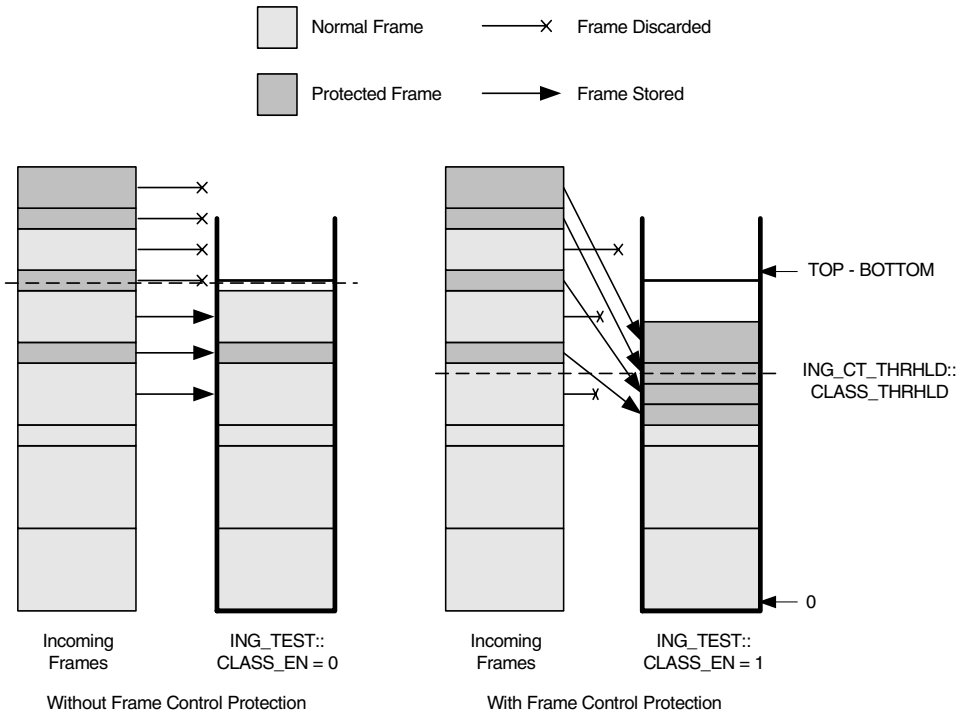


Figure 6. Effect of Control Frame Protection

The control frames that are to be protected are identified by a classifier, which is able to pick out frames associated with common Layer-2 and Layer-3 control protocols.

Layer-2 control frames are identified by having DMAC = 01-80-C2-00-00-XX or Type/Len = 0x8808; where XX can be matched to a number of defined groups, selected by configuring the protection_classes parameter. This includes Layer-2 protocols such as STP, MSTP, LACP and all GARP protocols including GMRP and GVRP.

Examples of Layer-3 control frames that can be identified and protected are shown in the following table. These are all IPv4 frames that have Type/Length 0x0800, possibly behind a tag. The IPv4 multicast frames use a DMAC in the 23-bit range 01-00-5E-00-00-00 to 01-00-5E-7F-FF-FF. The notation for this range is 01-00-5E-XX-XX-XX in the following table. A router has a router interface on each port. The MAC address of the router is called “Router-MAC” in the following table, and the IP addresses on such an interface are called “Router-IP”.

Table 3. Protectable Layer-3 Control Frames

Frame Type	DMAC	Type/ Length	IPv4 Protocol	IPv4 Destination Address	UDP/TCP Source or Destination Port
RIPv1	FF-FF-FF-FF-FF-FF FF-FF-FF-FF-FF-FF Router-MAC	0x0800	17 (UDP)	255.255.255.255 Net-Broadcast Router-IP	520
RIPv2	01-00-5E-00-00-09 Router-MAC	0x0800	17 (UDP)	224.0.0.9 Router-IP	520
OSPFv2	01-00-5E-00-00-05 01-00-5E-00-00-06 Router-MAC	0x0800	89 (OSPF)	224.0.0.5 224.0.0.6 Router-IP	
ISIS	01-80-C2-00-00-14 01-80-C2-00-00-15	Length			
BGPv4	Router-MAC	0x0800	6 (TCP)	Router-IP	179
VRRP	01-00-5E-00-00-12	0x0800	112	224.0.0.18	
IGMP	01-00-5E-00-00-01 01-00-5E-00-00-02 01-00-5E-XX-XX-XX	0x0800	2 (IGMP)	224.0.0.1 224.0.0.2 224.0.1.0 - 239.255.255.255	
DVMRP	01-00-5E-00-00-04 Router-MAC	0x0800	2 (IGMP)	224.0.0.4 Router-IP	
PIM	01-00-5E-00-00-0D	0x0800	103 (PIM)	224.0.0.13	
CBT	01-00-5E-00-00-0F	0x0800	7 (CBT)	224.0.0.15	
ICMP		0x0800	1 (ICMP)		

The Layer-3 control frames are identified by looking at the Layer 2 through 4 fields shown in Table 4.

Table 4. Frame Fields for Layer-3 Classification

Layer 2/3/4 Field	Size (Bits)	Description
mac_type_length1	16	The first Type/Length field in the MAC header
mac_type_length2	16	The second Type/Length field in the MAC header after the VLAN tag
ipv4_version	4	The version number in the IPv4 header
ipv4_protocol	8	The protocol field in the IPv4 header
ipv4_fragment_offset	13	The fragment offset field in the IPv4 header
ipv4_dip	32	The destination address in the IPv4 header
udp_tcp_dport	16	The destination port in the UDP/TCP header
udp_tcp_sport	16	The source port in the UDP/TCP header

Some of the frame fields are configurable while others are hard-coded. The configurable frame fields (per chip, not per port) are shown in Table 5.

Table 5. Configuration Options for the Schaumburg Device

Register	Size (Bits)	Description	Typical
ING_CL3_CTRL::IPV4_PROTO_EN	1	IPv4 protocol enable or disable	1
ING_CL3_L3::IPV4_PROT	8	IPv4 protocol value	89
ING_CL3_CTRL::IPV4_MC_EN	1	IPv4 multicast control enable or disable (224.0.0.x)	1
ING_CL3_CTRL::UDP_TCP_EN	1	UDP/TCP destination port enable or disable	1
ING_CL3_L4::PORT0	16	First UDP/TCP port value	520
ING_CL3_L4::PORT1	16	Second UDP/TCP port value	179

The following pseudo code shows the classification logic for Layer-3 control frames.

```
# By default, control frame protection is not done.
control_protect = 0;

# IPv4 frame classification, single VLAN tag allowed.
ipv4_frame = ((frame.mac_type_length1 == 0x0800 OR
               (frame.mac_type_length1 == 0x8100 AND
                frame.mac_type_length2 == 0x0800)) AND
              frame.ipv4_version == 4);

# IPv4 protocol filter
if (ipv4_frame AND
    chip.ipv4_proto_en AND
    (frame.ipv4_protocol == chip.ipv4_prot))
{
control_protect = 1;
}

# IPv4 multicast control filter (DIP 224.0.0.x)
if (ipv4_frame AND
      frame.ipv4_fragment_offset == 0 AND
      chip.ipv4_mc_en AND
    (frame.ipv4_dip & 0xFFFFFF00) == 0xE0000000)
{
control_protect = 1;
}

# UDP/TCP port filter. The port number is only valid
# if it is not a fragment and the protocol is TCP (6) or UDP (17).
if (ipv4_frame AND
    frame.ipv4_fragment_offset == 0 AND
    (frame.ipv4_protocol == 6 OR frame.ipv4_protocol == 17) AND
    chip.udp_tcp_enable AND
    (frame.udp_tcp_sport == chip.port0 OR
     frame.udp_tcp_sport == chip.port1 OR
     frame.udp_tcp_dport == chip.port0 OR
     frame.udp_tcp_dport == chip.port1))
{
control_protect = 1;
}
```

The classification result is the control_protect boolean, which indicates whether the frame is protected.

Layer-3 ISIS frames, which use a DMAC = 0x0180C2000014 or 0x0180C2000015, can be identified as a Layer-2 control frame.

1.3.6 Frame Size Fair Dropping

Generally, FIFO overflow is prevented by correctly configuring the watermarks. However, in applications where lossless flow control is not enabled, a FIFO overflows when the filling reaches the buffer size of TOP-BOTTOM. When this occurs, a frame that cannot be fully stored is discarded, so that no partial frames are forwarded through the FIFO. The memory previously occupied by the discarded frame is immediately available for new data.

If FIFO overflow occurs frequently, the overflow dropping and memory freeing procedure causes preferential dropping of large frames. If a port is heavily oversubscribed with traffic consisting of random size frames, all large frames are dropped and only small frames pass through. This is because the remaining free memory space, which is bombarded by different frame sizes, can easily be filled with small frames, but discards large frames. The left half of Figure 7 illustrates how large frames are dropped while small frames are passed through.

All the above is equally true when the classifier is enabled, the overflow happening at an earlier stage when the FIFO filling reaches CLASS_THRHLD, instead of the buffer size of TOP-BOTTOM.

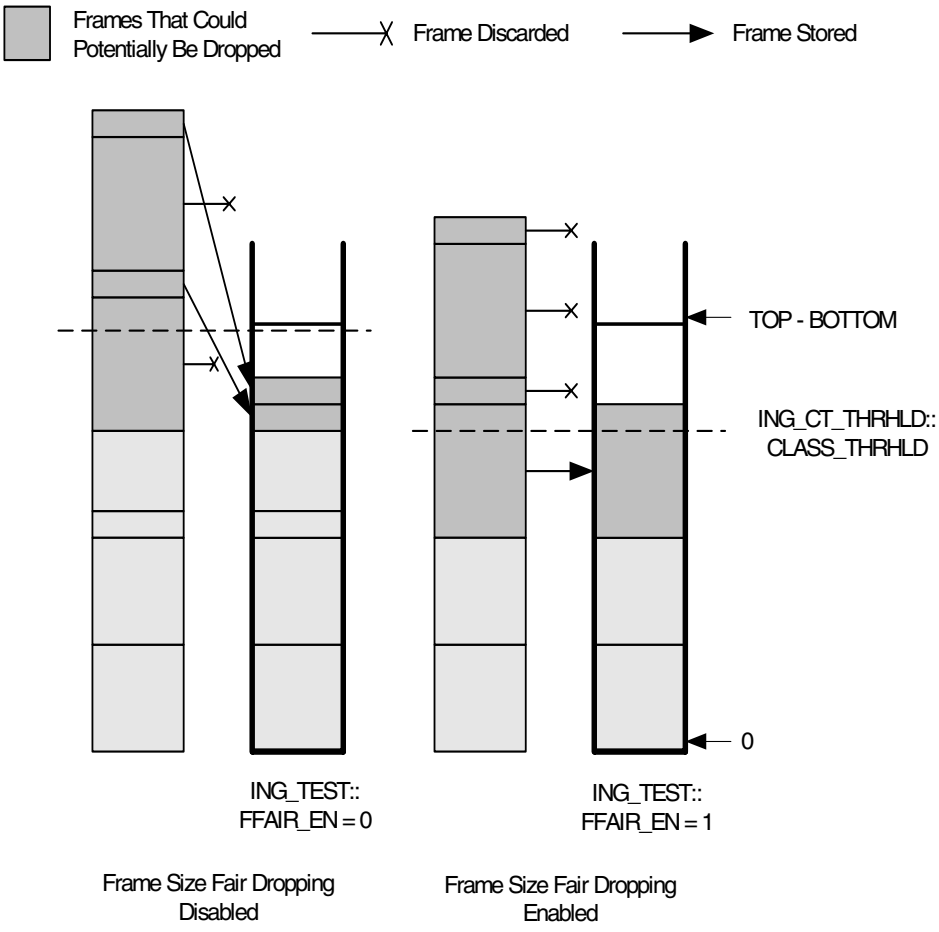


Figure 7. Frame Dropping Behavior

The frame size unfair dropping can be avoided by enabling frame size fair dropping, using ING_TEST::FFAIR_EN. The threshold normally used for the classifier, ING_CT_THRHLD::CLASS_THRHLD, is reused. Frames that start being stored below CLASS_THRHLD, are allowed to pass it, as illustrated on the right side of Figure 7. Therefore, there must be room for one maximum frame above CLASS_THRHLD, reducing the available FIFO memory. Dropping is frame size fair, as any size is still allowed to be received when the FIFO is close to being full, that is, close to CLASS_THRHLD.

If frame size fair dropping and frame classification is desired at the same time, there should be additional headroom above CLASS_THRHLD for the protection of classified frames.

1.3.7 Underflow

Generally, FIFO underflow should be prevented by correctly configuring the ING_CT_THRHLD::CT_THRHLD buffer threshold. In situations where the configuration is incorrect or the SPI-4.2 malfunctions, an egress FIFO may run empty while it is reading out a frame. This results in the current frame being aborted with an invalid CRC, and subsequent data is discarded up to the next valid start-of-frame indication.

1.4 Traffic Shaping and Policing

A set of “leaky bucket” configurations can be set to control the bandwidth out of the FIFO buffers in each direction. Each bucket is set to control the maximum bandwidth allowed for each FIFO buffer. In addition, there is a common bucket that controls the total bandwidth from all FIFO buffers for each directions.

The policer is positioned in the ingress direction and limits the received traffic bandwidth allowed for each port. It is used to make a fair allocation of the available SPI-4.2 bandwidth among the active ports. The policer buckets are located at the output of the ingress FIFO.

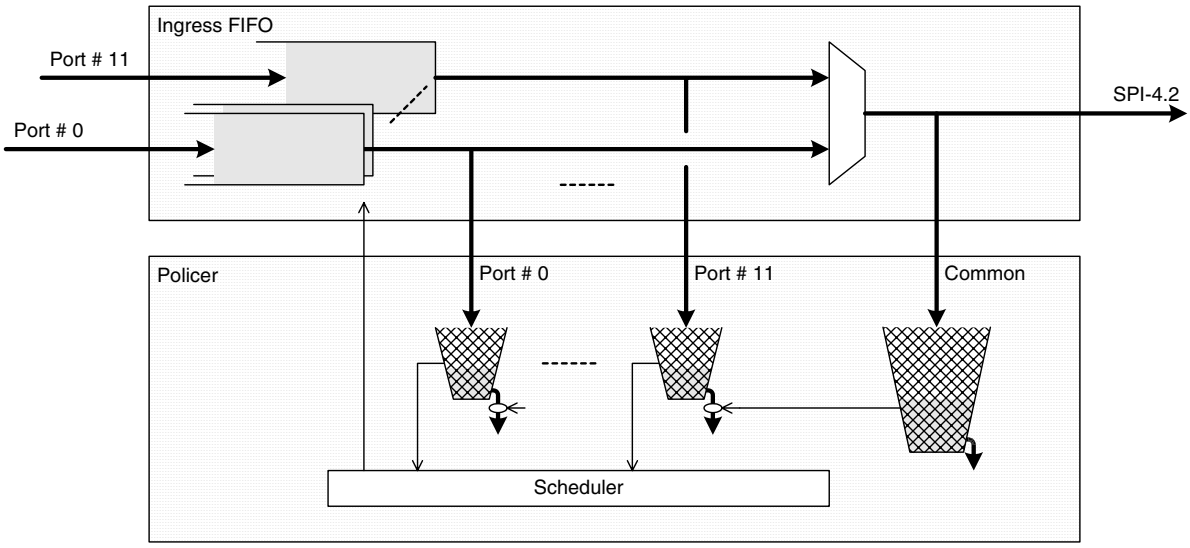


Figure 8. Ingress Traffic Policer

The shaper is positioned in the egress direction and limits the transmission bandwidth on each port. This is often done to limit congestion in downstream equipment. The shaper buckets are located at the output of the egress FIFO, where they directly limit the average rate and burstiness of the transmitted traffic on each port. They do this by adjusting the distance between the frames according to the desired bandwidth.

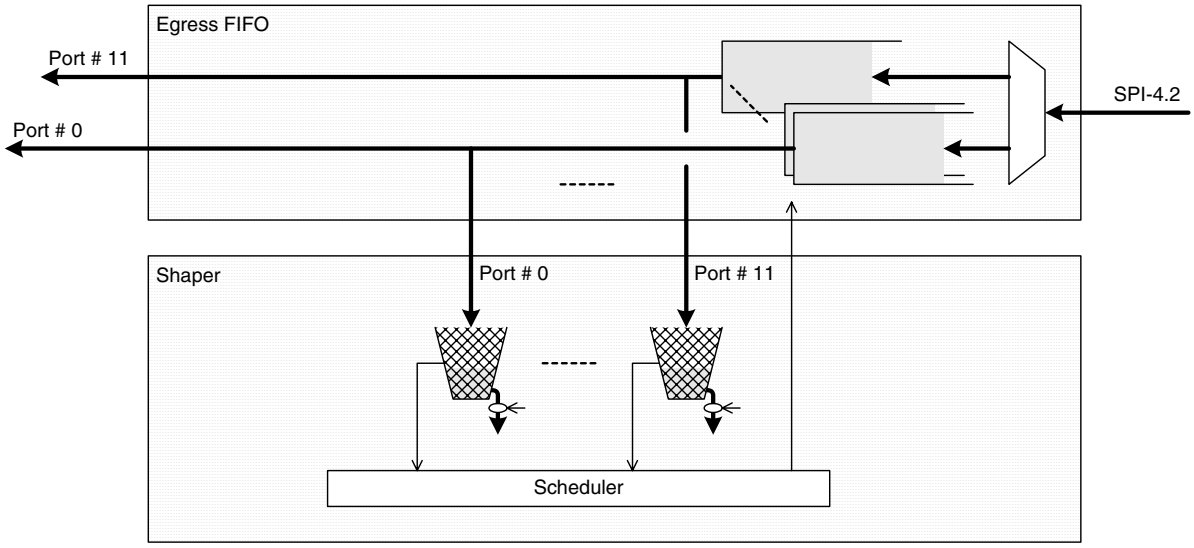


Figure 9. Egress Traffic Shaper

The actual function of the rate control is identical for the two directions.

1.4.1 Leaky Buckets

Each bucket is filled with the amount of traffic going out of the FIFO buffer, and is drained at a constant rate between 0 and 10 Gbps, called bucket rate. The bucket rate (BUCKET_RATE) is configured separately in increments of 150 kbps, together with a bucket level (BUCKET_LEVEL). Traffic out of the FIFO buffer may be scheduled as long as the bucket fill level is below the BUCKET_LEVEL. The BUCKET_RATE controls how large a bandwidth the bucket allows to pass through. The BUCKET_LEVEL controls how bursty traffic is allowed to be, because an empty bucket allows unlimited traffic to flow until the BUCKET_LEVEL is reached.

The common bucket is filled at the rate of the total traffic from the individual FIFO buffers. The common bucket is drained with a bucket rate between 0 and 20 Gbps. The bucket rate is configured separately in increments of 300 kbps. When the bucket level of the common bucket is above its BUCKET_LEVEL, the bucket rate of the individual buckets is reduced proportionally to the configured rates. This mechanism effectively reduces the bucket rates of the individual buckets proportionally to the missing bandwidth.

The BUCKET_RATE of the common bucket is also limited by the actual available bandwidth. In other words, the common ingress policer is limited by the SPI-4.2 interface bandwidth.

The leaky buckets are configured by using the shaper_bucket_rate and shaper_bucket_level parameters. For more information about BUCKET _LEVEL and BUCKET_RATE, see the Shaper Bucket register, Table 141, page 168 and the Common Shaper Bucket register, Table 142, page 168.

1.4.2 Ingress Bandwidth Fairness

In normal applications, the SPI-4.2 interface is running at a sufficiently high frequency, such as 400 MHz with 12 active ports, so that it is not oversubscribed. The ingress policer can be disabled because there is no fairness problem. However, in situations where the clock frequency is so low as to make the SPI-4.2 interface oversubscribed, bandwidth is distributed unfairly among the ports.This is because the ingress scheduler uses fixed-sized 32-byte bursts, also when the frame ends, even if this last burst only contains 1 valid byte. Thus, with the policer disabled, a port receiving only 65-byte frames receives twice the bandwidth of a port receiving only 64-byte frames. Bandwidth unfairness is aggravated when running in full-frame mode, where a full frame from each port is scheduled in round-robin fashion by the scheduler. Thus, a port with 1518-byte frames gets 96% of the available bandwidth if competing with a port receiving only 64-byte frames (where 96% = 1518 / (1518 + 64)).

With the ingress policer, this problem can be circumvented, and a fair distribution of the SPI-4.2 interface bandwidth among the active ports can be made. In burst-interleaved mode, this is achieved by setting the common shaper bandwidth just above the SPI-4.2 bandwidth.

In full-frame mode, such a setting still gives slight unfairness for frame mixtures containing frames that are less than 80 byte. To get full fairness for these situations, you must set the common mode shaper bandwidth to a reduced value, as shown in the table below. A consequence of this is that the effective SPI-4.2 bandwidth is reduced for all frame sizes, even though the interface supports up to 12.8 Gbps. In full-frame mode, you must therefore choose whether full fairness or maximum SPI-4.2 bandwidth is required.

Table 6. Ingress Policer Settings for Achieving Bandwidth Fairness

Register	Burst-Interleaved	Full-Frame				
	300 - 407[1]	300[1]	325[1]	350[1]	375[1]	400[1]
SHAPER_CONTROL:: SPI4_ADJUST	1	0	0	0	0	0
SHAPER_CONTROL:: FRM_LEN_ADJ	0x2	0x2	0x2	0x2	0x2	0x2
SHAPER_SLOW_RATE _CONTROL:: SLOW_DOWN_EN	1	1	1	1	1	1
COMMON_SHAPER_ BUCKET:: BUCKET_RATE	0xAD5E 13.000 Gbps	0x76AA 8.900 Gbps	0x7EAA 9.500 Gbps	0x86AA 10.100 Gbps	0x8D55 10.600 Gbps	0x92A0 10.997 Gbps
COMMON_SHAPER_ BUCKET::BUCKET_ LEVEL	0x2 256 Byte	0x2 256 Byte	0x2 256 Byte	0x2 256 Byte	0x2 256 Byte	0x2 256 Byte
Port shapers	Enabled	Disabled	Disabled	Disabled	Disabled	Disabled

1. SPI-4.2 Frequency (MHz)

1.4.3 Controlling Bandwith Distribution

If the sum of the configured bucket rates does not exceed the available bandwidth, each port is limited to, and guaranteed, its configured bucket rate, and this section does not apply.

If the sum of the configured bucket rates and the sum of the offered traffic bandwidth both exceed the available bandwidth or the allowed bucket rate of the common bucket, the bucket rates on the ports are reduced proportionally, as shown for port A, B, and C in Figure 10. The bucket rates are reduced to the point where the sum of the actual traffic rates is equal to the available bandwidth. This is the reason why the common bucket circle matches exactly the traffic bandwidth in Figure 10, Figure 11, Figure 12 and Figure 13. Ports with offered traffic below the reduced bucket rate are not affected, and ports where the reduced bucket rate exceeds the port speed as shown for port B and C in Figure 11 are not affected. Other ports are limited to the reduced bucket rate.

By carefully selecting the bucket rates, you can control the bandwidth distribution between the different ports. The figures on the following pages illustrate three ways to manage oversubscription by configuring the bucket rates for a 1-Gbps port and two 100-Mbps ports. The common bucket is typically set to the bandwidth of the SPI-4.2 interface. The dashed circle represents the configured bucket rate, and the solid circle is the resulting bucket rate due to adjustments made by the common bucket.

- Proportional share—Buckets are configured to match port speeds, which are 1 Gbps and 100 Mbps, respectively, in this example. With proportional share, each port gets its configured bucket rate reduced with the same proportion. If port C stops its transmission, port A gets 10/11 of 600 Mbps and port B gets 1/11 of 600 Mbps. Proportional share is the closest to an intuitive idea of fairness.

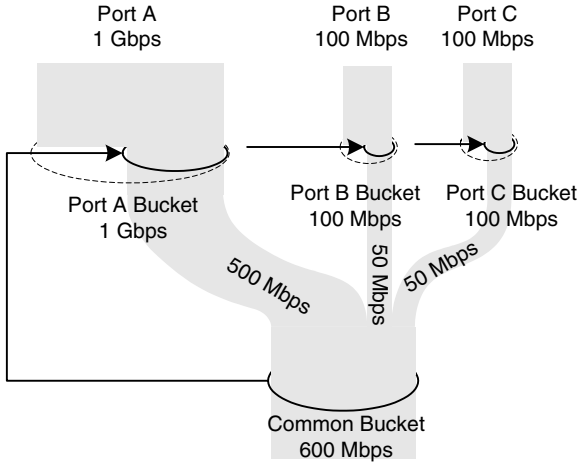


Figure 10. Proportional Bandwidth Sharing

- Equal share—Buckets rates are configured to the same value regardless of port speed. In this exampl,e the bucket rates are 1 Gbps, the rate of the 1G port is reduced to 400 Mbps, and the rate of the 100-Mbps port is not reduced at all. If the common bucket was only 200 Mbps, each port would get 67 Mbps.

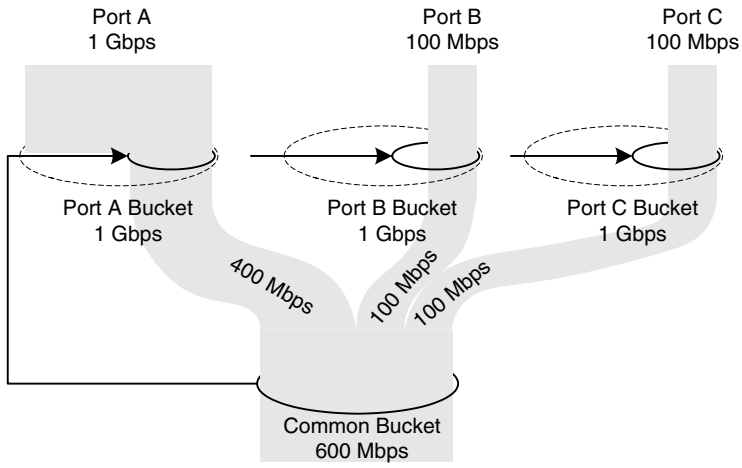


Figure 11. Equal Bandwidth Sharing

- Hybrid share—The configuration is a mixture of proportional share and equal share. In the example shown in Figure 12, two ports are configured to their actual port speeds, and the last port is configured at a rate higher than its limit. Port A gets $10/13$ of the available 600 Mbps, port B gets $2/13$, and port C gets $1/13$ of the bandwidth.

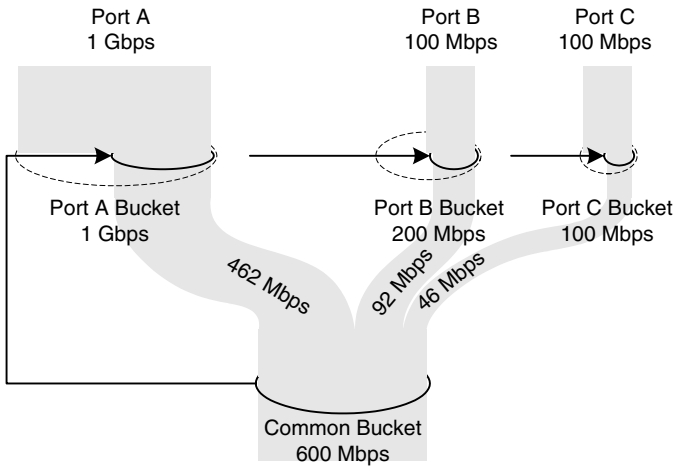


Figure 12. Hybrid Bandwidth Sharing

Configuring a bucket rate to a higher bandwidth than the bandwidth defined for the port protects that port from having its speed reduced. A port with a higher bucket rate configured maintains its higher speed longer than ports configured with lower bucket rates.

In most cases, a port is guaranteed a share of the available bandwidth proportional to the configured rates of all ports.

1.4.4 High-Priority Ingress Buckets

Each of the 12 ingress buckets can be configured for high priority. A high-priority bucket does not have its bandwidth adjusted by the common bucket.

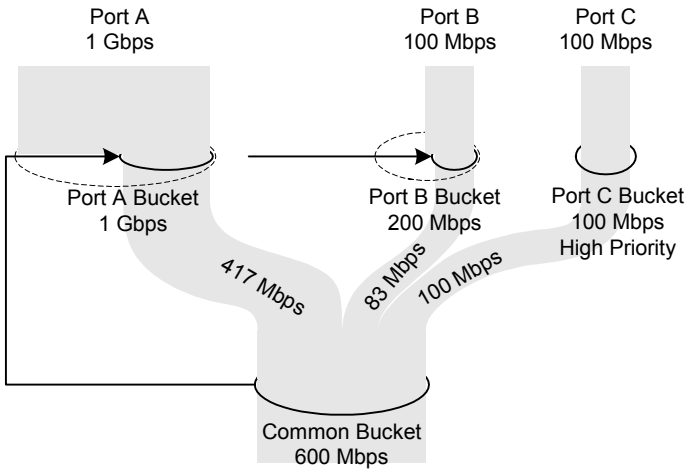


Figure 13. Hybrid Bandwidth Sharing with High-Priority Port

The bucket on port C is the only one of the three ports configured as high priority. The available bandwidth is 600 Mbps, of which port C receives 100 Mbps, unchanged by the common bucket. The low priority ports A and B share the remaining bandwidth. Therefore, port A gets $10/12$ of 500 Mbps, and port B gets $2/12$ of 500 Mbps.

If total bandwidth of the high-priority buckets exceeds the common bucket's bandwidth, the scheduler regulates the data bandwidth, not the buckets. This corresponds to having the buckets disabled and bandwidth distribution between ports not fair.

1.5 Statistics

The Schaumburg MAC includes counters to support the following standards:

- RMON 1 statistics group (RFC 2819)
- Ethernet MIB (IEEE Std 802.3-2002 Annex 30A counters)
- SNMP Interfaces group MIB (RFC 1213 and 1573)
- SNMP Ethernet-like group MIB (RFC 1643)

All statistics counters are 32 bits wide, allowing poll cycles up to 30 seconds.

The following table lists the receive statistics counters available in the Schaumburg device. Table 8, page 54, lists the transmit statistics counters available. For more information about each counter, see "Statistic Counters, Block 4," page 171.

Table 7. Receive Statistics Counters

Counter	Short Name
Bytes received (good, bad, and framing)	RX_IN_BYTES
Dropped frames in the receiver due to ingress FIFO overflow	ING_DROP_CNT[1]
Frames received with one or more symbol errors, excluding collision	RX_SYMBOL_CARRIER
Pause control frames received	RX_PAUSE
Control frames with unsupported opcode received	RX_UNSUP_OPCODE
Received bytes in good frames	RX_OK_BYTES
Received bytes in bad frames	RX_BAD_BYTES
Good unicast frames received	RX_UNICAST
Good multicast frames received	RX_MULTICAST
Good broadcast frames received	RX_BROADCAST
Frames received with CRC error only	RX_CRC
Frames received with alignment error	RX_ALIGNMENT
Undersized, well-formed frames received	RX_UNDERSIZE
Undersized frames with CRC error received	RX_FRAGMENTS
Frames with legal length field that do not match length of MAC client data	RX_IN_RANGE_LENGTH_ERROR
Frames with illegal length field (frames using type field are not counted here)	RX_OUT_OF_RANGE_LENGTH
Oversized, well-formed frames received	RX_OVERSIZE
Oversized frames with CRC error received	RX_JABBERS
64 bytes frames received	RX_SIZE64
65 to 127 bytes frames received	RX_SIZE65TO127
128 to 255 bytes frames received	RX_SIZE128TO255
256 to 511 bytes frames received	RX_SIZE256TO511
512 to 1023 bytes frames received	RX_SIZE5121023
1024 to 1518 bytes frames received	RX_SIZE1024TO1518
Frames received longer than1518 bytes, and not longer than Maximum Length Register (Maximum Length Register + 4 if the frame is VLAN tagged)	RX_SIZE1519TOMAX
The number of times an IPG shrink was detected	RX_IPG_SHRINK

1. This counter resides in the ingress FIFO. For more information, see "Ingress FIFO Buffer, Block 2," page 130.

Table 8. Transmit Statistics Counters

Counter	Short Name
Bytes transmitted (good, bad and framing)	TX_OUT_BYTES
Dropped frames in the transmitter due to egress FIFO overflow	EGR_DROP_CNT [1]

Table 8. Transmit Statistics Counters *(continued)*

Counter	Short Name
Frames dropped due to aging in egress FIFO	DROP_CNT [2]
Pause control frames transmitted	TX_PAUSE
Bytes transmitted successfully	TX_OK_BYTES
Unicast frames transmitted	TX_UNICAST
Multicast frames transmitted	TX_MULTICAST
Broadcast frames transmitted	TX_BROADCAST
Frames transmitted without errors after multiple collisions	TX_MULTI_COLL
Late collisions detected	TX_LATE_COLL
Frames lost due to excessive collisions	TX_XCOLL
Frames being deferred on first transmission attempt	TX_DEFER
Frames sent with excessive deferral	TX_XDEFER
The number of times CarrierSenseError is true at the end of a frame transmission	TX_CSENSE
X64 bytes frames transmitted	TX_SIZE64
X65 to 127 bytes frames transmitted	TX_SIZE65TO127
X128 to 255 bytes frames transmitted	TX_SIZE128TO255
X256 to 511 bytes frames transmitted	TX_SIZE256TO511
X512 to 1023 bytes frames transmitted	TX_SIZE5121023
X1024 to 1518 bytes frames transmitted	TX_SIZE1024TO1518
The number of frames transmitted longer than1518 bytes and not longer than Maximum Length Register (Maximum Length Register + 4 if the frame is VLAN tagged)	TX_SIZE1519TOMAX
Frames transmitted without errors after a single collision	TX_SINGLE_COLL
Frames sent successfully after 2 backoffs or collisions	TX_BACKOFF2
Frames sent successfully after 3 backoffs or collisions	TX_BACKOFF3
Frames sent successfully after 4 backoffs or collisions	TX_BACKOFF4
Frames sent successfully after 5 backoffs or collisions	TX_BACKOFF5
Frames sent successfully after 6 backoffs or collisions	TX_BACKOFF6
Frames sent successfully after 7 backoffs or collisions	TX_BACKOFF7
Frames sent successfully after 8 backoffs or collisions	TX_BACKOFF8
Frames sent successfully after 9 backoffs or collisions	TX_BACKOFF9
Frames sent successfully after 10 backoffs or collisions	TX_BACKOFF10
Frames sent successfully after 11 backoffs or collisions	TX_BACKOFF11
Frames sent successfully after 12 backoffs or collisions	TX_BACKOFF12
Frames sent successfully after 13 backoffs or collisions	TX_BACKOFF13

Table 8. Transmit Statistics Counters *(continued)*

Counter	Short Name
Frames sent successfully after 14 backoffs or collisions	TX_BACKOFF14
Frames sent successfully after 15 backoffs or collisions	TX_BACKOFF15
Number of times MAC transmit FIFO has dropped a frame because of an underrun	TX_UNDERRUN

1. This counter resides in the egress FIFO. For more information, see "Egress FIFO Buffer, Block 2," page 151.
2. This counter resides in the tri-speed MAC. For more information, see "Tri-Speed MAC, Block 1," page 118.

1.5.1 Mapping of RMON Statistics Group (RFC 2819)

The RMON statistics group is only defined for received frames because it was originally designed for a separate network monitor on a shared medium. In current practice, the RMON specifications are often used both for received as well as transmitted frames.

Although Ethernet maximum frames are specified in the RMON RFC as 1518 bytes, they can now be 1522 bytes due to tagging. To support various interpretations, the VLAN-tagged frames with lengths from 1519 bytes to 1522 bytes are counted in separate counters in the Schaumburg device. The Schaumburg device can be configured to treat VLAN-tagged frames with lengths from 1519 bytes to 1522 bytes as good frames or oversized frames.

Table 9. Mapping Statistics Counters to RMON Group (RFC 2819)

Counter	RX/TX	Implementation
etherStatsDropEvents	RX	ING_DROP_CNT[1]
etherStatsOctets	RX	RX_OK_BYTES + RX_BAD_BYTES
etherStatsPkts	RX	RX_UNICAST + RX_MULTICAST + RX_BROADCAST + RX_CRC + RX_ALIGNMENT + RX_UNDERSIZE + RX_OVERSIZE + RX_FRAGMENTS + RX_JABBERS + RX_IN_RANGE_LENGTH_ERROR + RX_OUT_OF_RANGE_LENGTH
etherStatsBroadcastPkts	RX	RX_BROADCAST
etherStatsMulticastPkts	RX	RX_MULTICAST
etherStatsCRCAlignErrors	RX	RX_CRC + RX_ALIGNMENT
etherStatsUndersizePkts	RX	RX_UNDERSIZE
etherStatsOversizePkts	RX	RX_OVERSIZE
etherStatsFragments	RX	RX_FRAGMENTS
etherStatsJabbers	RX	RX_JABBERS
etherStatsCollisions	RX	RX_CRC + RX_ALIGNMENT
etherStatsPkts64Octets	RX	RX_SIZE64
etherStatsPkts65to127Octets	RX	RX_SIZE65TO127
etherStatsPkts128to255Octets	RX	RX_SIZE128TO255

Table 9. Mapping Statistics Counters to RMON Group (RFC 2819) *(continued)*

Counter	RX/TX	Implementation
etherStatsPkts256to511Octets	RX	RX_SIZE256TO511
etherStatsPkts512to1023Octets	RX	RX_SIZE512TO1023
etherStatsPkts1024to1518Octets	RX	RX_SIZE1012TO1518
etherStatsTXDropEvents	TX	EGR_DROP_CNT + DROP_CNT + TX_XCOLL + TX_XDEFER + TX_LATE_COLL
etherStatsTXOctets	TX	TX_OK_BYTES
etherStatsTXPkts	TX	TX_UNICAST + TX_MULTICAST + TX_BROADCAST + TX_LATE_COLL
etherStatsTXBroadcastPkts	TX	TX_BROADCAST
etherStatsTXMulticastPkts	TX	TX_MULTICAST
etherStatsTXCollisions	TX	TX_SINGLE_COLL + 2 × TX_BACKOFF2 + 3 × TX_BACKOFF3 + . . . + 15 × TX_BACKOFF15 + 16 × TX_XCOLL
etherStatsTXPkts64Octets	TX	TX_SIZE64
etherStatsTXPkts65to127Octets	TX	TX_SIZE65TO127
etherStatsTXPkts128to255Octets	TX	TX_SIZE128TO255
etherStatsTXPkts256to511Octets	TX	TX_SIZE256TO511
etherStatsTXPkts512to1023Octets	TX	TX_SIZE512TO1023
etherStatsTXPkts1024to1518Octets	TX	TX_SIZE1024TO1518

1. Does not include frames dropped due to ingress aging.

1.5.2 Mapping of Ethernet MIB (IEEE Std 802.3-2000 Annex 30A Counters)

Counters based on collisions, deferring, and carrier sense are not applicable to the full-duplex mode.

Table 10. MAC Entity Mandatory Counters

Counter	RX/TX	Implementation
aFramesTransmittedOK	TX	TX_UNICAST + TX_MULTICAST + TX_BROADCAST
aSingleCollisionFrames	TX	TX_SINGLE_COLL
aMultipleCollisionFrames	TX	TX_MULTI_COLL
aFramesReceivedOK	RX	RX_UNICAST + RX_MULTICAST + RX_BROADCAST
aFrameCheckSequenceErrors	RX	RX_CRC
aAlignmentErrors	RX	RX_ALIGNMENT

Table 11. MAC Entity Recommended Counters

Counter	RX/TX	Implementation
aOctetsTransmittedOK	TX	TX_BYTES
aFramesWithDeferredXmissions	TX	TX_DEFER
aLateCollisions	TX	TX_LATE_COLL
aFramesAbortedDueToXSColls	TX	TX_XCOLL
aFramesLostDueToIntMACXmitError	TX	EGR_DROP_CNT
aCarrierSenseErrors	TX	TX_CSENSE
aOctetsReceivedOK	RX	RX_OK_BYTES
aFramesLostDueToIntMACRcvError	RX	ING_DROP_CNT

Table 12. MAC Entity Optional Counters

Counter	RX/TX	Implementation
aMulticastFramesXmittedOK	TX	TX_MULTICAST
aBroadcastFramesXmittedOK	TX	TX_BROADCAST
aMulticastFramesReceivedOK	RX	RX_MULTICAST
aBroadcastFramesReceivedOK	RX	RX_BROADCAST
aInRangeLengthErrors	RX	RX_IN_RANGE_LENGTH_ERROR
aOutOfRangeLengthField	RX	RX_OUT_OF_RANGE_LENGTH
aFrameTooLongErrors	RX	RX_OVERSIZE

Table 13. MAC Entity Array Counter

Counter	RX/TX	Implementation
aCollisionFrames	TX	TX_SINGLE_COLL
		TX_BACKOFF2
		TX_BACKOFF3
		TX_BACKOFF4
		TX_BACKOFF5
		TX_BACKOFF6
		TX_BACKOFF7
		TX_BACKOFF8
		TX_BACKOFF9
		TX_BACKOFF10
		TX_BACKOFF11
		TX_BACKOFF12
		TX_BACKOFF13
		TX_BACKOFF14
		TX_BACKOFF15

Table 14. MAC Entity Excessive Deferral Counter

Counter	RX/TX	Implementation
aFramesWithExcessiveDeferral	TX	TX_XDEFER

Table 15. DTE Physical Entity Recommended Counter

Counter	RX/TX	Implementation
aSymbolErrorDuringCarrier	RX	RX_SYMBOL_CARRIER

Table 16. DTE Physical Entity 10 Mbps Monitor Counter

Counter	RX/TX	Implementation
aSQETestErrors	RX	Not applicable

Table 17. DTE MAC Control Entity Recommended Counters

Counter	RX/TX	Implementation
aMACControlFramesTransmitted	TX	TX_PAUSE
aMACControlFramesReceived	RX	RX_PAUSE + RX_UNSUP_OPCODE
aUnsupportedOpcodesReceived	RX	RX_UNSUP_OPCODE

Table 18. DTE MAC Control Function Entity Recommended Counters

Counter	RX/TX	Implementation
aPauseMACControlFramesTransmitted	TX	TX_PAUSE
aPauseMACControlFramesReceived	RX	RX_PAUSE

Table 19. Mapping of SNMP Interfaces Group MIB (RFC 1213 and 1573)

Counter	RX/TX	Implementation
ifInOctets	RX	RX_IN_BYTES
ifInUcastPkts	RX	RX_UNICAST
ifInMulticastPkts	RX	RX_MULTICAST
ifInBroadcastPkts	RX	RX_BROADCAST
ifInNUcastPkts	RX	RX_MULTICAST + RX_BROADCAST
ifInDiscards	RX	ING_DROP_CNT[1]
ifInErrors	RX	RX_CRC + RX_ALIGNMENT + RX_UNDERSIZE + RX_OVERSIZE + RX_IN_RANGE_LENGTH_ERROR + RX_OUT_OF_RANGE_LENGTH
ifInUnknownProtos	RX	Not applicable
ifOutOctets	TX	TX_OUT_BYTES
ifOutUcastPkts	TX	TX_UNICAST
ifOutMulticastPkts	TX	TX_MULTICAST
ifOutBroadcastPkts	TX	TX_BROADCAST
ifOutNUcastPkts	TX	TX_MULTICAST + TX_BROADCAST
ifOutDiscards	TX	EGR_DROP_CNT + ING_DROP_CNT
ifOutErrors	TX	TX_LATE_COLL

1. Does not include frames dropped due to ingress aging.

Table 20. Mapping of SNMP Ethernet-like Group MIB (RFC 1643)

Counter	RX/TX	Implementation
dot3StatsAlignmentErrors	RX	RX_ALIGNMENT
dot3statsFCSErrors	RX	RX_CRC
dot3StatsSingleCollisionFrames	TX	TX_SINGLE_COLL
dot3StatsMultipleCollisionFrames	TX	TX_MULTI_COLL
dot3StatsSQETestErrors	RX	Not applicable
dot3StatsDeferredTransmissions	TX	TX_DEFER
dot3StatsLateCollisions	TX	TX_LATE_COLL
dot3StatsExcessiveCollisions	TX	TX_XCOLL
dot3StatsInternalMacTransmitErrors	TX	EGR_DROP_CNT + DROP_CNT
dot3StatsCarrierSenseErrors	TX	TX_CSENSE
dot3StatsFrameTooLongs	RX	RX_OVERSIZE
dot3StatsInternalMacReceiveErrors	RX	ING_DROP_CNT[1]

1. Does not include frames dropped due to ingress aging.

1.6 Test Features

The VSC7326 has a number of features that can be used to facilitate the debugging of a new board designed to use the Schaumburg device. These features are also useful for testing a system whose design is based on the Schaumburg device.

- PRBS pattern generation and checking for SPI-4.2. For more information, see "Pattern Checker and Generator," page 89.
- Generating and capturing wire-speed raw SPI-4.2 data bits without decoding.
- A large number of debug counters.
- Multiple internal loopback connections to enable segmented board testing.
- Wire-speed capture of frames from tri-speed MAC and SPI-4.2.
- Wire-speed continuous replay of frames to tri-speed MAC and SPI-4.2 (traffic generation).
- Mix traffic generation and real traffic to emulate high load situations.
- Per-port bandwidth reduction on all data interfaces using traffic shapers (available during traffic generation).

1.6.1 Loopback Connections

The following functional loopbacks are provided:

- SPI-4.2 inside loopback—ingress SPI-4.2 data to egress SPI-4.2 data
- FIFO host side loopback—ingress FIFO data to egress FIFO data
- RGMII inside loopback close to RGMII interface—Tx data to Rx data
- RGMII inside loop far from RGMII interface—Tx data to Rx data

The directions indicated above refer to the data directions. Corresponding flow control signals in the opposite direction are implied in each case. The SPI-4.2 inside loopback only loops the data bus however, not the status channel. It must therefore either be looped externally, or one must use a setup not requiring a valid status channel.

The loopbacks are enabled or disabled by using the configuration registers shown in Figure 14.

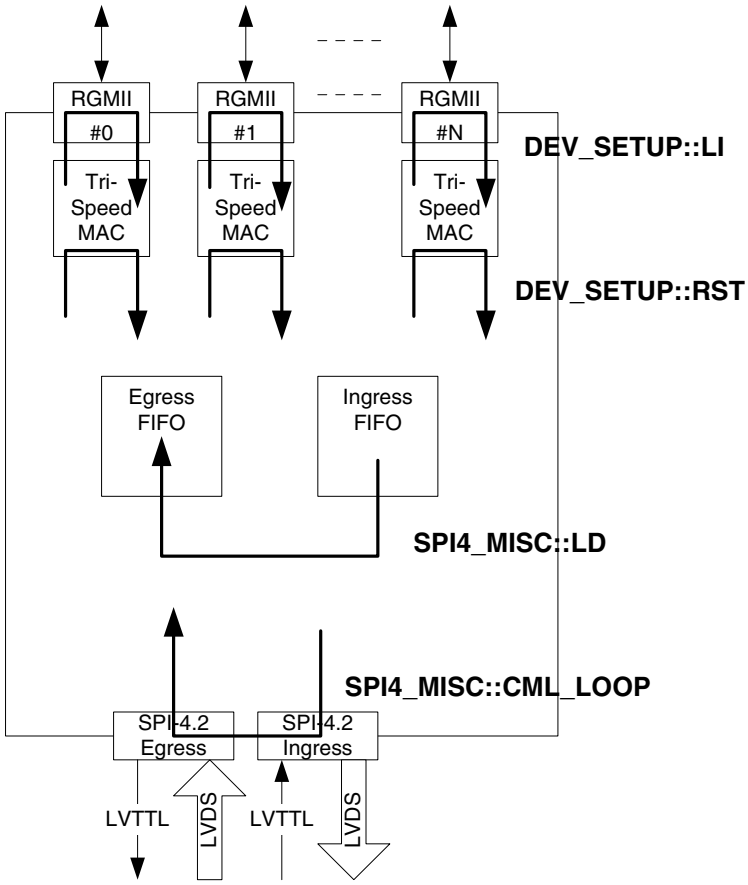


Figure 14. Internal Loopbacks and Corresponding Configuration Enable Bits

1.6.2 Frame Capture and Frame Replay

The FIFOs provide frame capture and frame replay features for use with in-circuit testing and debugging.

- Frame capture—This feature stores a sequence of frames received on an Ethernet port on ingress or a SPI-4.2 channel on egress in the respective FIFO buffer, making it available for inspection through the CPU interface.
- Frame replay—This feature plays out a sequence of frames from a FIFO buffer to the corresponding SPI-4.2 channel on ingress, or Ethernet port on egress, either a single sequence or repetitively. The frame sequence is written to the FIFO through the CPU interface, or captured and then replayed.

Frame replay can be done in various ways.

- Single shot—Data is replayed once.
- Normal replay—Data is replayed continuously, only limited by the interface bandwidth.
- Timed replay—Data is replayed with a timed pause between each replay. The timer being programmable from once every 107 nanosecond to once every 28 second.

For stress testing, replaying can be started simultaneously on all ports with a single register write.

1.6.3 SPI-4.2 Transparent Mode

On SPI-4.2 Frame Capture and Frame Replay can be done with the SPI-4.2 formatting suppressed, so-called Transparent mode. In this mode replay can control both SPI-4.2 control and data words, making it possible to generate arbitrary SPI-4.2 ingress patterns. Capturing in Transparent Mode stores both SPI-4.2 control and data, providing a true snapshot of the SPI-4.2 egress data bus activity.

1.7 Configuration Parameter Values

1.7.1 System Parameters

To clarify the relationship between functionality and register setup, some major mode parameter groups have been defined showing which bits or registers to set for the required functionality. These parameters are used in this datasheet when the function is being described, and are named identically with the MAC API.

For more information about application support, visit the Vitesse Web site at www.vitesse.com.

Table 21. Mapping

Parameter	Value	Description	Register
active_port_map		Defines the virtual port indexes for the active ports, equal to the SPI-4.2 channel ID,	REMAP_ING::REMAP_WR REMAP_ING::REMAP_IN REMAP_ING::REMAP_OUT REMAP_EGR::REMAP_WR REMAP_EGR::REMAP_IN REMAP_EGR::REMAP_OUT
	[port_map]	An array of physical port numbers. port_map[n] contains the number of the physical port that is treated as virtual port # n, which, in turn, is represented as bit n of a port index bit mask.	For each port: REMAP_WR = 1 REMAP_IN = port REMAP_OUT = channel
Configures the port-to-channel map (channel # p, virtual port # p), type: byte[24].			

The CPU parameter group controls CPU interface characteristics.

Table 22. CPU

Parameter	Value	Description	Register
cpu_si_read_wait_states		Configures wait states in the read cycle on the serial CPU interface, to ensure at least 1.3 µs latency between address and data.	SPI_INSERT_BYTES::ISBOR
	0, 1, 2, 3, 4, 5, 6, 7	Number of dummy-byte wait states.	ISBOR = value
Default parameter values are shown in bold text.			

The board parameter group defines options related to board design and layout

Table 23. Board Parameters

Parameter	Value	Description	Register
spi4_pin_swap		Configures the order of the SPI-4.2 data pins. Swapped order: bit 0 <–> bit 15 bit 1 <–> bit 14 (and so forth),	SPI4_MISC::WI SPI4_MISC::WE
	none	Both interfaces are straight.	WI = 0 WE = 0
	both	Both interfaces are swapped.	WI = 1 WE = 1
	ingress	The ingress interface (RDATA) is swapped.	WI = 1 WE = 0
	egress	The egress interface (TDATA) is swapped.	WI = 0 WE = 1
spi4_ing_data_clk_select		Configures the reference clock selection for the ingress data clock, RDCLK.	PLL_CLK_SPEED:: SPI4_XTAL_SEL
	high	The reference clock high range (internally divided by 4).	SPI4_XTAL_SEL = 1
	low	The reference clock low range.	SPI4_XTAL_SEL = 0
spi4_ing_data_clk_multiplier		Configures the reference clock selection for the ingress data clock, RDCLK.	PLL_CLK_SPEED:: SPI4_PLL_MULT
	12, 13, 14, 15, 16, 17, 18	Clock multiplication factor, multiplied to the internal reference clock.	SPI4_PLL_MULT = value
spi4_ing_data_clk_divider		Divides the SPI-4.2 ingress data path frequency by 2.	PLL_CLK_SPEED::SPI4_PLL_DIV2
	off	Disable.	SPI4_PLL_DIV2 = 0
	on	Enable - divided by 2.	SPI4_PLL_DIV2 = 1
spi4_ing_data_clk_misc_ divider		SPI-4.2 ingress data path clock divided by 2.	SPI4_MISC::CRCD2
	off	Unmodified data path frequency (normal operation).	CRCD2 = 0
	on	Half-speed operation (divided by 2).	CRCD2 = 1
spi4_ing_data_clk_phase		Configures the phase of the ingress data clock, RDCLK.	SPI4_MISC::RQC
	off	Output data on the clock edges.	RQC = 0
	on	Output data 90 degrees off the clock edges.	RQC = 1
spi4_ing_status_clk_phase		Configures the phase of the ingress status channel clock, RSCLK.	SPI4_ING_SETUP0::INVERTCLK
	off	Sample data on the rising edge.	INVERTCLK = 0
	on	Sample data on the falling edge.	INVERTCLK = 1
Default parameter values are shown in bold text.			

Table 23. Board Parameters *(continued)*

Parameter	Value	Description	Register
spi4_egr_status_clk_select		Selects the source for the egress status channel clock, TSCLK.	SPI4_EGR_SETUP0::TXSTATCLKSEL
	system_clk	System clock divided by 2.	TXSTATCLKSEL = 0
	data_clk	Egress data clock, TDCLK, divided by 4.	TXSTATCLKSEL = 1
spi4_egr_status_clk_phase		Configures the phase of the egress status channel clock, TSCLK.	SPI_EGR_SETUP0::INVERTCLK
	off	Output data on the rising edge.	INVERTCLK = 0
	on	Output data on the falling edge.	INVERTCLK = 0
Default parameter values are shown in bold text.			

1.7.2 MAC Parameters

The MAC parameter group defines the function of an Ethernet MAC indexed by (port_set).

Table 24. MAC

Parameter	Value	Description	Register
mac_mode		Configures the link speed and duplex mode for the port(s).	MODE_CFG::GIGA_MODE MODE_CFG::MODE DEV_SETUP::MODE
	1000	1 Gbps mode, full-duplex.	MODE = 1, GIGA_MODE = 1
	100	100 Mbps mode, full-duplex.	MODE = 2, GIGA_MODE = 0
	10	10 Mbps mode, full-duplex.	MODE = 3, GIGA_MODE = 0
	100hdx	100 Mbps mode, half-duplex.	DEV_SETUP::MODE = 2, GIGA_MODE = 0, MODE_CFG::MODE = 0
	10hdx	10 Mbps mode, half-duplex.	DEV_SETUP::MODE = 3, GIGA_MODE = 0, MODE_CFG::MODE = 0
mac_enable		Enables or disables data transfer in each direction of the MAC.	MODE_CFG::RX_EN MODE_CFG::TX_EN
	off	Disabled in both directions.	RX_EN = 0, TX_EN = 0
	on	Enabled in both directions.	RX_EN = 1, TX_EN = 1
	ingress	Enabled only in the ingress direction.	RX_EN = 1
	egress	Enabled only in the egress direction.	TX_EN = 1
mac_max_frame_length		Configures the maximum frame size.	MACLEN_CFG::MAXLEN
	64–16383 **1518**	The maximum frame size in bytes, excluding any 802.1Q tags.	MAXLEN = value
Default parameter values are shown in bold text. Configures the port-to-channel map (channel # p, virtual port # p), type: byte[24].			

1.7.3 CRC Parameters

The CRC parameter group defines the CRC generation and validation operation indexed by (port_set).

Table 25. CRC

Parameter	Value	Description	Register
ingress_crc_mode		Configures the ingress CRC checking mode.	NORMALIZER::NO_CRC DEV_DEBUG::KEEP_BAD
	keep	No CRC checking and forwarding of bad frames including CRC errors, runts, and other bad frames.	NO_CRC = 1 KEEP_BAD = 1
	check	The CRC is calculated and compared with the frame CRC. Frames with CRC errors are aborted.	NO_CRC = 0 KEEP_BAD = 0
egress_crc_mode		Configures the egress CRC generation.	DENORM::CRC_ADD DENORM::CRC_UPD CRC_CFG::CRC_IGN
	pass	No CRC operation. All frames are CRC checked.	CRC_ADD = 0 CRC_UPD = 0 CRC_IGN = 0
	add	The CRC is calculated and appended to the frame.	CRC_ADD = 1 CRC_UPD = 0 CRC_IGN = 1
	update	The CRC is calculated and written in the last four bytes of the frame.	CRC_ADD = 0 CRC_UPD = 1 CRC_IGN = 1
Default parameter values are shown in bold text.			

1.7.4 FIFO Parameters

The FIFO parameter group defines the FIFO configuration for a port in one direction indexed by (port_set, direction).

Table 26. FIFO

Parameter	Value	Description	Register
fifo_size		Allocates a memory area for the FIFO.	TOP_BOTTOM::TOP
	2, 4…max ingress **42** egress **12**	The size of the FIFO, measured in kbytes.	BOTTOM = (TOP of preceding FIFO) TOP = (BOTTOM + FIFO size/2)
fifo_read_mode		Configures the readout mode for store-and-forward or cut-through operation.	CT_THRHLD::CT_THRHLD
	store_forward	Operate in store-and-forward mode.	CT_THRHLD = ceil(max frame size/32)
	32, 64…max	Operate in cut-through mode; the value is the cut-through threshold level.	CT_THRHLD = value/32
fifo_protection_size		Configures the action of the ingress frame protection function.	ING_TEST::CLASS_EN CT_THRHLD::CLASS_THRHLD
	off	Protection is disabled.	CLASS_EN = 0
	32, 64…**512**…max	The number of bytes set aside for frame protection.	CLASS_EN = 1 CLASS_THRHLD = (fifo_size - prot_size/32)
Default parameter values are shown in bold text.			

The FIFO Watermarks parameter group defines the FIFO watermarks for a port in one direction. Ranges and default values depend on the direction and major mode, indexed by (port_set, direction).

Table 27. FIFO Watermarks

Parameter	Value	Description	Register
fifo_high_watermark		Configures the high watermark, at which the flow control state is entered.	HIGH_LOW_WM::HIGH_WM
	32, 64…max	The high watermark threshold level in bytes.	HIGH_WM = value/32
fifo_low_watermark		Configures the low watermark, at which the flow control state is exited.	HIGH_LOW_WM::LOW_WM
	32, 64…max	The low watermark threshold level in bytes.	LOW_WM = value/32
Default parameter values are shown in bold text.			

1.7.5 Shaping Parameters

The policing and shaping parameter group defines the behavior of the shaper or policer of a port, indexed by (port_set, direction).

Table 28. Policing and Shaping

Parameter	Value	Description	Register
shaper_bucket_rate		Configures the sustained rate of the ingress policers or egress shapers.	TRAFFIC_SHAPER_CONTROL:: CLEAR_BUCKETS TRAFFIC_SHAPER_BUCKET:: BUCKET_RATE
	off	Disable the shaper or policer.	CLEAR_BUCKETS = 1
	0…65535	The allowed sustained rate measured in units of 146484.375 bps (6828 corresponds to 1 Gbps).	CLEAR_BUCKETS = 0, BUCKET_RATE = value
shaper_bucket_level		Configures the bucket level of the ingress policers or egress shapers, at which data is being held back. This defines the allowed burstiness of the resulting data traffic.	TRAFFIC_SHAPER_BUCKET:: BUCKET_LEVEL
	0…**256**...65535	The bucket level threshold measured in units of 128 bytes.	BUCKET_LEVEL = value
Default parameter values are shown in bold text.			

The policer common group of parameters defines the behavior of the common policer on ingress.

Table 29. Policer Common

Parameter	Value	Description	Register
policer_bucket_rate		Configures the sustained rate of the ingress policers or egress shapers.	JOINT_TRAFFIC_SHAPER_BUCKET:: BUCKET_RATE
	0…65535	The allowed sustained rate measured in units of 292968.75 bps.	BUCKET_RATE = value
policer_bucket_level		Configures the bucket level of the ingress policers or egress shapers, at which data is being held back. This defines the allowed burstiness of the resulting data traffic.	JOINT_TRAFFIC_SHAPER_BUCKET:: BUCKET_LEVEL
	0…**256**..65535	The bucket level threshold measured in units of 128 bytes.	BUCKET_LEVEL = value

1.7.6 SPI-4.2 Parameters

The SPI-4.2 ingress group defines the behavior of the ingress SPI-4.2 interface.

Table 30. SPI-4.2 Ingress

Parameter	Value	Description	Register
spi4_ing_sch_mode		Selects the burst-scheduling mode on the ingress SPI-4.2 interface.	ING_CONTROL::CM, NH, LE NORMALIZER::NH, NLE
	normal	Normal mode, also known as burst-interleaved mode.	CM = 1
	frame_mode	Full-frame mode, also known as frame-interleaved mode, with policers enabled for frame-size fairness.	CM = 0, NH = 1, LE = 1 Ingress shaper_bucket_rate = 6826
spi4_ing_burst_size		Configures the maximum burst size allowed from the ingress FIFO.	SPI4_ING_SETUP1::MAXBURSTSIZE FIFO CONTROL::BURST_LEN
	32, 64…**128**…480	The maximum number bytes in a single burst.	MAXBURSTSIZE = value/32
spi4_ing_maxburst1		Configures the number of credits allowed from the ingress FIFO when the ingress status channel indicates STARVING, MaxBurst1.	SPI4_ING_SETUP1::MAXBURST1
	32, 64…**256**…4096	The number of bytes allowed; must be a multiple of 16.	MAXBURST1 = value/16
spi4_ing_maxburst2		Configures the number of credits allowed from the ingress FIFO when the ingress status channel indicates HUNGRY, MaxBurst2.	SPI4_ING_SETUP1::MAXBURST2
	32, 64…**256**…4096	The number of bytes allowed; must be a multiple of 16.	MAXBURST2 = value/16
spi4_ing_calendar_m		The number of repetitions of the status calendar before DIP2. Note that the value must not be lower than 1: CALENDAR_M = 1 (default)	SPI4_ING_SETUP0::CALENDAR_M
spi4_ing_linkup_limit		The number of consecutive good DIP2 to receive before accepting link. Note that the value must not be lower than 1: LINKUPLIMIT = 1 (default)	SPI4_ING_SETUP0::LINKUPLIMIT
spi4_ing_linkdown_limit		The number of consecutive bad DIP2 to receive before dropping link. Note that the value must not be lower than 1: LINKDOWNLIMIT = 1 (default)	SPI4_ING_SETUP0::LINKDOWNLIMIT
Default parameter values are shown in bold text.			

The SPI-4.2 egress parameter group controls the behavior of the egress SPI-4.2 interface.

Table 31. SPI-4.2 Egress

Parameter	Value	Description	Register
spi4_egr_status_indication		Configures which status indication is sent when the FIFO is able to receive data, that is, not in flow control state. When in flow control state, SATISFIED is indicated.	SPI4_EGR_SETUP0::CP
	starving	STARVING is indicated when ready.	CP = 00
	hungry	HUNGRY is indicated when ready.	CP = 01
spi4_egr_deskew_mode		Configures whether dynamic deskew is enabled.	SPI4_DSKW_CTRL_MODE:: DSKW_MODE_RST SPI4_DSKW_CTRL_MODE:: DSKW_MODE_EDGE SPI4_DSKW_CTRL_MODE:: DSKW_MODE_ENA SPI4_DSKW_CTRL_DIP4_ERR_THRS:: EGR_DIP4_FAIL SPI4_DSKW_CTRL_DIP4_WINDOW:: EGR_DIP4_WINDOW
	disabled		DSKW_MODE_ENA = 0
	enabled		DSKW_MODE_RST = 0x4 DSKW_MODE_EDGE = alpha of SPI-4.2 transmitter DSKW_MODE_ENA = 1 EGR_DIP4_FAIL = 15 EGR_DIP4_WINDOW = 30
spi4_egr_calendar_m			SPI4_EGR_SETUP0::CALENDAR_M
spi4_egr_alpha		Swap the egress training sequence repetition factor for the dynamic deskew.	SPI4_DSKW_CTRL_MODE:: DSKW_MODE_EDGE
	1...256		
Default parameter values are shown in bold text.			

The SPI-4.2 training parameter group controls the training pattern insertion in the ingress data path.

Table 32. SPI-4.2 Training

Parameter	Value	Description	Register
spi4_ing_training_mode		Configures if and when training sequences are to be issued on the ingress data path.	SPI4_ING_SETUP1::EN_AUTO_TRAIN SPI4_ING_SETUP1::SEND_TRAIN
	auto	Issue training sequences, in response to all 11s on the status channel.	EN_AUTO_TRAIN = 1
	off	Do not issue training sequences.	EN_AUTO_TRAIN = 0
	force	Issue training sequences continuously.	SEND_TRAIN = 1
spi4_ing_alpha		Configures the training sequence repetition factor.	SPI4_ING_SETUP1::ALPHA
	1… 255	The number of times that the 20-cycle training pattern is repeated in every training sequence.	ALPHA = value
spi4_ing_data_max_t		Configures the training sequence repetition period.	SPI4_ING_SETUP1:: TSPERIOD
	4, 8…262140	The maximum interval between scheduling of training sequences, measured in units of 4 SPI-4.2 data cycles.	TSPERIOD = value/4
Default parameter values are shown in bold text.			

1.7.7 MAC Flow Control Parameters

The MAC flow control parameter group defines the flow control operation of an Ethernet MAC, indexed by (port_set).

Table 33. MAC Flow Control

Parameter	Value	Description	Register
mac_flow_control		Enables or disables flow control in each direction of the MAC for ports running in full-duplex mode. In half-duplex mode, disabling of flow control must be done by setting ING_HIGH_LOW_WM::HIGH_WM to 0x3FFF.	PAUSE_CFG::PAUSE_EN PAUSE_CFG::TX_PAUSE_EN PAUSE_CFG::RX_PAUSE_EN
	off	Disabled in both directions.	PAUSE_EN = 0
	on	Enabled in both directions.	PAUSE_EN = 1
	obey	Enabled only in ingress direction.	PAUSE_EN = 0, RX_PAUSE_EN = 1
	generate	Enabled only in egress direction.	PAUSE_EN = 0, TX_PAUSE_EN = 1
mac_pause_mode		Configures the scheme for issuing pause frames in the egress direction.	PAUSE_CFG::TX_PAUSE_XONXOFF
	pause_time	Pause frames are issued at the beginning of the pause and periodically, every half mac_pause_time interval.	TX_PAUSE_XONXOFF = 0
	xonxoff	A pause frame is also issued when the pause ends.	TX_PAUSE_XONXOFF = 1
mac_pause_time		Configures the pause time value set in issued pause frames.	PAUSE_CFG::TX_PAUSE_VAL
	1…**65535**	Pause time, measures in units of 512 bit times.	TX_PAUSE_VAL = mac_pause_time
mac_address		The source MAC address to be used in issued pause frames.	MAC_ADR_HIGH_CFG:: MAC_ADR_LOW_CFG::
	<48bit address>	48-bit MAC address.	MAC_ADDR_HIGH = address<47..24> MAC_ADDR_LOW = address<23..0>
Default parameter values are shown in bold text.			

Table 34. Flow Control Mode

Parameter	Value	Description	Register
host_controlled_mode		Defines the overall egress flow control scheme, that is, what directly controls the egress SPI-4.2 status channel.	(Egress FIFO CONTROL) EGR_CONTROL::ETFC
	off	Fill-level controlled operation.	ETFC = 0
	egress	Host-controlled operation on egress.	EFTC = 1
Default parameter values are shown in bold text.			

1.7.8 Diagnostic Control Parameters

Table 35. Debug Parameters

Parameter	Value	Description	Register
loopback		Establish loopback connection test mode.	DEV_SETUP::LI DEV_SETUP::RST SPI4_MISC::LD SPI4_MISC::CML_LOOP
fifo_capture		Capture FIFO traffic for the SPI-4.2 interface.	SPI4_TEST::Log
fifo_ing_transparent		Establish transparent test mode for SPI-4.2.	SPI4_MISC::EN_TRANSP
fifo_read_data		Read FIFO buffer data after it is captured.	FIFO TEST:: MODE
fifo_normal		FIFO normal test mode.	FIFO TEST:: MODE
fifo_rx_stop		Stop receive traffic to the FIFO.	FIFO TEST:: MODE
fifo_tx_stop		Stop transmit traffic from the FIFO.	FIFO TEST:: MODE

1.7.9 GPIO Parameters

Table 36. GPIO Parameters

Parameter	Value	Description	Register
gpio_config		Setup direction of the GPIO pins.	GPIO_CTRL::PD7_0
	Input	GPIO pin is input.	PD7_0 = 0
	Output	GPIO pin is output.	PD7_0 = 1
gpio_in		Input value of GPIO pins.	
	0..255		GPIO_IN = Pin value
gpio_out		Output value of GPIO pins.	Pin value = GPIO_OUT
	0..255		Pin value = GPIO_OUT
Default parameter values are shown in bold text.			

The FIFO Common parameter group defines FIFO behavior common to all ports.

Table 37. FIFO Common

Parameter	Value	Description	Register
protection_classes		Configures the identification of frames to protect.	CLASSIFIER_CTRL::all
class_mask		Setup of classifier masks.	CLASSIFIER_CTRL

1.8 Initialization

To ensure proper operation of the FIFO RAMs, a RAM BIST must be executed during the initialization of the Schaumburg device. BIST execution is shown in TCL code below, assuming access to register writing function ht_spi_wr and register reading function ht_spi_rd.

The RAM BIST must be executed t_{rec} after reset has been released. The execution of the RAM BIST algorithm takes approximately 200 ms. For more information about the value of t_{rec}, see "Reset Timing," page 230.

```
# -------------------------
 # Schaumburg BIST
 # -------------------------
 # Enable BIST control ring
 ht_spi_wr system 15 0x04 0x0005
 # release reset on device1g
 ht_spi_wr device 0-11 0x0B 0x0000
 # start BIST
 foreach ram {13 14 20 21} {
   ht_bist_wr $ram 0x00 0x02
   ht_bist_wr $ram 0x01 0x01
 }
 run 200 ms
 foreach ram {13 14 20 21} {
   set res [ht_bist_rd $ram 0x02]
   set col [expr ([ht_bist_rd $ram 0x0e]<<8) + [ht_bist_rd $ram 0x0d]]
   if {($res & 3) != 0x3} {
     exb_error "BIST error in ram $ram, column: [format 0x%04x $col]"
   }
 }
 # enable RAM
 foreach ram {13 14 20 21} {
   ht_bist_wr $ram 0x00 0x00
 }
```

The definition of ht_bist_rd is as follows:

```
proc ht_bist_rd {moduleID address} {
   if {$address > 0x02} {
       echo "Unknown BIST address: $address";
   }
   set data 0x00[format "%02X" $address]00[format "%02X" $moduleID]
   # Prepare reading result register
   ht_spi_wr system 0x01 0x00 $data
   # Wait until read from "BIST Read Result" register is completed
   run 10 us
   set ret_val [ht_spi_rd system 0x01 0x01]
   if {[expr $ret_val >> 9] == 1} {
     echo "bist ring in read process"
   } elseif {[expr $ret_val >> 8] == 1} {
     echo "bist ring has read error"
   }
   return [expr $ret_val & 0xff]
}
```

The definition of ht_bist_wr is as follows:

```
proc ht_bist_wr {moduleID address val} {
   if {$address > 0x02} {
       echo "Unknown BIST address: $address";
   } elseif {$val > 255} {
       echo "Suspicous write, value out of range";
   }
   set data 0x01[format "%02X" $address][format "%02X" $val][format "%02X"
$moduleID]
   # Prepare write to a bist register
   ht_spi_wr system 0x01 0x00 $data
   # Wait until write is completed
   run 5 us
   set ret_val [ht_spi_rd system 0x01 0x00]
   if {[expr $ret_val >> 27] == 1} {
     echo "bist ring in write process"
   } elseif {[expr $ret_val >> 26] == 1} {
     echo "bist ring has write error"
   }
}
```

2 INTERFACES

2.1 Reduced Gigabit Media Independent Interfaces (RGMII)

Each Ethernet port features a Reduced Gigabit Media Independent Interface (RGMII).

RGMII features a reduced pin-count alternative to the GMII interface with four data pins per direction, supporting tri-speed operation as follows:

- 1000 Mbps at 125 MHz double data rate – data sampled on both clock edges
- 100 Mbps at 25 MHz, nibble oriented
- 10 Mbps at 2.5 MHz, nibble oriented

The port speed and duplex mode are selected using the mac_mode parameter.

2.2 SPI-4.2 Host Interface

The interface to the Host processor is a System Packet Interface Level 4 Phase 2 (SPI-4.2) interface as specified in the Implementation Agreement (OIF-SPI4-02.0) by the Optical Internetworking Forum.

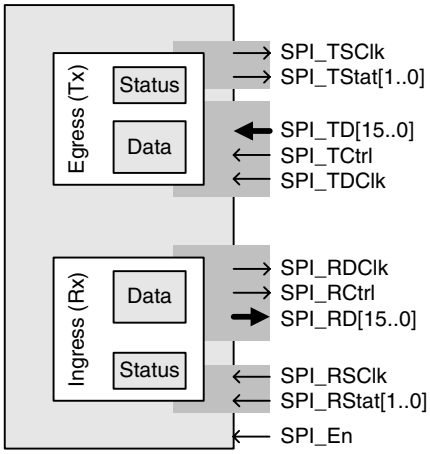


Figure 15. SPI-4.2 Interface Pin Overview

The SPI-4.2 interface provides a high-speed packet transfer interface between the MAC device and a Host device. An application example would be a network processor or a Switch System ASIC. The following is a list of SPI-4.2 interface features:

- Maximum 13.0-Gbps data transfer rate in each direction at 407 MHz
- 16-bit wide LVDS DDR data path
- 2-bit wide LVTTL FIFO status channels

- 75 to 407 MHz egress data clock range
- 75 to 112 MHz, 150 to 225 MHz, and 300 to 407 MHz ingress data clock range
- Dynamic deskew, 300 MHz to 407 MHz
- Burst-interleaved mode as well as full-frame mode
- Ingress full-stop flow control channel
- Sensible credit system during full-frame mode
- Bit-order reversal to simplify board layout
- Configurable mapping between SPI-4.2 channels and Ethernet ports
- DIP-4 error detection and optional frame drop
- CRC checker
- Pattern generator and checker validating SPI-4.2 data path integrity

2.2.1 Data Paths

SPI-4.2 provides two independent unidirectional 16-bit wide LVDS data paths at up to 407 MHz Double Data Rate (DDR). This means that the data bits are sampled on both the rising and the falling edge of the clock. Each data path provides data rates up to 12.8 Gbps.

- The egress data path (SPI_TD) operates with a DDR clock (SPI_TDClk) at frequencies of up to 407 MHz. The egress interface features dynamic deskew at clock frequencies above 300 MHz.
- The ingress data path (SPI_RD) operates with a DDR clock (SPI_RDClk) at a frequency in the ranges 75 to 112.5 MHz. 150 to 225 MHz, and 300 to 407 MHz. The frequency is selected by a reference clock frequency in the ranges 25.00 to 37.5 MHz and 66.67 to 125.0 MHz, and by configuration of the clock multiplier parameter (spi4_ing_data_clk_multiplier).

Data packets are transferred in bursts, separated by SPI-4.2 control words with packet boundary and channel identification. SPI_RCtrl and SPI_TCtrl indicate whether SPI_RD[15:0] and SPI_TD[15:0] are transferring a control word or a data word. The control word consists of two parts; one part covering previous data words, and the other part covering later data words. This is illustrated in Figure 16, which shows a 32-byte burst on port 5 starting a frame, followed by a 5-byte burst on port 2 that ends the frame. The clock is shown with a 90-degree shift and is enabled with SPI4_MISC::RQC. This is not according to the SPI-4.2 standard for the transmitting side, but it is the clock phase that is most convenient when transmitting to a SPI-4.2 receiver, which is placed nearby.

An example of a SPI-4.2 data path showing control word coverage range is shown in Figure 16.

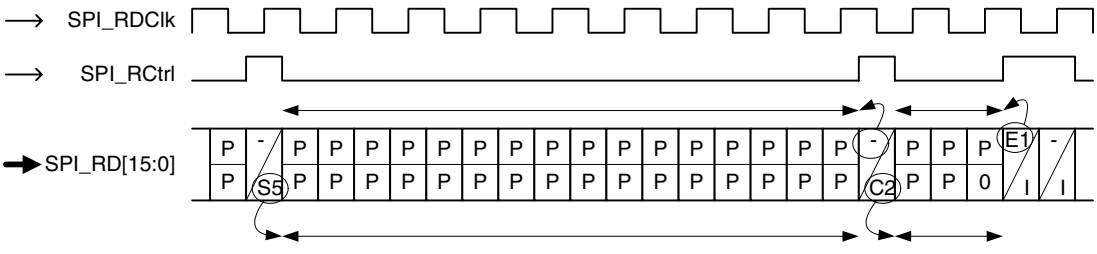


Figure 16. SPI-4.2 Data Path: Control Word Coverage Range Example

Figure 17, Figure 18, and Figure 19 show three different examples of SPI-4.2 data path transfers on port n.

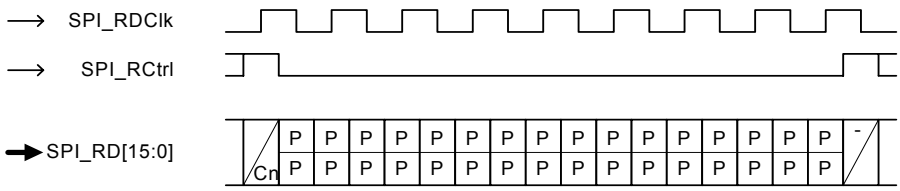


Figure 17. Transferring 32 Bytes of the Middle of a Frame

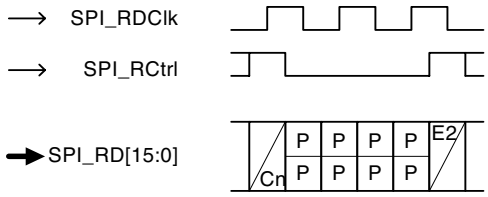


Figure 18. Transferring Last 8 Bytes of an OK Frame

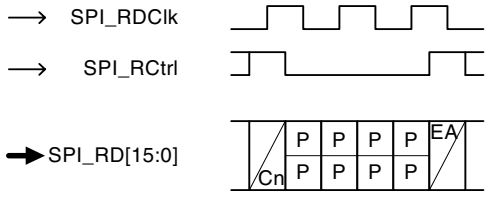


Figure 19. Transferring Last 7 or 8 Bytes of an Aborted Frame

Figure 20 shows the control word format and the possible values.

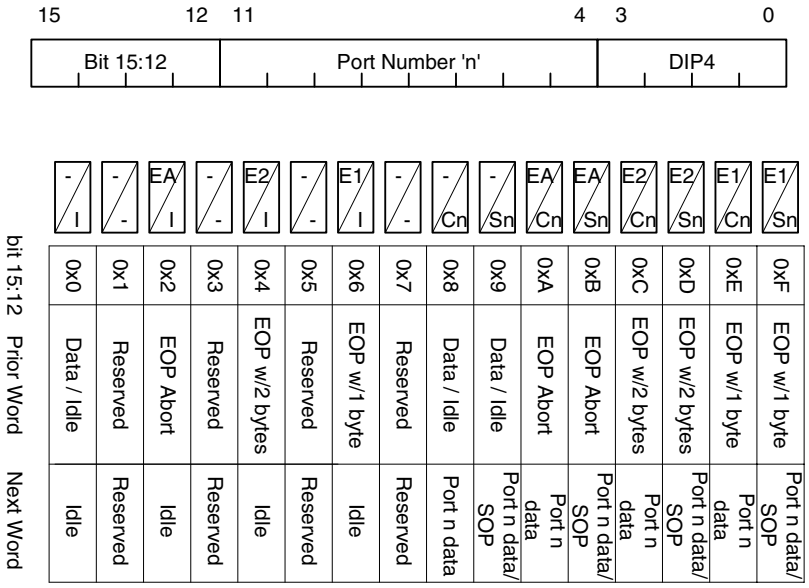


Figure 20. SPI-4.2 Control Word Format

The control words can also indicate an aborted frame. In the ingress direction, the Schaumburg transmits an EOP Abort control word when running in cut-through mode, and when the frame currently being forwarded on SPI-4.2 generates a frame reception error. In the egress direction, the Schaumburg discards frames that are terminated with an EOP Abort. For more information about cut-through and store-and-forward modes, see "Cut-Through and Store-and-Forward Mode," page 39.

Data words are transmitted in bursts, which are multiple of 32 bytes, except for the burst holding the end of packet (EOP). The maximum burst size in the ingress direction can be set to 32 bytes, 64 bytes, 128 bytes, and 256 bytes by configuring the spi4_ing_burst_size parameter.

The data path is protected by a DIP-4 parity field with each burst. It is configurable if bursts with DIP-4 errors result in the corresponding packet being aborted or discarded.

The order of the data pins can be reversed for easier board layout. The order is independently selectable in each direction by configuring the spi4_pin_swap parameter.

2.2.2 Burst-Interleaved Mode and Full-Frame Mode

Data from multiple ports can be transferred over SPI-4.2 ingress by two different modes, which are set in the spi4_ing_sch_mode parameter.

- Burst-interleaved — SPI-4.2 data bursts belonging to different ports can be intermixed or *interleaved*.
- Full-frame mode — SPI-4.2 finishes transmitting data from one port before it begins to transmit from another port.

The burst-interleaved mode is the standard mode for data transfer over SPI-4.2. This mode allows bursts from different channels to be interleaved regardless of packet boundaries.

In contrast, the Full-frame mode only switches the channel on frame boundaries. This mode is feasible for single-task packet processors that cannot handle burst-interleaved frames. In this mode, a frame can be scheduled for transfer if the relevant status channel does not indicate the SATISFIED condition. The frame is then transferred in its entirety regardless of status indication during transmission.

The egress mode is defined by the operation of the host processor.

Note: When running full-frame mode on ingress, the FIFOs must run in store-and-forward mode, and the ingress policer must be enabled. In full-frame mode, the number of uncongested ports transferred over SPI-4.2 is reduced from 12 ports to 10 ports.

The bandwidth reduction only occurs for small frames. The worst case of 10 uncongested ports occurs if all Ethernet ports receive frames with sizes below 72 bytes. For any normal frame size distribution, SPI-4.2 can support 12 uncongested ports.

For more information about the operation of the SPI-4.2 status channel, see "Status Channels," page 84 and "Status Channel Credit-Bypass," page 88.

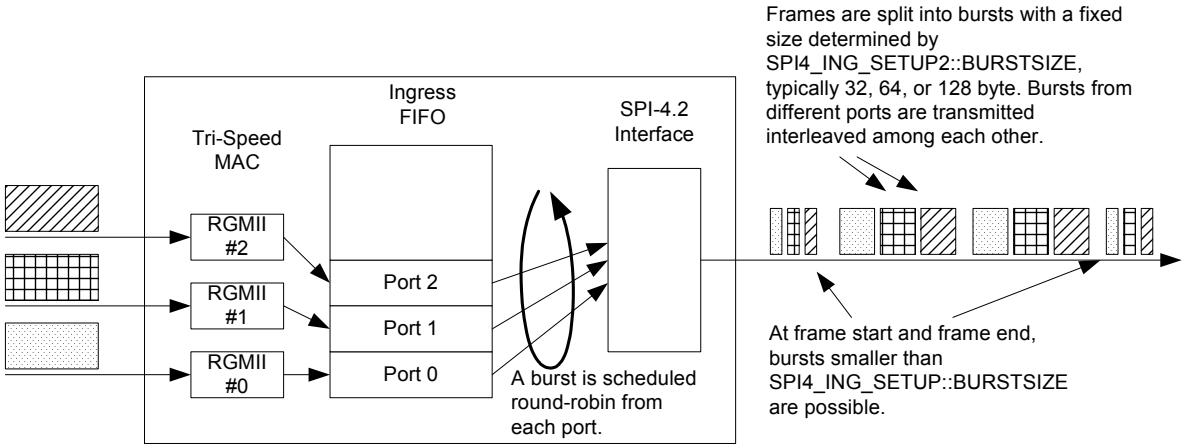


Figure 21. Burst-Interleaved Behavior

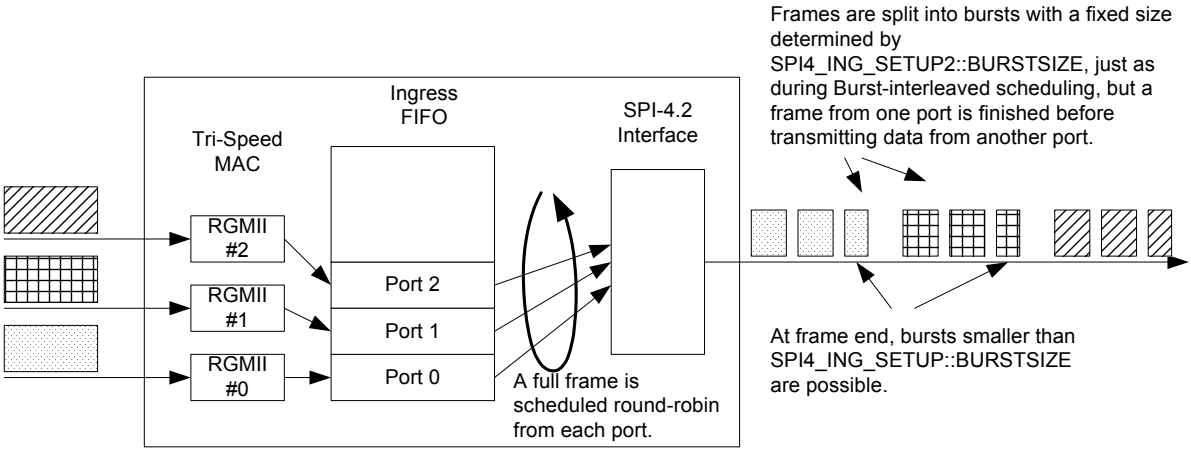


Figure 22. Full-Frame Mode Behavior

2.2.3 Status Channels

The data flow over the SPI-4.2 interface is controlled by the serial status channels. SPI-4.2 flow control information is transferred out-of-band on the 2-bit LVTTL status channels, independently for each direction. The status channels use a source synchronous clock at approximately one-fourth of the corresponding data clock frequency. The status channel is sampled on one clock edge only; therefore, the status signal bit rate is one-eighth of the data signal bit rate.

- The ingress path status channel (SPI_RStat) operates with a clock (SPI_RSClk) in the range from 0 MHz to 112 MHz.
- The status channel Calendar length, sequence and repetition factors are configurable in both directions using the {spi4_calendar_len, spi4_calendar_seq, spi4_calendar_m} parameters.
- The egress path status channel (SPI_TStat) operates with a clock (SPI_TSClk) at SPI_TDClk/4 or fixed at 75 MHz.

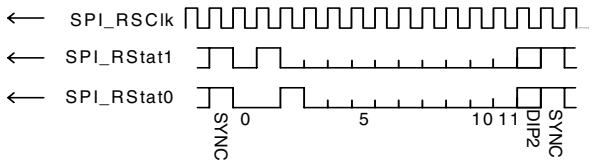


Figure 23. SPI-4.2 Status Channel Format Example

The format of the status channel, shown in Figure 23, is for a 12-port configuration, where 0 to 11 are the time slots for ports 0 to 11 used to indicate the flow control status. Port 1 is SATISFIED, port 2 is HUNGRY, and the remaining ports are STARVING. The last port is followed by a parity check called DIP2, followed by a '1 1' framing pulse called SYNC in Figure 23, indicating the end or start of one status channel period.

The SPI-4.2 status channel is credit-based; that is, each time status channel information is received on a port, only a certain amount of data may be transmitted, which is programmed by the spi4_ing_maxburst1 and spi4_ing_maxburst2 parameters (Table 38).

Table 38. Ingress Status Channel Indication and Corresponding Amount of Data Transmitted

SPI_RStat1	SPI_RStat0	Description
1	1	Reserved for the framing pulse.
1	0	**SATISFIED** Indicates that the corresponding port's FIFO is almost full. When SATISFIED is received, only transfers using the remaining previously granted bytes (if any) may be sent to the corresponding port until the next status update. No additional transfers to that port are permitted while SATISFIED is indicated.
0	1	**HUNGRY** When HUNGRY is received, transfers for up to spi4_ing_maxburst2 bytes or the remainder of what was previously granted (whichever is greater) may be sent to the corresponding port until the next status update.
0	0	**STARVING** Indicates that the buffer has additional space. When STARVING is received, transfers for up to spi4_ing_maxburst1 bytes may be sent to the corresponding port until the next status update.

The egress status channel operates with only two indication levels. SATISFIED indicates that the channel operates above the FIFO watermarks. When operating below the FIFO watermarks, the Schaumburg transmits either HUNGRY or STARVING, which is determined by the configuration of the spi4_egr_status_indication parameter. Figure 24 shows the relationship between the egress FIFO buffer fill level, configuration, and egress status channel signaling.

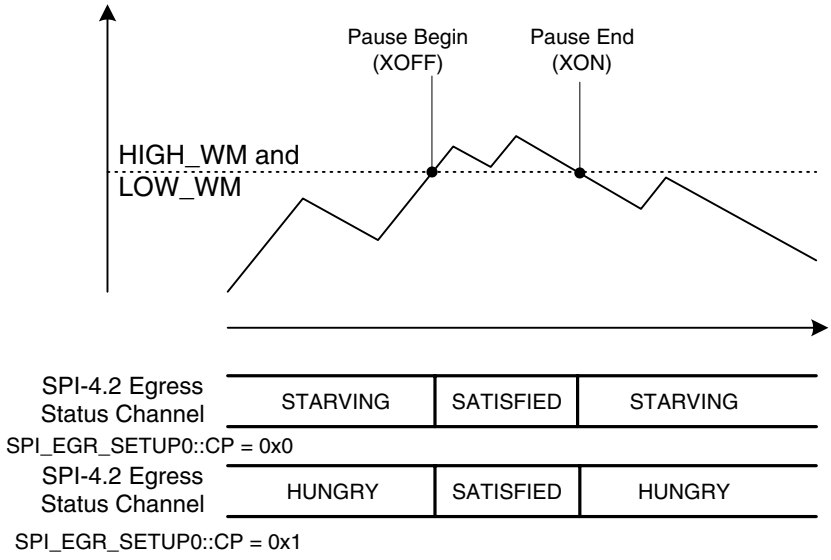


Figure 24. The Egress FIFO Buffer Fill Level, Configuration, and Egress Status Channel Signaling

2.2.4 Flow Control

The maximum burst size parameter spi4_ing_burst_size defines the size of the data bursts, except for the first and last one. The size of this parameter affects the overhead on the interface and the memory usage in the receiving FIFOs. A large burst size reduces the relative overhead but increases the amount of memory required for each FIFO. The recommended maximum burst size is 128 bytes.

The spi4_ing_maxburst1 and spi4_ing_maxburst2 parameters define the credit size and thus affect the receiving FIFO watermark settings. The larger the credits, the more memory is required above the watermarks. This allows the data transfer to be more bursty within each channel. The recommended value is 256 bytes. With a 100 MHz status channel clock and a calendar length of 12, you are allowed to transmit 256 bytes every 14 clock cycles (12 channels plus DIP2 plus sync); that is, 256 bytes every 140 ns resulting in 14.6 Gbps. Because this is above the Ethernet rate of 1 Gbps, the SPI-4.2 status channel does not limit the bandwidth. The ingress credit parameters are configured by setting the spi4_ing_maxburst1and spi4_ing_maxburst2 parameters.

2.2.5 Channel-to-Port Mapping

The default relationship between the SPI-4.2 channels and the Ethernet ports uses a one-to-one mapping for the SPI channel numbers and the Ethernet port numbers. The mapping can be changed using the active_port_map parameter. Changing the mapping can be helpful when a subset of the Ethernet ports is in use. This allows any set of Ethernet ports to be represented as a consecutive range of SPI-4.2 channel numbers.

Figure 25 shows a port remapping example where RGMII port 0 and 6 are mapped to SPI-4.2 port numbers 0 and 1.

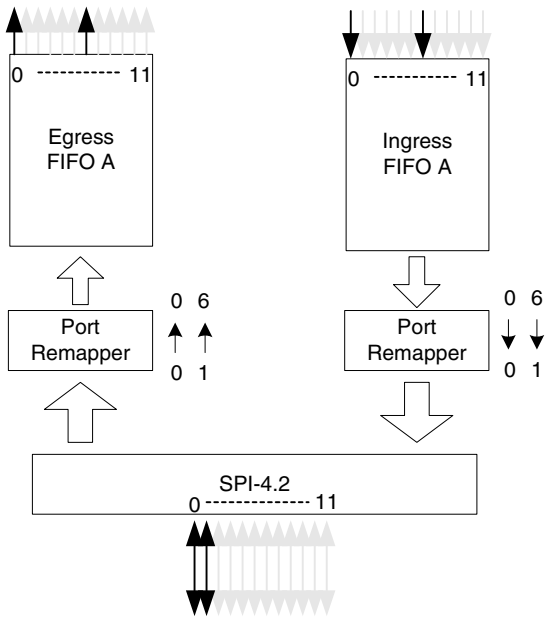


Figure 25. Port Remapping Example: RGMII Ports to SPI-4.2 Ports

2.2.6 Dynamic Deskew

Dynamic deskew is supported in both the ingress and egress data path directions.

The ingress path direction can insert training sequences into its data path under normal operations. This is shown in Figure 26 with spi4_ing_alpha = 4. Notice that extra idle control words can be inserted at the beginning. The training factor, the repetition interval, and the repetition sequence period can be set by configuring the spi4_ing_training_mode, spi4_ing_alpha, spi4_ing_data_max_t parameters.

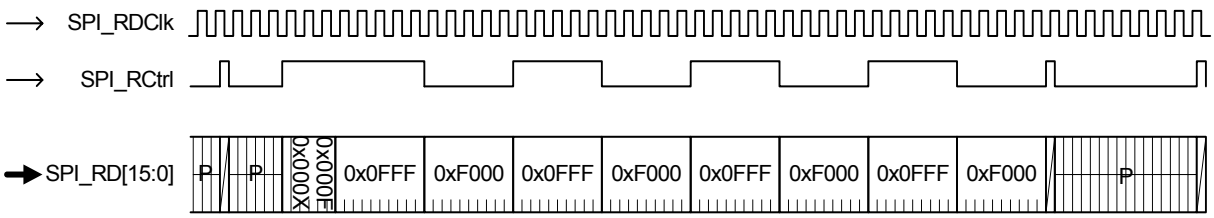


Figure 26. SPI-4.2 Data Path Example, Showing Training Sequence Insertions

Training sequences can also be inserted as a response to status channel signaling or per request from the CPU. Training sequences are set using the spi4_ing_training_mode parameter.

Figure 27 shows SPI-4.2 ingress after reset, and illustrates how enabling the status channel causes the transmission of repeated training patterns to be stopped. Repeated training patterns are transmitted as long as the status channel is '11'. The transmission of training patterns stops after SPI4_ING_SETUP0::LINKUPLIMIT good DIP2s (which is 2 in this example). The status channel length is 14 with port 5 and port 11 indicating SATISFIED and all other ports indicating STARVING.

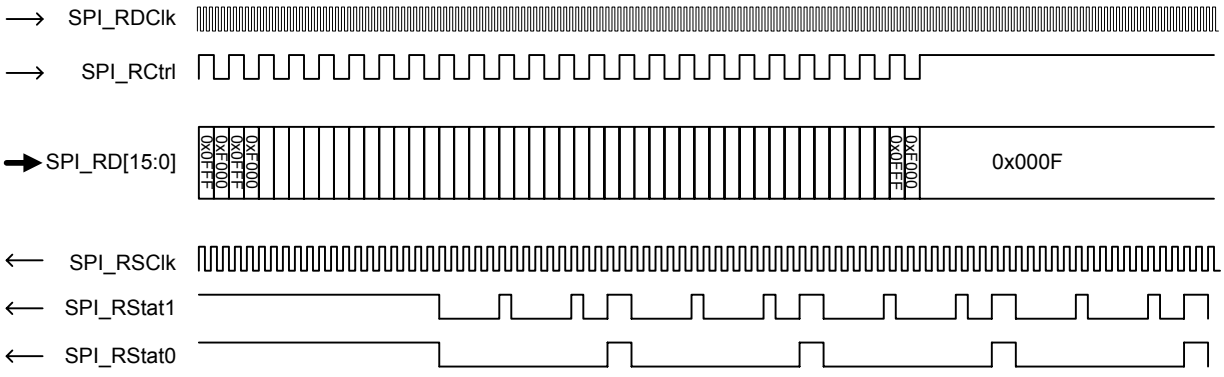


Figure 27. SPI-4.2 Ingress Start-Up After Reset

The egress direction can set the dynamic deskew of the data path for clock frequencies above 300 MHz and is enabled using the spi4_egr_deskew_mode configuration parameter. For lower clock frequencies, dynamic deskew must be disabled. The dynamic deskew circuitry is capable of aligning a skew of ±1 bit time, as required by the SPI-4.2 standard.

The quality of the egress data path is continuously monitored by the DIP-4 error detection feature. In case of excess errors, a training sequence for resynchronization is requested using the status channel.

2.2.7 Ingress Full-Stop Flow Control Channel

Some NPU applications make a full-stop on the SPI-4.2 ingress data flow. On the Schaumburg, it is possible to use the top-most status channel as a full-stop status channel. When the Schaumburg channel operation indicates the SATISFIED condition, the ingress data is completely stopped. When channel operation indicates STARVING or HUNGRY, the ingress data flow is controlled by the lower-numbered status channels.

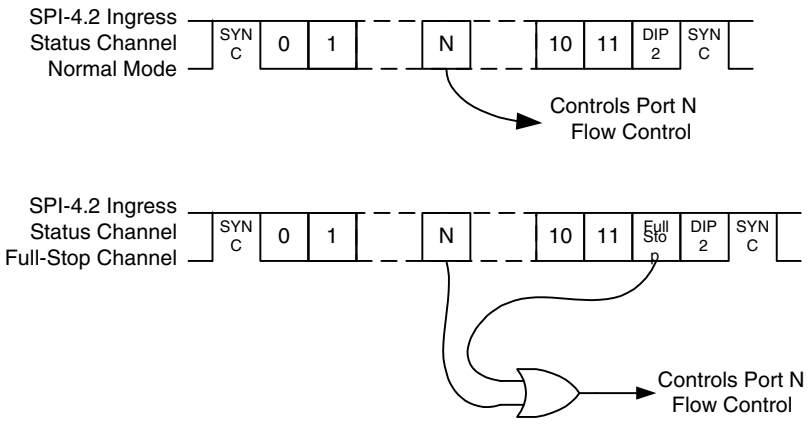


Figure 28. Full-Stop Channel

Figure 28 shows the format of the status channel in a situation where 12 RGMII ports are active. The top-half of the figure shows a configuration with normal-mode, and the bottom-half shows a configuration with a full-stop channel. The OR gate commands a stop on port N if either status channel N or the full-stop channel indicate SATISFIED.

The status calendar length cannot have a length shorter than the number of active ports. It is therefore not possible to define a calendar length of 1 that is solely used as a full-stop channel.

2.2.8 Status Channel Credit-Bypass

In full-frame mode, the credit-based SPI-4.2 flow control system is not well suited to control data from individual ports. Instead the Schaumburg ingress status channel can be set in a Credit Bypass mode. This causes the status channel slot to directly enable or disable data output from a port buffer. When doing full-frame mode in the ingress direction, Credit Bypass should always be enabled. In the egress direction, the NPU should behave in a similar way.

2.2.9 Pattern Checker and Generator

The SPI-4.2 interface contains a sophisticated pattern generator and pattern checker, which are controlled by the SPI4_TEST register. The generator transmits on the SPI_RD[15:0] and SPI_RCtrl pins, either from a linear feedback sequence register (LFSR) of 31 or 23 bits, or a 64-bit user pattern. The 31-bit and 23-bit LFSRs are maximum length, meaning that they generate a bit stream that repeats after $2^{31} - 1$ and $2^{23} - 1$ bits, and can be considered as random. For this reason, they are called PRBS-31 and PRBS-23 (pseudo-random binary sequence).

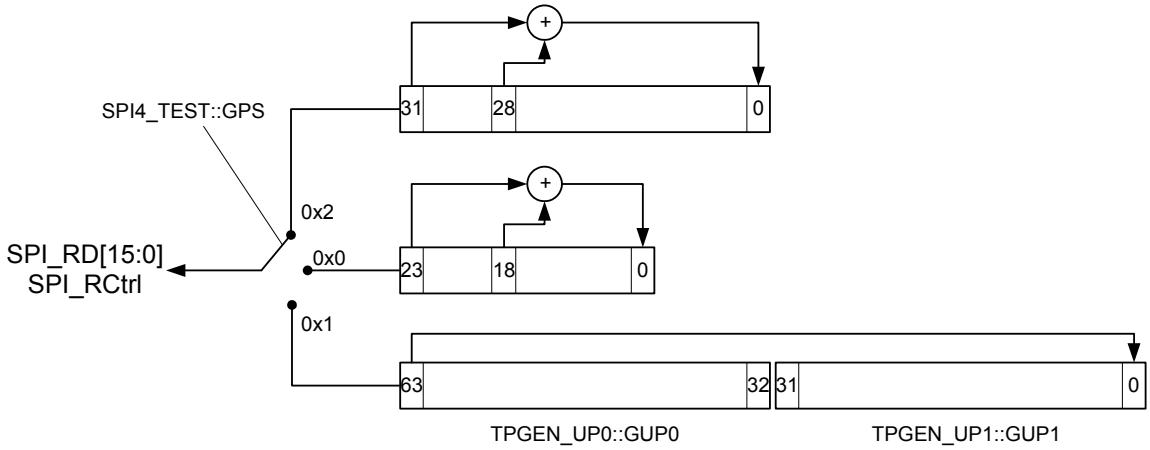


Figure 29. SPI-4.2 Ingress Pattern Generator

The generator transmits a serial stream on SPI_RD0, which is delayed one SPI-4.2 period (one-half of a SPI-4.2 clock cycle) on SPI_RD1, two periods on SPI_RD2, and 15 periods on SPI_RD15. SPI_RCtrl is last and detects the serial stream delayed 16 SPI-4.2 periods. Most standard parallel bit error rate test (BERT) systems should be able to generate or check these sequences.

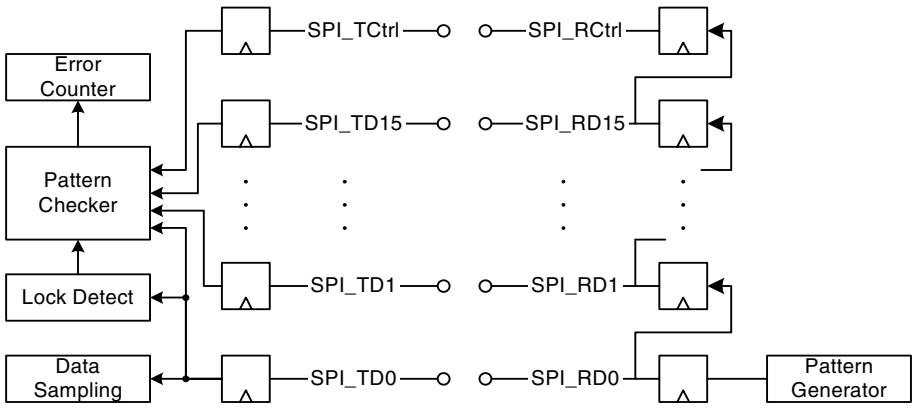


Figure 30. SPI-4.2 Pattern Generator Bit Skewing and Pattern Checker Blocks

The pattern checker resides in the SPI-4.2 egress direction and is located after the dynamic deskew block. It must be programmed to expect the same pattern as the generator is transmitting. Before any checking can be done, the checker must obtain lock. For the PRBS checkers, lock is obtained after seeing 87 correct bits (PRBS-23) or 95 correct bits (PRBS-31). The user pattern checker obtains lock after seeing from 64 to 512 correct bits. Lock is indicated with SPI4_STICKY::PRL_STK for the PRBS checker, and SPI4_STICKY::PAL_STK for the user pattern checker. Lock is achieved by looking at SPI_TD0 only.

Note: The pattern checker can obtain lock on an all-zero pattern, and TPERR_CNT::PCEC starts counting. To ensure lock is obtained on the correct pattern, force the pattern checker out of lock momentarily after applying the test pattern. This can be done using SPI4_TEST::FNL.

When lock is obtained, the error events counter TPERR_CNT::PCEC starts counting. The error events can either be the number of bit errors on a single data line or the number of words where any of the 17 bits do not match. The type of event to be counted is selected by SPI4_TEST::CHECK_LANE.

A data sampling circuitry is associated with the pattern checker. It can sample 64 bits of SPI_TD0. Sampling starts on the falling edge of SPI4_TEST::SP. The sampled bits are read in TPSAM0::SP0 and TPSAM_P1::SP1.

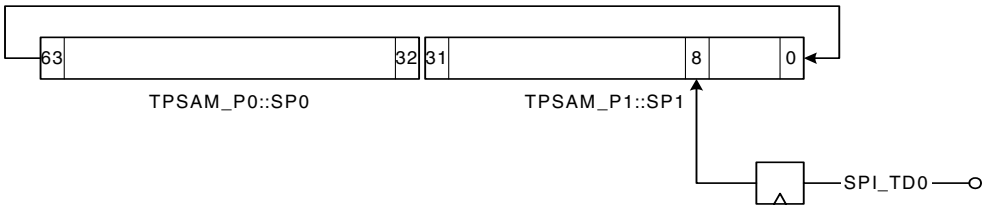


Figure 31. SPI-4.2 Egress Data Sampling Circuitry

2.3 CPU Interfaces

There are two CPU interfaces in the Schaumburg device. The first is a 16-bit parallel interface that can be connected directly to the address and data buses of most CPUs. The second is a serial interface that uses only four wires.

All registers in the Schaumburg device are 32-bits wide. Three types of registers exist: read only, write only, and read/write. Normal configuration registers are read/write types. This type of register is used where read back of the current setting is needed or where a write is needed to clear sticky bits and counters. Very few registers are write only. Write-only registers are used mostly to trigger an event. An example is global soft reset of the Schaumburg device. Read-only registers are used to pass internal status information from the device to the CPU system.

To keep the configuration registers of the Schaumburg well organized, the addresses of the individual registers are divided into three parts as shown in Figure 32. The three parts are described here:

- 3-bit block address points to the overall functional module to which the register belongs.
- 4-bit subblock address points to the submodule underneath it.
- 8-bit address points out the specific register address within the subblock.

For more information about the blocks and subblocks, see "Register Information," page 111.

Block	Subblock	Register Address	LSW/ MSW
A15:13	A12:9	A8:1	A0

Figure 32. Address Mapping

2.3.1 Parallel Interface

The parallel interface has four modes of operation.

- Intel microprocessor type with a two-wire control interface with a shared read/write signal. In two-wire control mode, the PI_nRd pin must be tied to VSS.
- Two-wire control interface with an additional PI_nOE output enable to support multiplexed data and address pins for microprocessors with a limited number of I/O pins.
- PowerPC (Motorola/IBM) microprocessor type with a three-wire control interface (individual read and write signals).
- Three-wire control interface with an additional PI_nOE output enable to support multiplexed data and address pins for microprocessors with a limited number of I/O pins.

Note: The parallel interface addressing is in 16-bit units. Most microprocessors address in 8-bit units. On an 8-bit oriented CPU, the Schaumburg PI_Addr0 must be connected to CPU A1. The remaining bits are addressed in a similar manner.

The registers are distributed to the modules they control and accessed through a serial link, as shown Figure 33. The parallel interface only accesses the local registers (LOCAL_DATA, LOCAL_STATUS) and a local address register directly.

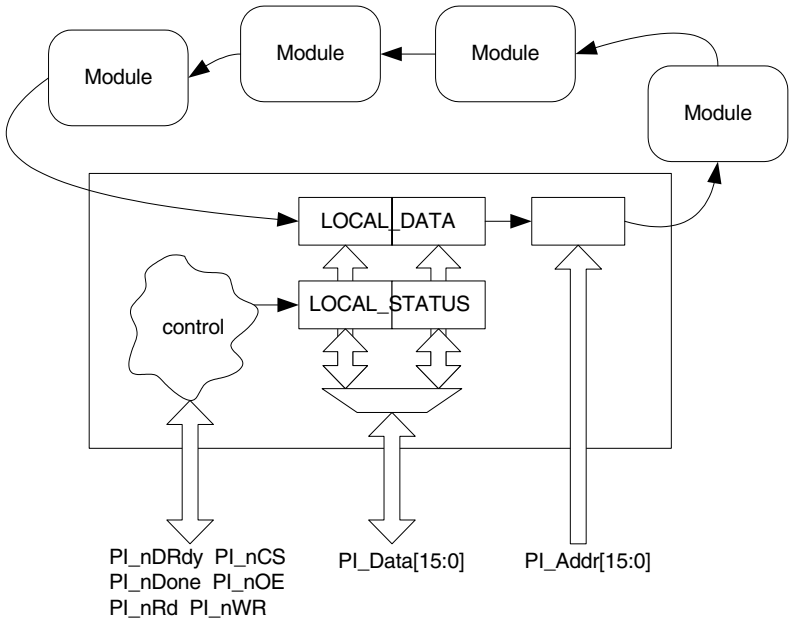


Figure 33. Register Access Through Parallel Interface

2.3.1.1 Write Operations

To write a register in the Schaumburg, the CPU must write both the LSW and MSW of the register in two consecutive accesses to the Schaumburg, as shown in Figure 34. The order of writing LSW and MSW is immaterial. The actual internal access on the serial link is made when both parts of the register have been written. The mapping of the two 16-bit halves into the CPU addresses is dependant on whether little endian or big endian is selected in the parallel interface setup register.

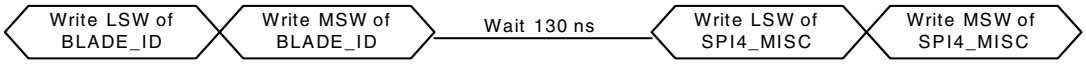


Figure 34. Parallel Interface Register Write

The duration of 130 ns represents the time needed for the parallel interface to transmit the content of the local registers to the serial link. The delay of the serial link is 1.00 µs, but write operations can be spaced closer due to pipelining on the serial link.

2.3.1.2 Read Operations

Due to the long internal access latency on the serial link, the content of a register is read using posted reads. Figure 35 shows a posted read operation using the CHIP_ID register as an example. A more detailed posted read is shown in Figure 70, page 249.

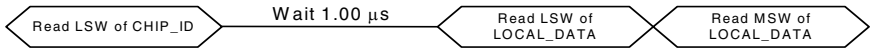


Figure 35. Parallel Interface Register Read

A posted read access is initiated when either the LSW or MSW of a register is read. The result from this 16-bit read operation should be discarded. After 1.00 µs, which is the delay of the internal serial link, the result arrives at the parallel interface.

The value of the requested 32-bit register is transferred to LOCAL_DATA, which is accessible without the use of posted reads. The status from the internal read is put into LOCAL_STATUS. The pin PI_nDRdy and LOCAL_STATUS::RO indicates when the transfer to LOCAL_DATA::DATA has completed. Detection can be made by monitoring the PI_nDRdy pin or by polling LOCAL_STATUS::RO. Because the internal read operation latency is deterministic, waiting until data is known to be ready is also an option. The value of the register can be read by reading both the LSB and MSB of LOCAL_DATA:: DATA. The data output always comes from the LOCAL_DATA or LOCAL_STATUS, and only the two lower bits of the address are decoded to select between them.

Accessing the LOCAL_DATA or LOCAL_STATUS does not initiate a new internal read operation.

2.3.1.3 8-Bit Mode

The parallel interface (PI) can be configured to run in 8-bit mode, allowing it to work with CPUs only equipped with 8-bit interface. This is configured through PI_TRANSFER_SEL::TM0-3. In this mode, all register accesses are split into four bytes accesses. The byte select is not added to the address bus, so in effect ,a register is read through multiple accesses to the same address, and an 8-bit CPU should thus have its address lines 1..16 attached to the Schaumburg.

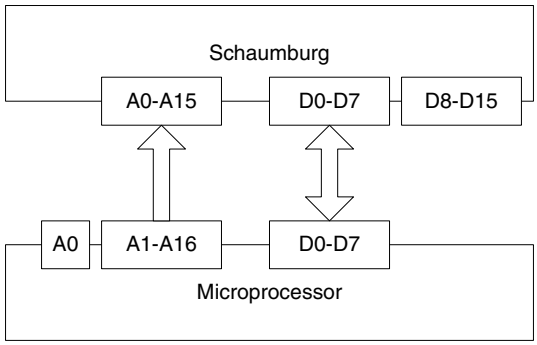


Figure 36. Connecting the Schaumburg to an 8-Bit Microprocessor

As an example, assuming big endian mode, CHIP_ID (block 7, subblock 15, address 0x00) is read as:

Table 39. PI Bus 8-Bit Datawidth Example

Access Number	Schaumburg Address	CPU Bus Address	Data
0	0xfe00	0x1FC00	Most significant byte
1	0xfe00	0x1FC01	Second most significant byte
2	0xfe01	0x1FC02	Second least significant byte
3	0xfe01	0x1FC03	Least significant byte

2.3.2 Serial Interface

The serial interface (SI) is a simple four-wire interface with a clock frequency of up to 24 MHz. It consists of four signal lines: an input clock (SI_Clk), an enable signal (SI_nEN), and a data signal in each direction (SI_DI and SI_DO). SI_DO changes on the negative SI_Clk edge, and SI_DI is sampled on the positive SI_Clk edge.

The relationship of these signals is shown in Figure 37 and Figure 38.

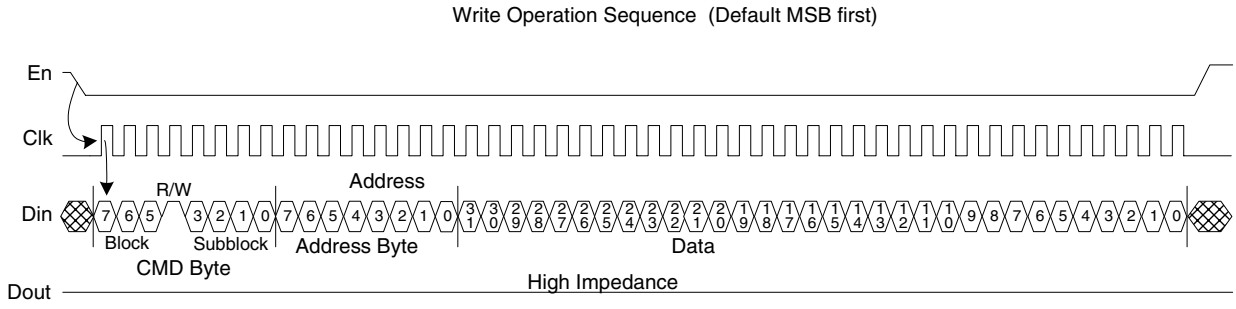


Figure 37. Write Timing Diagram

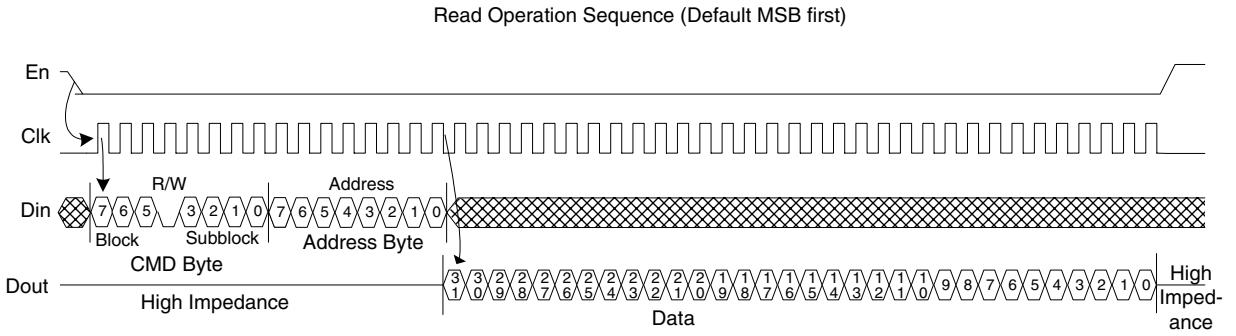


Figure 38. Read Timing Diagram

SI_nEN low starts an operation consisting of 48 clock cycles: 8 for command/block/subblock, 8 for module/address, and 32 for data. In a read operation, SI_DO is driven. Otherwise, it is kept in a tri-state mode. The interface can be configured to run in big endian or little endian mode, and the transmitted bit order is also configurable.

The command format on the serial peripheral interface is listed in Table 40.

Table 40. Serial Interface Format

Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Block			RW	Subblock			
1	Register Address							
2	Data byte 0							
3	Data byte 1							
4	Data byte 2							
5	Data byte 3							

When performing consecutive read/write operations of Nx32 data bits, the address automatically increments. This can accelerate access to consecutive addresses in the memory area.

The Schaumburg can be set up to insert a user-definable number of bytes (0 to 7) between byte 1 and byte 2 in read operations. This is needed if using a serial peripheral interface clock faster than 0.5 MHz. The extra bytes are not used during write operations, and the Schaumburg interprets extra bytes as a block-write operation using the auto-increment-address feature. When block-reading, the extra bytes are inserted between every 32-bit word.

Table 41. Serial Interface Dummy Bytes

Clock Frequency	Number of Dummy Bytes
≤ 0.5 MHz	0
≤ 8 MHz	1
≤ 16 MHz	2
≤ 24 MHz	3

The insertion of dummy bytes is necessary because of latency during register read. When the Schaumburg receives the last address bit, a register read access is initialized. The latency for a read register access is approximately 1 µs.

The clock frequency can be up to 24 MHz. However, dummy bytes must be inserted with a clock frequency higher than 0.5 MHz (giving a delay of at least 1 µs) before reading valid data.

For example, where the clock frequency = 10 MHz (clock cycle time = 0.1 µs), the smallest number of dummy bits before reading valid data is 1 µs / 0.1 µs = 10 bit. In this case, the number of dummy bytes must be set to 2 bytes.

Note: Because the negative clock edge is used for clocking out data from the serial interface, clock frequencies between 0.5 MHz and 1 MHz must also have one dummy byte inserted.

2.4 MII Management Interface

Two separate IEEE Std 802.3 compliant MII management interfaces, also known as Management Data Input/Output (MDIO), are provided for management access to Ethernet PHY devices. Each MII management interface can control up to 32 logical PHY devices, but two separate interfaces are provided for easier board layout to multiple physical PHY devices.

2.4.1 MIIM Physical Interface and Protocol

The MIIM physical interface is a two-wire interface consisting of a data and a clock pin called MDC (clk) and MDIO (data). The MDIO pin is bidirectional pin used for both data in and data out. The direction of the MDIO pin is determined by the protocol as shown in the following figures.

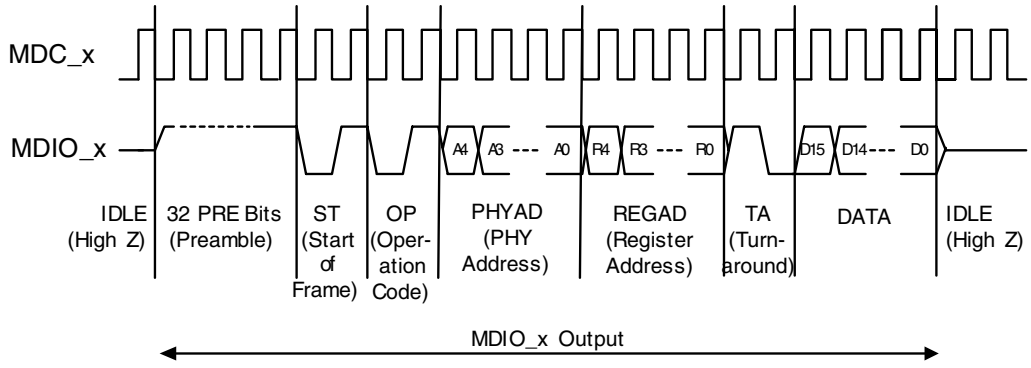


Figure 39. MIIM Protocol, Write PHY

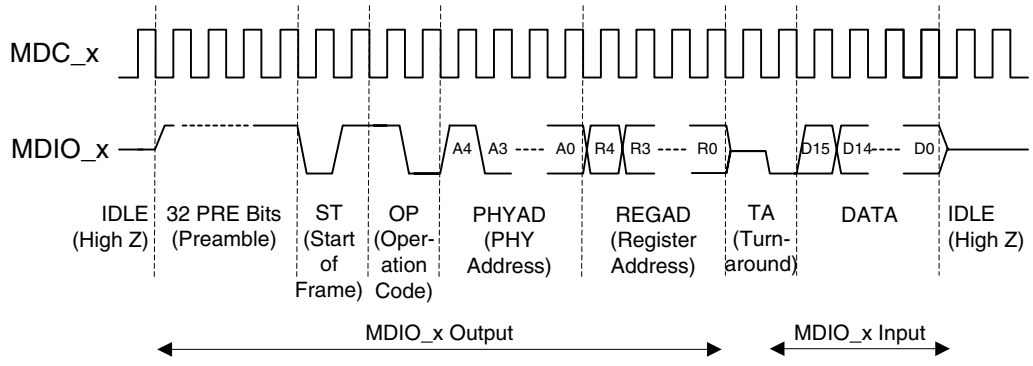


Figure 40. MIIM Protocol, Read PHY

The MII Management Clock (MDC) is running with the frequency configured in the MIIM_PRESCALE register, programmable from 290 kHz to 12.5 MHz. When the MII management interface is idle, the MDIO pin is tri-stated.

Writing to the MIIM_CMD register starts a MIIM access, and the transition at the physical interface starts with a 32 bits preamble sequence, which is used to establish synchronization. After the preamble, the start of a new frame is indicated with a '01' transition.

The sequence of the frame bits is operation code, PHY address, register address, and data, which corresponds to MIIM_CMD::OPR, MIIM_CMD::PHY_ADDR, MIIM_CMD::PHY_REG_ADDR and MIIM_DATA::PHY_DATA.

Data is separated from the register address with two turnaround bits. For a read operation, the first turnaround bit is tri-stated and the second bit, which is driven by the PHY, is '0'. For a write operation, the first turnaround bit is 1 and the second bit is 0.

2.5 General-Purpose I/O – GPIO Interface

The Schaumburg features eight general-purpose LVTTL pins, which may be configured individually as input or output pins. These application-specific pins may, for example, be used to control LEDs or monitor external signals.

The GPIO pins are configured and accessed using the register access gpio_config, gpio_in and gpio_out.

2.6 JTAG Interface

The Schaumburg provides an IEEE Std 1149.1 standard Test Access Port (TAP) and associated circuitry to enable boundary scan testing operations.

2.7 Clock Overview

The Schaumburg has a number of clock inputs, both for clock references and specific interface clocks, as well as clock outputs for specific interfaces. These are summarized in Table 42 through Table 44.

Table 42. Reference Clock Inputs

Clock Input	Description	Range
Core clock reference	Reference for generating the core clock	25 MHz or 125 MHz
SPI-4.2 ingress data reference	Reference for generating the SPI-4.2 ingress data clock	25.00-37.5 MHz or 66.67-125.0 MHz

Table 43. Clock Inputs

Clock Input	Description	Range
RGMII receive clocks	RGMII receive clock, per port.	2.5 MHz, 25 MHz, or 125 MHz
SPI-4.2 egress data clock	SPI-4.2 egress data DDR clock.	75-407 MHz
SPI-4.2 ingress status clock	SPI-4.2 ingress status clock. Sampling phase can be shifted 180º.	0-112 MHz
CPU SI clock	CPU serial interface clock.	0-24 MHz
JTAG	JTAG TAP interface clock.	0-10 MHz

Table 44. Clock Output

Clock Output	Description	Range
SPI-4.2 ingress data clock (SPI_TDClk)	SPI-4.2 ingress data DDR clock. Derived from SPI-4.2 ingress data reference clock by a configurable clock multiplier; can be phase shifted 90°.	75-112.5 MHz, 150-225 MHz, or 300-407 MHz
SPI-4.2 egress status clock (SPI_TSClk)	SPI-4.2 egress status clock. Derived from SPI-4.2 egress data clock (TDClk /4) or from the core clock (75 MHz); can be phase shifted 180°.	18.75-102 MHz
RGMII transmit clocks (RGMII[11:0]_Tx_Clk)	RGMII transmit clock, per port. Derived from the core clock reference.	2.5 MHz, 25 MHz, or 125 MHz
MII management clocks (MDC)	MII management data clock, one per MIIM interface. Derived from the core clock by a configurable prescaler.	0.29-12.5 MHz

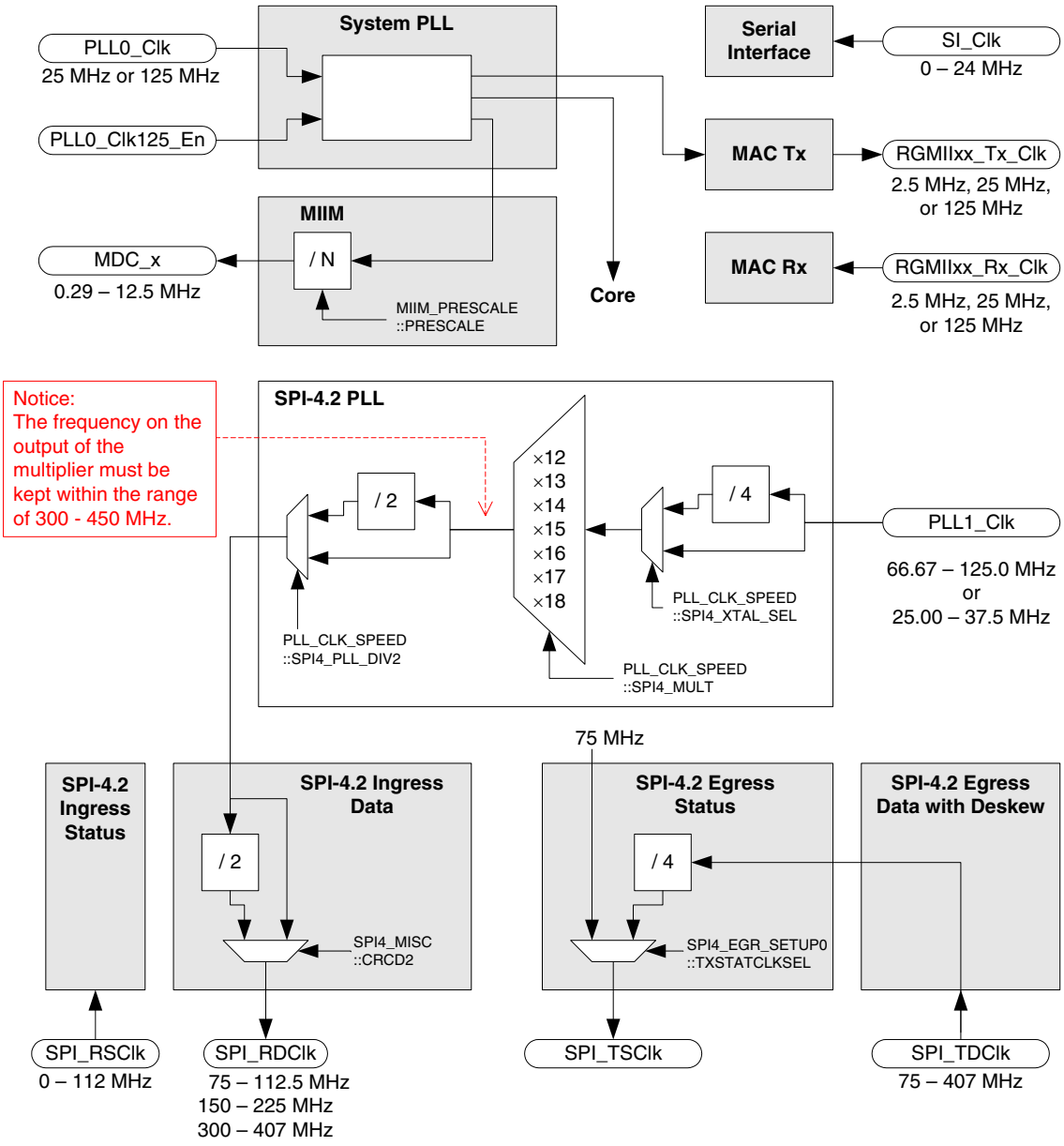


Figure 41. Clock Pins, Internal Clock Distribution, and Associated Configuration Bits

3 APPLICATIONS

3.1 Ethernet Line Card

A typical application for Schaumburg is a 12-port Ethernet line card for a modular switch or router, as depicted in Figure 42. On the line card, Schaumburg typically connects to one or more of the following:

- Copper PHYs and or SerDes and optics using RGMII and MII management
- Host processor using SPI-4.2, typically an ASIC or a network processor (NP) with a backplane interface
- Management CPU through a parallel or serial interface

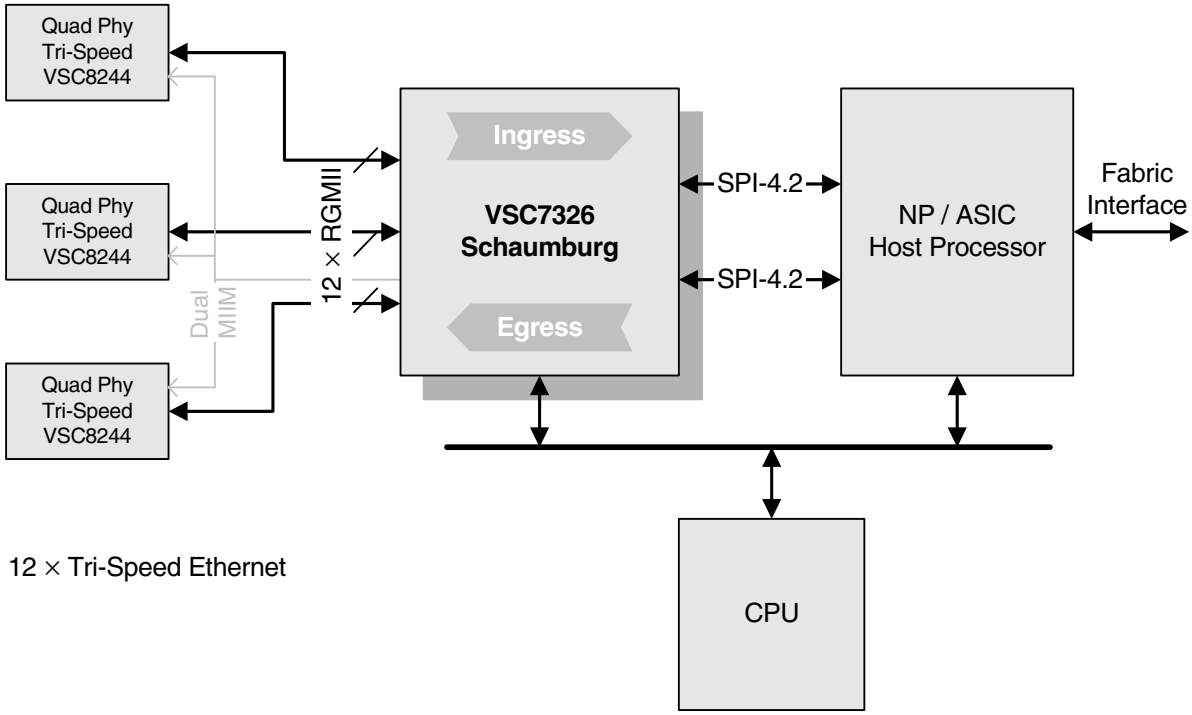


Figure 42. 12-Port Tri-Speed Ethernet Line Card

3.2 Ethernet over SONET Access Card

Schaumburg is well suited for Ethernet over SONET access applications. In the example shown in Figure 43, Schaumburg is connected to:

- SerDes and optics using RGMII and MII management
- SONET framer using SPI-4.2; for instance, the VSC9118 OC-192 virtual concatenation framer
- Board management CPU using parallel or serial interface

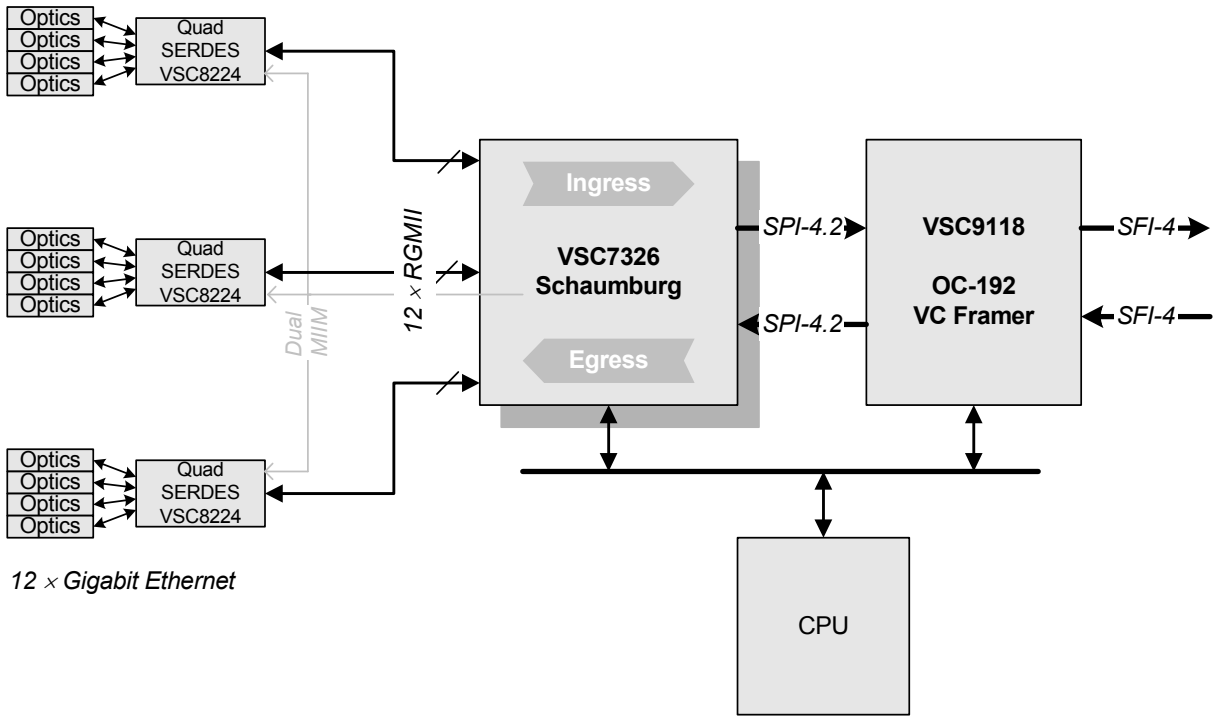


Figure 43. 12-Port Ethernet over SONET Access Card

This application provides a high port count for EOS access. The virtual concatenation enables flexible fine-granularity provisioning of bandwidth between the Ethernet ports. The traffic policing feature in Schaumburg allows an even finer granularity of bandwidth limitations.

The large memory of the Schaumburg device allows the use of Ethernet flow control to enforce the bandwidth, without frame loss, on long Ethernet links. In this case, a large memory is defined as more than a 5 km average at 100 Mbps.

3.3 Flow Control

Schaumburg offers flow control mechanisms on Ethernet and on SPI-4.2, in both directions. Figure 44 provides an overview of the flow control loops related to each port.

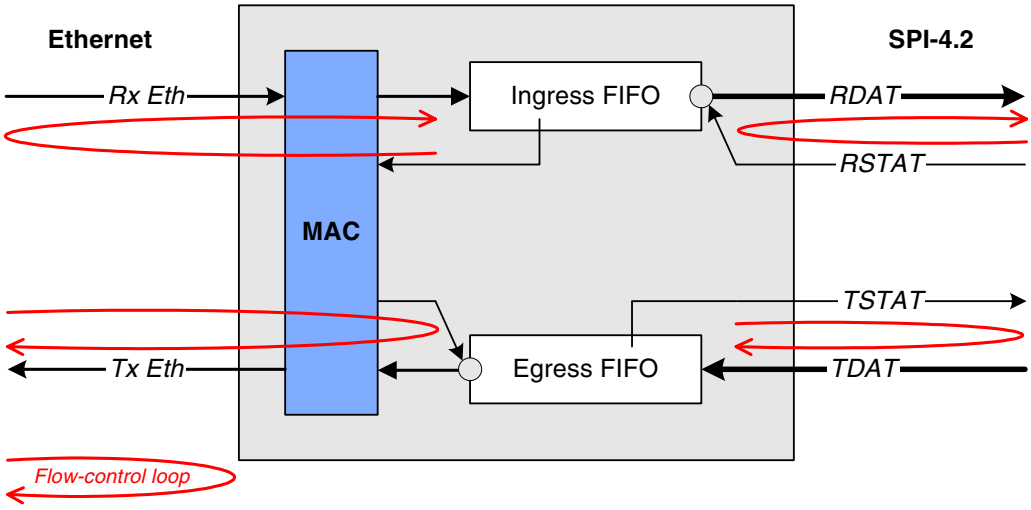


Figure 44. Flow Control Overview

Ethernet flow control is carried out with pause frames in full-duplex mode and with collisions in half-duplex mode. Each port can be independently configured for flow control. For information about configuring the MAC for flow control, see "Ethernet Flow Control," page 34.

SPI-4.2 flow control is an integral part of the SPI-4.2 operation, signalled over the status channels in each direction. For information about the configuration of SPI-4.2 flow control, see "Flow Control," page 86.

3.3.1 Normal Operation

During normal operation within the fill-level controlled mode (set with the host_controlled_mode parameter), the flow control loops function independently on each side of the FIFOs. However, a flow control stop at the output side consequently causes the FIFO to cross the fill level if it continues to receive data and thereby causes flow control on the input side.

Table 45 summarizes causes and actions for each of the flow control loops.

Table 45. Flow Control Loops

Loop	Cause for Push Back	Action
Ethernet, ingress	Exceeded ingress FIFO watermarks	Issue pause frame (full-duplex), or backpressure by collisions (half-duplex)
SPI-4.2, ingress	Received SATISFIED indication	Stop scheduling ingress FIFO
SPI-4.2, egress	Exceeded egress FIFO watermarks	Indicate SATISFIED on the status channel
Ethernet, egress	Received pause frame	Stop scheduling egress FIFO

In the ingress direction, the MAC enters and exits the flow control state depending on the configurable thresholds in the ingress FIFO, known as the High Watermark (HIGH_WM) and the Low Watermark (LOW_WM).

The MAC enters flow control state when the ingress FIFO exceeds the HIGH_WM, and exits flow control state when the FIFO falls below the LOW_WM. The distance between the two watermarks provides a hysteresis that reduces the risk of issuing excess pause frames, while occasionally fluctuating around the watermarks.

A flow control operation, in full-duplex mode, in which no data is lost, is possible, provided that enough memory space is available above the HIGH_WM to accommodate the flow control tail. Here, the term flow control tail refers to the amount of data that may be received until the MAC at the far-end has reacted on a pause frame, and stops the transmission. For more information about setting watermarks, see "High and Low Watermarks," page 40.

In the egress direction, the SPI-4.2 status channel carries the flow control indications based on the egress FIFO watermarks. The function is similar to what happen in ingress, as described in the previous paragraph. When the HIGH_WM is exceeded, the status channel indicates the SATISFIED condition until the amount of traffic in the buffer falls below the LOW_WM. When this occurs, the status channel returns to the HUNGRY or STARVING condition per the configuration.

3.3.2 Host-Controlled Operation

In host-controlled mode, entering and exiting flow control state is signaled to and from the host using the SPI-4.2 status channels. The following table summarizes causes and actions for each of the flow control loops. In principle, host-controlled operation is possible in both egress and ingress direction, but Schaumburg only supports it for the egress direction.

Table 46. Flow Control Loops in Host-Controlled Mode

Loop	Cause for Push Back	Action
Egress	Received pause frame	Indicate SATISFIED on the SPI-4.2 status channel

In the egress direction, the MAC signals the flow control state to the host by indicating the SATISFIED condition for that port. The flow control state is entered when a pause frame is received, and the state is

maintained for the length of the pause specified in the pause frame or until a pause frame is received with the value zero, whichever occurs first.

In host-controlled mode, the flow control on the egress SPI-4.2 interface is overlaid by the flow control state, in the sense that the SATISFIED condition may be signaled due to the flow control state, as well as the fill level of the FIFO.

The host control mode can be enabled for the egress direction by configuring the host_controlled_mode parameter.

3.3.3 Flow Control Latency

The full-duplex Ethernet flow control loop has an inherent latency from pause frame scheduling, transmission, and reaction time. This results in a flow control tail at XOFF. In this case, the term flow control tail refers to the amount of data that may arrive after crossing the HIGH_WM. Similarly, the flow control latency at XON causes a delay from the moment the LOW_WM is crossed until the first data is received.

Using pause frame flow control, it is possible to guarantee an operation where no data is lost in the ingress direction — provided that the FIFOs are configured to accommodate the flow control tail. The FIFOs should also be configured to hold enough data to prevent underflow when the pause is over.

Table 47 shows the details of the flow control latency contributions. The contributions depend on the link length (LEN) and the maximum frame size (MTU). The link speed is assumed to be 1 Gbps.

Table 47. Ingress Flow Control Latency Contributions

Ethernet Flow Control Latency	Latency at XOFF Pause Begins	Latency at XON Pause Ends
MAC latency – FIFO threshold to Tx data	128B	128B
Worst case Tx latency – Wait for current frame to end	1 × MTU	1 × MTU
Tx latency – Transmit pause frame	84B	84B
Link delay – 5 bit/m wire, 5.5 bit/m fiber at 1 Gbps	0.7B × LEN	0.7B × LEN
MAC, far end – Maximum respond time	128B	128B
Tx, far end – Wait for current frame to end (XOFF only)	1 × MTU	
Link delay – 5 bit/m wire, 5.5 bit/m fiber at 1 Gbps	0.7B × LEN	0.7B × LEN
Rx latency – Rx to FIFO	44B	44B
Required above HIGH_WM / below LOW_WM	384B + 1.4B × LEN + 2 × MTU	384B + 1.4B × LEN + 1 × MTU

3.4 Watermark Calculations

3.4.1 Ingress High and Low Watermarks

The settings of the watermarks for full-duplex operation depend on the requirements for performance when flow control is enabled. These requirements include the following:

- Wire-speed must operate without activating excess flow control. (No waste)
- Wire-speed must operate without underflow after pauses. (No underflow)
- Lossless flow control operation must be active. (No overflow)

The sizes of these contributions depend on numerous parameters, primarily the data transfer mode (cut-through, store-and-forward, full-frame mode), the maximum frame size (MTU), and the link speed and link length (LEN). These metrics are described in Table 48 and Table 49.

You can avoid wasting bandwidth by triggering flow control under normal operating conditions in situations where there is no congestion. To do this, the HIGH_WM must be set high enough to accommodate the normal fluctuations resulting from the scheduling of frames out of the FIFO.

Table 48. Ingress HIGH_WM Calculations

Contributions to Avoid Excess Flow Control	Cut-Through	Store-and-Forward	Full-Frame
SPI-4.2 scheduling margin, 4 × SPI-4.2 burst size	4 × Burst	4 × Burst	4 × Burst
Cut-through or store-and-forward buffering	CUT_THRU	1 × MTU	1 × MTU
Full-frame scheduling latency of 12 ports on SPI-4.2			1.2 × MTU
Required below HIGH_WM	4 × Burst + CUT_THRU	4 × Burst + 1 × MTU	4 × Burst + 2.2 × MTU

To avoid overflow and underflow in the flow control loop, there must be enough buffer space above the HIGH_WM and below the LOW_WM. For more information, see "Flow Control Latency," page 105.

The distance between the LOW_WM and HIGH_WM must be at least one kilobyte to reduce bandwidth consumption by pause frames on the Ethernet egress line.

Figure 49 provides examples of the watermark calculations. Each example assumes 12 x 1 Gbps configuration with lossless flow control and equal size buffers.

Table 49. Ingress HIGH_WM and LOW_WM Examples

Parameter	Example 1	Example 2	Example 3
Mode	Store-and-forward	Store-and-forward	Full-frame
Max frame size, MTU	1522 B	9600 B	9600 B
Link length, LEN	12 km	550 m	550 m
Ingress Memory and Watermarks			
Total per port	43008 B	43008 B	43008 B
Above HIGH_WM	20228 B	20354 B	20354 B
HIGH_WM	22784 B	22656 B	22656 B
Available margin HIGH_WM-LOW_WM	4060 B	11870 B	2046 B
LOW_WM	18720 B	10784 B	20608 B
LOW_WM — wire-speed operation – no underflow	1522 B 18706 B	9088 B 10754 B	20608 B 10754 B

The watermarks are configured through the fifo_high_watermark and fifo_low_watermark parameters.

3.4.2 Egress High and Low Watermarks

The egress watermark calculations are somewhat simpler than the ingress watermark calculations because there are fewer options. This is because overflow and underflow are considered unacceptable over an SPI-4.2 interface.

To avoid asserting flow control in normal operation, the HIGH_WM must be set high enough to accommodate the normal fluctuations from the host's scheduling of data over the SPI-4.2 interface. Assuming that this scheduling is no worse than a round-robin scheduling of bursts or frames, this configuration is also sufficient to avoid underflow.

Table 50. Egress HIGH_WM Calculation for Wire-Speed Operation

Contributions to Avoid Excess Flow Control	Cut-Through	Store-and-Forward Full-Frame
SPI-4.2 scheduling margin	4 × Burst	4 × Burst
Cut-through or store-and-forward buffering	CUT_THRU	1 × MTU
Required below HIGH_WM	4 × Burst + CUT_THRU	4 × Burst + 1 × MTU

To avoid overflow, there must be enough buffer space above the HIGH_WM to accommodate the amount of data that may arrive after crossing the HIGH_WM (due to the latency in the SPI-4.2 flow control and scheduling loop). This is specified in Table 51.

Table 51. Egress Flow Control Latency Contributions

Contributions to Avoid Overflow	Cut-Through Store-and-Forward	Full-Frame
Host L_max – reaction time in the host (1µs)	100 B	1600 B
SPI-4.2 buffers – internal pipelines etc.	500 B	400 B
SPI-4.2 scheduling unit	MaxBurst	1 × MTU
Required above HIGH_WM	600 B + MaxBurst	2000 B + 1 × MTU

The data flow can be stopped in the middle of a frame. This can cause a deadlock situation if the threshold is programmed incorrectly. A deadlock occurs if the buffer fill level is above LOW_WM, but the buffer does not initiate frame transmission because it is in one of the following conditions:

- In store-and-forward mode and the frame has not been completely received.
- In cut-through mode but the fill level is below CUT_THRU.

To avoid this situation, LOW_WM must be set sufficiently high, as specified by the "Required to avoid deadlock" row in Table 52.

Table 52. Egress LOW_WM Calculations

Contributions	Cut-Through	Store-and-Forward	Full-Frame
Required to avoid deadlock	CUT_THRU	1 × MTU	0 B
Required to avoid MAC underrun due to host L_max reaction time (1 µs)	128 B	128 B	128 B
Required below LOW_WM	max(CUT_THRU,128B)	1 × MTU	128 B

LOW_WM and HIGH_WM can be set to the same value. Hysteresis is not required for SPI-4.2 flow control, because unlike Ethernet flow control, changing SPI-4.2 flow control state does not consume bandwidth.

Table 53 exemplifies the watermark calculations. Both examples assume 12 × 1 Gbps configuration with lossless flow control and equal size buffers. A MaxBurst of 256 B is assumed.

Table 53. Egress HIGH_WM and LOW_WM Calculation Examples

Parameter	Example 1	Example 2	Example 3
Mode	Store-and-forward	Store-and-forward	Full-frame
Maximum frame size, MTU	1522 B	9600 B	9600 B
Active ports	12	12	12
Ingress Memory and Watermarks			
Total per port	12288 B	12288 B	24576 B
Above HIGH_WM	856 B	856 B	11600 B

Table 53. Egress HIGH_WM and LOW_WM Calculation Examples *(continued)*

Parameter	Example 1	Example 2	Example 3
HIGH_WM (see Table 51)	11456 B	11456 B	12992 B
Available margin HIGH_WM-LOW_WM	9896 B	1832 B	3376 B
LOW_WM (see Table 52)	1536 B	9600 B	9600 B
LOW_WM — wire-speed operation	1522 B	9600 B	9600 B

Example 3 in Table 53 indicates that the combination of full-frame mode and jumbo frames consumes a significant amount of egress memory. Adequate memory is available for 12 ports with this combination.

4 Register Information

Table 54. Tri-Speed MAC, Block 1, Subblock 0–11

Address	Register Name	Short Name
0x00	"Mode Configuration," page 119	MODE_CFG
0x01	"Pause Configuration," page 120	PAUSE_CFG
0x02	"Maximum Length Configuration," page 121	MAXLEN_CFG
0x03	"MAC Address Configuration, Bit 47:24," page 121	MAC_ADDR_HIGH_CFG
0x04	"MAC Address Configuration, Bit 23:0," page 122	MAC_ADDR_LOW_CFG
0x05	"Tri-Speed MAC Frame Normalizer," page 122	NORMALIZER
0x0A	"Sticky Bits," page 123	STICK_BIT
0x0B	"Tri-Speed MAC Clock/Reset Setup," page 125	DEV_SETUP
0x0C	"Drop Counter," page 126	DROP_CNT
0x0E	"Tri-Speed MAC Debug," page 127	DEV_DEBUG
0x15	"Frame Denormalization," page 127	DENORM
0x16	"Device 1 G Debug," page 128	DEV_DBG
0x18	"TX Inter-Frame Gap Configuration," page 129	TX_IFG
0x19	"Advanced Half-Duplex Configuration," page 129	ADV_HDX_CFG

Table 55. Ingress FIFO Buffer, Block 2, Subblock 0

Address	Register Name	Short Name
0x00	"Mode and Test," page 130	ING_TEST
0x10	"FIFO Buffer Top and Bottom," page 134	ING_TOP_BOTTOM
0x20	"Write Pointer," page 135	ING_TAIL
0x30	"Read Pointer," page 135	ING_HEAD
0x40	"Flow Control Watermarks," page 135	ING_HIGH_LOW_WM
0x50	"Cut-Through Threshold," page 136	ING_CT_THRHLD
0x60	"Drop and CRC Error Counter," page 137	ING_DROP_CNT
0x70	"Input Side Debug Classification Control," page 137	ING_DEBUG_BUF_CNT
0x0C	"Ingress Layer 3-4 Classification Control," page 137	ING_CL3_CTRL
0x1C	"Ingress Layer 3-4 Classification Layer 3 Field," page 138	ING_CL3_L3
0x2C	"Ingress Layer 3-4 Classification Layer 4 Fields," page 138	ING_CL3_L4
0x0D	"Ingress Unicast and Multicast Filtering Enable," page 138	ING_FFILT_UM_EN
0x1D	"Ingress Broadcast and EtherType Filtering Enable," page 139	ING_FFILT_BE_EN
0x2D	"Ingress Value Filter Address," page 140	ING_FFILT_VAL0

Table 55. Ingress FIFO Buffer, Block 2, Subblock 0 *(continued)*

Address	Register Name	Short Name
0x3D	"Ingress Value Filter Address," page 140	ING_FFILT_VAL1
0x4D	"Ingress Mask Filter Address," page 140	ING_FFILT_MASK0
0x5D	"Ingress Mask Filter Address," page 140	ING_FFILT_MASK1
0x6D	"Ingress Mask Filter Address," page 140	ING_FFILT_MASK2
0x7D	"Ingress EtherType Filter Values," page 141	ING_FFILT_ETYPE
0x0E	"SRAM_ADDR," page 142	ING_SRAM_ADDR
0x1E	"SPRAM_WR_STRB," page 143	ING_SRAM_WR_STRB
0x2E	"SRAM_RD_STRB," page 143	ING_SRAM_RD_STRB
0x3E	"SRAM_DATA_0," page 144	ING_SRAM_DATA_0
0x4E	"SRAM_DATA_1," page 144	ING_SRAM_DATA_1
0x5E	"SRAM_DATA_2," page 144	ING_SRAM_DATA_2
0x6E	"SRAM_DATA_3," page 144	ING_SRAM_DATA_3
0x7E	"SRAM_DATA_BLK_TYPE," page 145	ING_SRAM_DATA_BLK_TYPE
0x0F	"Ingress FIFO Master Control," page 145	ING_CONTROL
0x1F	"Ingress Aging Timer," page 147	ING_AGE_TIMER
0x2F	"Ingress Aging Timer Increment," page 147	ING_AGE_INC
0x3F	"Ingress Output Side Debug Counter Control," page 148	ING_DEBUG_OUT
0x4F	"Output Side Debug Counter," page 148	ING_DEBUG_CNT
0x5F	"Per Port Inhibit Control," page 148	PPORT_INH
0x6F	"Classifier Control," page 149	CLASSIFIER_CTRL

Table 56. Egress FIFO Buffer, Block 2, Subblock 1

Address	Register Name	Short Name
0x00	"Mode and Test," page 151	EGR_TEST
0x10	"FIFO Buffer Top and Bottom," page 154	EGR_TOP_BOTTOM
0x20	"Write Pointer," page 155	EGR_TAIL
0x30	"Read Pointer," page 155	EGR_HEAD
0x40	"Flow Control Watermarks," page 155	EGR_HIGH_LOW_WM
0x50	"Cut-Through Threshold," page 156	EGR_CT_THRHLD
0x60	"Drop Counter," page 157	EGR_DROP_CNT
0x70	"Input Side Debug Counter," page 157	EGR_DEBUG_BUF_CNT
0x0E	"SRAM_ADDR," page 158	EGR_SRAM_ADDR
0x1E	"SRAM_WR_STRB," page 158	EGR_SRAM_WR_STRB

Table 56. Egress FIFO Buffer, Block 2, Subblock 1 *(continued)*

Address	Register Name	Short Name
0x2E	"SRAM_RD_STRB," page 159	EGR_SRAM_RD_STRB
0x3E	"SRAM_DATA_0," page 159	EGR_SRAM_DATA_0
0x4E	"SRAM_DATA_1," page 159	EGR_SRAM_DATA_1
0x5E	"SRAM_DATA_2," page 160	EGR_SRAM_DATA_2
0x6E	"SRAM_DATA_3," page 160	EGR_SRAM_DATA_3
0x7E	"SRAM_DATA_BLK_TYPE," page 160	EGR_SRAM_DATA_BLK_TYPE
0x0F	"Egress FIFO Master Control," page 160	EGR_CONTROL
0x1F	"Egress Aging Timer," page 162	EGR_AGE_TIMER
0x2F	"Egress Aging Timer Increment," page 163	EGR_AGE_INC
0x3F	"Egress Output Side Debug Counter Control," page 163	EGR_DEBUG_OUT
0x4F	"Output Side Debug Counter," page 164	EGR_DEBUG_CNT
0x5F	"Per Port Inhibit Control," page 164	PPORT_INH

Table 57. Egress Shaper and Ingress Policer, Block 2, Subblock 2–3

Address	Register Name	Short Name
0x00	"Traffic Shaper Control," page 164	SHAPER_CONTROL
0x01	"Shaper Priority," page 166	SHAPER_PRIO
0x02	"Timer," page 166	SHAPER_TIMER
0x03	"Clear Buckets," page 167	SHAPER_CLR_BUCKETS
0x04	"Traffic Slow Rate Control," page 167	SHAPER_SLOW_RATE_CONTROL
0x20	"Traffic Shaper Bucket Setting," page 168	SHAPER_BUCKET
0x38	"Common Traffic Shaper Bucket Setting," page 168	COMMON_SHAPER_BUCKET

Table 58. MII-Management, Block 3, Subblock 0–1

Address	Register Name	Short Name
0x00	"MII-M Status," page 169	MIIM_STATUS
0x01	"MII-M Command," page 170	MIIM_CMD
0x02	"MII-M Data," page 170	MIIM_DATA
0x03	"MII-M MDC Pre-scale," page 171	MIIM_PRESCALE

Table 59. Statistic Counters, Block 4, Subblock 0–11

Address	Register Name	Short Name
0x00	"Rx Byte Counter," page 171	RX_IN_BYTES_CNT
0x01	"Rx Symbol Carrier Error Counter," page 171	RX_SYMBOL_CARRIER_ERR_CNT
0x02	"Rx Pause Frame Counter," page 172	RX_PAUSE_CNT
0x03	"Rx Control Frame Counter," page 172	RX_UNSUP_OPCODE_CNT
0x04	"Rx OK Byte Counter," page 172	RX_OK_BYTES_CNT
0x05	"Rx Bad Byte Counter," page 172	RX_BAD_BYTES_CNT
0x06	"Rx Unicast Frame Counter," page 172	RX_UNICAST_CNT
0x07	"Rx Multicast Frame Counter," page 173	RX_MULTICAST_CNT
0x08	"Rx Broadcast Frame Counter," page 173	RX_BROADCAST_CNT
0x09	"Rx CRC Error Counter," page 173	RX_CRC_ERR_CNT
0x0A	"Rx Alignment Error Counter," page 173	RX_ALIGNMENT_ERR_CNT
0x0B	"Rx Undersize Counter (Valid Frame Format)," page 173	RX_UNDERSIZE_CNT
0x0C	"Rx Undersize Counter (CRC Error)," page 174	RX_FRAGMENTS_CNT
0x0D	"Rx In-Range Length Error Counter," page 174	RX_IN_RANGE_LENGTH_ERR_CNT
0x0E	"Rx Out-Of-Range Length Error Counter," page 174	RX_OUT_OF_RANGE_ERR_CNT
0x0F	"Rx Oversize Counter (Valid Frame Format)," page 174	RX_OVERSIZE_CNT
0x10	"Rx Jabbers Counter," page 175	RX_JABBERS_CNT
0x11	"Rx 64 Byte Frame Counter," page 175	RX_SIZE64_CNT
0x12	"Rx 65 to 127 Byte Frame Counter," page 175	RX_SIZE65TO127_CNT
0x13	"Rx 128 to 525 Byte Frame Counter," page 175	RX_SIZE128TO255_CNT
0x14	"Rx 256 to 511 Byte Frame Counter," page 175	RX_SIZE256TO511_CNT
0x15	"Rx 512 to 1023 Byte Frame Counter," page 176	RX_SIZE512TO1023_CNT
0x16	"Rx 1024 to 1518 Byte Frame Counter," page 176	RX_SIZE1024TO1518_CNT
0x17	"Rx 1519 To Maximum Length Byte Frame Counter," page 176	RX_SIZE1519TOMAX_CNT
0x18	"Tx Byte Counter," page 176	TX_OUT_BYTES_CNT
0x19	"Tx Pause Frame Counter," page 176	TX_PAUSE_CNT
0x1A	"Tx OK Byte Counter," page 177	TX_OK_BYTES_CNT
0x1B	"Tx Unicast Frame Counter," page 177	TX_UNICAST_CNT
0x1C	"Tx Multicast Frame Counter," page 177	TX_MULTICAST_CNT
0x1D	"Tx Broadcast Frame Counter," page 177	TX_BROADCAST_CNT
0x1E	"Tx Multiple Collision Frame Counter," page 177	TX_MULTI_COLL_CNT
0x1F	"Tx Late Collision Counter," page 178	TX_LATE_COLL_CNT
0x20	"Tx Excessive Collision Counter," page 178	TX_XCOLL_CNT
0x21	"Tx First Defer Counter," page 178	TX_DEFER_CNT

Table 59. Statistic Counters, Block 4, Subblock 0–11 *(continued)*

Address	Register Name	Short Name
0x22	"Tx Excessive Defer Counter," page 178	TX_XDEFER_CNT
0x23	"Tx Carrier Sense Error Counter," page 178	TX_CSENSE_CNT
0x24	"Tx 64 Byte Frame Counter," page 179	TX_SIZE64_CNT
0x25	"Tx 65 to 127 Byte Frame Counter," page 179	TX_SIZE65TO127_CNT
0x26	"Tx 128 to 255 Byte Frame Counter," page 179	TX_SIZE128TO255_CNT
0x27	"Tx 256 to 511 Byte Frame Counter," page 179	TX_SIZE256TO511_CNT
0x28	"Tx 512 to 1023 Byte Frame Counter," page 179	TX_SIZE512TO1023_CNT
0x29	"Tx 1024 to 1518 Byte Frame Counter," page 180	TX_SIZE1024TO1518_CNT
0x2A	"Tx 1519 to Maximum Length Byte Frame Counter," page 180	TX_SIZE1519TOMAX_CNT
0x2B	"Tx Single Collision Counter," page 180	TX_SINGLE_COLL_CNT
0x2C	"Tx 2 Backoff Counter," page 180	TX_BACKOFF2_CNT
0x2D	"Tx 3 Backoff Counter," page 180	TX_BACKOFF3_CNT
0x2E	"Tx 4 Backoff Counter," page 181	TX_BACKOFF4_CNT
0x2F	"Tx 5 Backoff Counter," page 181	TX_BACKOFF5_CNT
0x30	"Tx 6 Backoff Counter," page 181	TX_BACKOFF6_CNT
0x31	"Tx 7 Backoff Counter," page 181	TX_BACKOFF7_CNT
0x32	"Tx 8 Backoff Counter," page 181	TX_BACKOFF8_CNT
0x33	"Tx 9 Backoff Counter," page 182	TX_BACKOFF9_CNT
0x34	"Tx 10 Backoff Counter," page 182	TX_BACKOFF10_CNT
0x35	"Tx 11 Backoff Counter," page 182	TX_BACKOFF11_CNT
0x36	"Tx 12 Backoff Counter," page 182	TX_BACKOFF12_CNT
0x37	"Tx 13 Backoff Counter," page 182	TX_BACKOFF13_CNT
0x38	"Tx 14 Backoff Counter," page 183	TX_BACKOFF14_CNT
0x39	"Tx 15 Backoff Counter," page 183	TX_BACKOFF15_CNT
0x3A	"Tx Underflow Counter," page 183	TX_UNDERRUN_CNT
0x3C	"Rx Inter-Packet Gap Shrink Counter," page 183	RX_IPG_SHRINK_CNT
0x3E	"Statistics Tri-Speed MAC Sticky Bit," page 183	STAT_STICKY1G
0x3F	"Initialize Statistics," page 184	STAT_INIT

Table 60. SPI-4.2 Host Interface, Block 5, Subblock 0

Address	Register Name	Short Name
0x00	"Master Configuration," page 185	SPI4_MISC
0x01	"CML Status," page 187	SPI4_STATUS

Table 60. SPI-4.2 Host Interface, Block 5, Subblock 0 *(continued)*

Address	Register Name	Short Name
0x02	"Ingress Status Channel Setup," page 188	SPI4_ING_SETUP0
0x03	"Ingress Data Training Setup," page 189	SPI4_ING_SETUP1
0x04	"Ingress Data Burst Sizes Setup," page 189	SPI4_ING_SETUP2
0x05	"Egress Status Channel Setup," page 190	SPI4_EGR_SETUP0
0x08	"Egress Status Port Setup," page 191	SPI4_EGR_STAT_PORT_SETUP
0x09	"Ingress Status Port Setup," page 192	SPI4_INGR_STAT_PORT_SETUP
0x0A	"Ingress Effective FIFO Size," page 192	SPI4_INGR_EFF_FIFO_SIZE
0x0B	"Ingress Invert Bits," page 192	SPI4_INGR_INV_BITS
0x0C	"Ingress Control Mask," page 192	SPI4_INGR_CNTR_MASK
0x10	"SPI-4.2 Debug Counter 0," page 193	C0
0x11	"SPI-4.2 Debug Counter 1," page 193	C1
0x12	"SPI-4.2 Debug Counter 2," page 193	C2
0x13	"SPI-4.2 Debug Counter 3," page 193	C3
0x14	"SPI-4.2 Debug Counter 4," page 193	C4
0x15	"SPI-4.2 Debug Counter 5," page 194	C5
0x16	"SPI-4.2 Debug Counter 6," page 194	C6
0x17	"SPI-4.2 Debug Counter 7," page 194	C7
0x18	"SPI-4.2 Debug Counter 8," page 194	C8
0x1A	"Debug Counters Setup," page 195	SPI4_DBG_SETUP
0x20	"Test Setup," page 198	SPI4_TEST
0x21	"Test Pattern Generator User Pattern0," page 200	TPGEN_UP0
0x22	"Test Pattern Generator User Pattern1," page 200	TPGEN_UP1
0x23	"Test Pattern Checker User Pattern0," page 200	TPCHK_UP0
0x24	"Test Pattern Checker User Pattern1," page 201	TPCHK_UP1
0x25	"Sampled Pattern0," page 201	TPSAM_P0
0x26	"Sampled Pattern1," page 201	TPSAM_P1
0x27	"Pattern Checker Error Counter," page 201	TPERR_CNT
0x30	"Sticky Bits Register 0," page 202	SPI4_STICKY
0x31	"Sticky Bits Register 1," page 203	SPI4_CRDT_STICKY
0x33	"Ingress Status Channel Granted Credit Value," page 204	SPI4_DBG_GRANT
0x34	"Core Egress Inhibit," page 204	SPI4_DBG_EGR_INH
0x35	"Core Ingress Inhibit," page 204	SPI4_DBG_INGR_INH
0x36	"Sampled Ingress Status Channel 1," page 204	SPI4_DBG_STATUS1
0x37	"Sampled Ingress Status Channel 2," page 205	SPI4_DBG_STATUS2

Table 60. SPI-4.2 Host Interface, Block 5, Subblock 0 *(continued)*

Address	Register Name	Short Name
0x40	"SPI-4.2 Deskew Control Mode," page 206	SPI4_DSKW_CTRL_MODE
0x4B	"SPI-4.2 Deskew Control DIP-4 Error Threshold," page 207	SPI4_DSKW_CTRL_DIP4_ERR_THRS
0x4C	"SPI-4.2 Deskew Control DIP-4 Window," page 208	SPI4_DSKW_CTRL_DIP4_WINDOW
0x4D	"SPI-4.2 Deskew Control Synchronization," page 208	SPI4_DSKW_CTRL_SYNC

Table 61. BIST Access Registers, Block 7, Subblock 1

Address	Register Name	Short Name
0x00	"RAM BIST Command," page 209	RAM_BIST_CMD
0x01	"RAM BIST Read Status and Read Result," page 210	RAM_BIST_RESULT

Table 62. Indirect BIST Registers

Address	Register Name	Short Name
0x00	"Wrapper Control Module Configuration," page 210	WCM_CFG
0x01	"Wrapper Control Module Command," page 211	WCM_CMD_CTRL
0x02	"Wrapper Control Module Status," page 211	WCM_STATUS

Table 63. Primary System Control, Block 7, Subblock 15

Address	Register Name	Short Name
0x00	"Chip ID," page 213	CHIP_ID
0x01	"Blade ID," page 213	BLADE_ID
0x02	"Global Soft Reset," page 213	SW_RESET
0x04	"Memory Control Register," page 214	MEMLOCK_CTRL
0x07	"Interface Mode," page 214	IFACE_MODE
0x0F	"SI Insert Bytes," page 214	SI_INSERT_BYTES
0x18	"SI Transfer Select," page 215	SI_TRANSFER_SEL
0x19	"Clock Speed Selection," page 215	PLL_CLK_SPEED
0x1D	"GPIO Control," page 217	GPIO_CTRL
0x1E	"GPIO Output," page 217	GPIO_OUT
0x1F	"GPIO Input," page 218	GPIO_IN
0x20	"Parallel CPU Interface Transfer Select," page 218	PI_TRANSFER_SEL

Table 64. Secondary System Control, Block 7, Subblock 2

Address	Register Name	Short Name
0x03	"Host Interface Select," page 219	HOST_INF_SELECT
0x06	"Master Scheduler Configuration," page 219	MSCH
0x08	"Master Scheduler Sync Clear," page 221	MSCH_SYNC_CLR
0x09	"Master Scheduler Status," page 221	MSCH_STAT
0x0A	"Egress CRC Error Count," page 221	EGR_CRC_CNT
0x0B	"Egress CRC Checker Configuration," page 221	CRC_CFG
0x21	"Ingress Port Remapper," page 222	REMAP_ING
0x22	"Egress Port Remapper," page 223	REMAP_EGR

Table 65. PI Local Registers, Block 7, Subblock 15

Address	Register Name	Short Name
0xFE	"Local PI Data," page 223	LOCAL_DATA
0xFF	"Local PI Status," page 223	LOCAL_STATUS

4.1 Register Details

Unspecified fields in the following tables always return a value of 0 when read. When writing, they should always be set to 0. The mode column of each table shows the access to the field as follows:

R/W	Read and Write
R/O	Read only
W/O	Write only (read dummy)

4.2 Tri-Speed MAC, Block 1

The ports are indexed with the subblock number.

4.2.1 Mode Configuration

Table 66. MODE_CFG (Address 0x00), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
25	WEXC_DIS	R/W	Internal debug use. Do not change default value.	0x0
24	OB_EN	R/W	Internal debug use. Do not change default value.	0x0
23:16	SEED	R/W	Internal debug use. Do not change default value.	0x000
15	LEN_DROP	R/W	Configures the MAC receiver to drop frames with in-range and out-of-range errors. These are errors where the Ethernet Type/Len field designates a length, and this length is either illegal (out-of-range error) or does not match the actual frame length (in-range error). Today, the Type/Len field is rarely used to encode frame length information. 0: Disabled (recommended). 1: Enabled.	0x0
14	SEED_LOAD	R/W	Internal debug use. Do not change default value.	0x0
13:10	IFG2	R/W	Second part of half-duplex Rx to Tx Inter-frame Gap. RX-to-TX_IFG = 4.25 + (MODE_CFG::IFG2 + MODE_CFG::IFG1)/2 bytes. Transitions within IFG2 are ignored on CRS. 0x8: Recommended value.	0x09
9:6	IFG1	R/W	First part of half-duplex Rx to Tx Inter Frame Gap. The sum of MODE_CFG::IFG2 and MODE_CFG::IFG1 times the RX to TX IFG. Timing is restarted within IFG2 if CRS has multiple high-low transitions, that is, it is noisy. 0x6: Recommended value.	0x06
5	RESV	R/W	Reserved	0x0
4	VLAN_AWR	R/W	VLAN Aware. Enables the MAC to work in a VLAN aware environment. The MAC allows a VLAN-tagged frame to be MAXLEN_CFG::MAX_LEN + 4 bytes. 0: VLAN disabled. 1: VLAN enabled.	0x1
3:2	MODE	R/W	Selects the MAC duplex operation. 0x0: 10/100 Mbps half-duplex 0x1: 10/100 Mbps full-duplex 0x2: Not valid 0x3: 1000 Mbps full-duplex	0x3

Table 66. MODE_CFG (Address 0x00), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
1	RX_EN	R/W	Enables MAC receiver. 0: Disabled. 1: Enabled.	0x0
0	TX_EN	R/W	Enables MAC transmitter. 0: Disabled. 1: Enabled.	0x0

4.2.2 Pause Configuration

Table 67. PAUSE_CFG (Address 0x01), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
19	TX_PAUSE_EN	R/W	Tx Pause Enable for ports in full-duplex mode. Enables flow control in the Tx direction. The MAC generates pause frames with PAUSE_CFG::TX_PAUSE_VAL when the high watermark in the FIFO buffer is reached (see ING_HIGH_LOW_WM::HIGH_WM). 0: Disabled. 1: Enabled.	0x0
18	RX_PAUSE_EN	R/W	Rx Pause Enable. Enables flow control in Rx direction. The MAC reacts to incoming pause frames and flow control in the Tx direction. 0: Disabled. 1: Enabled.	0x0
17	TX_PAUSE_XONOFF	R/W	Tx Pause Zero on deassert. Determines whether or not a pause control frame with pause value zero is transmitted when congestion in the ingress FIFO buffer has stopped (low watermark is reached). 0: No pause Frame of zero value is transmitted. 1: A pause Frame of value zero is transmitted when flow control becomes inactive (recommended).	0x1
16	PAUSE_EN	R/W	Pause Enable. Enables flow control in both directions for full-duplex ports. Setting this bit takes precedence over PAUSE_CFG::TX_PAUSE_EN and PAUSE_CFG::RX_PAUSE_EN 0: Flow control disabled. PAUSE_CFG::TX_PAUSE_EN and PAUSE_CFG::RX_PAUSE_EN can be used. 1: Flow control enabled in both directions. In half-duplex mode disabling of flow control must be done by setting EGR_HIGH_LOW_WM::HIGH_WM_WM to 0x3FFF	0x0

Table 67. PAUSE_CFG (Address 0x01), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
15:0	TX_PAUSE_VAL	R/W	Tx Pause Value. The pause value inserted in each pause control frame in the Tx direction. It is also used to schedule the transmission of pause control frames when flow control is enabled and active. As long as flow control is asserted (high watermark in the ingress FIFO buffer is reached), a pause frame is sent out every (0.5 × TX_PAUSE_VAL) to ensure that the remote transmitter is paused. If flow control is enabled, the pause value must be set to a value > 1, based on the link characteristics. The value is specified in multiples of 512 bit times, called pause quanta. 0x0000: Not allowed. 0xFFFF: Recommended if PAUSE_CFG::TX_PAUSE_XONOFF is enabled.	0xFFFF

4.2.3 Maximum Length Configuration

Table 68. MAXLEN_CFG (Address 0x02), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
15:0	MAX_LEN	R/W	The maximum frame length accepted by the MAC receiver. If the length is exceeded, it is indicated in RX_OVERSIZE_CNT::RX_OVERSIZE. The maximum length is automatically adjusted to accommodate maximum sized frames containing a VLAN tag when the MAC is configured to be VLAN aware (MODE_CFG::VLAN_AWR = 1).	0x05EE

4.2.4 MAC Address Configuration, Bit 47:24

Table 69. MAC_ADDR_HIGH_CFG (Address 0x03), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
23:0	MAC_ADDR_HIGH	R/W	Bit 47:24 of MAC address used when generating pause control frames with the specified MAC address and when filtering incoming pause control frames.	0x0000000

4.2.5 MAC Address Configuration, Bit 23:0

Table 70. MAC_ADDR_LOW_CFG (Address 0x04), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
23:0	MAC_ADDR_LOW	R/W	Bit 23:0 of MAC address used when generating pause control frames with the specified MAC address, and when filtering incoming pause control frames.	0x0000000

4.2.6 Tri-Speed MAC Frame Normalizer

Table 71. NORMALIZER (Address 0x05), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
31:16	RSV00	R/W	Reserved	0x0000
15:12	RSV01	R/W	Reserved	0x00
11	RSV02	R/W	Reserved	0x1
10	CHAIN_LOOP	R/W	Enables chain loop. 0: Device receives data from the RGMII port if there is data on the RGMII interface. 1: Device ignores data from RGMII port. It receives data from an adjacent device even if there is data on the RGMII interface.	0x0
9	NO_CRC	R/W	Disable the Ethernet CRC updating function in the ingress path. 0: Ethernet CRC always updated in the ingress path. 1: CRC updating disabled.	0
8	RSV04	R/W	Reserved	0x0
7	RSV05	R/W	Reserved	0x0
6	DROP_PAUSE	R/W	Drop Pause. Enables the normalizer to filter out pause frames. 0: Forward pause frames. 1: Drop pause frames. These frames are not forwarded to the host interface (recommended).	0x1
5	DROP_CTRL	R/W	Drop Control. Enables the normalizer to filter out MAC control frames other than pause frames. MAC control frames are defined as follows: DMAC = 01-80-C2-00-00-01, T/L = 0x8808, opcode = 0x0000–0xFFFF (except 0x0001 which is a pause frame) 0: Forward MAC control frames. 1: Drop MAC control frames. These frames are not forwarded to the host interface (recommended).	0x1
4	RSV06	R/W	Reserved	0x0

Table 71. NORMALIZER (Address 0x05), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
3	RSV07	R/W	Reserved	0x0
2	NLE	R/W	Non-Length Enqueue mode. When a frame ends, the tri-speed MAC re-transmitted the beginning of the frame with updated length information in the normalized header. This updated header overwrites the original header in the FIFO before the FIFO passes the frame on. If NLE = 0 Ingress FIFO must be setup in store-and-forward mode, NORMALIZER::NH must be set to 1, and ING_CONTROL::LE and ING_CONTROL::NH must be set to 1. 0: Retransmit beginning of frame (full-frame mode). 1: No retransmit (burst-interleaved mode).	0x1
1	NH	R/W	Insert Normalized Header. Inserts a 9-byte header in front of received frames. If enabled, ING_CONTROL::NH must also be set. 0: Header insert disabled. 1: Header insert enabled.	0x0
0	PH	R/W	For debug purpose only. Must be set to 0x0.	0x0

4.2.7 Sticky Bits

Table 72. STICK_BIT (Address 0x0A), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
25	STK_RDPTR_IS_SAME_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
24	STK_RDPTR_IS_SAME_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
23	STK_FIFO_INH_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
22	STK_FIFO_INH_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
21	STK_REQUEST_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
20	STK_REQUEST_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
19	STK_RESTART_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
18	STK_RESTART_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
17	STK_DATA_READY_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
16	STK_DATA_READY_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
15	STK_SAME_DATA_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
14	STK_SAME_DATA_HI_STK	R/W	Sticky bit used for debug purposes.	0x0

Table 72. STICK_BIT (Address 0x0A), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
13	STK_NO_DATA_LO_STK	R/W	Sticky bit used for debug purposes.	0x1
12	STK_NO_DATA_HI_STK	R/W	Sticky bit used for debug purposes.	0x0
11	STK_RECOVER_SAME_DATA_STK	R/W	Sticky bit used for debug purposes.	0x0
10	STK_RECOVER_NO_DATA_STK	R/W	Sticky bit used for debug purposes.	0x0
9	CFG_AGE2	R/W	Count age dropped frames, in second buffer. Both CFG_AGE2 and STICK_BIT CFG_AGE must be set to count all egress frames dropped due to aging. Frames are counted with DROP_CNT::DROP_CNT. 0: No counting. 1: Count number of dropped frames (recommended).	0x0
8	DROP_AGE2_STK	R/W	Sticky bit, reset on write of 1 to this bit, frame dropped due to aging in 2nd buffer. 0: No event. 1: Event seen since reset or the last time a 1 was written.	0x0
6	CFG_ABORT	R/W	Count frames transmitted as aborted frames by the MAC. Frames are counted with DROP_CNT::DROP_CNT. Aborted frames are transmitted in the following conditions: • The frame size is >32 bytes and <64 bytes, but padding was not done, that is, not enabled for this frame. • The egress FIFO is in cut-through mode, and either an EOP/abort was received on SPI-4.2, or the frame had a CRC error. • The retransmit buffer experienced an under-run condition. 0: No counting. 1: Count number of transmitted abort frames (recommended).	0x0
5	CFG_SHORT	R/W	Count drops frames too short to be padded, that is frame sizes that are ≤32 bytes. Note that frame sizes that are ≤16 bytes are dropped and counted by the egress FIFO. Frames are counted with DROP_CNT::DROP_CNT. 0: No counting. 1: Count number of dropped frames (recommended).	0x0
4	CFG_AGE	R/W	Count age dropped egress frames. Frames are counted with DROP_CNT::DROP_CNT. 0: No counting. 1: Count number of dropped frames.	0x0

Table 72. STICK_BIT (Address 0x0A), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
3	DROP_NOPAD_STK	R/W	Sticky bit, reset on write of 1 to this bit, frame dropped due to padding not performed. 0: No event. 1: Event seen since reset, or last time 1 was written.	0x0
2	DROP_SHORT_STK	R/W	Sticky bit, reset on write of 1 to this bit, frame dropped due to too short size (≤ 32 bytes) for padding to be performed. 0: No event. 1: Event seen since reset, or last time 1 was written.	0x0
1	DROP_AGE_STK	R/W	Sticky bit, reset on write of 1 to this bit, frame dropped due to aging. 0: No event. 1: Event seen since reset, or last time 1 was written.	0x0
0	NDR_STK	R/W	Used for debug purposes.	0x0

4.2.8 Tri-Speed MAC Clock/Reset Setup

Table 73. DEV_SETUP (Address 0x0B), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
10	LI	R/W	Loopback Internally, enables loopback of the RGMII signals. The looped back data is transmitted. 0: No loopback. 1: RGMII loopback active.	0x0
9	LE	R/W	Used for debug purposes. Must be set to 0x0.	0x0
7	RMODE_XOR	R/W	Controls the decoding of the RX_ERR signal on the RGMII Interface, and the behavior of TX_EN during transmission. This bit must be set to 0x1 to ensure standard compliant RGMII behavior. 0: If RX_ERR is asserted on the falling edge of RX_CLK, the tri-speed MAC indicates a RGMII error. RX_ERR is ignored on rising edges of the RX_CLK. 1: If RX_ERR on falling edge of RX_CLK is different from RX_ERR on rising edge of RX_CLK within one Rx clock cycle, the tri-speed MAC indicates a RGMII error (recommended).	0x0
6	INV_CLK	R/W	Invert RGMII clock. 0: Do not invert the clock (recommended). 1: Invert the clock.	0x0

Table 73. DEV_SETUP (Address 0x0B), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
2:1	MODE	R/W	RGMII Mode setup. May be changed only when the tri-speed MAC is disabled, that is, when MODE_CFG::RX_EN and MODE_CFG::TX_EN are low. After the mode is changed the tri-speed MAC should be reset. 0x0: No clock 0x1: 1 Gbps 0x2: 100 Mbps 0x3: 10 Mbps	0x0
0	RST	R/W	Master reset for tri-speed MAC. 0: Tri-speed MAC is running. 1: Tri-speed MAC is reset. Note: Changes on the RGMII_RX_CLK require a reset of the MAC. It is therefore recommended to always reset the MAC after an external PHY is linked up. RST Loopback. When this bit is asserted, egress FIFO data to this port is looped to the ingress FIFO. For the loopback to work properly, the ingress FIFO must run in SME (start-mid-end) mode (ING_CONTROL::LE = 0), because the egress FIFO output always sends SME. In addition, for the RST loopback to work in full-frame mode, frames must also have the normalized header prepended when coming from the egress FIFO. Due to these restrictions, it is recommended to use bit 10 (DEV_SETUP::LI) instead.	0x1

4.2.9 Drop Counter

Table 74. DROP_CNT (Address 0x0C) Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	DROP_CNT	R/W	Indicates the number of frames that are discarded (not passed to the MAC) by the transmitter due to either aging or invalid frame length. Writing any value to the register clears the counter. The exact type of counting is controlled by STICK_BIT::CFG_AGE2, STICK_BIT::CFG_ABORT, STICK_BIT::CFG_SHORT and STICK_BIT::CFG_AGE.	0x00000000

4.2.10 Tri-Speed MAC Debug

Table 75. DEV_DEBUG (Address 0x0E), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
1	SET_FAIL	R/W	Enable proper generation of SPI-4.2 EOP/Abort for frames with CRC errors, or for suppressed MAC control frames. Should only be set when the ingress FIFO is in cut-through mode. When set, ING_TEST::NO_DROP_IN_FRM should also be set. 0: Do not generate EOP/Abort for bad ingress frames. 1: Generate EOP/Abort for bad ingress frames (recommended).	0x0
0	KEEP_BAD	R/W	Keep bad frames. Bad frames are frames with length < 64 bytes, frames with CRC errors, and frames with length > MAXLEN_CFG::MAX_LEN. If MODE_CFG::LEN_DROP is set, the errors described in the MODE_CFG::LEN_DROP bitfield, are also considered as bad frames. 0: Discard bad frames (recommended). 1: Do not discard bad frames. (NORMALIZER::NO_CRC must be set to avoid CRC updating).	0x0

4.2.11 Frame Denormalization

Table 76. DENORM (Address 0x15), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
5	CRC_UPD	R/W	Update CRC. The CRC calculator replaces the last four bytes of the egress frame with a correct Frame Check Sequence. Enabling CRC_UPD causes frames with sizes 33–63 bytes to be padded with 0x00 to achieve a 64-byte frame size. DENORM::CRC_ADD and CRC_UPD are mutually exclusive. 0: CRC update disabled (recommended). 1: CRC update enabled.	0x0

Table 76. DENORM (Address 0x15), Block 1, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
4	CRC_ADD	R/W	Add CRC. The CRC calculator appends a correct Frame Check Sequence to the frame. Enabling CRC_ADD causes frames with sizes 33–59 bytes to be padded with 0x00, to achieve a 64-byte frame size. The CRC is added on all frames. CRC_ADD and DENORM::CRC_UPD are mutually exclusive, and cannot be set high simultaneously. 0: CRC add disabled (recommended). 1: CRC add enabled.	0x0
1	EXP_N_HEAD	R/W	Used for debug purposes. Must be set to 0x0.	0x0
0	EXP_P_HEAD	R/W	Used for debug purposes. Must be set to 0x0.	0x0

4.2.12 Device 1 G Debug

Table 77. DEV_DBG (Address 0x16), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
31	FORCE_64BDONE	R/W	Internal debug use.	0x0
30	FORCE_INIT	R/W	Internal debug use.	0x0
15:8	RECOVER_DELAY	R/W	Use the following configuration according to the speed mode. 1 Gbps: 0x04 100 Mbps: 0x20 10 Mbps: 0xFF	0x0FF
7	RECOVER_INIT_SAME_DATA	R/W	Must be set to 1.	0x0
6	RECOVER_INIT_NO_DATA	R/W	Must be set to 1.	0x0
5	RECOVER_64B_SAME_DATA	R/W	Must be set to 1.	0x0
4	RECOVER_64B_NO_DATA	R/W	Must be set to 1.	0x0
2	DIS_AGE2_DROP	R/W	Internal debug use.	0x0
1	GNAT_DIS_29	R/W	Internal debug use.	0x0
0	AUTO_64B_DONE	R/W	Internal debug use.	0x0

4.2.13 TX Inter-Frame Gap Configuration

Table 78. TX_IFG (Address 0x18), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
4:0	TX_IFG	R/W	Adjust the minimum inter frame gap (IFG) between two transmitted frames. The unit of TX_IFG is MAC clocks. To achieve a 12-byte IFG, the following values should be used: 0x05: in gigabit mode. 0x11: in 10/100 Mbit full-duplex mode. 0x11: in 10/100 Mbit half-duplex mode.	0x05

4.2.14 Advanced Half-Duplex Configuration

Table 79. ADV_HDX_CFG (Address 0x19), Block 1, Subblock 0–11

Bit	Name	Mode	Description	Default
19:16	SLOT_TIMOUT	R/W	When a half-duplex (HDX) collision occurs, the total backoff time is decreased SLOT_TIMEOUT × 4 bit times. 0x7: Backoff time is decreased by 7 × 4 bit times (recommended).	0x07
14:12	FAST_DEFER_DLY	R/W	Faster defer delay. Controls the delay between going to IDLE state and when the MAC is allowed to defer transmission in half-duplex. 0x3: Recommended value.	0x3
8	FAST_HDX_EN	R/W	Enable fast half-duplex. Enables improved half-duplex (HDX) features such as fast collision detection, fast collision jamming, fast HDX flow control, and fast deferring. 0: Disable fast HDX features. 1: Enable fast HDX features (recommended).	0x1
3:0	LCOLPOS	R/W	Late Collision Position. Adjusts the border between a collision and a late collision in increments of 1 byte. According to IEEE Std 802.3 section 21.3, this border is allowed on data byte 56 (counting frame data from 1). This means that a frame experiencing a collision on data byte 55 is always retransmitted, and a frame experiencing a collision on data byte 57 is never retransmitted. For each higher LCOLPOS value, the border is moved 1 byte higher. 0x01 : Recommended value.	0x02

4.3 Ingress FIFO Buffer, Block 2

The subblock is always 0 for the ingress FIFO. Some registers are controlling overall FIFO buffer behavior; others are duplicated in the 12 FIFO buffers. Registers specific to a FIFO buffer (within a subblock) are addressed using the lowest nibble in the 7 bit register address. For example the ING_TEST register in FIFO buffer 5 has the address 0x05.

4.3.1 Mode and Test

Table 80. ING_TEST (Addresses 0x00–0x0B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31	T15	R/W	DEBUG_BUF_CNT event sets ING_TEST::MODE = TX_STOP.	0x0
30	T14	R/W	DEBUG_BUF_CNT event sets ING_TEST::MODE = RX_STOP.	0x0
29	S8_STK	R/W	Reserved, default value may not be changed.	0x1
28:26	RESV	R/W	Reserved, default value may not be changed.	0x00
25	FFAIR_EN	R/W	Enable frame fairness dropping. Normally, when a FIFO buffer is oversubscribed and in dropping mode (flow control not enabled), the frame dropping feature drops large frames with higher probability and is therefore unfair. This can be avoided by setting FFAIR_EN=1 and ING_CT_THRHLD::CLASS_THRHLD so there is room for a maximum frame above the classifier threshold. If protection of control frames is also desired, ING_CT_THRHLD::CLASS_THRHLD must be reduced with the buffer amount desired for control frame protection. 0: Low priority frames are dropped as soon as the fill level is above ING_CT_THRHLD::CLASS_THRHLD. 1: low priority frames are allowed to cross the ING_CT_THRHLD::CLASS_THRHLD threshold. If enabled CLASS_EN must also be set.	0x0
24	T8	R/W	Valid plane read. Notice that when transmitting in full-frame mode, 2 and 3 plane frames count as 4 planes.	0x0
23	T7	R/W	Plane discarded.	0x0
22	T6	R/W	Complete ultra short frame received (SE).	0x0
21	T5	R/W	Complete frame received (SMES and size >4 planes).	0x0
20	T4	R/W	Complete frame received (SMES and size ≤ 4 planes).	0x0

Table 80. ING_TEST (Addresses 0x00–0x0B), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
19	T3	R/W	ENDD/PARTIAL plane received (SMES).	0x0
18	T2	R/W	MID plane received (SME, SMES).	0x0
17	T1	R/W	START plane received (SME, SMES).	0x0
16	T0	R/W	Complete frame received (SME).	0x0
15	S7_STK	R/W	Set when reading inside frame where writing of the frame is not yet completed. This bit is typically set when running in cut-through mode. It should not be set in store-and-forward mode (see ING_CT_THRHLD::CT_THRHLD).	0x0
14	S6_STK	R/W	FIFO buffer full. Setting MODE to REPLAY triggers S6_STK each time the first plane is output. Note: Is always 0x0 if classifier dropping is enabled. (ING_TEST::CLASS_EN is set).	0x0
13	S5_STK	R/W	FIFO buffer empty. It is set after reset.	0x1
12	S4_STK	R/W	Condition ING_TEST::S7_STK AND the written frame was aborted.	0x0
11	S3_STK	R/W	Flow control was asserted by the FIFO buffer.	0x0
10	S2_STK	R/W	At least one plane was marked for aging.	0x0
9	S1_STK	R/W	Frame abort or illegal SMES sequence received when in length store mode. It is set when a FIFO buffer full condition is cancelled, and a plane was not of type START	0x0
8	S0_STK	R/W	Frame abort or illegal SME sequence received when not in length-store mode. It is set when a FIFO buffer full condition is cancelled, and a plane was not of type START.	0x0
7	CLASS_EN	R/W	Classifier dropping Enabled. Used together with the ING_CT_THRHLD::CLASS_THRHLD threshold. 0: Classifier has no influence on frame dropping. 1: FIFO buffer drops non-marked frames if the fill level is above ING_CT_THRHLD::CLASS_THRHLD. Marked frames are only dropped if the FIFO buffer is full.	0x0

Table 80. ING_TEST (Addresses 0x00–0x0B), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
6	COUNT_SUPR_CT	R/W	Suppress ING_DROP_CNT::ING_DROP_CNT counting of frames that are in the process of being transmitted. 0: ING_DROP_CNT::ING_DROP_CNT counts frames causing buffer overflow, both those where transmission has not begun, and those that are currently being transmitted. 1: ING_DROP_CNT::ING_DROP_CNT only counts overflow frames, of which transmission has not started.	0x0
5	COUNT_FAIL	R/W	Enables counting of error frames in ING_DROP_CNT::ING_DROP_CNT. ING_DROP_CNT::ING_DROP_CNT would normally only count frames dropped due to buffer overflow. (Failed frames are counted by other counters.) Error frames are frames that the tri-speed MAC has marked as bad frames either because of CRC or size error. These failures are also counted in the MAC statistics block. 0: ING_DROP_CNT::ING_DROP_CNT counts frames dropped due to buffer overflow. 1: ING_DROP_CNT::ING_DROP_CNT also counts failure marked frames.	0x0
4	NO_DROP_IN_FRM	R/W	Prevents FIFO buffer from discarding remaining part of frame when it has started to transmit a frame that is received with an error. When the frame is not discarded, it is also not counted by ING_DROP_CNT::ING_DROP_CNT. In the ingress direction, it must be set if SPI-4.2 is to generate proper EOP with abort for frames with CRC errors. DEV_DEBUG::SET_FAILshould contain same value as NO_DROP_IN_FRM. 0: Remaining part of frame dropped. 1: Never drop or count a frame that is currently transmitting.	0x0

Table 80. ING_TEST (Addresses 0x00–0x0B), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
3:0	MODE	R/W	Selects the FIFO buffer operational mode. 0x0: NORMAL. The FIFO buffer attempts to store everything it receives and transmit everything stored. 0x1: RX_STOP. The FIFO buffer is flushed. All frames from the FIFO buffer are transmitted normally. No new frames are stored. 0x2: CLEAR_PTR. No new frames are stored in the FIFO buffer, and no frames are sent out of the FIFO buffer. HEAD and ING_TAIL::TAIL pointers are initialized to the ING_TOP_BOTTOM::BOTTOM value. While in CLEAR_PTR mode, ING_DROP_CNT::ING_DROP_CNT counts discarded frames. 0x3: TX_STOP. Frame transmitting is stopped immediately, which can also happen in the middle of a frame. Note that frames can still be received and are stored normally. 0x4: REPLAY. Starts a cyclic replay of the frames in the FIFO buffer. Data between ING_TOP_BOTTOM::BOTTOM and ING_TOP_BOTTOM::TOP – 1 is output continuously, reading from the ING_HEAD::HEAD location, and the FIFO buffer rejects any data presented on the inputs. When stopping the REPLAY test mode with the NORMAL or RX_STOP mode, output continues until the HEAD pointer reaches the ING_TAIL::TAIL pointer. With an initial value of ING_HEAD::HEAD = ING_TAIL::TAIL = 0, this implies that REPLAY outputs an integer number of FIFO buffer sizes. If the size of the frame data is less than the FIFO buffer size, the REPLAY function outputs whatever is stored in the FIFO buffer RAM anyway. In such a situation in the egress direction, the memory not used by frames should be filled with START, MID planes. 0xC: RTT, Replay To Tail. Starts a cyclic replay of the frames in the FIFO buffer. Data between ING_TOP_BOTTOM::BOTTOM and ING_TAIL::TAIL-1 is output continuously, reading from the ING_HEAD::HEAD location, and the FIFO buffer rejects any data presented on the inputs. When stopping the REPLAY test mode with the NORMAL or RX_STOP mode, output continues until the ING_HEAD::HEAD pointer reaches 0. If the size of the frame data is less than the FIFO buffer size, the REPLAY function outputs whatever is stored in the FIFO buffer RAM anyway.	0x02

S0_STK through S7_STK are sticky bits used for diagnostics. Each bit is set upon a certain condition. The sticky bit is only cleared with a reset of the FIFO, or by writing a 1 to it. The bits T0 through T15 enable DEBUG_BUF_CNT to count the events specified by the “T”bit.

4.3.2 FIFO Buffer Top and Bottom

Table 81. ING_TOP_BOTTOM (Addresses 0x10–0x1B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
23:16	TOP	R/W	Defines the top of the memory area used by the FIFO buffer.	0x000
7:0	BOTTOM	R/W	Defines the bottom of the memory area used by the FIFO buffer.	0x000

The ingress FIFO is an 4-Mbit buffer. The 4-Mbit buffer is shared between a maximum of 12 RGMII ports.

ING_TOP_BOTTOM::TOP and ING_TOP_BOTTOM::BOTTOM must be defined for each RGMII port. The size and placement of each FIFO buffer can be individually defined through the ING_TOP_BOTTOM::TOP and ING_TOP_BOTTOM::BOTTOM pointers. The granularity of ING_TOP_BOTTOM::TOP and ING_TOP_BOTTOM::BOTTOM pointers is 2048 bytes, with ING_TOP_BOTTOM::TOP allowed to wrap around.

The FIFO buffer uses memory starting from ING_TOP_BOTTOM::BOTTOM × 2048 bytes and ending with ING_TOP_BOTTOM::TOP × 2048 – 32 bytes. This means that the ING_TOP_BOTTOM::BOTTOM pointer of the adjacent FIFO buffer may be set to the same value as ING_TOP_BOTTOM::TOP on the previous buffer.

When writing to this register, ING_CONTROL::CLR must be 1, or MODE in the ING_TEST register must be set to CLEAR_PTR.

The FIFO buffer size is calculated as
(ING_TOP_BOTTOM::TOP – ING_TOP_BOTTOM::BOTTOM) × 2048 bytes.

The maximum a FIFO buffer can contain is
(ING_TOP_BOTTOM::TOP – ING_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes.

After reset, all 12 FIFO buffers have ING_TOP_BOTTOM::TOP = 0x00 and ING_TOP_BOTTOM::BOTTOM = 0x00. This makes all 12 FIFO buffers occupy the same 4-Mbit buffer area. For even distribution of memory between 12 ports, use the following configuration:

register 0x10 = 0x00150000, register 0x11 = 0x002A0015, register 0x1B = 0x00FC00E7

Note that all the FIFO buffer size calculations do not include the additional data used internally to keep track of frame boundaries.

4.3.3 Write Pointer

Table 82. ING_TAIL (Addresses 0x20–0x2B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
13:0	TAIL	R/W	TAIL points to a free 32-byte location, where the data received by the FIFO buffer is written. Writing to TAIL is only possible if ING_CONTROL::CLR=0, and immediately causes the FIFO buffer content from ING_HEAD::HEAD up to and including ING_TAIL::TAIL – 1 to be output (assuming MODE = NORMAL in the ING_TEST register). If ING_HEAD::HEAD equals TAIL, the FIFO buffer is empty.	0x0000

4.3.4 Read Pointer

Table 83. ING_HEAD (Addresses 0x30–0x3B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
13:0	HEAD	R/W	HEAD points to an occupied 32-byte location, ready to be transmitted. Under normal usage HEAD should not be set. Instead, use ING_TEST::MODE = CLEAR, which initializes HEAD to 64 × ING_TOP_BOTTOM::BOTTOM. Writing to HEAD is only possible if ING_CONTROL::CLR = 0. If HEAD equals ING_TAIL::TAIL, the FIFO buffer is empty.	0x0000

4.3.5 Flow Control Watermarks

Table 84. ING_HIGH_LOW_WM (Addresses 0x40–0x4B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
29:16	LOW_WM	R/W	Defines the low watermark for flow control.	0x3FFF
13:0	HIGH_WM	R/W	Defines the high watermark for flow control.	0x3FFF

When the FIFO buffer content is equal to or greater than the ING_HIGH_LOW_WM::HIGH_WM number of 32-byte entries, flow control is asserted. When the FIFO buffer contains less than ING_HIGH_LOW_WM::LOW_WM number of 32-byte entries, flow control is released.

Note that if a suppression of frame fragments is selected (setting ING_CT_THRHLD::CT_THRHLD = 0), then ING_HIGH_LOW_WM::LOW_WM must be set to at least one plane higher than the maximum frame size. This is to avoid a deadlock situation.

If the ING_HIGH_LOW_WM::LOW_WM and ING_HIGH_LOW_WM::HIGH_WM default values are used, the MAC flow control must be explicitly disabled by writing to a MAC pause control register.

When calculating FIFO utilization, note that the maximum a FIFO buffer can contain is (ING_TOP_BOTTOM::TOP – ING_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes. Frame storage in bytes is rounded up to the nearest byte count divisible by 32. The Ethernet frame length is enhanced by 9 bytes when transmitting in full-frame mode, and the frames occupy at least 128 bytes. For more information, see "Watermark Calculations," page 106.

4.3.6 Cut-Through Threshold

Table 85. ING_CT_THRHLD (Addresses 0x50–0x5B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
29:16	CLASS_THRHLD	R/W	Classifier Drop Threshold. Used when ING_TEST::CLASS_EN = 1. Number of 32-byte entries allowed in a FIFO buffer from normal priority frames before data is discarded. When the FIFO buffer storage exceeds the CLASS_THRHLD, only data from frames classified as high priority is allowed. On an oversubscribed buffer, CLASS_THRHLD is used to protect control frames. For example, setting CLASS_THRHLD to 0x530 gives a 384 bytes reserved at the top of the FIFO buffer for control frames only.	0x0000
13:0	CT_THRHLD	R/W	Cut Through Threshold. Number of 32-byte entries at the start of a frame, which must be present in the FIFO buffer before the output algorithm "sees" the frame. When the number of entries reaches the CT_THRHLD value, the FIFO buffer begins to transmit the frame. Values higher than 1 can wait for a certain amount of data before the FIFO starts to transmit. In full-frame mode, this parameter must be set to 0. Setting CT_THRHLD to 0 makes the buffer wait for a complete frame before it is forwarded. This is useful if frames with CRC errors must be suppressed. 0: Store-and-forward mode 1–2: Cut-through mode >2: Partly store-and-forward mode and partly cut-through mode	0x0001

4.3.7 Drop and CRC Error Counter

Table 86. ING_DROP_CNT (Addresses 0x60–0x6B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	ING_DROP_CNT	R/W	Counts the number of frames dropped due to FIFO buffer overflow. ING_DROP_CNT also counts once, when the FIFO buffer is put in RX_STOP mode (see register ING_TEST::MODE). A FIFO buffer can maximum contain (ING_TOP_BOTTOM::TOP – ING_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes. The ING_TEST::S6_STK bit indicates that a FIFO buffer full condition has occurred. Setting ING_TEST::COUNT_FAIL also counts frames that are detected by the MAC to be in error, that is, CRC failure or in-range error. Notice, however, that such failed frames are also counted by MAC counters.	0x00000000

4.3.8 Input Side Debug Classification Control

Table 87. ING_DEBUG_BUF_CNT (Addresses 0x7-0x7B), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	DEBUG_BUF_CNT	R/W	Counts the number of trigger events in a single FIFO buffer. These are all related to the input side of the FIFO buffer. The events to be counted are set in the ING_TEST register (the T# bits). Note that the counter does not saturate. Simultaneous events are counted just once. To count events related to the output side of the FIFO, see the registers ING_DEBUG_OUT and ING_DEBUG_CNT::DEBUG_CNT.	0x00000000

4.3.9 Ingress Layer 3-4 Classification Control

Table 88. ING_CL3_CTRL (Address 0x0C), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
16	UDP_TCP_EN	R/W	Enable classification of TCP and UDP frames with a port number matching either ING_CL3_L4::PORT0 or ING_CL3_L4::PORT1. Only IPv4 frames are considered. Due to pipeline optimization in the classifier, UDP frames with a IP header length IHL ≥ 7 and UDP payload data ≤ 12 bytes, are also classified as high priority.	0x0

Table 88. ING_CL3_CTRL (Address 0x0C), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
9	IPV4_PROTO_EN	R/W	Enable classification of IPv4 frames with a IPv4 protocol field matching ING_CL3_L3::IPV4_PROT.	0x0
8	IPV4_MC_EN	R/W	Enable classification of IPv4 frames with multicast destination IP = 224.0.0/24.	0x0

Note: A frame identified by IPV4_MC_EN or the IPV4_PROTO_EN, or the UDP_TCP_EN classifier is marked as high priority, and is not dropped when the FIFO buffer fill level is above the classifier threshold. For classification to work, ING_CT_THRHLD::CLASS_THRHLD must be programmed and ING_TEST::CLASS_EN must be = 1.

4.3.10 Ingress Layer 3-4 Classification Layer 3 Field

Table 89. ING_CL3_L3 (Address 0x1C), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
7:0	IPV4_PROT	R/W	Programmable IPv4 protocol value for protection of IPv4 frames. Used when ING_CL3_CTRL::IPV4_PROT_EN = 1.	0x00

4.3.11 Ingress Layer 3-4 Classification Layer 4 Fields

Table 90. ING_CL3_L4 (Address 0x2C), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:16	PORT1	R/W	Programmable TCP/UDP destination or source port number 1. Used when ING_CL3_CTRL::UDP_TCP_EN = 1.	0x0000
15:0	PORT0	R/W	Programmable TCP/UDP destination or source port number 0. Used when ING_CL3_CTRL::UDP_TCP_EN = 1.	0x0000

4.3.12 Ingress Unicast and Multicast Filtering Enable

Table 91. ING_FFILT_UM_EN (Address 0x0D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:28	PSEL	R/W	Port Select. Indirect addressing for the mask filter ING_FFILT_VAL0::VAL_A0 and ING_FFILT_MASKx. Only values 0–11 allowed.	0x00

Table 91. ING_FFILT_UM_EN (Address 0x0D), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
27:16	MASK_EN	R/W	Enable mask filtering. Each bit corresponds to one FIFO buffer. When enabled, all frames not matching the port specific mask address are dropped. Dropping takes precedence over non-dropping by other filters.	0x0000
13	VALSEL	R/W	Value filter select. Indirect addressing of one of two possible filters, which is accessed through ING_FFILT_VAL0 and ING_FFILT_VAL1.	0x0
12	FFILT_EN	R/W	Frame filtering enable. Global enable bit for all frame filtering features.	0x0
11:0	VAL_EN	R/W	Enable value filtering. Each bit corresponds to one FIFO buffer. When enabled, all frames not matching the port specific address, are dropped. Note that both unicast and multicast addresses can be matched. Dropping takes precedence over non-dropping by other filters.	0x0000

4.3.13 Ingress Broadcast and EtherType Filtering Enable

Table 92. ING_FFILT_BE_EN (Address 0x1D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
27:16	ETYPE_EN	R/W	Enable EtherType filtering. Each bit corresponds to one FIFO buffer. When enabled, all frames with a Type/Length field that match one of the programmable EtherType values are dropped. Dropping takes precedence over non-dropping by other filters. For example if both EtherType filtering and unicast filtering is enabled, and the packet contains a matching unicast DMAC (would normally not be dropped), and a matching Type/Length value (should be dropped), the packet is dropped.	0x0000
11:0	BROAD_EN	R/W	Enable broadcast filtering. Each bit corresponds to one FIFO buffer. When enabled, all frames with a broadcast DMAC are dropped. Dropping takes precedence over non-dropping by other filters.	0x0000

4.3.14 Ingress Value Filter Address

Table 93. ING_FFILT_VAL0 (Address 0x2D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:5	VAL_A1	R/W	DMAC address filter bits 31:5. The bits are global and applies to all 12 ports. VAL_A1 accesses two independent filters. ING_FFILT_UM_EN::VALSEL selects which one is accessed.	0x0000000
4:0	VAL_A0	R/W	DMAC address filter bits 4:0. The bits are unique for each port. Bits 7:0 are contained in the last received DMAC byte. Each port has two independent unicast filters. ING_FFILT_UM_EN::PSEL selects for which port these bits apply, and ING_FFILT_UM_EN::VALSEL selects which of the two filters is accessed through VAL_A0.	0x00

Table 94. ING_FFILT_VAL1 (Address 0x3D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
15:0	VAL_A2	R/W	DMAC address filter bits 47:32. The bits are global and applies to all 12 ports. Bits 47:40 are contained in the first received DMAC byte. Bit 40 is the DMAC I/G bit. VAL_A2 bit 8 must match the DMAC I/G bit before this filter consideres if the frame should be dropped. It is therefore possible to filter both unicast or multicast frames.	0x0000

4.3.15 Ingress Mask Filter Address

Table 95. ING_FFILT_MASK0 (Address 0x4D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:16	MMASK_A0	R/W	DMAC mask bits 15:0. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply.	0x0000
15:0	ADDR_A0	R/W	DMAC address filter bits 15:0. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply. ADDR_A0 is only compared at the bit positions where MMASK_A0 = 1. ADDR_A0 must contain 0 at the locations where ING_FFILT_MASK0::MMASK_A0 is 0, otherwise the filter never matches.	0x0000

Table 96. ING_FFILT_MASK1 (Address 0x5D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:16	MMASK_A1	R/W	DMAC mask bits 31:16. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply.	0x0000
15:0	ADDR_A1	R/W	DMAC address filter bits 31:16. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply. ADDR_A1 is only compared at the bit positions where ING_FFILT_MASK1::MMASK_A1 = 1. ADDR_A1 must contain 0 at the locations where ING_FFILT_MASK1::MMASK_A1 is 0, otherwise the filter never matches.	0x0000

Table 97. ING_FFILT_MASK2 (Address 0x6D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:16	MMASK_A2	R/W	DMAC mask bits 47:32. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply.	0x0000
15:0	ADDR_A2	R/W	DMAC address filter bits 47:32. The bits are unique for each port. ING_FFILT_UM_EN::PSEL selects for which port these bits apply. Bits 47:40 are contained in the first received DMAC byte. Bit 40 is the DMAC I/G bit. ADDR_A2 bit 8 must match the DMAC I/G bit before this filter considers whether the frame should be dropped. It is therefore possible to filter either unicast or multicast frames. ADDR_A2 must contain 0 at the locations where ING_FFILT_MASK2::MMASK_A2 is 0, otherwise the filter never matches.	0x0000

4.3.16 Ingress EtherType Filter Values

Table 98. ING_FFILT_ETYPE (Address 0x7D), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:16	ETYP1	R/W	EtherType filter value 2.	0x0000

Table 98. ING_FFILT_ETYPE (Address 0x7D), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
15:0	ETYP0	R/W	EtherType filter value 1. If EtherType filtering is enabled, frames with a Type/Length field that match either ETYP0 or ING_FFILT_ETYPE::ETYP1 are dropped. If you only want to filter against a single EtherType value set ING_FFILT_ETYPE::ETYP1 := ETYP0.	0x0000

4.3.17 SRAM_ADDR

Table 99. ING_SRAM_ADDR (Address 0x0E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
14:0	SRAM_ADDR	R/W	The address of the data to be accessed in the FIFO buffer memory. The address is in 16-byte units, therefore two consecutive addresses correspond to one address as indexed by ING_HEAD::HEAD and ING_TAIL::TAIL. When writing frames to the FIFO RAM, these frames must start at an even SRAM_ADDR value, that is, on a 32-byte boundary. A restriction on the packing of frames applies when the frames have to be output in full-frame mode (ING_CONTROL::CM = 0). Frames must have a spacing of at least 128 bytes. This means that after a 64-byte frame, the next 64 bytes are ignored and not transmitted by the FIFO.	0x0000

4.3.18 SPRAM_WR_STRB

Table 100. ING_SRAM_WR_STRB (Address 0x1E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_WR_STRB	W/O	A write to this address initiates a write to the FIFO RAM. Data contained in the ING_SRAM_DATA_0::SRAM_DATA_0, ING_SRAM_DATA_1::SRAM_DATA_1 , ING_SRAM_DATA_2::SRAM_DATA_2, ING_SRAM_DATA_3::SRAM_DATA_3 and ING_SRAM_DATA_BLK_TYPE::BLK_TYPE is written to the location contained in the ING_SRAM_ADDR::SRAM_ADDR. The write is delayed until a timeslot with an unused plane is detected in the normal input data stream to the FIFO. Note that data in SRAM_DATA_x must be bit-swapped within each byte. This is in contrast to an RGMII interface application where data must be bit-swapped to the same byte.	0x00000000

4.3.19 SRAM_RD_STRB

Table 101. ING_SRAM_RD_STRB (Address 0x2E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_RD_STRB	W/O	A “write” to this address initiates a “read” signal to the FIFO RAM. Data is read from the address contained in ING_SRAM_ADDR::SRAM_ADDR and stored in ING_SRAM_DATA_0::SRAM_DATA_0, ING_SRAM_DATA_1::SRAM_DATA_1 , ING_SRAM_DATA_2::SRAM_DATA_2, ING_SRAM_DATA_3::SRAM_DATA_3 and ING_SRAM_DATA_BLK_TYPE::BLK_TYPE. If the FIFO is putting out data while a “read” is performed, the frame data is destroyed. All FIFO buffers must be empty or put into test mode (by setting ING_TEST::MODE= TX_STOP) prior to performing a “read”. Note that data in SRAM_DATA_x must be bit-swapped within each byte. This is in contrast to an RGMII interface application where data must be bit-swapped to the same byte.	0x00000000

4.3.20 SRAM_DATA_0

Table 102. ING_SRAM_DATA_0 (Address 0x3E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_0	R/W	Byte 0 to 3 of the data read or to be written to the FIFO RAM. Byte 0 is the last byte (of byte 0 to 15) received on the RGMII interface. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.3.21 SRAM_DATA_1

Table 103. ING_SRAM_DATA_1 (Address 0x4E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_1	R/W	Byte 4 to 7 of the data read or to be written to the FIFO RAM. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.3.22 SRAM_DATA_2

Table 104. ING_SRAM_DATA_2 (Address 0x5E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_2	R/W	Byte 8 to 11 of the data read or to be written to the FIFO RAM. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.3.23 SRAM_DATA_3

Table 105. ING_SRAM_DATA_3 (Address 0x6E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_3	R/W	Byte 12 to 15 of the data read or to be written to the FIFO RAM. Byte 15 (bit 24 to 31 in this register) is the first byte received on the RGMII interface. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.3.24 SRAM_DATA_BLK_TYPE

Table 106. ING_SRAM_DATA_BLK_TYPE (Address 0x7E), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
1:0	BLK_TYPE	R/W	The bits determining the type of the stored 16 bytes. START must be used for the first 16 bytes in a frame. All remaining 16-byte blocks must be marked with MID, except for the last data, which is marked with END if the frame ends on a 16-byte boundary. Otherwise PARTIAL is used. 0x0: START 0x1: MID 0x2: PARTIAL, byte 0 in SRAM_DATA_0 indicates number of valid bytes 0x3: END	0x0

4.3.25 Ingress FIFO Master Control

Table 107. ING_CONTROL (Address 0x0F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:28	OUT_PORT_OFFSET	R/W	Reserved, must be set to 0x0.	0x00
27:24	INP_PORT_OFFSET	R/W	Reserved, must be set to 0xA.	0x00
21	EXT_SCH_EN	R/W	Reserved, must be set to 0x1.	0x0
20	EXT_FRM_MODE	R/W	External scheduler in full-frame mode. 0: Burst-interleaved mode. 1: Full-frame mode.(Also set MSCH::BURSTINTLV = 1).	0x0
19	SRR	R/W	Reserved, default value may not be changed.	0x0
18	FAIL_IGN	R/W	Fail Ignore. When set, this bit ignores fail signaling from the tri-speed MAC. For example, frames with in-range or CRC errors are not dropped or counted by ING_DROP_CNT::ING_DROP_CNT. Such frames can cause an EOP/Abort on the SPI-4.2 interface.	0x0
17:16	MUX	R/W	Selects the source of inhibit signal to the FIFO buffers. 0x0: Inhibit signals are sourced by SPI-4.2 interface.	0x0
15	ETFC	R/W	Must be 0x0 when SPI-4.2 is active.	0x0
14	TC	R/W	Turbo Change. Enables the internal FIFO scheduling algorithm. Must be set to 0x0.	0x0

Table 107. ING_CONTROL (Address 0x0F), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
13	LE	R/W	Length Enqueue mode. Enable support for retransmit of normalized header with updated length field. Used in full-frame mode (ING_CONTROL::CM = 0). Must be set to the opposite value of NORMALIZER::NLE. 0: a frame is received as START, MID, END. 1: a frame is received as START, MID, END, START. The last START plane is a retransmission of the first plane, but with correct length information.	0x0
12	IGI	R/W	Ignore Inhibit. When this bit is set, the inhibit input to the FIFO is ignored. This bit is used for debugging purposes only. Note that setting IGI disables shaper functionality. 0: Normal mode. Inhibit to the FIFO stops transmission (recommended). 1: The inhibit input to the FIFO is ignored.	0x0
11	ITFC	R/W	Ingress Transparent Flow Control. When enabled, the flow control status received from the SPI-4.2 status channel is passed on and generates egress flow control frames. If proper transparent flow control mode, both ING_CONTROL::IGI and SPI4_ING_SETUP2::CREDITBYPASS must be set. 0: Disabled (recommended). 1: Enabled.	0x0
10	RT	R/W	Internal debug feature. Do not change default value.	0x0
9	PH	R/W	Reserved, do not change default value.	0x0
8	NH	R/W	Reserved.	0x0
7	SS	R/W	Suppress Slow_Enable. Suppresses the full stop (slow_enable) signal to the FIFO. Slow_Enable causes the output to stop from all 12 FIFO buffers. In the ingress direction SS is only used by the common shaper. 0: Slow_enable is active and can stop FIFO transmission (recommended). 1: Slow_enable is suppressed.	0x0
6	CLR	R/W	Clears all FIFO Buffers. When reset is initiated, a state machine starts clearing the FIFO RAM locations. During this process nothing is written to the RAM. The CLR bit should not be cleared until 120 µs after the reset. 0: FIFO is active. 1: FIFO pointers are cleared. FIFO is not active.	0x1
5	M10G	R/W	Not used. Must be set to 0x0.	0x1

Table 107. ING_CONTROL (Address 0x0F), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
4	CM	R/W	Not used in Ingress FIFO. Should always be 0x1. The scheduling method is controlled by MSCH::BURSTINTLV.	0x1
3:0	BURST_LEN	R/W	Reserved, should always be set to 0x1.	0x1

4.3.26 Ingress Aging Timer

Table 108. ING_AGE_TIMER (Address 0x1F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	AGE_TIMER	R/W	The Age Timer counter is incremented for every second system clock by the amount set for ING_AGE_INC::AGE_INC. When the 32 bit AGE_TIMER counter overflows, an age time tick is generated. Frames stored in the FIFO buffer for longer than the timeout period are marked as aged. The timeout period is > TIME_TICK and ≤ (2 × TIME_TICK). TIME_TICK = 2×2^{32} / (150.0 MHz × AGE_INC). The time tick updates two shadow registers in each of the 12 FIFO Buffers.	0x00000000

4.3.27 Ingress Aging Timer Increment

Table 109. ING_AGE_INC (Address 0x2F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
27:0	AGE_INC	R/W	The Aging Increment adds a unit to the ING_AGE_TIMER::AGE_TIMER every second time the system clocks. When the 32 bit ING_AGE_TIMER::AGE_TIMER counter overflows, an age time tick is generated. Frames stored in the FIFO buffer for longer than the timeout duration are marked as aged. The timeout period is > TIME_TICK and ≤ (2 × TIME_TICK). TIME_TICK = 2×2^{32} / (150.0 MHz × AGE_INC). The time out duration depends on the arrival of the frame relative to the age time tick. All 12 buffers in the FIFO are aged by the same timer. Using 0x0000000 disables aging of frames. 0x0000074 ages frames stored for more than 1 second.	0x00000000

4.3.28 Ingress Output Side Debug Counter Control

Table 110. ING_DEBUG_OUT (Address 0x3F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
23:22	T7_T6	R/W	Unused.	0x0
21	T5	R/W	Reserved.	0x0
20	T4	R/W	Read 32 valid bytes (= 1 plane) (burst-interleaved and full-frame mode).	0x0
19:16	T3_T0	R/W	Reserved.	0x00
15:11	S7_S3_STK	R/W	Unused.	0x00
10	S2_STK	R/W	Non-allowed plane seen on the input side. The first halfplane has valid = 0. The second plane has valid = 1. This signal indicates a problem in the device1g normalizer.	0x0
9	S1_STK	R/W	Reserved.	0x0
8	S0_STK	R/W	Reserved.	0x0

S0_STK–S7_STK are sticky bits used for diagnostics. Each bit is set upon a certain condition. The sticky bit is only cleared with a reset of the FIFO, or by writing a 1 to it.

Each of the T0–T7 bits enables counting by ING_DEBUG_CNT of certain event specified.

4.3.29 Output Side Debug Counter

Table 111. ING_DEBUG_CNT (Address 0x4F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
31:0	DEBUG_CNT	R/W	Counts the number of FIFO trigger events. These are set in the ING_DEBUG_OUT register. Simultaneous events are counted just once.	0x00000000

4.3.30 Per Port Inhibit Control

Table 112. PPORT_INH (Address 0x5F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
11:0	PPORT_IGI	R/W	Per Port Ignore Inhibit. Each bit corresponds to a single FIFO Buffer. When set, the inhibit input is ignored for this port. Shaper functionality for this port will not be possible. Used for debug purposes.	0x0000

4.3.31 Classifier Control

Table 113. CLASSIFIER_CTRL (Address 0x6F), Block 2, Subblock 0

Bit	Name	Mode	Description	Default
24	TYPE_FIELD_MASK	R/W	Mark MAC control frames with type field 0x8808 as control frames to be protected in the FIFO. 0: Do not mark MAC control frames. 1: Mark MAC control frames.	0x0
23:16	DMAC_MASK_TWO	R/W	Bit mask used to select MAC addresses reserved for GARP protocols (DMAC 0x0180c2000020 to 0x0180c200002f). Currently only GMRP (0x..20) and GVRP (0x..21) are assigned. Bit 16 to bit 22 selects marking of frames with DMAC in the range of 0x0180c2000020 to 0x0180c2000026, where the number in the last 4 bits of the DMAC is equal to the number of the bit as shown below. Bit 23 selects the entire range of 0x0180c2000020 to 0x0180c200002F. Bit 16 → 0x0180c2000020 Bit 17 → 0x0180c2000021 Bit 18 → 0x0180c2000022 Bit 19 → 0x0180c2000023 Bit 20 → 0x0180c2000024 Bit 21 → 0x0180c2000025 Bit 22 → 0x0180c2000026 Bit 23 → 0x0180c200002X	0x00
15:8	DMAC_MASK_ONE	R/W	Bit mask used to select MAC addresses reserved for bridge management (DMAC 0x0180c2000010 to 0x0180c200001f). This is similar to CLASSIFIER_CTRL::DMAC_MASK_ZERO, but the IEEE has removed this range in the latest Provider Bridge additions. This classifier can also be used to identify ISIS frames, which use a DMAC=0x0180c2000014 or 0x0180c2000015. Bit 8 to bit 14 selects marking of frames with DMAC in the range of 0x0180c2000010 to 0x0180c2000016, where the number in the last 4 bits of the DMAC is equal to the number of the bit as shown below. Bit 15 selects the entire range of 0x0180c2000010 to 0x0180c200001F. Bit 8 → 0x0180c2000010 Bit 9 → 0x0180c2000011 Bit 10 → 0x0180c2000012 Bit 11 → 0x0180c2000013 Bit 12 → 0x0180c2000014 Bit 13 → 0x0180c2000015 Bit 14 → 0x0180c2000016 Bit 15 → 0x0180c200001X	0x00

Table 113. CLASSIFIER_CTRL (Address 0x6F), Block 2, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
7:0	DMAC_MASK_ZERO	R/W	Bridge frames selection. Bit mask used to select frames with a DMAC address in the range 0x0180c2000000 to 0x0180c200000f are marked as control frames to be protected in the FIFO. Bit 0 to bit 6 selects marking of frames with DMAC in the range of 0x0180c2000000 to 0x0180c2000006, where the number in the last 4 bits of the DMAC is equal to the number of the bit as shown below. Bit 7 selects the entire range of 0x0180c2000000 to 0x0180c200000F. Bit 0 → 0x0180c2000000 Bit 1 → 0x0180c2000001 Bit 2 → 0x0180c2000002 Bit 3 → 0x0180c2000003 Bit 4 → 0x0180c2000004 Bit 5 → 0x0180c2000005 Bit 6 → 0x0180c2000006 Bit 7 → 0x0180c200000X	0x00

4.4 Egress FIFO Buffer, Block 2

The subblock is always 1 for the egress FIFO. Some registers are controlling overall FIFO buffer behavior; others are duplicated in the 12 FIFO buffers. Registers specific to a FIFO buffer (within a subblock) are addressed using the lowest nibble in the 7-bit register address, that is, the EGR_TEST register in FIFO buffer 5 has the address 0x05.

4.4.1 Mode and Test

Table 114. EGR_Test (Addresses 0x00–0x0B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31	T15	R/W	EGR_DEBUG_CNT::DEBUG_CNT event sets EGR_TEST::MODE = TX_STOP.	0x0
30	T14	R/W	EGR_DEBUG_CNT::DEBUG_CNT event sets EGR_TEST::MODE = RX_STOP.	0x0
29	S8_STK	R/W	Set when Above Threshold signal to GFPT block is asserted.	0x1
28:25	T12_T9	R/W	Not used.	0x00
24	T8	R/W	Valid plane read.	0x0
23	T7	R/W	Plane discarded.	0x0
22	T6	R/W	Complete ultra short frame received (SE).	0x0
21	T5	R/W	Complete frame received (SMES and size > 4 planes).	0x0
20	T4	R/W	Complete frame received (SMES and size ≤ 4 planes).	0x0
19	T3	R/W	ENDD/PARTIAL plane received (SMES).	0x0
18	T2	R/W	MID plane received (SME, SMES).	0x0
17	T1	R/W	START plane received (SME, SMES).	0x0
16	T0	R/W	Complete frame received (SME).	0x0
15	S7_STK	R/W	Set when reading inside frame where writing of the frame is not yet completed.	0x0
14	S6_STK	R/W	FIFO buffer full. Setting EGR_TEST::MODE = REPLAY triggers S6_STK each time the first plane is output.	0x0
13	S5_STK	R/W	FIFO buffer empty. It is set after reset.	0x1
12	S4_STK	R/W	Condition EGR_TEST::S7_STK AND the written frame was aborted.	0x0
11	S3_STK	R/W	Flow control was asserted by the FIFO buffer.	0x0
10	S2_STK	R/W	At least one plane was marked for aging.	0x0
9	S1_STK	R/W	Frame abort or illegal SMES sequence received when in length store mode.	0x0

Table 114. EGR_Test (Addresses 0x00–0x0B), Block 2, Subblock 1 *(continued)*

Bit	Name	Mode	Description	Default
8	S0_STK	R/W	Frame abort or illegal SME sequence received when not in length-store mode.	0x0
6	COUNT_SUPR_CT	R/W	Suppresses EGR_DROP_CNT::DROP_CNT counting of frames that are in the process of being transmitted. Should be set to achieve an exact count when the EGR_DROP_CNT::DROP_CNT is combined with DROP_CNT::DROP_CNT counter. 0: EGR_DROP_CNT::DROP_CNT counts frames causing buffer overflow, both those where transmission has not begun, and those that are currently being transmitted. 1: EGR_DROP_CNT::DROP_CNT only counts overflow frames that have not yet started transmitting.	0x0
5	COUNT_FAIL	R/W	Enables counting of failure frames in EGR_DROP_CNT::DROP_CNT. EGR_DROP_CNT::DROP_CNT would normally only count frames dropped due to buffer overflow, because failed frames are counted by other counters. Used for debugging purposes. 0: EGR_DROP_CNT::DROP_CNT counts frames dropped due to buffer overflow (recommended). 1: EGR_DROP_CNT::DROP_CNT also counts failure marked frames.	0x0
4	NO_DROP_IN_FRM	R/W	Prevents FIFO buffer from discarding remaining part of frame when it has started to transmit a frame this is received with an error. When the frame is not discarded, it is also not counted by EGR_DROP_CNT::DROP_CNT. When the egress direction runs in cut-through mode and this bit is set, frames received with EOP/Abort or CRC errors are transmitted if transmission started before that last part of the frame was received. 0: Remaining part of frame dropped (recommended). 1: Never drop or count a frame that is currently transmitting.	0x0

Table 114. EGR_Test (Addresses 0x00–0x0B), Block 2, Subblock 1 *(continued)*

Bit	Name	Mode	Description	Default
3:0	MODE	R/W	Selects the FIFO buffer operational mode. 0x0: NORMAL. The FIFO buffer attempts to store everything it receives, and transmit everything stored. 0x1: RX_STOP. The FIFO buffer is flushed. All frames from the FIFO buffer are transmitted normally. No new frames are stored. 0x2: CLEAR_PTR. No new frames are stored in the FIFO buffer and no frames are sent out of the FIFO buffer. EGR_HEAD::HEAD and pointers are initialized to the EGR_TOP_BOTTOM::BOTTOM value. While in CLEAR_PTR mode, EGR_DROP_CNT::DROP_CNT counts discarded frames. 0x3: TX_STOP. Frame transmitting is stopped immediately, which can also happen in the middle of a frame. Note that frames can still be received, and are stored normally. 0x4: REPLAY. Starts a cyclic replay of the frames in the FIFO buffer. Data between EGR_TOP_BOTTOM::BOTTOM and EGR_TOP_BOTTOM::TOP-1 is output continuously, reading from the EGR_HEAD::HEAD location, and the FIFO buffer rejects any data presented on the inputs. When stopping the REPLAY test mode with the NORMAL or RX_STOP mode, output continues until the EGR_HEAD::HEAD pointer reaches the TAIL pointer. With an initial value of EGR_HEAD::HEAD = EGR_TAIL::TAIL = 0, this implies that replay will output an integer number of FIFO buffer sizes. If the size of the frame data is less than the FIFO buffer size, the replay function outputs whatever is stored in the FIFO buffer RAM. In such a situation in the egress direction, the memory not used by frames should be filled with START, MID planes. 0xC: RTT. Replay To Tail. Starts a cyclic replay of the frames in the FIFO buffer. Data between EGR_TOP_BOTTOM::BOTTOM and EGR_TAIL::TAIL-1 is output continuously, reading from the EGR_HEAD::HEAD location, and the FIFO buffer rejects any data presented on the inputs. When stopping the REPLAY test mode with the NORMAL or RX_STOP mode, output continues until the EGR_HEAD::HEAD pointer reaches 0. If the size of the frame data is less than the FIFO buffer size, the replay function outputs whatever is stored in the FIFO buffer RAM anyway. In such a situation in the egress direction, the memory not used by frames should be filled with START, MID planes.	0x02

S0_STK–S7_STK are sticky bits for debugging. Each bit is set upon a certain condition. The sticky bit is only cleared with a reset of the FIFO, or by writing a 1 to it. The bits numbered T0–T15 enable DEBUG_BUF_CNT to count the events specified by the “T” bit.

4.4.2 FIFO Buffer Top and Bottom

Table 115. EGR_TOP_BOTTOM (Addresses 0x10–0x1B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
22:16	TOP	R/W	Defines the top of the memory area used by the FIFO buffer.	0x00
6:0	BOTTOM	R/W	Defines the bottom of the memory area used by the FIFO buffer.	0x00

The egress FIFO is a 1.125-Mbit buffer. The 1.125-Mbit buffer is shared between a maximum of 12 RGMII ports. EGR_TOP_BOTTOM::TOP and EGR_TOP_BOTTOM::BOTTOM must be defined for each RGMII port. The size and placement for each FIFO buffer can be individually defined through the EGR_TOP_BOTTOM::TOP and EGR_TOP_BOTTOM::BOTTOM pointers. The granularity of EGR_TOP_BOTTOM::TOP and EGR_TOP_BOTTOM::BOTTOM pointers is 2048 bytes.

The FIFO buffer uses memory starting from EGR_TOP_BOTTOM::BOTTOM × 2048 bytes and ending with EGR_TOP_BOTTOM::TOP × 2048 – 32 bytes, which means that the adjacent FIFO buffer's EGR_TOP_BOTTOM::BOTTOM pointer may be set to the same as EGR_TOP_BOTTOM::TOP on the previous buffer.

While writing to this register, EGR_CONTROL::CLR must be equal to 1, or the MODE in the EGR_TEST register must be set to CLEAR_PTR.

The FIFO buffer size is calculated as (EGR_TOP_BOTTOM::TOP – EGR_TOP_BOTTOM::BOTTOM) × 2048 bytes. The maximum a FIFO buffer can contain is (EGR_TOP_BOTTOM::TOP – EGR_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes.

After reset, all 12 FIFO buffers have EGR_TOP_BOTTOM::TOP = 0x00 and EGR_TOP_BOTTOM::BOTTOM = 0x00, which makes all 12 FIFO buffers occupy the same buffer area. For even distribution of memory between 12 ports, use the following configuration:

register 0x10 = 0x00060000, register 0x11 = 0x000C0006, ..., register 0x1B = 0x00480042

Note that all the FIFO buffer size calculations do not include the additional data used internally to keep track of frame boundaries.

4.4.3 Write Pointer

Table 116. EGR_TAIL (Addresses 0x20–0x2B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
12:0	TAIL	R/W	TAIL points to a free 32-byte location where the data received by the FIFO buffer is written. Writing to TAIL is only possible if EGR_CONTROL::CLR = 0, and immediately causes the FIFO buffer content from EGR_HEAD::HEAD up to and including TAIL – 1 to be output (assuming MODE = NORMAL in the ING_TEST register). If EGR_HEAD::HEAD equals TAIL, the FIFO buffer is empty.	0x0000

4.4.4 Read Pointer

Table 117. EGR_HEAD (Addresses 0x30–0x3B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
12:0	HEAD	R/W	HEAD points to an occupied 32-byte location, ready to be transmitted. Under normal usage, HEAD should not be set. Instead, use EGR_TEST::MODE = CLEAR. This initializes HEAD to 64 × EGR_TOP_BOTTOM::BOTTOM. Writing to HEAD is only possible if EGR_CONTROL::CLR = 0. If HEAD equals EGR_TAIL::TAIL, the FIFO buffer is empty.	0x0000

4.4.5 Flow Control Watermarks

Table 118. EGR_HIGH_LOW_WM (Addresses 0x40–0x4B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
28:16	LOW_WM	R/W	Defines the low watermark for flow control.	0x1FFF
12:0	HIGH_WM	R/W	Defines the high watermark for flow control.	0x1FFF

When the FIFO buffer is greater than or equal to the EGR_HIGH_LOW_WM::HIGH_WM number of 32-byte entries, flow control is asserted. When the FIFO buffer contains less than EGR_HIGH_LOW_WM::LOW_WM number of 32-byte entries, flow control is released.

Note that if a suppression of frame fragments is selected (setting EGR_CT_THRHLD::CT_THRHLD = 0), then EGR_HIGH_LOW_WM::LOW_WM must be set to at least one plane higher than the maximum frame size. This avoids a deadlock situation.

If the EGR_HIGH_LOW_WM::LOW_WM and EGR_HIGH_LOW_WM::HIGH_WM default values are used, the MAC flow control must be explicitly disabled, by writing to a MAC pause control register. When calculating FIFO utilization, note that the maximum a FIFO buffer can contain is (EGR_TOP_BOTTOM::TOP – EGR_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes. Frame storage in bytes is rounded up to the nearest byte count divisible by 32.

4.4.6 Cut-Through Threshold

Table 119. EGR_CT_THRHLD (Addresses 0x50–0x5B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
29:16	RESERVED	R/W	Reserved.	0x0000
12:0	CT_THRHLD	R/W	Cut-Through Threshold. The number of 32-byte entries at the start of a frame, which must be present in the FIFO buffer before the output algorithm 'sees' the frame. When the number of entries reaches the CT_THRHLD value, the FIFO buffer begins to transmit the frame. Values higher than 1 can be used to wait for a certain amount a data before the FIFO starts to transmit. In store-and-forward mode, this parameter has no effect. Setting CT_THRHLD = 0 makes the buffer wait for a complete frame before it is forwarded. This is the recommended setting to avoid tri-speed MAC underrun. 0: Store-and-forward mode (recommended) 1-2: Cut-through mode >2: Partly store-and-forward and cut-through mode.	0x0001

4.4.7 Drop Counter

Table 120. EGR_DROP_CNT (Addresses 0x60–0x6B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	DROP_CNT	R/W	Counts the number of frames dropped due to FIFO buffer overflow. DROP_CNT also counts once, when the FIFO buffer is put in RX_STOP mode.At maximum, a FIFO buffer can contain (EGR_TOP_BOTTOM::TOP – EGR_TOP_BOTTOM::BOTTOM) × 2048 – 128 bytes. The S6_STK bit in the EGR_TEST register indicates that a FIFO buffer full condition has occurred. Setting EGR_TEST::COUNT_FAIL also counts frames that are dropped due to CRC failure, and frames terminated with EOP/Abort on SPI-4.2. Frame fragments not terminated with an EOP or EOP/Abort are dropped silently. EGR_TEST::S0_STK indicate if such a situation has happened.	0x00000000

4.4.8 Input Side Debug Counter

Table 121. EGR_DEBUG_BUF_CNT (Addresses 0x70–0x7B), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	DEBUG_BUF_CNT	R/W	Counts the number of trigger events in a single FIFO buffer. These are all related to the input side of the FIFO Buffer. The events to be counted are set in the EGR_TEST register (the T# bits). Note that the counter does not saturate. Simultaneous events are counted just once. To count events related to the output side of the FIFO, see the registers EGR_DEBUG_OUT and EGR_DEBUG_CNT.	0x00000000

4.4.9 SRAM_ADDR

Table 122. EGR_SRAM_ADDR (Address 0x0E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
13:0	SRAM_ADDR	R/W	The address of in the FIFO buffer memory. The address is in 16-byte units, therefore two consecutive addresses corresponds to one address as indexed by HEAD, TAIL. When writing frames to the FIFO RAM, these must start at an even SRAM_ADDR value, that is, on a 32-byte boundary.	0x0000

4.4.10 SRAM_WR_STRB

Table 123. EGR_SRAM_WR_STRB (Address 0x1E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_WR_STRB	W/O	A write to this address initiates a FIFO RAM write. Data contained in EGR_SRAM_DATA_0::SRAM_DATA_0, EGR_SRAM_DATA_1::SRAM_DATA_1 , EGR_SRAM_DATA_2::SRAM_DATA_2, EGR_SRAM_DATA_3::SRAM_DATA_3 and EGR_SRAM_DATA_BLK_TYPE::BLK_TYPE are written to the location contained in EGR_SRAM_ADDR::SRAM_ADDR. The write is delayed until a timeslot with an unused plane is detected in the normal input data stream to the FIFO. Note that the data in SRAM_DATA_x must be bit-swapped within each byte compared to the same byte on, for example the RGMII interface.	0x00000000

4.4.11 SRAM_RD_STRB

Table 124. EGR_SRAM_RD_STRB (Address 0x2E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_RD_STRB	W/O	A write to this address initiates a FIFO RAM read. Data is read from the address contained in EGR_SRAM_ADDR::SRAM_ADDR, and stored inEGR_SRAM_DATA_0::SRAM_DATA_0, EGR_SRAM_DATA_1::SRAM_DATA_1 , EGR_SRAM_DATA_2::SRAM_DATA_2, EGR_SRAM_DATA_3::SRAM_DATA_3 and EGR_SRAM_DATA_BLK_TYPE::BLK_TYPE. If the FIFO is transmitting data while a read is performed, frame data is destroyed. All FIFO buffers must be empty or put in EGR_TEST::MODE = TX_STOP prior to performing a read. Note that the data in SRAM_DATA_x must be bit-swapped within each byte compared to the same byte on, for example the RGMII interface.	0x00000000

4.4.12 SRAM_DATA_0

Table 125. EGR_SRAM_DATA_0 (Address 0x3E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_0	R/W	Byte 0 to 3 of the data read or to be written to the FIFO RAM. Byte 0 is the last byte (of byte 0 to 15) received on the RGMII interface. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.4.13 SRAM_DATA_1

Table 126. EGR_SRAM_DATA_1 (Address 0x4E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_1	R/W	Byte 4 to 7 of the data read or to be written to the FIFO RAM. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.4.14 SRAM_DATA_2

Table 127. EGR_SRAM_DATA_2 (Address 0x5E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_2	R/W	Byte 8 to 11 of the data read or to be written to the FIFO RAM. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.4.15 SRAM_DATA_3

Table 128. EGR_SRAM_DATA_3 (Address 0x6E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	SRAM_DATA_3	R/W	Byte 12 to 15 of the data read or to be written to the FIFO RAM. Byte 15 (bit 24 to 31 in this register) is the first byte received on the RGMII interface. Each byte is bit-swapped relative to the usual IEEE byte notation convention.	0x00000000

4.4.16 SRAM_DATA_BLK_TYPE

Table 129. EGR_SRAM_DATA_BLK_TYPE (Address 0x7E), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
1:0	BLK_TYPE	R/W	The bits determining the type of the stored 16 bytes. START must be used for the first 16 bytes in a frame. All remaining 16-byte blocks must be marked with MID, except for the last data, which is marked with END, if the frame ends on a 16-byte boundary, otherwise PARTIAL is used. 0x0 = START 0x1 = MID 0x2 = PARTIAL, byte 0 in SRAM_DATA_0 indicates number of valid bytes 0x3 = END	0x0

4.4.17 Egress FIFO Master Control

Table 130. EGR_CONTROL (Address 0x0F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:28	OUT_PORT_OFFSET	R/W	Reserved, must be set to 0xA.	0x00
27:24	INP_PORT_OFFSET	R/W	Reserved, must be set to 0x0.	0x00

Table 130. EGR_CONTROL (Address 0x0F), Block 2, Subblock 1 *(continued)*

Bit	Name	Mode	Description	Default
21	EXT_SCH_EN	R/W	Not used in Egress FIFO, must be set to 0x0.	0x0
20	EXT_FRM_MODE	R/W	Not used in Egress FIFO, must be set to 0x0.	0x0
19	SRR	R/W	Debug. Default value may not be changed.	0x0
18	FAIL_IGN	R/W	Fail Ignore. When set, ignores the fail signal from the CRC checker, or the SPI-4.2 egress block.	0x0
17:16	MUX	R/W	Selects the source of inhibit signal to the FIFO buffers. 0x1: Inhibit signals are sourced by tri-speed MACs.	0x0
15	ETFC	R/W	Egress Transparent Flow Control enable. When enabled, the egress flow control status is OR'ed to the EGR_HIGH_LOW_WM::HIGH_WM/ EGR_HIGH_LOW_WM::LOW_WM generated and passed on to the SPI-4.2 egress status channel. RX_PAUSE_EN or PAUSE_EN in device register PAUSE_CFG must be set, and for a transparent setup EGR_HIGH_LOW_WM::HIGH_WM and EGR_HIGH_LOW_WM::LOW_WM should be set to the maximum value. Note that even though the information about a flow control frame received on the ingress line is passed transparently on, the egress Ethernet data stream is stopped immediately. 0: Disabled. 1: Enabled.	0x0
14	TC	R/W	Not used in Egress FIFO, must be set to 0x0.	0x0
13	LE	R/W	Not used in Egress FIFO, must be set to 0x0.	0x0
12	IGI	R/W	Ignore Inhibit. Debug purposes only. When set, the inhibit input to the FIFO is ignored. Setting IGI disables shaper functionality. 0: Normal mode. Inhibit to the FIFO stops transmission (recommended). 1: The inhibit input to the FIFO is ignored.	0x0
11	ITFC	R/W	Not used in Egress FIFO, must be set to 0x0.	0x0
10	RT	R/W	Internal debug feature. Do not change the default value.	0x0
9	PH	R/W	Internal debug feature. Do not change the default value.	0x0

Table 130. EGR_CONTROL (Address 0x0F), Block 2, Subblock 1 *(continued)*

Bit	Name	Mode	Description	Default
8	NH	R/W	Normalized header. Used by the FIFO when transmitting in full-frame mode. Setting should correspond with the settings in the MAC devices. Setting the NH bit is only necessary when EGR_CONTROL::CM = 0. 0: frames are without normalized header. 1: frames are with normalized header.	0x0
7	SS	R/W	Not used in Egress FIFO. Should always be set to 0x1.	0x0
6	CLR	R/W	Clear all FIFO Buffers. When reset is released, a state machine starts clearing the FIFO RAM locations. During this process, writing to the RAM is blocked. The CLR bit should therefore not be cleared until after 120 µs after reset has been released. 0: FIFO is active. 1: FIFO pointers are cleared. FIFO is not active.	0x1
5	M10G	R/W	Not used. Must be set to 0x0.	0x1
4	CM	R/W	Internal debug use. Must be set to 0x1.	0x1
3:0	BURST_LEN	R/W	Internal debug use. Must be set to 0x1.	0x01

4.4.18 Egress Aging Timer

Table 131. EGR_AGE_TIMER (Address 0x1F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	AGE_TIMER	R/W	The Age Timer counter is incremented for every second system clock by EGR_AGE_INC::AGE_INC amount. When the 32 bit AGE_TIMER counter overflows, an age time tick is generated. Frames that have been stored in the FIFO buffer for longer than timeout duration are marked as aged. The timeout duration is > TIME_TICK and ≤ 2 × TIME_TICK. TIME_TICK = 2×2^{32} / (150.0 MHz × AGE_INC). The time tick updates two shadow registers in each of the 12 FIFO Buffers.	0x00000000

4.4.19 Egress Aging Timer Increment

Table 132. EGR_AGE_INC (Address 0x2F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
27:0	AGE_INC	R/W	The Aging Increment value to added to EGR_AGE_INC::AGE_TIMER every second system clock. When the 32 bit AGE_TIMER counter overflows, an age time tick is generated. Frames that have been stored in the FIFO buffer for longer than timeout duration are marked as aged. The timeout duration is > TIME_TICK and ≤ 2 × TIME_TICK, TIME_TICK = 2×2^{32} / (150.0 MHz × AGE_INC). The exact time out duration is dependent on the arrival of the frame relative to the age time tick. All 12 buffers in the FIFO are aged by the same timer. Using 0x0000000 disables aging of frames. 0x0000074 ages frames stored for more than 1 second.	0x00000000

4.4.20 Egress Output Side Debug Counter Control

Table 133. EGR_DEBUG_OUT (Address 0x3F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
23:22	T7_T6	R/W	Unused.	0x0
21	T5	R/W	Reserved.	0x0
20	T4	R/W	Read 32 valid bytes (= 1 plane) (burst-interleaved and full-frame mode).	0x0
19:16	T3_T0	R/W	Reserved.	0x00
15:11	S7_S3_STK	R/W	Unused.	0x00
10	S2_STK	R/W	Non-allowed plane seen on the input side, with first halfplane having valid = 0, and second plane valid = 1. Indicates reception of SPI-4.2 junk data.	0x0
9	S1_STK	R/W	Reserved.	0x0
8	S0_STK	R/W	Reserved.	0x0

S0_STK to S7_STK are sticky bits for debugging. Each bit is set upon a certain condition. The sticky bit is only cleared with a reset of the FIFO, or by writing a 1 to it. Each of the T0 to T7 bits enables counting by EGR_DEBUG_CNT of certain event specified.

4.4.21 Output Side Debug Counter

Table 134. EGR_DEBUG_CNT (Address 0x4F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
31:0	DEBUG_CNT	R/W	Counts the number of FIFO trigger events. These are set in the EGR_DEBUG_OUT register. Simultaneous events are counted just once.	0x00000000

4.4.22 Per Port Inhibit Control

Table 135. PPORT_INH (Address 0x5F), Block 2, Subblock 1

Bit	Name	Mode	Description	Default
27:16	PPORT_IPT	R/W	Internal debug use. Must be set to 0x000.	0x0000
11:0	PPORT_IGI	R/W	Internal debug use. Must be set to 0x000.	0x0000

4.5 Shaper and Policer, Block 2

There is one bucket (shaper) for each port buffer plus one common bucket for all buffers. Subblock 2 controls all ingress traffic buckets, subblock 3 controls all egress traffic buckets.

4.5.1 Traffic Shaper Control

Table 136. SHAPER_CONTROL (Address 0x00), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
31:26	FRM_LEN_ADJ	R/W	Frame Length Adjustment for per port shaper. This value is added (modulus 64) to the frame length counter, after the first 16 bytes of data are received from the FIFO. A preamble header or a normalized header is also counted in the frame length. Example: Shaping is desired to take the Ethernet 12-byte IPG and 8-byte preamble into account, and the frames have a 9-byte normalized header: FRM_LEN_ADJ = 12 + 8 – 9 = 11.	0x14

Table 136. SHAPER_CONTROL (Address 0x00), Block 2, Subblock 2–3 *(continued)*

Bit	Name	Mode	Description	Default
25	SPI4_ADJUST	R/W	Should be set if the total bandwidth is limited by the SPI-4.2 bandwidth, and fairness share of bandwidth between the ports should be obtained. Note: To obtain fairness between ports, the individual shapers have to be set, that is, to the wire speed of the port, and COMMON_SHAPER_BUCKET::BUCKET_RATE = 0xAD5E (13 Gbps).	0x0
24:23	COMMON_STOP_CFG	R/W	Stop Configuration for common Shaper, 0x0: Common shaper does not flow-control FIFOs directly when common shaper level is reached, but flow controls indirectly through the individual shapers (common shaper is used to give fairness). (recommended) 0x1: Common Shaper flow controls FIFO on a per-port basis. 0x2: Common Shaper flow controls FIFO on all ports at once basis. 0x3: Not allowed.	0x0
22	RESERVED	R/W	Must be set to 0x0 if shapers are used.	0x1
21	EE	R/W	Enable at EOF: Clear link level enable at end of frame. Should be set for egress shapers and cleared for ingress shapers.	0x0
20	IE	R/W	Inhibit at EOF: Set per port inhibit signals at end of frame in FIFO burst-interleaved mode. Should be set for egress shapers, and cleared for ingress shapers.	0x0
19:15	PORT_OFFSET	R/W	Must be set to 0xC for egress shaper and 0x0 for ingress shaper.	0x00
12	ADD_HIGH_PRIO_TO_COMMON_SHAPER	R/W	When cleared the common shaper counts bandwidth for all ports, otherwise it only counts low priority ports.	0x1
11	CLR_TO_BLVL	R/W	Clear to bucket level. When set, a clear bucket prefills the bucket to SHAPER_BUCKET::BUCKET_LEVEL. The prefill level is kept unchanged, that is, not diminished by the SHAPER_BUCKET::BUCKET_RATE, until leaky bucket sees the first data. Setting this bit avoids the initial bandwidth overshoot when the leaky bucket is started the first time. This bit must to set/cleared before clearing the SHAPER_CLR_BUCKETS::CLEAR_BUCKETS bits.	0x0

Table 136. SHAPER_CONTROL (Address 0x00), Block 2, Subblock 2–3 *(continued)*

Bit	Name	Mode	Description	Default
5:0	FRM_LEN_ADJ_COMMON	R/W	Frame Length Adjustment for common shaper. This value is added (modulus 64) to the frame length counter, after the first 16 bytes of data are received from the FIFO. A preamble header or a normalized header is also counted in the frame length. Example: Shaping is desired to take the Ethernet 12-byte IPG, and 8-byte preamble into account, and the frames have a 9-byte normalized header: FRM_LEN_ADJ_COMMON = 12 + 8 – 9 = 11.	0x00

4.5.2 Shaper Priority

Table 137. SHAPER_PRIO (Address 0x01), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
11:0	PRIO	R/W	Each bit sets the priority for the corresponding bucket when set high. If a bucket is set to high priority, it does not respond on the common bucket. Note: When setting a shaper to high priority, make sure that the total bandwidth for the high priority ports is below the common shaper bandwidth.	0x0000000

4.5.3 Timer

Table 138. SHAPER_TIMER (Address 0x02), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
31	EN	R/W	Enable Timer. When the timer is enabled ports are stopped when timer value is zero.	0x0
30:0	VALUE	R/W	Current value of timer. The timer can be used for testing or debugging. The timer can turn on all low priority shapers for a specified amount of time with very high accuracy. When the timer is enabled, its value decreases by 1 for each system clock cycle (6.66 ns). Thus ports can be set up to stop transmitting in the interval from 0 to 14.32 seconds.	0x00000000

4.5.4 Clear Buckets

Table 139. SHAPER_CLR_BUCKETS (Address 0x03), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
24	COMMON_CLEAR_BUCKETS	R/W	Clears the common bucket (disables the common shaper).	0x1
11:0	CLEAR_BUCKETS	R/W	Each bit clears the corresponding bucket when set high (disable shaper).	0xFFF

4.5.5 Traffic Slow Rate Control

Table 140. SHAPER_SLOW_RATE_CONTROL (Address 0x04), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
12	SLOW_DOWN_EN	R/W	The shapers can stop the frame rate in two ways. One way is by simply signalling full stop or full speed. The other way to do it, is a three step method where the shaper either signals full stop, slow down or full speed. Setting this bit to zero sets shapers to use the full stop / full speed method. Setting this bit to one sets shapers to use the three step method. This bit should be set when using shapers to give SPI-4.2 bandwidth fairness.	0x0
11:8	FRM_SLOW_DOWN_RATIO	R/W	When SHAPER_SLOW_RATE_CONTROL::SLOW_DOWN_EN is set to 1, this register determines how much the rate is slowed down (compared to full speed). The rate can be calculated as -- pct slow down = (100 – value × 100/16) % 0x0 : Frame rate is slowed down 100 %. (Full stop) 0x1 : Frame rate is slowed down to (100 – 1 × 100 / 16) 93.8 % 0x2 : Frame rate is slowed down to (100 – 2 × 100 / 16) 87.5 % 0xF : Frame rate is slowed down to (100 – 15 × 100 /16) 6.25 % This is needed when using shapers to give SPI-4.2 bandwidth fairness, and should be set to its default value.	0x0A

Table 140. SHAPER_SLOW_RATE_CONTROL (Address 0x04), Block 2, Subblock 2–3 *(continued)*

Bit	Name	Mode	Description	Default
3:0	FRM_SLOW_DOWN_LVL	R/W	When SHAPER_SLOW_RATE_CONTROL::SLOW_DOWN_EN is set to 1, the shaper uses the full stop, slow down, and then full speed mode. The slow down period is determined by this register. When the shaper level is above SHAPER_BUCKET::BUCKET_LEVEL and below SHAPER_BUCKET::BUCKET_LEVEL + FRM_SLOW_DOWN_LVL, the frame rate is slowed down. When the shaper level is above SHAPER_BUCKET::BUCKET_LEVEL + FRM_SLOW_DOWN_LVL, the frame rate is set to full stop. This is needed when using shapers to give SPI-4.2 bandwidth fairness, and should be set to its default value.	0x0A

4.5.6 Traffic Shaper Bucket Setting

Table 141. SHAPER_BUCKET (Addresses 0x20–0x2B), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
31:16	BUCKET_LEVEL	R/W	Defines the fill level of the leaky bucket, where the traffic shaper backpressures into the core of the FIFO. The range of BUCKET_LEVEL is (0 – 65535) × 128 bytes.	0x7FFF
15:0	BUCKET_RATE	R/W	Defines the sustained rate for the leaky bucket. The bandwidth for this bucket is BUCKET_RATE × 146484.375 bps. The bucket counts all bytes coming out from a single FIFO buffer. This includes all eventual header information attached to the frame information.	0xFFFF

4.5.7 Common Traffic Shaper Bucket Setting

Table 142. COMMON_SHAPER_BUCKET (Address 0x38), Block 2, Subblock 2–3

Bit	Name	Mode	Description	Default
31:16	BUCKET_LEVEL	R/W	Defines the fill level of the leaky bucket, where the traffic shaper backpressures into the core of the FIFO. The range of BUCKET_LEVEL is (0 – 65535) × 128 bytes. Recommended value is BUCKET_LEVEL = 0x2.	0x7FFF

Table 142. COMMON_SHAPER_BUCKET (Address 0x38), Block 2, Subblock 2–3 *(continued)*

Bit	Name	Mode	Description	Default
15:0	BUCKET_RATE	R/W	Defines the sustained rate for the leaky bucket. The bandwidth for this bucket which is BUCKET_RATE × 292968.75 bps. The bucket counts all bytes coming out of the FIFO regardless of which FIFO buffer the data belongs to. This includes all eventual header information attached to the frame information. When using shapers for obtaining bandwidth fairness BUCKET_RATE has to be set slightly higher than the SPI-4.2 bandwidth.	0xFFFF

4.6 MII-Management, Block 3

MIIM0 is placed in subblock 0, and MIIM1 is placed in subblock 1.

4.6.1 MII-M Status

Table 143. MIIM_STATUS (Address 0x00), Block 3, Subblock 0–1

Bit	Name	Mode	Description	Default
4	MIIM_BUSY	R/O	MIIM I/F busy. Indicates that the MII-Management interface is busy (serving a pending read/write operation).	0x0
3	OPR_PEND	R/O	Operation Pending. A CPU initiated operation is pending — waiting for the MII-M I/F to be idle.	0x0
2	OPR_PROG	R/O	Operation in Progress. Indicates that a CPU initiated operation is in progress.	0x0
1:0	OPCODE	R/O	Opcode of last command.	0x0

To determine when the next command can be issued, proceed as follows:

- READ—Before issuing a read command to MIIM_CMD, verify that the field MIIM_STATUS::OPR_PEND is 0. After issuing the read command, the value in MIIM_DATA is valid when fields MIIM_STATUS::MIIM_BUSY, MIIM_STATUS::OPR_PEND and MIIM_STATUS::OPR_PROG are all-zeros.
- WRITE—Before issuing a write command to MIIM_CMD, verify that field MIIM_STATUS::OPR_PEND is 0.

4.6.2 MII-M Command

Table 144. MIIM_CMD (Address 0x01), Block 3, Subblock 0–1

Bit	Name	Mode	Description	Default
31:16	PHY_DATA	R/W	Contains the data written the specified register in the selected PHY during write operation.	0x0000
13:9	PHY_ADDR	R/W	Specifies the address of the PHY / the Port Address in the MMD.	0x00
8:4	PHY_REG_ADDR	R/W	Specifies the address of the register in the selected PHY	0x00
3:2	OPR	R/W	Determines the operation type. 0x0: Reserved. 0x1: Write. 0x2: Read. 0x3: Reserved.	0x0
1:0	MODE	R/W	Determines the access mode. 0x0: Reserved. 0x1: PHY access [802.3 2000, clause 22]. 0x2: Reserved. 0x3: Reserved.	0x0

4.6.3 MII-M Data

Table 145. MIIM_DATA (Address 0x02), Block 3, Subblock 0–1

Bit	Name	Mode	Description	Default
16	RS	R/O	Indicates that the previous read operation was successful and read data is valid. 1: Read failed. 0: Read succeeded.	0x0
15:0	PHY_DATA	R/O	Contains the data read from the specified register in the selected PHY during read operations. The register contains valid data when MIIM_STATUS::MIIM_BUSY, MIIM_STATUS::OPR_PEND and MIIM_STATUS::OPR_PROG are all-zeros. This register is also updated on a MIIM write operation, with the MIIM data written.	0x0000

4.6.4 MII-M MDC Pre-scale

Table 146. MIIM_PRESCALE (Address 0x03), Block 3, Subblock 0–1

Bit	Name	Mode	Description	Default
5:0	PRESCALE	R/W	MDC pre-scale factor. Contains the MDC clock pre-scale factor that is used to generate the MDC clock frequency. Pre-scale = roundup ((75 MHz / MDC_FREQ) – 1) IEEE Std 802.3 states that the maximum frequency of MDC is 2.5 MHz. This requirement results in a pre-scale factor that equals 0x1D.	0x1D

4.7 Statistic Counters, Block 4

All statistics counters are listed as read and write. However, when a write is issued to a specific counter at the same time the counter is incremented, the value written to the counter is ignored. In this case, the value in the counter would not be accurate. For this reason the write feature is only for diagnostic purposes and should not be used for regular operation. In most cases STAT_INIT::CLR_ALL should be used instead.

4.7.1 Rx Byte Counter

Table 147. RX_IN_BYTES_CNT (Address 0x00), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_IN_BYTES	R/W	RX_IN_BYTES: The number of nibbles received (good, bad, and framing) in 10 Mbps or 100 Mbps mode or the number of bytes received (good, bad, and framing) in 1 Gbps mode. Note: Framing is Preamble + SFD.	0x00000000

4.7.2 Rx Symbol Carrier Error Counter

Table 148. RX_SYMBOL_CARRIER_ERR_CNT (Address 0x01), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SYMBOL_CARRIER	R/W	RX_SYMBOL_CARRIER: The number of frames received with one or more symbol errors, excluding collision.	0x00000000

4.7.3 Rx Pause Frame Counter

Table 149. RX_PAUSE_CNT (Address 0x02), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_PAUSE	R/W	Number of pause control frames received.	0x00000000

4.7.4 Rx Control Frame Counter

Table 150. RX_UNSUP_OPCODE_CNT (Address 0x03), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_UNSUP_OPCODE	R/W	Number of control frames with unsupported OPCODE received. Unsupported refers to all control frames other than pause control frames.	0x00000000

4.7.5 Rx OK Byte Counter

Table 151. RX_OK_BYTES_CNT (Address 0x04), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_OK_BYTES	R/W	The number of received bytes in good frames. Does not include preamble or SFD. All frames with OK CRC are counted, including oversize and undersize frames and frames with a length error.	0x00000000

4.7.6 Rx Bad Byte Counter

Table 152. RX_BAD_BYTES_CNT (Address 0x05), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_BAD_BYTES	R/W	The number of received bytes in bad frames. Does not include preamble or SFD. Same as RX_OK_BYTES, but includes frames with CRC errors. Thus, for all fragments that contain an SFD, the bytes after the SFD are counted.	0x00000000

4.7.7 Rx Unicast Frame Counter

Table 153. RX_UNICAST_CNT (Address 0x06), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_UNICAST	R/W	The number of good unicast frames received.	0x00000000

4.7.8 Rx Multicast Frame Counter

Table 154. RX_MULTICAST_CNT (Address 0x07), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_MULTICAST	R/W	The number of good multicast frames received.	0x00000000

4.7.9 Rx Broadcast Frame Counter

Table 155. RX_BROADCAST_CNT (Address 0x08), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_BROADCAST	R/W	The number of good broadcast frames received.	0x00000000

4.7.10 Rx CRC Error Counter

Table 156. RX_CRC_ERR_CNT (Address 0x09), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_CRC	R/W	The number of frames received with CRC errors only.	0x00000000

4.7.11 Rx Alignment Error Counter

Table 157. RX_ALIGNMENT_ERR_CNT (Address 0x0A), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_ALIGNMENT	R/W	The number of frames received with Alignment errors. An alignment error is a frame that has a. An uneven amount of nibbles counting from the SFD and b. A CRC error when ignoring the last nibble.	0x00000000

4.7.12 Rx Undersize Counter (Valid Frame Format)

Table 158. RX_UNDERSIZE_CNT (Address 0x0B), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_UNDERSIZE	R/W	The number of undersize well-formed frames received.	0x00000000

4.7.13 Rx Undersize Counter (CRC Error)

Table 159. RX_FRAGMENTS_CNT (Address 0x0C) Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_FRAGMENTS	R/W	The number of undersize frames with CRC errors received. Single SFD at 1 Gbps are not counted.	0x00000000

4.7.14 Rx In-Range Length Error Counter

Table 160. RX_IN_RANGE_LENGTH_ERR_CNT (Address 0x0D), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_IN_RANGE_LENGTH_ ERROR	R/W	The number of frames with a legal length field that does not match length of MAC client data.	0x00000000

4.7.15 Rx Out-Of-Range Length Error Counter

Table 161. RX_OUT_OF_RANGE_ERR_CNT (Address 0x0E), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_OUT_OF_RANGE_ERROR	R/W	The number of frames with an illegal length field (frames using type field are not counted here).	0x00000000

4.7.16 Rx Oversize Counter (Valid Frame Format)

Table 162. RX_OVERSIZE_CNT (Address 0x0F), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_OVERSIZE	R/W	The number of oversize well-formed frames received. A frame is oversize if: a. length is > MAX_LEN_CFG::MAX_LEN *and* frame is not VLAN tagged, or b. length is > MAX_LEN_CFG::MAX_LEN + 4 *and* frame is VLAN tagged *and* MAC is configured to be VLAN aware (MODE_CFG::VLAN_AWR = 1). The length does not include preamble and SFD, but includes CRC.	0x00000000

4.7.17 Rx Jabbers Counter

Table 163. RX_JABBERS_CNT (Address 0x10), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_JABBERS	R/W	The number of oversize frames with CRC errors received.	0x00000000

4.7.18 Rx 64 Byte Frame Counter

Table 164. RX_SIZE64_CNT (Address 0x11), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE64	R/W	The number of 64-byte frames received.	0x00000000

4.7.19 Rx 65 to 127 Byte Frame Counter

Table 165. RX_SIZE65TO127_CNT (Address 0x12), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE65TO127	R/W	The number of 65-byte to 127-byte frames received.	0x00000000

4.7.20 Rx 128 to 525 Byte Frame Counter

Table 166. RX_SIZE128TO255_CNT (Address 0x13), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE128TO255	R/W	The number of 128-byte to 255-byte frames received.	0x00000000

4.7.21 Rx 256 to 511 Byte Frame Counter

Table 167. RX_SIZE256TO511_CNT (Address 0x14), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE256TO511	R/W	The number of 256-byte to 511-byte frames received.	0x00000000

4.7.22 Rx 512 to 1023 Byte Frame Counter

Table 168. RX_SIZE512TO1023_CNT (Address 0x15), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE_512_TO_1023	R/W	The number of 512-byte to 1023-byte frames received.	0x00000000

4.7.23 Rx 1024 to 1518 Byte Frame Counter

Table 169. RX_SIZE1024TO1518_CNT (Address 0x16), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE_1024_TO_1518	R/W	The number of 1024-byte to 1518-byte frames received.	0x00000000

4.7.24 Rx 1519 To Maximum Length Byte Frame Counter

Table 170. RX_SIZE1519TOMAX_CNT (Address 0x17), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_SIZE_1519_TO_MAX	R/W	The number of frames received longer than 1518 bytes and not longer than MAXLEN_CFG::MAX_LEN, or not longer than MAXLEN_CFG::MAX_LEN + 4, if the frame is VLAN tagged.	0x00000000

4.7.25 Tx Byte Counter

Table 171. TX_OUT_BYTES_CNT (Address 0x18), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_OUT_BYTES	R/W	The number of bytes transmitted (good, bad and framing) when in 10/100 Ethernet mode.	0x00000000

4.7.26 Tx Pause Frame Counter

Table 172. TX_PAUSE_CNT (Address 0x19), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_PAUSE	R/W	The number of pause control frames transmitted.	0x00000000

4.7.27 Tx OK Byte Counter

Table 173. TX_OK_BYTES_CNT (Address 0x1A), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_OK_BYTES	R/W	The number of bytes transmitted successfully. Does not include preamble or SFD. Note that unlike RX_OK_BYTES the counter also counts bytes for frames with CRC errors.	0x00000000

4.7.28 Tx Unicast Frame Counter

Table 174. TX_UNICAST_CNT (Address 0x1B), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_UNICAST	R/W	The number of unicast frames transmitted.	0x00000000

4.7.29 Tx Multicast Frame Counter

Table 175. TX_MULTICAST_CNT (Address 0x1C), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_MULTICAST	R/W	The number of multicast frames transmitted.	0x00000000

4.7.30 Tx Broadcast Frame Counter

Table 176. TX_BROADCAST_CNT (Address 0x1D), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BROADCAST	R/W	The number of broadcast frames transmitted.	0x00000000

4.7.31 Tx Multiple Collision Frame Counter

Table 177. TX_MULTI_COLL_CNT (Address 0x1E), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_MULTI_COLL	R/W	The number of frames transmitted without errors after multiple collisions when in 10/100 Ethernet mode.	0x00000000

4.7.32 Tx Late Collision Counter

Table 178. TX_LATE_COLL_CNT (Address 0x1F), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_LATE_COLL	R/W	The number of late collisions detected when in 10/100 Ethernet mode.	0x00000000

4.7.33 Tx Excessive Collision Counter

Table 179. TX_XCOLL_CNT (Address 0x20), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_XCOLL	R/W	The number of frames lost due to excessive collision when in 10/100 Ethernet mode.	0x00000000

4.7.34 Tx First Defer Counter

Table 180. TX_DEFER_CNT (Address 0x21), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_DEFER	R/W	The number of frames being deferred on first transmission attempt when in 10/100 Ethernet mode.	0x00000000

4.7.35 Tx Excessive Defer Counter

Table 181. TX_XDEFER_CNT (Address 0x22), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_XDEFER	R/W	The number of frames sent with excessive deferral when in 10/100 Ethernet mode.	0x00000000

4.7.36 Tx Carrier Sense Error Counter

Table 182. TX_CSENSE_CNT (Address 0x23), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_CSENSE	R/W	The number of times CarrierSenseError is true at the end of a frame transmission when in 10/100 Ethernet mode.	0x00000000

4.7.37 Tx 64 Byte Frame Counter

Table 183. TX_SIZE64_CNT (Address 0x24), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_64	R/W	The number of 64-byte frames transmitted.	0x00000000

4.7.38 Tx 65 to 127 Byte Frame Counter

Table 184. TX_SIZE65TO127_CNT (Address 0x25), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_65_TO_127	R/W	The number of 65-byte to 127-byte frames transmitted.	0x00000000

4.7.39 Tx 128 to 255 Byte Frame Counter

Table 185. TX_SIZE128TO255_CNT (Address 0x26), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_128_TO_255	R/W	The number of 128-byte to 255-byte frames transmitted.	0x00000000

4.7.40 Tx 256 to 511 Byte Frame Counter

Table 186. TX_SIZE256TO511_CNT (Address 0x27), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_256_TO_511	R/W	The number of 256-byte to 511-byte frames transmitted.	0x00000000

4.7.41 Tx 512 to 1023 Byte Frame Counter

Table 187. TX_SIZE512TO1023_CNT (Address 0x28), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_512_TO_1023	R/W	The number of 512-byte to 1023-byte frames transmitted.	0x00000000

4.7.42 Tx 1024 to 1518 Byte Frame Counter

Table 188. TX_SIZE1024TO1518_CNT (Address 0x29), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_1024_TO_1518	R/W	The number of 1024-byte to 1518-byte frames transmitted.	0x00000000

4.7.43 Tx 1519 to Maximum Length Byte Frame Counter

Table 189. TX_SIZE1519TOMAX_CNT (Address 0x2A), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SIZE_1519_TO_MAX	R/W	The number of frames received longer than 1518 bytes and not longer than MAXLEN_CFG::MAX_LEN, or not longer than MAXLEN_CFG::MAX_LEN + 4, if the frame is VLAN tagged.	0x00000000

4.7.44 Tx Single Collision Counter

Table 190. TX_SINGLE_COLL_CNT (Address 0x2B), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_SINGLE_COL	R/W	The number of frames transmitted without errors after a single collision.	0x00000000

4.7.45 Tx 2 Backoff Counter

Table 191. TX_BACKOFF2_CNT (Address 0x2C), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF2	R/W	Number of frames sent successfully after 2 backoffs or collisions.	0x00000000

4.7.46 Tx 3 Backoff Counter

Table 192. TX_BACKOFF3_CNT (Address 0x2D), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF3	R/W	Number of frames sent successfully after 3 backoffs or collisions.	0x00000000

4.7.47 Tx 4 Backoff Counter

Table 193. TX_BACKOFF4_CNT (Address 0x2E), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF4	R/W	Number of frames sent successfully after 4 backoffs or collisions.	0x00000000

4.7.48 Tx 5 Backoff Counter

Table 194. TX_BACKOFF5_CNT (Address 0x2F), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF5	R/W	Number of frames sent successfully after 5 backoffs or collisions.	0x00000000

4.7.49 Tx 6 Backoff Counter

Table 195. TX_BACKOFF6_CNT (Address 0x30), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF6	R/W	Number of frames sent successfully after 6 backoffs or collisions.	0x00000000

4.7.50 Tx 7 Backoff Counter

Table 196. TX_BACKOFF7_CNT (Address 0x31), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF7	R/W	Number of frames sent successfully after 7 backoffs or collisions.	0x00000000

4.7.51 Tx 8 Backoff Counter

Table 197. TX_BACKOFF8_CNT (Address 0x32), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF8	R/W	Number of frames sent successfully after 8 backoffs or collisions.	0x00000000

4.7.52 Tx 9 Backoff Counter

Table 198. TX_BACKOFF9_CNT (Address 0x33), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF9	R/W	Number of frames sent successfully after 9 backoffs or collisions.	0x00000000

4.7.53 Tx 10 Backoff Counter

Table 199. TX_BACKOFF10_CNT (Address 0x34), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF10	R/W	Number of frames sent successfully after 10 backoffs or collisions.	0x00000000

4.7.54 Tx 11 Backoff Counter

Table 200. TX_BACKOFF11_CNT (Address 0x35), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF11	R/W	Number of frames sent successfully after 11 backoffs or collisions.	0x00000000

4.7.55 Tx 12 Backoff Counter

Table 201. TX_BACKOFF12_CNT (Address 0x36), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF12	R/W	Number of frames sent successfully after 12 backoffs or collisions.	0x00000000

4.7.56 Tx 13 Backoff Counter

Table 202. TX_BACKOFF13_CNT (Address 0x37), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF13	R/W	Number of frames sent successfully after 13 backoffs or collisions.	0x00000000

4.7.57 Tx 14 Backoff Counter

Table 203. TX_BACKOFF14_CNT (Address 0x38), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF14	R/W	Number of frames sent successfully after 14 backoffs or collisions.	0x00000000

4.7.58 Tx 15 Backoff Counter

Table 204. TX_BACKOFF15_CNT (Address 0x39), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_BACKOFF15	R/W	Number of frames sent successfully after 15 backoffs or collisions.	0x00000000

4.7.59 Tx Underflow Counter

Table 205. TX_UNDERRUN_CNT (Address 0x3A), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	TX_UNDERRUN	R/W	Number of times MAC transmit FIFO has dropped a frame because of under run. Can be caused by using cut-through mode with too small a threshold in the egress FIFO.	0x00000000

4.7.60 Rx Inter-Packet Gap Shrink Counter

Table 206. RX_IPG_SHRINK_CNT (Address 0x3C), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	RX_IPG_SHRINK	R/W	The number of times an IPG shrink was detected. Single SFD's at 1 Gbps are ignored. An IPG shrink is when the inter-packet gap (idle period between frames) is less than 12 bytes.	0x00000000

4.7.61 Statistics Tri-Speed MAC Sticky Bit

Table 207. STAT_STICKY1G (Address 0x3E), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
11	CEXT_STK	R/W	Indicates that a carrier extend was detected.	0x0
10	CEXTE_STK	R/W	Indicates that a carrier extend error was detected.	0x0

Table 207. STAT_STICKY1G (Address 0x3E), Block 4, Subblock 0–11 *(continued)*

Bit	Name	Mode	Description	Default
9	JUNK_STK	R/W	Indicates that junk was received (bytes not recognized as a frame).	0x0
8	IPGS_STK	R/W	Indicates that an inter packet gap shrink wad detected (IPG < 12 bytes).	0x0
7	PRES_STK	R/W	Indicates that a preamble shrink was detected (preamble < 8 bytes).	0x0
4	RXMIT_STK	R/W	Indicates that the transmit MAC asked host for a frame retransmission.	0x0
3	JAM_STK	R/W	Indicates that the transmit host issued a jamming signal.	0x0
2	ABORT_STK	R/W	Indicates that the transmit host initiated abort was executed.	0x0
1	FIFO_O_STK	R/W	Indicates that the MAC transmit FIFO has overrun.	0x0
0	FLENO_STK	R/W	Indicates that the transmit frame length including the 8 byte preamble has overrun (frame > 64 kilobyte).	0x0

Writing a bit mask to this register clears the sticky bits at the positions where the bit mask is 1.

4.7.62 Initialize Statistics

Table 208. STAT_INIT (Address 0x3F), Block 4, Subblock 0–11

Bit	Name	Mode	Description	Default
31:0	CLR_ALL	W/O	A write to this field in any of the subblocks 0x0 through 0xB clears the statistics in all subblocks from 0x0 to 0xB.	0x00000000

4.8 SPI-4.2 Host Interface, Block 5

4.8.1 Master Configuration

Table 209. SPI4_MISC (Address 0x00), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
30	INGCLK	R/W	Ingress Clock Clear. Asserts reset and freezes the clock on the SPI-4.2 data ingress block not using the system clock. 0: Normal operation. 1: Reset active, clock frozen.	0x1
29	EGRCLK	R/W	Egress Clock Clear. Asserts reset and freezes the clock on the SPI-4.2 egress block not using the system clock. This reset should preferably be released after any kind of startup junk data on the SPI-4.2 data bus has passed. Otherwise counts may occur in the DIP-4 error counter (C8::CNT), or in the CRC error counter (EGR_CRC_CNT::EGR_ERR_CNT), first time data passes through the egress SPI-4.2. 0: Normal operation. 1: Reset active, clock frozen.	0x1
28	ISYSCLK	R/W	Ingress System Clock Clear. Asserts reset on SPI-4.2 ingress block using the system clock. Asserting ISYSCLK and INGCLK at the same time resets the ingress part of the SPI-4.2 block. 0: Normal operation. 1: Reset active.	0x0
27	ESYSCLK	R/W	Egress System Clock Clear. Asserts reset on SPI-4.2 egress block using the system clock. Asserting ESYSCLK and EGRCLK as the same time resets the egress part of the SPI-4.2 block. 0: Normal operation. 1: Reset active.	0x0
26	COLLAPSER_DIS	R/W	Internal debug use. Do not change the default value.	0x0
25	TRAINSUPR_DIS	R/W	Internal debug use. Do not change the default value.	0x0
24:20	PACKET_EXTEND	R/W	Internal debug use. Do not change the default value.	0x00
19	LD	R/W	Loopback connection internally in Schaumburg. Data can be injected on both the SPI-4.2 and the RGMII side. When enabled, SPI4_TEST::FORCE_HUNGRY should also be set. 0: Normal operation. 1: Data is looped from SPI-4.2 egress to SPI-4.2 ingress and from ingress FIFO to egress FIFO.	0x0

Table 209. SPI4_MISC (Address 0x00), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
18	SD	R/W	Swap Data. Swaps the bits within each byte. For compliant SPI-4.2 operation SD must be 1. 0: SPI-4.2 bit 8 and bit 0 are Ethernet most significant bits. 1: SPI-4.2 bit 15 and bit 7 are Ethernet most significant bits (recommended).	0x1
17	DI	R/W	Internal debug use. Do not change the default value.	0x0
16	DE	R/W	Internal debug use. Do not change the default value.	0x0
15:11	BURST_EXTEND	R/W	Internal debug use. Do not change the default value.	0x00
10	EN_DIP4	R/W	Enable packet-discard upon DIP-4 errors on the egress input data. DIP-4 errors are counted by the debug counters regardless of the setting of this bit. 0: DIP-4 Discarding disabled. 1: DIP-4 Discarding enabled.	0x1
9	WI	R/W	Word Direction Ingress. Swaps the bits on the SPI-4.2 ingress output data pins. 0: Normal mode, pin 0-15 carries data bits 0-15. 1: Swapped, pin 0-15 carries data bits 15-0.	0x0
8	WE	R/W	Word Direction Egress. Swaps the bits on the SPI-4.2 egress input data pins. 0: Normal mode, pin 0-15 carries data bits 0-15. 1: Swapped, pin 0-15 carries data bits 15-0.	0x0
7	WD	R/W	Word Direction Egress Deskew. Swaps the bits on the SPI-4.2 egress deskew input data pins. 0: Normal mode, pin 0-15 carries data bits 0-15. 1: Swapped, pin 0-15 carries data bits 15-0.	0x0
6	INGR_DISABLE	R/W	Disable the ingress dataflow when set.	0x0
5	EN_TRANSP	R/W	Enable transparent mode, where data is copied directly to the SPI-4.2 ingress bus. Every 16 bit on the 128 bit data bus generates one SPI-4.2 word. The control word bit is generated by bit 11 in the 16 bit word, and bit 11 on the SPI-4.2 bus is generated by bit 10. 0: Normal operation. 1: Transparent mode enabled. Note: To enable transparent mode, the following conditions must also be met: Set MSCH::CYCLEMIN to 0. Set ING_CONTROL::BURST_LEN to 15. Set EGR_CONTROL::BURST_LEN to 15. Set SPI4_TEST::FORCE_HUNGRY to 1.	0x0

Table 209. SPI4_MISC (Address 0x00), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
4	CRCD2	R/W	SPI-4.2 ingress path data clock divided by 2. 0: Normal operation, unmodified ingress data path frequency. 1: half speed operation, the ingress data path frequency is divided by 2.	0x0
3	CML_RES	R/W	CML Reset 0: holds the SPI-4.2 CML logic reset. 1: Normal operation.	0x0
2	RQC	R/W	SPI-4.2 Ingress Data Alignment. In the normal situation the SPI-4.2 standard requires a shift between the data and clock. Used if two SPI-4.2 devices are connected directly without any external board delay. 0: Data and clock are shifted relatively according to SPI-4.2. 1: The SPI-4.2 ingress clock is shifted 90 degrees relative to situation when RQC = 0.	0x0
1	CML_LOOP	R/W	CML loopback connection 0: Normal operation. 1: The SPI-4.2 data outputs are internally looped to the SPI-4.2 data inputs.	0x0
0	CML_EN	R/W	CML Enable 0: Powers the SPI-4.2 CML block down. 1: Normal operation.	0x0

4.8.2 CML Status

Table 210. SPI4_STATUS (Address 0x01), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
1	CMLPWD	R/O	CML Powered Down Status. The power down status of the CML logic. The value is equal to NOT (SPI4_MISC::CML_EN AND pin/SPI4_EN) where pin/SPI4_EN is the value of the external enable pin. 0: CML logic active 1: CML logic powered down.	0x1
0	CMLEN	R/O	CML Enable is the value of the external SPI4_EN pin. 0: CML logic is powered down 1: Normal operation.	0x1

4.8.3 Ingress Status Channel Setup

Table 211. SPI4_ING_SETUP0 (Address 0x02), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
27:24	MAXONES	R/W	Defines number of consecutive “1 1”s before the status is in link error state. Trigger to the link error event.	0x04
23:20	LINKUPLIMIT	R/W	Defines number of consecutive good DIP2 to receive before accepting link. Zero is an illegal value.	0x01
19:16	LINKDOWNLIMIT	R/W	Defines number of consecutive bad DIP2 to receive before dropping link. Zero is an illegal value.	0x01
15:12	CALENDAR_M	R/W	Defines the number of times the calendar is repeated before the DIP2. It is not recommended to use a CALENDAR_M > 1, because this enhances the latency for the flow control messages. A credit value is not accepted before the DIP2 value arrives.	0x01
11	INVERTCLK	R/W	Invert TCLK. If enabled the sampling of the ingress status channel data is made on the falling edge. 0: Sample the incoming status on rising edge (recommended). 1: Sample the incoming status on falling edge.	0x0
10	RXSTATCLEAR	R/W	Clear and freeze the ingress status channel 0: Normal operation. 1: The status clock is frozen.	0x1
8	LINK_STOP_EN	R/W	When enabled, the top most status channel functions as a SPI-4.2 link level stop/go signal. SATISFIED on this channel stops all SPI-4.2 ingress transmission, HUNGRY/STARVING enables SPI-4.2 transmission. When enabled, the other status channels have their usual behavior. Note the link level channel is always the topmost, regardless of the port remapper setting.	0x0
4:0	CALENDAR_LEN	R/W	The length of the ingress status calendar. Ports outside the calendar are stopped, therefore SPI4_ING_SETUP0::CALENDAR_LEN must always be ≥ number of active ports.	0x18

4.8.4 Ingress Data Training Setup

Table 212. SPI4_ING_SETUP1 (Address 0x03), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
25	EN_AUTO_TRAIN	R/W	Enable the transmission of training sequences when the ingress SPI-4.2 status channel is in link down state. Only applicable if the other end has a dynamic deskew circuit. Setting this bit makes it possible for the deskew block at the receiving end to signal that it has lost sync and wants training sequences for retraining.	0x1
24	SEND_TRAIN	R/W	Transmit continuously training patterns.	0x0
23:16	ALPHA	R/W	Sets how many times the 20 cycle training pattern should be repeated when transmitting a training sequence. This parameter is designated by an alpha in the SPI-4.2 standard. 0: Reserved. 1: A training sequence consists of only 1 training pattern. N: A training sequence consists of N training pattern.	0x001
15:0	TSPERIOD	R/W	Maximum Interval between scheduling of training sequences designated as DATA_MAX_T in the SPI-4.2 standard, in steps of 4 cycles. 0: Feature is disabled. No training sequences are scheduled. 1: A training sequence is scheduled about every 4 cycles. N: A training sequence is scheduled about every N × 4 cycles.	0x0000

4.8.5 Ingress Data Burst Sizes Setup

Table 213. SPI4_ING_SETUP2 (Address 0x04), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	MAXBURST2	R/W	Defines the allowed number of 16-bytes blocks to send when HUNGRY status is received.	0x010
23:16	MAXBURST1	R/W	Defines the allowed number of 16-bytes blocks to send when STARVING status is received.	0x010
8	RF	R/W	For debug use. Do not change default value.	0x0

Table 213. SPI4_ING_SETUP2 (Address 0x04), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
4	CREDITBYPASS	R/W	When set, the credit based system is bypassed. When bypassed a HUNGRY or STARVING enables FIFO buffer output, and a SATISFIED disables FIFO buffer output. In the bypass state SPI4_ING_SETUP2::MAXBURST1 and SPI4_ING_SETUP2::MAXBURST2 have no effect. Transmission of Ethernet flow control can be controlled directly from the SPI-4.2 status channel, if this bit is set together with the PPORT_INH::PPORT_IPT and PPORT_INH::PPORT_IGI bits.	0x0
3:0	BURSTSIZE	R/W	Defines the maximum burst size in 32 bytes increments. Setting BURSTSIZE = 0 disables burst size enforcing, and generates SPI-4.2 data burst the same size as what is output by the ingress FIFO. When transmitting burst-interleaved, it must be set to the same value as the BURSTLEN field in the MSCH register. When transmitting in full-frame mode, BURSTSIZE determines solely the SPI-4.2 burst size.	0x08

4.8.6 Egress Status Channel Setup

Table 214. SPI4_EGR_SETUP0 (Address 0x05), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:28	FSV	R/W	Fast Stop Value. Typical fast stop value should be set to 0x8, for frames smaller than 64 bytes set to 0x4.	0x08
27	FSE	R/W	Fast stop enable. 0: Normal operation 1: fast stop enabled, used when frames smaller than 64 bytes are sent to Schaumburg over the SPI-4.2 interface to prevent internal overflow.	0x1
26	PSD	R/W	Internal debug use. Do not change the default value.	0x0

Table 214. SPI4_EGR_SETUP0 (Address 0x05), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
17:16	CP	R/W	Schaumburg uses two states on the status channel bus. Either "send no more data" = SATISFIED, or "ready to receive data" = HUNGRY / STARVING. CP defines if is should be HUNGRY or STARVING that is signaled when the internal FIFO buffer is in "ready to receive data" state. 0x0: Ready to receive data is flagged on the status channel as STARVING. 0x1: Ready to receive data is flagged on the status channel as HUNGRY.	0x0
15:12	CALENDAR_M	R/W	Defines the number of times the calendar is repeated before sending the DIP2. 0x1: Recommended.	0x01
11	INVERTCLK	R/W	Output the status channel data on falling edge. 0: Output the status on rising edge (recommended). 1: output the status on falling edge.	0x0
10	TXSTATCLEAR	R/W	Clear and freeze the egress status channel 0: Normal operation. 1: The status channel is frozen.	0x1
9	TXSTATCLKSEL	R/W	Status clock selection bit. 0: The outgoing SPI-4.2 status channel clock 75 MHz. 1: The outgoing SPI-4.2 status channel clock is the incoming egress data clock divided by 4.	0x0
4:0	CALENDAR_LEN	R/W	The length of the egress status calendar. Values greater than 24 causes a calendar length of 24.	0x18

4.8.7 Egress Status Port Setup

Table 215. SPI4_EGR_STAT_PORT_SETUP (Address 0x08), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
16	PORTMAPREAD	W/O	Read indication, if set the read from the egress address table. 0: Write operation. 1: Read operation.	0x0
15:8	ADDRESS	R/W	The index address of the port map table to which the port number value should be written. The maximum value is 23. After the first read, subsequent reads auto-increment ADDRESS.	0x000
7:0	PORTMAPVALUE	R/W	The value to be written in the port map table. The maximum value is 23.	0x000

4.8.8 Ingress Status Port Setup

Table 216. SPI4_INGR_STAT_PORT_SETUP (Address 0x09), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
16	PORTMAPREAD	W/O	Read indication, if set the read from the ingress address table. 0: Write operation. 1: Read operation.	0x0
15:8	ADDRESS	R/W	The index address of the port map table to which the port number value should be written. The maximum value is 23. After the first read, subsequent reads auto-increment ADDRESS.	0x000
7:0	PORTMAPVALUE	R/W	The value to be written in the port map table. The maximum value is 23.	0x000

4.8.9 Ingress Effective FIFO Size

Table 217. SPI4_INGR_EFF_FIFO_SIZE (Address 0x0A), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
4:0	OB_EFF_FIFO_SIZE	R/W	Internal debug use. Do not change the default value.	0x1F

4.8.10 Ingress Invert Bits

Table 218. SPI4_INGR_INV_BITS (Address 0x0B), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
16:0	OB_INV_BITS	R/W	Invert the SPI-4.2 ingress data bits and control bit. This feature facilitates PCB design. Bits 0 to 15 invert the data lines 0 to 15. Bit 16 inverts the control line.	0x0000

4.8.11 Ingress Control Mask

Table 219. SPI4_INGR_CNTR_MASK (Address 0x0C), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
23:0	OB_CNTR_MASK	R/W	Specifies which ports are included in the port based debug counting. Only valid when SPI4_DBG_SETUP::DEBCON = 0xF.	0x0000000

4.8.12 SPI-4.2 Debug Counter 0

Table 220. C0 (Address 0x10), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.13 SPI-4.2 Debug Counter 1

Table 221. C1 (Address 0x11), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.14 SPI-4.2 Debug Counter 2

Table 222. C2 (Address 0x12), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.15 SPI-4.2 Debug Counter 3

Table 223. C3 (Address 0x13), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.16 SPI-4.2 Debug Counter 4

Table 224. C4 (Address 0x14), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.17 SPI-4.2 Debug Counter 5

Table 225. C5 (Address 0x15), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.18 SPI-4.2 Debug Counter 6

Table 226. C6 (Address 0x16), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.19 SPI-4.2 Debug Counter 7

Table 227. C7 (Address 0x17), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.20 SPI-4.2 Debug Counter 8

Table 228. C8 (Address 0x18), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CNT	R/W	Debug counter. Records the number of trigger events determined by SPI4_DBG_SETUP::DEBCON.	0x00000000

4.8.21 Debug Counters Setup

Table 229. SPI4_DBG_SETUP (Address 0x1A), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
15:8	ICGS	R/W	Ingress Credit Grant Select. Selects one of 24 SPI-4.2 ingress status channel granted credit counters. The value of the granted credit counter can be read from SPI4_DBG_GRANT::GCV. 0-23: The selected SPI-4.2 port.	0x000
5	CNTFREEZE	R/W	Freezes all debug counters.	0x0
4	CNTRESET	R/W	Resets all debug counters. 0 = Reset counters. 1 = Enable counters	0x1
3.0	DEBCON	R/W	Master counter controller. Controls what events are counted. C0-C8 refers to the counter registers at 0x10–0x18. The hexadecimal number in the parenthesis refers to bit 15:12 in the SPI-4.2 Control Word. For information about the counter register settings, see Table 230, "Counter Register Settings for a Given SPI4_DBG_SETUP::DEBCON Value," page 195.	0x7

Table 230. Counter Register Settings for a Given SPI4_DBG_SETUP::DEBCON Value

DEBCON Value	Counter Registers	Events Counted
0x0	C0	Egress payload data word in a burst.
	C1	Egress SOB control words (0x8, 0xA, 0xC, 0xE).
	C2	Egress EOP/Abort control words (0x2, 0xA, 0xB).
	C3	Egress EOP with 1 byte valid (0x6, 0xE, 0xF).
	C4	Egress EOP with 2 bytes valid (0x4, 0xC, 0xD).
	C5	Egress IDLE control words (0x0 and Port Address bits 11:4 = 0).
	C6	Egress Training control words and training data words (0x0 and Port Address bits 11:4 = 1).
	C7	Egress (End of) Extension control word (0x1, 0x7).
	C8	Egress Dip4 error seen by the egress data path.

Table 230. Counter Register Settings for a Given SPI4_DBG_SETUP::DEBCON Value *(continued)*

DEBCON Value	Counter Registers	Events Counted
0x1	C0	Egress collapsed EOB and SOB received.
	C1	Egress EOB and SOB 2 control word with no idle words in between.
	C2	Egress EOB and SOB 2 control word with 1 idle words in between.
	C3	Egress EOB and SOB 2 control word with 2 idle words in between.
	C4	Egress EOB and SOB 2 control word with 3 idle words in between.
	C5	Egress EOB and SOB 2 control word with 4 idle words in between.
	C6	Egress EOB followed by more than 4 idle words.
	C7	Reserved.
	C8	Egress Dip4 error seen by the egress data path.
0x2	C0	Unknown control word (0x3, 0x5).
	C1	Unknown data word between burst boundaries.
	C2	Egress SOP for all ports.
	C3	Egress EOP for all ports.
	C4	Reserved.
	C5	Egress sync fail indications from the deskew block.
	C6	Reserved.
	C7	Reserved.
	C8	Egress Dip4 error seen by the egress data path.
0x3	C0	C0 Ingress payload data word in a burst.
	C1	C1 Ingress EOB and SOP control words (0x8, 0x9).
	C2	C2 Ingress EOP/Abort control words (0x2, 0xA, 0xB).
	C3	C3 Ingress EOP with 1 byte valid control words (0x6, 0xE, 0xF).
	C4	C4 Ingress EOP with 2 bytes valid control words (0x4, 0xC, 0xD).
	C5	C5 Ingress IDLE control word control words (0x0 and Port Address bits 11:4 = 0).
	C6	Ingress Training control words (0x0 and Port Address bits 11:4 = 1). The training data words are also counted in this counter.
	C7	Ingress (End of) Extension control words (0x1, 0x7).
	C8	Egress Dip4 error seen by the egress data path.

Table 230. Counter Register Settings for a Given SPI4_DBG_SETUP::DEBCON Value *(continued)*

DEBCON Value	Counter Registers	Events Counted
0x4	C0	Ingress collapsed EOB and SOB received.
	C1	Ingress EOB and SOB 2 control words with no idle words in between.
	C2	Ingress EOB and SOB 2 control words with 1 idle word in between.
	C3	Ingress EOB and SOB 2 control words with 2 idle word in between.
	C4	Ingress EOB and SOB 2 control words with 3 idle word in between.
	C5	Ingress EOB and SOB 2 control words with 4 idle words in between.
	C6	Ingress SOB preceded by more than 4 idle words.
	C7	Reserved.
	C8	Egress Dip4 error seen by the egress data path.
0x5	C0	Ingress Dip2 errors.
	C1	Ingress Dip2 valid indications.
	C2	Ingress Link up status indications.
	C3	Ingress Link down status indications.
	C4	Ingress Link error event received consecutive "11" defined in the configurations register (SPI4_ING_SETUP0::MAXONES).
	C5	Ingress Link sync error, the sync bits are not "11".
	C6	Ingress Link sync, the sync bits are "11".
	C7	Reserved.
	C8	Egress Dip4 error seen by the egress data path.
0x7	C4	Egress payload data word in a burst.
	C5	Egress EOB and SOP control words (0x8, 0x9).
	C6	Egress EOP/Abort control words (0x2, 0xA, 0xB).
	C7	Egress EOP with 1 or 2 bytes valid (0x4, 0x6, 0xC, 0xD, 0xE, 0xF).
	C8	Egress Dip4 error seen by the egress data path.
0xF	C0	Port specific count of number of ingress payload data in bytes.
	C1	Port specific count of number of ingress frames.
	C2	Port specific count of number of ingress EOPA.
	C3 to C8	Reset.

4.8.22 Test Setup

Table 231. SPI4_TEST (Address 0x20), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31	LOG	R/W	Enables logging of SPI-4.2 egress data path. When enabled, all data path data is passed through together with SPI-4.2 control words. Data can be read from the egress FIFO. If logging is started while data is passing through the SPI-4.2, the device can enter a state where no data is stored in the egress FIFO. In this case logging must be restarted until capture is seen. Note: SPI4_MISC::EN_DIP4 must be cleared. 0: logging disabled. 1: logging enabled.	0x0
30	FORCE_HUNGRY	R/W	Simulate HUNGRY received on all ports. When SPI4_TEST::FORCE_HUNGRY is set, the status channel is ignored and no training sequence is transmitted. Therefore SPI4_TEST::FORCE_HUNGRY should not be enabled before the dynamic deskew block on the receiving end has acquired synchronization on the initial sequence of training patterns. 0: Normal mode. 1: SPI-4.2 ingress status channel is forced to HUNGRY for all ports.	0x0
25	DDT	R/W	Disable data through to the SPI-4.2 interface. It is recommended that this bit be set to 1 when PRBS/user pattern checker is used. 0: Data is both received by the pattern checker and the SPI-4.2 egress interface. 1: Data is only received by the pattern checker.	0x0
24:20	CHECK_LANE	R/W	Selects which events are counted in TPERR_CNT::PCEC. 0-15: Reports bit errors on the selected data bit (0-15) of the SPI-4.2, that is, pins SPI_TD0 to SPI_TD15. 16: Reports bit errors on the control bit of the SPI-4.2, that is, pin SPI_TCtrl. 17: Reports word errors on the SPI-4.2 interface. If the 17 bit bus has one or more bit errors, the error counter is incremented. Covers all the pins SPI_TD0-15 and SPI_TCtrl.	0x00
19	SP	R/W	Sample pattern. Sample 64 bit of data on SPI-4.2 bus and place it in TPSAM_P0::SP0 and TPSAM_P1::SP1. The data is sampled on the falling edge of the SP bit.	0x0

Table 231. SPI4_TEST (Address 0x20), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
18	FNL	R/W	Force No Lock. The pattern checker can lock on an all-zero pattern, typically seen during start-up. The user should therefore momentarily force the checker out of lock, after the test pattern is applied. 0x0: The pattern checker is allowed to lock on the incomming pattern. 0x1: The pattern checker is forced out of lock.	0x0
17:16	CPS	R/W	Checker PRBS pattern select. Configuration of the pattern checker in the SPI-4.2 egress direction. Enable DDT when pattern checker is used. 0x0: Use the PRBS-23 pattern checker. 0x1: Use a 64 bit user-definable pattern checker. Configuration of the user pattern checker is performed in TPCHK_UP0::GUP0 and TPCHK_UP1::GUP1 . 0x2: Use the PRBS-31 pattern checker. 0x3: Same as 0x1. The result of the checker can be viewed in the SPI4_STICKY::PRL_STK and SPI4_STICKY::PAL_STK. Note: To force the pattern checker out of lock, shift to another checker pattern and clear SPI4_STICKY register.	0x0
2:1	GPS	R/W	Generator PRBS pattern select. Configuration of the pattern generator in the SPI-4.2 ingress direction. 0x0: Use the PRBS-23 pattern generator. 0x1: Use a 64 bit user-definable pattern generator. Configuration of the user pattern generator is performed in TPGEN_UP0::GUP0 and TPGEN_UP1::GUP1. 0x2: Use the PRBS-31 pattern generator. 0x3: Same as 0x1.	0x0
0	TE	R/W	Test Enable. Start the pattern generator. 0: SPI-4.2 operates in normal mode. 1: SPI-4.2 transmits test patterns from the test pattern generator. If the lanes are looped back to the egress path the pattern checker can verify the pattern. See result in SPI4_STICKY register and TPERR_CNT::PCEC.	0x0

4.8.23 Test Pattern Generator User Pattern0

Table 232. TPGEN_UP0 (Address 0x21), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	GUP0	R/W	Generator user pattern; bit 63:32. Defines one half of the user pattern to send on the SPI-4.2 interface. The sequence is sent serially with bit 63 first on all 17 wires. There is a skew of one bit between each of the 17 wires. The user pattern programmed to this register is loaded into the user pattern generator when the generator is switched from another pattern (PRBS) to user pattern. So if the generator already generates user pattern, the generator has to be switched to PRBS pattern and back to user pattern to get the new programmed user pattern effective	0x00000000

4.8.24 Test Pattern Generator User Pattern1

Table 233. TPGEN_UP1 (Address 0x22), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	GUP1	R/W	Generator user pattern; bit 31:0. Defines one half of the user pattern to send on the SPI-4.2 interface. The sequence is sent serially with bit 63 first on all 17 wires. There is a skew of one bit between each of the 17 wires. The user pattern programmed to this register is loaded into the user pattern generator when the generator is switched from another pattern (PRBS) to user pattern. So if the generator already generates user pattern the generator has to be switched to PRBS pattern and back to user pattern to get the new programmed user pattern effective	0x00000000

4.8.25 Test Pattern Checker User Pattern0

Table 234. TPCHK_UP0 (Address 0x23), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CUP0	R/W	Checker user pattern; bit 63:32. Defines one half of the user pattern to expect on the SPI-4.2 interface. The sequence is received serially with bit 0 first. A skew of one bit between each of the 17 bits is required.	0x00000000

4.8.26 Test Pattern Checker User Pattern1

Table 235. TPCHK_UP1 (Address 0x24), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	CUP1	R/W	Checker user pattern; bit 31:0. Defines one half of the user pattern to expect on the SPI-4.2 interface. The sequence is received serially with bit 0 first. A skew of one bit between each of the 17 bits is required.	0x00000000

4.8.27 Sampled Pattern0

Table 236. TPSAM_P0 (Address 0x25), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SP0	R/O	Sampled pattern; bit 63:32. The top 32 bits of the sampled pattern.	0x00000000

4.8.28 Sampled Pattern1

Table 237. TPSAM_P1 (Address 0x26), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	SP1	R/O	Sampled pattern; bit 31:0. The lowest 32 bits of the sampled pattern.	0x00000000

4.8.29 Pattern Checker Error Counter

Table 238. TPERR_CNT (Address 0x27), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:0	PCEC	R/W	Pattern checker error counter. Counts the number of events defined by the SPI4_TEST register. During normal operation, PCEC counts because the PRBS checker can lock on an all-zero pattern. This counting does not indicate any errors and can safely be ignored.	0x00000000

4.8.30 Sticky Bits Register 0

Table 239. SPI4_STICKY (Address 0x30), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
30	SPI4_OB_PBACK_STK	R/W	SPI-4.2 ingress push back asserted (slow enable).	0x0
25	NORMALIZERFIFO_FULL_STK	R/W	Indicates that the FIFO is full. 0: Data in FIFO. 1: FIFO is full.	0x0
24	NORMALIZERFIFO_EMPTY_ STK	R/W	Always 0x0.	0x0
23	PLANEBUILDERFIFO_FULL_ STK	R/W	Indicates that the FIFO is full. 0: Data in FIFO. 1: FIFO is full.	0x0
22	PLANEBUILDERFIFO_EMPTY_ STK	R/W	Always 0x0.	0x0
21	FILTER_USIZE_STK	R/W	Frames with 16 bytes or less are filtered out on SPI-4.2 egress. 0: No undersize frames. 1: one or more undersize frames.	0x0
20	SYNC_FAIL_STK	R/W	Indicates that synchronization is lost, meaning that more than a certain amount of DIP-4 errors within a configurable window, has occurred. 0: Sync okay. 1: Synchronization lost on SPI-4.2 egress data path.	0x0
19	DIP4_ERR_STK	R/W	DIP-4 error has occurred. 0: No error observed since last time bit was cleared. 1: DIP-4 error observed since last time bit was cleared.	0x0
18	PRL_STK	R/W	PRBS locked. Once the checker has locked, it stays in the lock state until the checker mode is changed (PRBS ↔ user-defined pattern). 0: Not locked. 1: Locked.	0x0
17	PAL_STK	R/W	User defined pattern locked. User defined pattern locked. Once the checker has locked, it stays in the lock state until the checker mode is changed (PRBS ↔ user-defined pattern). 0: Not locked. 1: Locked.	0x0
16	L16_STK	R/W	If 1, an error has occurred on lane 16.	0x0
15	L15_STK	R/W	If 1, an error has occurred on lane 15.	0x0
14	L14_STK	R/W	If 1, an error has occurred on lane 14.	0x0

Table 239. SPI4_STICKY (Address 0x30), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
13	L13_STK	R/W	If 1, an error has occurred on lane 13.	0x0
12	L12_STK	R/W	If 1, an error has occurred on lane 12.	0x0
11	L11_STK	R/W	If 1, an error has occurred on lane 11.	0x0
10	L10_STK	R/W	If 1, an error has occurred on lane 10.	0x0
9	L9_STK	R/W	If 1, an error has occurred on lane 9.	0x0
8	L8_STK	R/W	If 1, an error has occurred on lane 8.	0x0
7	L7_STK	R/W	If 1, an error has occurred on lane 7.	0x0
6	L6_STK	R/W	If 1, an error has occurred on lane 6.	0x0
5	L5_STK	R/W	If 1, an error has occurred on lane 5.	0x0
4	L4_STK	R/W	If 1, an error has occurred on lane 4.	0x0
3	L3_STK	R/W	If 1, an error has occurred on lane 3.	0x0
2	L2_STK	R/W	If 1, an error has occurred on lane 2.	0x0
1	L1_STK	R/W	If 1, an error has occurred on lane 1.	0x0
0	L0_STK	R/W	If 1, an error has occurred on lane 0.	0x0

For all sticky bits, the events can only set the bit. To clear the sticky bit in this register you must write "0". Note that writing to this register sets all bits to the written value.

4.8.31 Sticky Bits Register 1

Table 240. SPI4_CRDT_STICKY (Address 0x31), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	RSV	R/W	Reserved, do not change default value.	0x000
23:0	CRDT_VIO_STK	R/W	Credit Violation Sticky Bit. Indicates that the credit counter was zero while the data path was transferring data. For all bits, a read does not alter the bits, while a write puts the write value into the bits. Bit N = 0: The credit counter for SPI-4.2 port N is OK. Bit N = 1: The credit counter for SPI-4.2 port N was zero, but ingress data was observed anyway.	0x0000000

4.8.32 Ingress Status Channel Granted Credit Value

Table 241. SPI4_DBG_GRANT (Address 0x33), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
15:0	GCV	R/O	Granted Credit Value shows the current value of the granted credits for a single SPI-4.2 ingress port. The port number is selected by SPI4_DBG_SETUP::ICGS.	0x0000

4.8.33 Core Egress Inhibit

Table 242. SPI4_DBG_EGR_INH (Address 0x34), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	RSV	R/W	Reserved, do not change default value.	0x000
11:0	EGRINH	R/W	Egress Inhibit. The value of the 12 inhibit signals going from the egress FIFO to SPI-4.2 egress, to be transmitted on the SPI-4.2 egress status channel. 0: The egress status channel sends out HUNGRY/STARVING 1: The egress status channel sends out SATISFIED.	0x0000000

4.8.34 Core Ingress Inhibit

Table 243. SPI4_DBG_INGR_INH (Address 0x35), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
11:0	INGRINH	R/W	Ingress Inhibit. The value of the 12 inhibit signals going from the SPI-4.2 ingress to ingress FIFO, to be received from the SPI-4.2 ingress status channel. 0: The ingress status channel sends out HUNGRY/STARVING 1: The ingress status channel sends out SATISFIED.	0x0FFFFFF

4.8.35 Sampled Ingress Status Channel 1

Table 244. SPI4_DBG_STATUS1 (Address 0x36), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	RSV	R/W	Reserved	0x000

Table 244. SPI4_DBG_STATUS1 (Address 0x36), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
23:22	P11	R/O	The current sampled 2 bit status channel value on the SPI-4.2 ingress status channel 11. '00': STARVING. '01': HUNGRY. '10': SATISFIED. '11': Reserved for framing or to indicate a disabled status link.	0x3
21:20	P10	R/O	Current sampled ingress channel 10.	0x3
19:18	P9	R/O	Current sampled ingress channel 9.	0x3
17:16	P8	R/O	Current sampled ingress channel 8.	0x3
15:14	P7	R/O	Current sampled ingress channel 7.	0x3
13:12	P6	R/O	Current sampled ingress channel 6.	0x3
11:10	P5	R/O	Current sampled ingress channel 5.	0x3
9:8	P4	R/O	Current sampled ingress channel 4.	0x3
7:6	P3	R/O	Current sampled ingress channel 3.	0x3
5:4	P2	R/O	Current sampled ingress channel 2.	0x3
3:2	P1	R/O	Current sampled ingress channel 1.	0x3
1:0	P0	R/O	Current sampled ingress channel 0.	0x3

4.8.36 Sampled Ingress Status Channel 2

Table 245. SPI4_DBG_STATUS2 (Address 0x37), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	RES	R/W	Reserved	0x000
23:22	P23	R/O	The current sampled 2 bit status channel value on the SPI-4.2 ingress status channel 23. '00': STARVING. '01': HUNGRY. '10': SATISFIED. '11': Reserved for framing or to indicate a disabled status link.	0x3
21:20	P22	R/O	Current sampled ingress channel 22.	0x3
19:18	P21	R/O	Current sampled ingress channel 21.	0x3
17:16	P20	R/O	Current sampled ingress channel 20.	0x3
15:14	P19	R/O	Current sampled ingress channel 19.	0x3
13:12	P18	R/O	Current sampled ingress channel 18.	0x3
11:10	P17	R/O	Current sampled ingress channel 17.	0x3
9:8	P16	R/O	Current sampled ingress channel 16.	0x3

Table 245. SPI4_DBG_STATUS2 (Address 0x37), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
7:6	P15	R/O	Current sampled ingress channel 15.	0x3
5:4	P14	R/O	Current sampled ingress channel 14.	0x3
3:2	P13	R/O	Current sampled ingress channel 13.	0x3
1:0	P12	R/O	Current sampled ingress channel 12.	0x3

4.8.37 SPI-4.2 Deskew Control Mode

Table 246. SPI4_DSKW_CTRL_MODE (Address 0x40), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
23:20	DSKW_MODE_RST	R/W	Controls the selection of the sample points in the delay line when (re)syncing. The delay lines hold data of multiple clock cycles. 0x0: initial sample point most near the center (recommended). 0x2: initial sample point 1 UI before center. 0x4: initial sample point 2 UI before center.	0x00
19:16	DSKW_MODE_SEL	R/W	For debug use. Do not change the default value. When DSKW_MODE_SEL = 0, the DDS does *not* update its select pointer at all (neither at initial training nor during repeated training). The select pointer is used to select the tap that is used to sample the incoming data. If DSKW_MODE_SEL is switched from 1 (default) to 0 after the DDS is synched, the DDS is prevented from tracking the skew at repeated training. If DSKW_MODE_SEL is switched to 0 BEFORE the DDS is synched, the DDS cannot select an appropriate sampling point at all.	0x01
15:12	DSKW_MODE_EDGE	R/W	The number of training patterns used (expected) by the deskew block for averaging within a training sequence. The transmitting end must be programmed to an adequate alpha parameter to provide (at least) this number of training patterns within a training sequence. Only the following values should be used. Values not listed give 1 training pattern on average. 0x1: Average of 1 training pattern. 0x2: Average of 1 training pattern. 0x3: Average of 4 training patterns. 0x4: Average of 8 training patterns. 0x5: Average of 16 training patterns. 0x6: Average of 32 training patterns. 0x7: Average of 64 training patterns. 0x8: Average of 128 training patterns. 0x9: Average of 256 training patterns.	0x02

Table 246. SPI4_DSKW_CTRL_MODE (Address 0x40), Block 5, Subblock 0 *(continued)*

Bit	Name	Mode	Description	Default
11:8	DSKW_MODE_CLK	R/W	For debug use. Do not change the default value.	0x02
0	DSKW_MODE_ENA	R/W	Enable dynamic deskew. SPI4_MISC::EGRCLK must be 1 while changing this bit. 0: Disabled, data is passed though untouched. 1: Enable dynamic deskew.	0x0

4.8.38 SPI-4.2 Deskew Control Force 2

Table 247. SPI4_DSKW_CTRL_FRC2 (Address 0x43), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
31:24	DSKW_DATA_FRC_DELAY90	R/W	Debug use only. Do not change default value.	0x00
23:19	DSKW_DELAY90_OFFSET	R/W	Offset that is added to measured delay_90 value (Range +15/–16; Coding two's complement). Recommended value is 1.	0x00
18:17	DSKW_RSV	R/W	Debug use only. Do not change default value.	0x00
16	DSKW_DIS_CLK_GATE	R/W	Debug use only. Do not change default value.	0x00
15:8	DSKW_DATA_FRC_EDGE	R/W	Debug use only. Do not change default value.	0x00
3	DSKW_FRC_SYNC_ACK	R/W	Debug use only. Do not change default value.	0x00
2	DSKW_FRC_DELAY90	R/W	Debug use only. Do not change default value.	0x00
1	DSKW_FRC_SEL_TCTL	R/W	Debug use only. Do not change default value.	0x00
0	DSKW_FRC_EDGE_TCTL	R/W	Debug use only. Do not change default value.	0x00

4.8.39 SPI-4.2 Deskew Control DIP-4 Error Threshold

Table 248. SPI4_DSKW_CTRL_DIP4_ERR_THRS (Address 0x4B), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
3:0	EGR_DIP4_FAIL	R/W	Threshold for number of incorrect DIP-4 control words detected within the DIP-4 window before declaring the egress data path FAIL. 0-15: Number of incorrect DIP-4 control words accepted without entering FAIL state.	0x00

4.8.40 SPI-4.2 Deskew Control DIP-4 Window

Table 249. SPI4_DSKW_CTRL_DIP4_WINDOW (Address 0x4C), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
5:0	EGR_DIP4_WINDOW	R/W	Window size to be used when examine DIP-4 error threshold. 0: Disable DIP-4 error monitoring. 1-63: Number of DIP-4 code words to be evaluated.	0x00

4.8.41 SPI-4.2 Deskew Control Synchronization

Table 250. SPI4_DSKW_CTRL_SYNC (Address 0x4D), Block 5, Subblock 0

Bit	Name	Mode	Description	Default
3	EGR_FRC_DSKW_RESYNC	R/W	Force reset of the deskew block. Deskew block restarts synchronization to a continuous training pattern.	0x0
2	EGR_DIS_DSKW_RESYNC_ON_FAIL	R/W	Disable reset of the deskew block upon entering the FAIL state.	0x0
1	EGR_DIS_FAIL_ON_DSKW_SYNC	R/W	Disable going into fail when the deskew block signals that it is not capable of obtaining synchronization.	0x0

4.9 BIST Access, Block 7

BIST access to any memory is only possible if reset is released on all internal subblocks containing memory. The bitfields listed in Table 271 must be cleared before doing any BIST accesses.

Table 251. BIST Access Registers Bitfields

Block	Subblock	Register	Bitfield
5	0	SPI4_MISC	SPI4_MISC::INGCLK = 0 and SPI4_MISC::EGRCLK = 0
1	0–11	DEV_SETUP	DEV_SETUP::RST = 0

Schaumburg can perform a self-test of the internal memories (RAM BIST). Accesses of registers used for RAM BIST are of an indirect nature. That is, for accessing these indirect registers, address, data, and access type have to be specified by writing to a direct register. After writing to the direct register, the indirect register access is performed internally. For a read, the result is returned to a direct register.

For information about running the RAM BIST, see “Initialization,” page 76.

4.9.1 RAM BIST Command

Table 252. RAM_BIST_CMD (Address 0x00), Block 7, Subblock 1

Bit	Name	Mode	Description	Default
27	WR_IN_PG	R/O	Write in progress. 0: BIST is ready for new access. 1: BIST register write is in progress.	0x0
26	WR_ERR	R/O	Write error. 0: Last write access was processed. 1: Last write access failed.	0x0
25	WR_ERR_STK	R/W	Write error sticky bit. The bit is cleared when a 1 is written. 0: No write failures. 1: At least one of the write accesses failed.	0x0
24	TYPE	R/W	Access type. 0: BIST register read is initiated. Result is read from register RAM_BIST_RESULT. 1: BIST register write is initiated.	0x0
23:16	ADDR	R/W	BIST register address. Selects which of the indirect registers within a BIST module to access. 0x00: WCM_CFG 0x01: WCM_CMD_CTRL 0x02: WCM_STATUS	0x000
15:8	DATA	R/W	Write data to indirect BIST register.	0x000
7:0	MOD_ID	R/W	BIST Module ID. Selects which BIST module to access.	0x000

Table 253. RAM Module IDs

RAM Module ID (MOD_ID)	Name
0	dev1g_a__0
1	dev1g_a__1
2	dev1g_a__2
3	dev1g_a__3
4	dev1g_a__4
5	dev1g_a__5
6	dev1g_a__6
7	dev1g_a__7
8	dev1g_a__8
9	dev1g_a__9
10	dev1g_a__10

Table 253. RAM Module IDs *(continued)*

RAM Module ID (MOD_ID)	Name
11	dev1g_a__11
12	stat_engine1g_a
13	fifo_egress_a_sram_a
14	fifo_egress_a_sram_b
15	spi4p2_normalizer_sram0
16	spi4p2_normalizer_sram1
17	spi4p2_plane_builder_sram0
18	spi4p2_plane_builder_sram1
19	spi4p2_ob_denormalizer
20	fifo_ingress_a_sram_a
21	fifo_ingress_a_sram_b

4.9.2 RAM BIST Read Status and Read Result

Table 254. RAM_BIST_RESULT (Address 0x01), Block 7, Subblock 1

Bit	Name	Mode	Description	Default
9	RD_IN_PG	R/O	Read in progress. 0: RAM_BIST_RESULT::RD_ERR and RAM_BIST_RESULT::RD_DATA are valid. 1: BIST read access still in progress. RAM_BIST_RESULT::RD_ERR and RAM_BIST_RESULT::RD_DATA are invalid.	0x0
8	RD_ERR	R/O	BIST read status. 0: BIST register read succeeded. 1: BIST register read failed. RAM_BIST_RESULT::RD_DATA is invalid.	0x0
7:0	RD_DATA	R/O	Read data from the indirect BIST register.	0x000

4.10 Indirect BIST

4.10.1 Wrapper Control Module Configuration

Table 255. WCM_CFG (Address 0x00)

Bit	Name	Mode	Description	Default
4	REPAIR_DISABLE	R/W	Reserved.	0x0

Table 255. WCM_CFG (Address 0x00) *(continued)*

Bit	Name	Mode	Description	Default
3	MEM_DISABLE	R/W	Disable and power down memory. Used for Iddq testing. 0: Memory is enabled. 1: Memory is disabled, applies to both ports on dual-port RAMs.	0x0
2	FORCE_RDY	R/W	Force memory ready: 0: Force not applied. 1: Force memory ready (must not be set if FORCE_NOT_RDY is 1).	0x0
1	FORCE_NOT_RDY	R/W	Force memory not ready: 0: Force not applied. 1: Force memory not ready (must not be set if FORCE_RDY is 1).	0x0
0	PORT_SEL	R/W	Reserved. Must always be set to 0x0.	0x0

4.10.2 Wrapper Control Module Command

Table 256. WCM_CMD_CTRL (Address 0x01)

Bit	Name	Mode	Description	Default
2	WR_ACCESS	R/W	Internal debug use. Do not change default value.	0x0
1	RD_ACCESS	R/W	Internal debug use. Do not change default value.	0x0
0	RUN_BIST	R/W	Force test engine to run BIST. WCM_CFG::FORCE_NOT_RDY must be set to 1 before starting the BIST. When RUN_BIST is changed from 0 to 1 the BIST is run from the beginning. The bit clears automatically when BIST is done.	0x0

4.10.3 Wrapper Control Module Status

Table 257. WCM_STATUS (Address 0x02)

Bit	Name	Mode	Description	Default
3	MEM_RDY	R/O	Indicates whether memory is ready to be accessed by normal operation. This bit must be 1 before the memory is accessed by normal operation; otherwise it is in test mode and ignores the normal operation. 0: Memory is not ready. 1: Memory is ready.	0x0
2	RESERVED	R/O	Internal debug use.	0x0

Table 257. WCM_STATUS (Address 0x02) *(continued)*

Bit	Name	Mode	Description	Default
1	BIST_OK	R/O	This bit tells whether BIST ran successfully or not. This bit is only valid when BIST_DONE is 1.	0x0
0	BIST_DONE	R/O	Status of BIST. 0: BIST is running or has not been run yet 1: BIST is done	0x0

4.11 Primary System Control, Block 7, Subblock 15

4.11.1 Chip ID

Table 258. CHIP_ID (Address 0x00), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31:28	REV_ID	R/O	Revision ID	0x01
27:12	PART_ID	R/O	Part number	0x7326
11:1	MFCT_ID	R/O	Manufacture ID (0x74 for Vitesse)	0x074
0	ONE	R/O	Always = 1	0x1

4.11.2 Blade ID

Table 259. BLADE_ID (Address 0x01), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
7:0	BLADE_ID	R/W	General purpose read write register, which can be used, for example, for the ID of the blade where Schaumburg is located.	0x0FF

4.11.3 Global Soft Reset

Table 260. SW_RESET (Address 0x02), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31	RST1	R/W	Global reset1	0x0
30:1	NOT_USED	R/O	Unused	0x00000000
0	RST0	R/W	Global reset0	0x0

When both SW_RESET::RST0 and SW_RESET::RST1 are set to 1, a complete reset is performed including the serial and parallel interfaces. Normal operation is when one or both of these bits are 0. After a reset is performed, the SW_RESET::RST0 and SW_RESET::RST1 bits automatically return to the default value.

Note: All interfaces are shut down by default after a global reset. Each interface must then be enabled individually.

4.11.4 Memory Control Register

Table 261. MEMLOCK_CTRL (Address 0x04), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
2	BIST_DESELECT_B	R/W	Must be set to 1.	0x0
1	MLC	R/W	For internal debug use only.	0x0
0	MLS	R/W	For internal debug use only.	0x0

4.11.5 Interface Mode

Table 262. IFACE_MODE (Address 0x07), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
4	STRP_HDR	R/W	Strip header. Remove the 9-byte normalized header before sending frames to the SPI-4.2 interface. This feature is intended for full-frame output only, where the header is unwanted (ingress only). 0: Do not remove header. 1: Remove header.	0x0

4.11.6 SI Insert Bytes

Table 263. SI_INSERT_BYTES (Address 0x0F), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
2:0	ISBOR	R/W	Insert bytes on read. Defines how many dummy bytes are inserted in front of data on all reads from the serial interface (SI). This feature is used when speeding up the SI clock frequency. The response time is 1 µs on reads from SI. If the clock frequency needs to be higher than 0.5 MHz, the answer can be delayed by a programmable number of bytes, keeping the required response time at 1 µs or higher. After changing this parameters wait for 2 µs before accessing the serial interface again, to allow for the configuration signal to be updated internally.	0x0

4.11.7 SI Transfer Select

Table 264. SI_TRANSFER_SEL (Address 0x18), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31	L7	R/W	Bit order.	0x1
28	E7	R/W	Byte order.	0x1
27	E6	R/W	Byte order.	0x1
24	L6	R/W	Bit order.	0x1
23	L5	R/W	Bit order.	0x1
20	E5	R/W	Byte order.	0x1
19	E4	R/W	Byte order.	0x1
16	L4	R/W	Bit order.	0x1
15	L3	R/W	Bit order.	0x1
12	E3	R/W	Byte order.	0x1
11	E2	R/W	Byte order.	0x1
8	L2	R/W	Bit order.	0x1
7	L1	R/W	Bit order.	0x1
4	E1	R/W	Byte order.	0x1
3	E0	R/W	Byte order.	0x1
0	L0	R/W	Bit order.	0x1

Notes:
1. All E*n* and L*n* bits must be set.
2. Byte order in word (E*n*):
   0: Little endian. Least significant byte first.
   1: Big endian. Most significant byte first.
3. Bit order in byte (L*n*):
   0: Least significant bit first.
   1: Most significant bit first.
4. After changing these configurations, the serial interface should not be accessed within the first 2 µs after the configurations change (to allow for the configuration to be updated internally).

4.11.8 Clock Speed Selection

Table 265. PLL_CLK_SPEED (Address 0x19), Block 7 Subblock 15

Bit	Name	Mode	Description	Default
18	SPI4_PLL_LOCK_STATUS_STK	R/W	Sticky bit which is set when SPI-4.2 PLL loses lock. Cleared by writing a 1.	0x0
17	SYS_PLL_LOCK_STATUS_STK	R/W	Sticky bit which is set when system PLL loses lock. Cleared by writing a 1.	0x0

Table 265. PLL_CLK_SPEED (Address 0x19), Block 7 Subblock 15 *(continued)*

Bit	Name	Mode	Description	Default
16:15	SPI4_PLL_BS_CTRL	R/W	Pll jitter performance adjust. Do not change default value.	0x2
14	SPI4_XTAL_SEL	R/W	Selects the SPI-4.2 PLL clock divider. For a frequency calculation, see PLL_CLK_SPEED::SPI4_PLL_MULT. 0: SPI-4.2 PLL clock divider is 1. 1: SPI-4.2 PLL clock divider is 4.	0x1
13:11	SPI4_PLL_MULT	R/W	Selects the SPI-4.2 PLL clock multiplier. The table lists the clock multiplier as a function of SPI4_PLL_MULT setting. The resulting SPI-4.2 ingress data path frequency is PLL1_Clk × (12 + SPI4_PLL_MULT) / (3 × PLL_CLK_SPEED::SPI4_XTAL_SEL + 1). For example, setting SPI4_PLL_CLK_SPEED::XTAL_SEL = 0, SPI4_PLL_MULT = 3, and feeding the SPI-4.2 PLL with a 26.04167 MHz frequency, results in a 390.625 MHz SPI-4.2 data clock. It is possible to generate frequencies > 407 MHz, but proper operation is not guaranteed. Furthermore, the output frequencies can be divided by 2 or 4, using SPI4_MISC::CRCD2, and PLL_CLK_SPEED::SPI4_PLL_DIV2. 25 MHz / 100 MHz / 125 MHz 0: 300 / 300 / 375 1: 325 / 325 / 406.25 2: 350 / 350 / 437.5 3: 375 / 375 / N/A 4: 400 / 400 / N/A 5: 425 / 425 / N/A 6: 450 / 450 / N/A 7: Not used.	0x4
10	SPI4_PLL_PWR_ON	R/W	Power on PLL. 0 = power off (PLL not running) 1 = power on (PLL running)	0x1
9	SPI4_PLL_BP	R/W	Bypass the PLL to provide a low-speed functional test of the Schaumburg device.	0x0
8	SPI4_PLL_FORCE_LOW	R/W	Forces the PLL VCO frequency to the lowest possible frequency. The lock signal is deasserted when the PLL VCO is forced low. 0: Normal mode. 1: PLL VCO forced to lowest frequency.	0x0
7	SPI4_PLL_DIV2	R/W	Divide the SPI-4.2 ingress data path frequency by 2. Additional divide-by-2 can be achieved by SPI4_MISC::CRCD2.	0x0

4.11.9 GPIO Control

Table 266. GPIO_CTRL (Address 0x1D), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
7:0	PD7_0	R/W	Pin direction of GPIO7-GPIO0 0: Pin is input. Value can be read in GPIO_IN::I0-I7. 1: Pin is output. Value is controlled by GPIO_OUT::O0-O7.	0x000

4.11.10 GPIO Output

Table 267. GPIO_OUT (Address 0x1E), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
7	O7	R/W	Output value on GPIO7 when configured as output.	0x0
6	O6	R/W	Output value on GPIO6 when configured as output.	0x0
5	O5	R/W	Output value on GPIO5 when configured as output.	0x0
4	O4	R/W	Output value on GPIO4 when configured as output.	0x0
3	O3	R/W	Output value on GPIO3 when configured as output.	0x0
2	O2	R/W	Output value on GPIO2 when configured as output.	0x0
1	O1	R/W	Output value on GPIO1 when configured as output.	0x0
0	O0	R/W	Output value on GPIO0 when configured as output.	0x0

4.11.11 GPIO Input

Table 268. GPIO_IN (Address 0x1F), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
7	I7	R/O	Value on GPIO7.	0x0
6	I6	R/O	Value on GPIO6.	0x0
5	I5	R/O	Value on GPIO5.	0x0
4	I4	R/O	Value on GPIO4.	0x0
3	I3	R/O	Value on GPIO3.	0x0
2	I2	R/O	Value on GPIO2.	0x0
1	I1	R/O	Value on GPIO1.	0x0
0	I0	R/O	Value on GPIO0.	0x0

4.11.12 Parallel CPU Interface Transfer Select

Table 269. PI_TRANSFER_SEL (Address 0x20), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31	DP7	R/W	Done polarity.	0x1
30	TM7	R/W	Transfer mode.	0x0
28	E7	R/W	Byte order.	0x1
27	E6	R/W	Byte order.	0x1
25	TM6	R/W	Transfer mode.	0x0
24	DP6	R/W	Done polarity.	0x1
23	DP5	R/W	Done polarity.	0x1
22	TM5	R/W	Transfer mode.	0x0
20	E5	R/W	Byte order.	0x1
19	E4	R/W	Byte order.	0x1
17	TM4	R/W	Transfer mode.	0x0
16	DP4	R/W	Done polarity.	0x1
15	DP3	R/W	Done polarity.	0x1
14	TM3	R/W	Transfer mode.	0x0
12	E3	R/W	Byte order.	0x1
11	E2	R/W	Byte order.	0x1
9	TM2	R/W	Transfer mode.	0x0
8	DP2	R/W	Done polarity.	0x1
7	DP1	R/W	Done polarity.	0x1

Table 269. PI_TRANSFER_SEL (Address 0x20), Block 7, Subblock 15 *(continued)*

Bit	Name	Mode	Description	Default
6	TM1	R/W	Transfer mode.	0x0
4	E1	R/W	Byte order.	0x1
3	E0	R/W	Byte order.	0x1
1	TM0	R/W	Transfer mode.	0x0
0	DP0	R/W	Done polarity.	0x1

Notes:
1 All E*n*, TM*n* and DP*n* bits must be set.
2 Byte order in word (E*n*):
0: Little endian. Least significant 16 bit transmitted on the low address.
1: Big endian. Most significant 16 bit transmitted on the low address.
3. Transfer Mode for CPU interface (TM*n*):
0: 16-bit transfer mode.
1: 8-bit transfer mode.
4 Done Polarity (DP*n*):
0: Done is active high.
1: Done is active low.

4.12 Secondary System Control, Block 7, Subblock 2

4.12.1 Host Interface Select

Table 270. HOST_INF_SELECT (Address 0x03), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
7	RESERVED	R/W	Reserved.	0x0
6	RSV2	R/W	Reserved.	0x0
5	RESERVED	R/W	Reserved.	0x0
4	RESERVED	R/W	Reserved.	0x0
3	RESERVED	R/W	Reserved.	0x0

4.12.2 Master Scheduler Configuration

Table 271. MSCH (Address 0x06), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
13:12	HDRLEN	R/W	Reserved. Default value may not be changed.	0x2

Table 271. MSCH (Address 0x06), Block 7, Subblock 2 *(continued)*

Bit	Name	Mode	Description	Default
11:10	CYCLEMIN	R/W	Cycle minimum. Minimum amount of time between scheduling of the same port in burst-interleaved mode, in units of 4 planes. The round trip delay from SPI-4.2 credit system issuing a stop until stop of data is 13 planes. To be SPI-4.2 compliant, 4 × CYCLEMIN should therefore be ≥ (13 – MSCH::BURSTLEN)/4 + 1.	0x2
9:8	RSV1	R/W	Reserved.	0x3
4	BURSTINTLV	R/W	Burst-interleaved enable. Selects output interleaving method when transmitting from the 12 FIFO buffers. This can either be full-frame or burst-interleaved mode. The parameters MSCH::BURSTLEN and ING_CT_THRHLD::CT_THRHLD control the burst-interleaving behavior. When transmitting in full-frame mode, a normalized header with correct length information must be present. NORMALIZER::NLE must be set correctly. 1: Burst-interleaved mode, a maximum of 32 × MSCH::BURSTLEN bytes are transmitted from each FIFO buffer. The FIFO buffers are served round robin. Empty FIFO buffers are skipped. 0: Full-frame mode, a complete frame is transmitted before switching to the next FIFO buffer. ING_CT_THRHLD::CT_THRHLD must be set to 0. The maximum frame size supported is 16 kilobytes – 1.	0x1
3:0	BURSTLEN	R/W	Burst Length in 32-byte units. Determines the SPI-4.2 maximum configured payload data transfer size, and must be set to the same value as SPI4_ING_SETUP2::BURSTSIZE. It has no effect if MSCH::BURSTINTLV = 0. Recommended settings are BURSTLEN = 4 (128 byte), SPI4_ING_SETUP2::MAXBURST1 = 8, and SPI4_ING_SETUP2::MAXBURST2 = 8.	0x2

4.12.3 Master Scheduler Sync Clear

Table 272. MSCH_SYNC_CLR (Address 0x08), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
11:0	SYNC_CLR	R/W	Clears the full-frame out-of-sync (MSCH_STAT::SYNC_LOST_STK). Each bit 11:0 corresponds to FIFO buffer 11:0. Setting a SYNC_CLR bit clears the corresponding MSCH_STAT::SYNC_LOST_STK bit. If the SYNC_CLR bit is = 1, the full-frame output scheduling does not stop outputting from bad FIFO buffers.	0x0000000

4.12.4 Master Scheduler Status

Table 273. MSCH_STAT (Address 0x09), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
11:0	SYNC_LOST_STK	R/W	Sticky bits indicating if full-frame transmission synchronization is lost. The frame is only checked for a START plane at the beginning. Each bit corresponds to FIFO buffers 11:0.	0x0000000

4.12.5 Egress CRC Error Count

Table 274. EGR_CRC_CNT (Address 0x0A), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
31:0	EGR_ERR_CNT	R/W	Egress CRC Error Count. Number of Ethernet CRC errors detected in frames received from the host interface.	0x00000000

4.12.6 Egress CRC Checker Configuration

Table 275. CRC_CFG (Address 0x0B), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
5	CRC_IGN	R/W	CRC Ignore. Disables the CRC checker. Frames with CRC errors, which enter the egress FIFO through the host interface, are not discarded. 0: Check the CRC field for errors. If EGR_CONTROL::FAIL_IGN is 0, frames with CRC errors are discarded. 1: Ignore the CRC field. Frames with CRC errors are forwarded.	0x0

Table 275. CRC_CFG (Address 0x0B), Block 7, Subblock 2 *(continued)*

Bit	Name	Mode	Description	Default
4	HDR	R/W	Normalized Header Present. If frames that enter the host interface have a normalized header, this bit must be set (unless CRC_CFG::CRC_IGN is set). 0: No header present. 1: Normalized header present.	0x0
3:0	CRC_HDR_SZ	R/W	Header Size. Header size information to the egress CRC checker. If CRC_CFG::HDR is 1, the size of the header must be written to CRC_HDR_SZ. The 9-byte normalized header is present.	0x00

4.12.7 Ingress Port Remapper

Table 276. REMAP_ING (Address 0x21), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
16	REMAP_WR	R/W	Write to the ingress port remapper table. After reset the remapping table is initialized with a one-to-one mapping: input port N mapped to output port N. 0: No action. 1: Write the REMAP_ING::REMAP_IN, REMAP_ING::REMAP_OUT pair to the remapper table.	0x0
12:8	REMAP_IN	R/W	The input port number equal to FIFO buffer number.	0x00
4:0	REMAP_OUT	R/W	The remapped output port number seen on the SPI-4.2 interface.	0x00

4.12.8 Egress Port Remapper

Table 277. REMAP_EGR (Address 0x22), Block 7, Subblock 2

Bit	Name	Mode	Description	Default
16	REMAP_WR	R/W	Write to the egress port remapper table. After reset the remapping table is initialized with a one-to-one mapping: input port N mapped to output port N. 0: No action. 1: Write the REMAP_EGR::REMAP_IN, REMAP_EGR::REMAP_OUT pair to the remapper table.	0x0
12:8	REMAP_IN	R/W	The input port number received from SPI-4.2 interface.	0x00
4:0	REMAP_OUT	R/W	The remapped output port number determining FIFO buffer number and the RGMII interface.	0x00

4.13 PI Local Registers, Block 7, Subblock 15

The following registers are only visible on the parallel interface. For more information, see "Parallel Interface," page 91.

4.13.1 Local PI Data

Table 278. LOCAL_DATA (Address 0xFE), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31:0	DATA	R/O	Data read from internal register. After a posted read of an internal register is issued, the value of the internal register is present when bit LOCAL_STATUS::RO = 1. Reading this register resets LOCAL_STATUS::RO.	0x00000000

4.13.2 Local PI Status

Table 279. LOCAL_STATUS (Address 0xFF), Block 7, Subblock 15

Bit	Name	Mode	Description	Default
31:16	AAD	R/O	Last address where an access was dropped due to overload of the internal register access system.	0x0000

Table 279. LOCAL_STATUS (Address 0xFF), Block 7, Subblock 15 *(continued)*

Bit	Name	Mode	Description	Default
2	SAO	R/O	Sticky access okay is set low if the current access was dropped due to overload of the internal register access system. It is set high after reading the LSB of the status register (the lowest 16 bit of this register). When doing multiple writes, checking this bit indicates if any of those writes went wrong.	0x0
1	RO	R/O	Reply okay is set low if the current read access of an internal register is dropped due to overload of the internal register access system. It is set high if the access went okay.	0x0
0	RSVO	R/O	Reserved.	0x1

5 ELECTRICAL SPECIFICATIONS

Specifications listed in the following tables are guaranteed over the recommended operating conditions listed in Table 307 unless otherwise noted.

5.1 DC Characteristics

The DC specifications for the Schaumburg device are covered in this section.

5.1.1 DC Specifications for PLL0_Clk and PLL1_Clk Signal

The reference clock inputs can be fed by using one of the following three signal types:

- LVDS signal
- LVPECL signal
- LVTTL signal by floating the opposite pin

When the inputs are in the differential mode, they can be DC-coupled or AC-coupled. If the inputs are AC-coupled, the internal bias is approximately 1.4 V.

Table 280. DC Specifications for PLLx_Clk in LVTTL Mode

Symbol	Parameter	Condition		Minimum	Typical	Maximum	Unit
		Supply	Input/Output				
V_{IH}	Input high voltage			2.0		3.6	V
V_{IL}	Input low voltage			–0.3		0.80	V
I_{IH}	Input high current	V_{DD_IO25} = minimum	V_I = 3.3 V			280	µA
I_{IL}	Input low current	V_{DD_IO25} = maximum	V_I = 0.0 V	–210			µA
C_I	Input capacitance				3		pF

Table 281. DC Specifications for PLLx_Clk in LVDS/LVPECL Mode

Symbol	Parameter	Minimum	Typical	Maximum	Unit
V_I	Input voltage range, V_{IA} or V_{IB}	0		3000	mV
V_{IC}	Input voltage common mode range	50		V_{DD_IO25} – 100	mV
$\|V_{ID}\|$	Input differential voltage	100		1300	mV
V_{IDTH}	Input differential threshold	–100		100	mV
C_I	Input capacitance		3		pF

5.1.2 DC Specifications for SPI-4.2 Data Channels

The LVDS outputs and inputs have been designed to meet or exceed the DC requirement of the ANSI/TIA/EIA-644-A-2001 LVDS standards modified to SPI-4.2 requirements.

All LVDS outputs and inputs comply with the specifications in Table 282 and Table 283.

Figure 45 illustrates the definitions of the LVDS potential, and Figure 46 shows the transmit test circuit. Figure 47, page 227, shows the LVDS DC definitions.

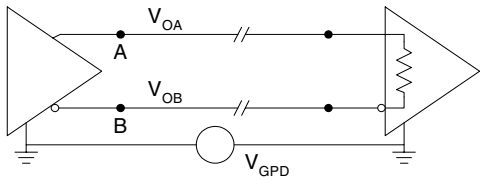


Figure 45. LVDS Potential Definitions

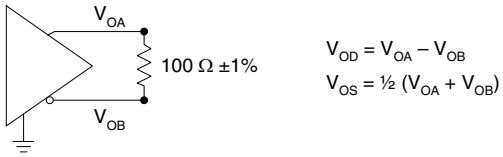


Figure 46. LVDS DC Transmit Test Circuit

Table 282. DC Specifications for LVDS Drivers

Symbol	Parameter	Condition	Minimum	Maximum	Unit
		Input/Output			
$\|V_{OD}\|$	Output differential voltage	R_L = 100 Ω ±1%	250	500	mV
V_{OS}	Output offset voltage	R_L = 100 Ω ±1%	1050	1375	mV
R_O	DC output impedance, single ended[1]	V_C = 1.0 V and 1.4 V	40	140	Ω
ΔR_O	R_O mismatch between A and B[1,2]	V_C = 1.0 V and 1.4 V		15	%
$\Delta\|V_{OD}\|$	Change in $\|V_{OD}\|$ between 0 and 1	R_{load} = 100 Ω ±1%		25	mV
ΔV_{OS}	Change in V_{OS} between 0 and 1	R_{load} = 100 Ω ±1%		50	mV
I_{OSA}, I_{OSB}	Output current[1, 3]	Driver shorted to GND		24	mA
I_{OSAB}	Output current[1, 3]	Drivers shorted together		12	mA

1. For a definition of the parameter and accompanying test circuit, see IEEE 1596.3-1996.
2. Matching of reflection cooefficients.
3. Parameters are named I_{sa}, I_{sb} and I_{sab} in IEEE 1596.3-1996.

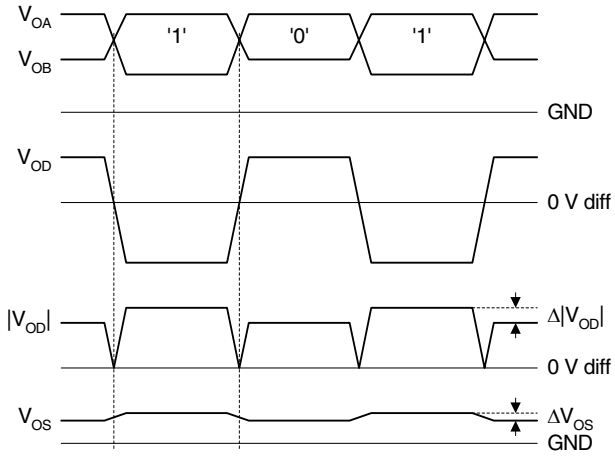


Figure 47. LVDS DC Definitions

Where,

$\Delta|V_{OD}| = | |V_{OAH} - V_{OBL}| - |V_{OBH} - V_{OAL}| |$

$\Delta V_{OS} = | ½(V_{OAH} + V_{OBL}) - ½(V_{OAL} + V_{OBH}) |$

Table 283. DC Specifications for LVDS Receivers with Built-In Termination

Symbol	Parameter	Condition	Minimum	Typical	Maximum	Unit
		Input/Output				
V_I	Input voltage range, V_{IA} or V_{IB}	$\|V_{GPD}\|$ < 825 mV	0		2400	mV
V_{IC}	Input voltage common mode range	$\|V_{ID}\|$ = 100 mV	50		2350	mV
$\|V_{ID}\|$	Input differential voltage	$\|V_{GPD}\|$ < 825 mV	100		600	mV
V_{IDTH}	Input differential threshold	$\|V_{GPD}\|$ < 825 mV	–100		100	mV
R_{IN}	Receiver differential input impedance		80		120	Ω
C_I	Input capacitance			3		pF

5.1.3 DC Specifications for RGMII and MII Management

The input and output values of the RGMII and MII Management interfaces meet the requirements of the JEDEC JESD8-5 2.5 V CMOS interface standard. All RGMII and MII Management input and output values comply with the specifications in the following table.

Table 284. DC Specifications for RGMII and MII Management

Symbol	Parameter	Condition		Minimum	Typical	Maximum	Unit
		Supply	Input/Output				
V_{OH}	Output high voltage	V_{DD_IO25} = minimum	I_{OH} = –1.0 mA	2.0			V
V_{OL}	Output low voltage	V_{DD_IO25} = Minimum[1]	I_{OL} = 1.0 mA			0.40	V
V_{IH}	Input high voltage			1.70		V_{DD_IO25} + 0.3	V
V_{IL}	Input low voltage			–0.3		0.70	V
I_{IH}	Input high current	V_{DD_IO25} = Minimum	V_I = 2.5 V			5	µA
I_{IL}	Input low current	V_{DD_IO25} = Maximum	V_I = 0.4 V	–5			µA
C_I	Input capacitance				5		pF

1. Worst-case V_{OL} is achieved at minimum supply voltage.

5.1.4 DC Specifications for PI, SI, JTAG, SPI-4.2 Status, and GPIO

The input and output values listed in the following table meet the requirements of the LVTTL and LVCMOS JEDEC JESD8-B (September 1999) standard except for the input leakage current in the low state.

Table 285. DC Specifications for PI, SI, JTAG, SPI-4.2 Status, and GPIO

Symbol	Parameter	Condition		Minimum	Typical	Maximum	Unit
		Supply	Input/Output				
V_{OH}	Output high voltage	V_{DD_OUT33} = minimum	I_{OH} = –2 mA	2.4			V
V_{OL}	Output low voltage	V_{DD_OUT33} = minimum[1]	I_{OL} = 2 mA			0.40	V
V_{IH}	Input high voltage			2.0		V_{DD_OUT33} + 0.3[2]	V
V_{IL}	Input low voltage			–0.3		0.8	V
I_{OH}	Output high current					-3.65	mA
I_{OL}	Output low current			3.65			mA
I_{IH}	Input high current	V_{DD_IO25} = minimum[1]	V_I = 3.3 V			5	µA
I_{IL}	Input low current	V_{DD_IO25} = maximum	V_I = 0.0 V	–75			µA

Table 285. DC Specifications for PI, SI, JTAG, SPI-4.2 Status, and GPIO *(continued)*

Symbol	Parameter	Condition		Minimum	Typical	Maximum	Unit
		Supply	Input/Output				
C_I	Input capacitance				5		pF

1. Worst-case VOL is achieved at minimum supply voltage.
2. V_{IH}(max) = 5.5 V for JTAG input signals

The following Schaumburg inputs and outputs comply with the specifications in Table 285.

JTAG_nTRST	PI_Addr[15:0]	PI_nOE	SI_DO	SPI_TSClk
JTAG_TCK	PI_Data[15:0]	PI_nRd	SI_nEn	SPI_TStat[1:0]
JTAG_TDI	PI_nCS	PI_nWR	SPI_En	GPIO[7:0]
JTAG_TDO	PI_nDone	SI_Clk	SPI_RSClk	
JTAG_TMS	PI_nDRdy	SI_DI	SPI_RStat[1:0]	

5.1.5 DC Specifications for Miscellaneous

The Miscellaneous I/Os are LVTTL inputs except that the high input voltage is reduced.

Table 286. DC Specifications for Miscellaneous

Symbol	Parameter	Condition		Minimum	Typical	Maximum	Unit
		Supply	Input/Output				
V_{IH}	Input high voltage			2.0		V_{DD_IO25} + 0.5	V
V_{IL}	Input low voltage			–0.3		0.8	V
I_{IH}	Input high current	V_{DD_IO25} = minimum	V_I = 2.5 V			1	µA
I_{IL}	Input low current	V_{DD_IO25} = maximum	V_I = 0.0 V	–1			µA
C_I	Input capacitance				5		pF

The following Schaumburg inputs and outputs comply with the specifications in Table 286.

Test_Enable	Test_Mode	nReset	Reserved_53	Reserved_54

5.2 AC Characteristics

The AC specifications for the Schaumburg device are covered in this section.

5.2.1 Reset Timing

The following figure shows the nReset signal waveform and required measurement points for the timing specification.

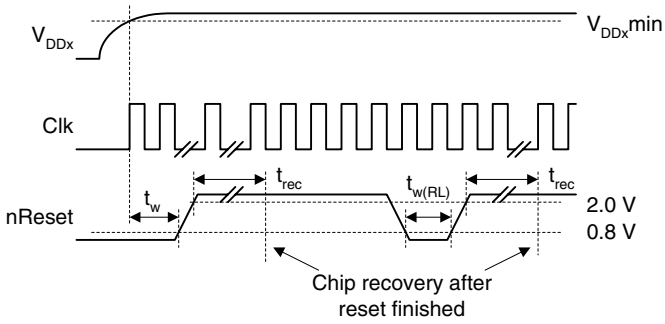


Figure 48. nReset Signal Timing Parameters

The nReset assertion time during power up is measured from when the later of the following conditions occur:

- Power supplies have reached their recommended operation condition levels, and the clock signal is stable.
- The nReset signal enters the transition region.

The signal applied to the nReset input must comply with the requirements listed in the following table at the reset pin of the Schaumburg device.

Table 287. AC Specifications for nReset

Symbol	Parameter	Condition	Minimum	Maximum	Unit
t_w	nReset assertion time after all power supplies and clock stable	See Figure 48.	20		ms
t_{rec}	Recovery time from reset inactive to first register read or write[1]	See Figure 48.		100	µs
$t_{w(RL)}$	Reset pulse width	See Figure 48.	20		ns
t_r, t_f	Rise time and fall time of nReset	0.8 V to 2.0 V.		10	ns

1. After reset release, the internal FIFO RAM clearing takes 210 µs, during which data cannot pass through the device. You must always run the BIST algorithm after reset release. For more information about the BIST algorithm, see "Initialization," page 76. The execution of the BIST algorithm takes approximately 200 ms, after which the device is fully operational.

5.2.2 PLL Timing

The signal applied to the PLLx_Clk input should comply with the requirements listed in the following table at the PLL pins of the Schaumburg device.

Table 288. AC Specifications for PLLx Input Clock

Symbol	Parameter	Condition	Minimum	Typical	Maximum	Unit
f	PLL0 nominal clock frequency	PLL0_Clk125_En = low		25		MHz
		PLL0_Clk125_En = high		125		MHz
f	PLL1 nominal clock frequency	Bit SPI4_XTAL_SEL = 0.	25		37.5	MHz
		Bit SPI4_XTAL_SEL = 1.	66.67		125	MHz
	Clock frequency tolerance	PLL0_Clk	–50		+50	ppm
		PLL1_Clk		See note[1].		ppm
	Clock duty cycle	LVTTL @ 1.4 V.	40		60	%
		LVDS, LVPECL @ V_{ID} = 0 V.	40		60	%
t_r, t_f	Clock rise time and fall time	LVTTL 0.8 V to 2.0 V.			5	ns
		LVDS, LVPECL 20% – 80% of V_S[2].			2	ns
$t_{j(PLL0)}$	PLL0 cycle-to-cycle period jitter p-p				75	ps
$t_{j(PLL1)}$	PLL1 cycle-to-cycle period jitter p-p				50[3]	ps

1. Refer to your own design requirements for SPI_RDClk frequency tolerance. The PLL1_Clk input tolerance should equal the required output tolerance of SPI_RDClk.
2. For the definition of V_S, see Figure 54.
3. Use of PLL-based reference clock oscillator must be avoided. For more information, see "SPI-4.2 Jitter Considerations," page 293 in Design Considerations.

The internal PLL uses the positive-going edge. This edge has to be free of excess jitter. Any jitter on this edge is passed to the output clocks.

5.2.3 MII Management

All AC specifications for the MII Management interface are designed to meet the requirements of IEEE Std 802.3-2002 (clause 22.2-4).

The timing reference levels are 0.7 V and 1.7 V.

The following figure shows the MII Management waveforms and the required measurement points for the signals.

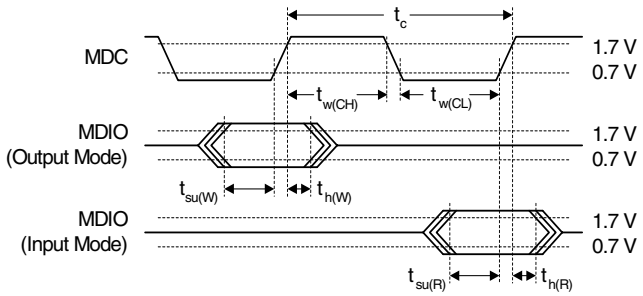


Figure 49. MII Management Waveforms

The setup time of the MDIO relative to the rising edge of the MDC is defined as the length of time between when the MDIO exits and remains out of the switching region and when the MDC enters the switching region. The hold time of the MDIO relative to the rising edge of the MDC is defined as the length of time between when the MDC exits the switching region and when the MDIO enters the switching region.

All MII management transmit signals comply with the specifications in the following table. The MDIO signal receive requirements must be provided at the signal receive pin of the Schaumburg device.

MDIO is tri-stated between accesses.

Table 289. AC Specifications for MII Management

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	MDC frequency		0.29	12.5	MHz
t_c	MDC cycle time		80[1]		ns
$t_{w(CH)}$	MDC time high	C_L = 50 pF	35		ns
$t_{w(CL)}$	MDC time low	C_L = 50 pF	35		ns
$t_{su(W)}$	MDIO setup to MDC on output mode	C_L = 50 pF	15		ns
$t_{h(W)}$	MDIO hold from MDC on output mode	C_L = 50 pF	15		ns
$t_{su(R)}$	MDIO setup to MDC on input mode	C_L = 50 pF on MDC	15		ns
$t_{h(R)}$	MDIO hold from MDC on input mode	C_L = 50 pF	0		ns

1. The Schaumburg device supports MDC clock speeds up to 12.5 MHz to allow faster communication with PHYs. If the standard frequency of 2.5 MHz is used, the MII management interface is designed to meet and exceed the IEEE Std 802.3 requirements of MDC minimum high and low times of 160 ns, and a MDC minimum cycle time of 400 ns. These requirements are not possible to fulfill at higher frequencies.

5.2.4 RGMII (10/100/1000 Mbps)

All AC specifications for the RGMII interface are compliant with the HP RGMII/RTBI draft 1.3 and meet the requirement in that document.

The skew between the clock and the Tx and Rx signals is defined as the amount of time it takes for the signals and the clock to reach the threshold value of 1.25 V. This is a double-data rate interface, so both the rising edge and falling edge of the clock are used.

The following two figures show the RGMII transmit and receive waveforms and required measurement points for the different signals. All AC timing requirements are specified relative to 1.25 V. All measurements are made at the measurement point shown in Figure 52, page 234.

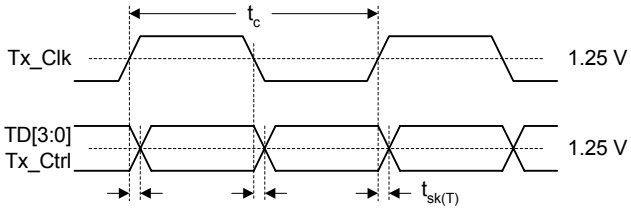


Figure 50. RGMII Transmit Waveforms

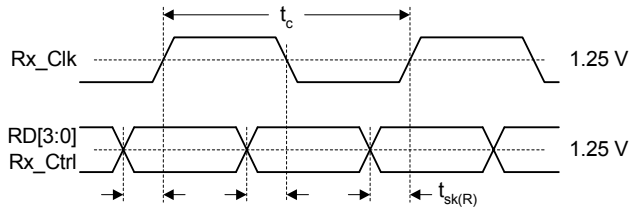


Figure 51. RGMII Receive Waveforms

All RGMII transmit signals comply with the specifications in the following two tables when measured using the test circuit shown in Figure 52. All RGMII receive signal requirements are requested at the device balls.

Table 290. AC Specifications for RGMII 1000 Mbps

Symbol	Parameter	Condition	Minimum	Typical	Maximum	Unit
f	Clock frequency	1000 Mbps		125		MHz
	Clock frequency stability		–50		+50	ppm
t_c	Clock cycle time	@ 1.25 V	7.2	8	8.8	ns
	Clock duty cycle	@ 1.25 V	45	50	55	%
t_r	RGMII signal rise time	20% – 80%[1]			0.75	ns
t_f	RGMII signal fall time	80% – 20%[1]			0.75	ns
$t_{sk(T)}$	Data to clock output skew	@ 1.25 V	–500	0	500	ps

Table 290. AC Specifications for RGMII 1000 Mbps *(continued)*

Symbol	Parameter	Condition	Minimum	Typical	Maximum	Unit
$t_{sk(R)}$	Data to clock input skew	@ 1.25 V	1		2.6	ns

1. Measured relative to worst-case DC output levels.

Table 291. AC Specifications for RGMII 10/100 Mbps

Symbol	Parameter	Condition	Minimum	Typical	Maximum	Unit
f	Clock frequency	100 Mbps		25		MHz
		10 Mbps		2.5		MHz
	Clock frequency stability		–50		+50	ppm
t_c	Clock cycle time	100 Mbps	36	40	44	ns
		10 Mbps	360	400	440	ns
	Clock duty cycle	@ 1.25 V	40	50	60	%
t_r	RGMII signal rise time	20 – 80%[1]			0.75	ns
t_f	RGMII signal fall time	80 – 20%[1]			0.75	ns
$t_{sk(T)}$	Data to clock output skew	@ 1.25 V	–500	0	500	ps
$t_{sk(R)}$	Data to clock input skew	@ 1.25 V	1			ns

1. Measured relative to worst-case DC output levels.

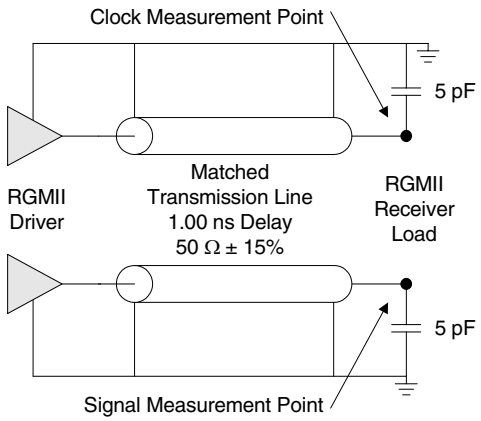


Figure 52. RGMII Test Circuit

5.2.5 SPI-4.2 Interface

All AC specifications for the Schaumburg SPI-4.2 interface are compliant with the OIF-SPI4-02.0 and meet the requirement defined in the standard.

The following figure shows the SPI-4.2 signal groups and their directions. The status signals are LVTTL and the data signals are LVDS (differential).

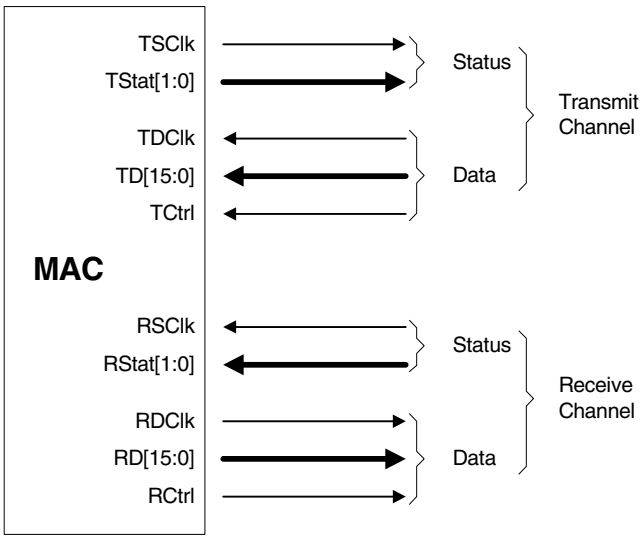


Figure 53. SPI-4.2 Signal Groups and Directions

All SPI-4.2 LVDS signal AC timing requirements are specified relative to the 0 V differential. All LVTTL signal AC timing requirements are specified relative to 1.4 V.

The unit-timing interval, t_{ui}, is defined as half the average period time for the clock signal.

The clock rise time (t_r) and fall time (t_f) parameters and other transient performance specifications are defined in Figure 54. The voltage swing, V_S, is defined as the difference between the steady state high and low voltage of the differential signal.

The SPI-4.2 LVDS parameters are referenced at the package edge, not the ball edge, as shown in Figure 55.

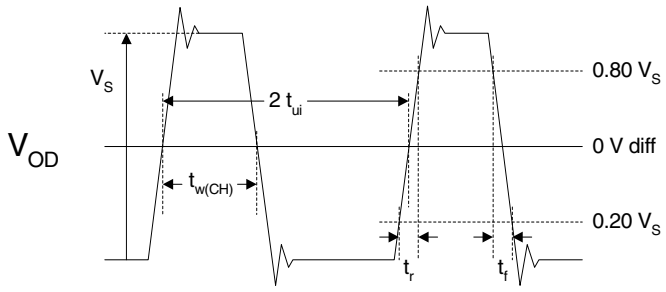


Figure 54. Transient Parameters of the Differential LVDS Signals

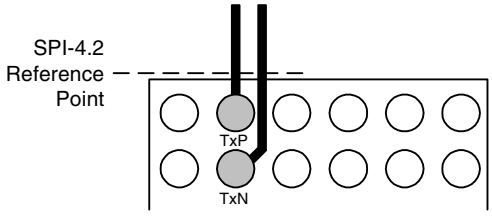


Figure 55. SPI-4.2 LVDS Reference Point

5.2.5.1 SPI-4.2 Transmit Data Channel (Static Alignment Mode), Egress

Figure 56 shows the SPI-4.2 data transmit waveforms and required measurement points for the different signals. Notice that the transmit channel signals are inputs to the Schaumburg device.

The setup time of a SPI-4.2 signal relative to the clock edge is defined as the length of time between when the differential signal exits and remains out of the switching region and when the clock passes the threshold value (0 V differential).

The hold time of an SPI-4.2 signal relative to the clock edge is defined as the length of time between when the clock passes the threshold voltage (0 V differential) and when the differential signals enter the switching region.

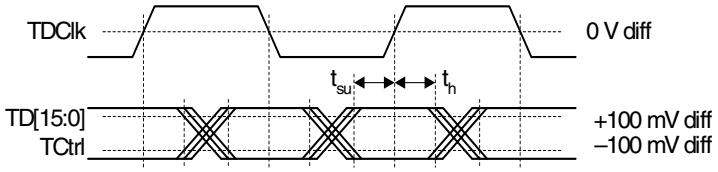


Figure 56. SPI-4.2 Transmit Data Channel Interface Timing Definitions (Device Inputs)

All signals applied to the SPI-4.2 transmit data channel pins (device inputs) should comply with the requirements in the following table that are referenced at the package edge, not the ball, as shown in Figure 55.

Table 292. SPI-4.2 Transmit Data Channel Interface Timing (Static Alignment)[1]

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	TDClk frequency		75	407	MHz
	TDClk duty cycle	@ 0 V differential	45	55	%
t_{su}	Data and control setup time		225		ps
t_h	Data and control hold time		155		ps
t_r, t_f[2]	Rise time and fall time for all signals	20% – 80% of V_S	100	0.36 t_{ui}	ps

1. AC requirements are defined at the package edge under the condition that the routing guidelines for the SPI-4.2 interface are followed when routing from ball to the package edge and that the differential impedance is 100 Ω. For information about the routing guidelines, see "SPI-4.2 Interface," page 297.
2. Rise times and fall times exclude reflections.

5.2.5.2 SPI-4.2 Transmit Data Channel (Dynamic Alignment Mode), Egress

The data channel signals should comply with the requirements in the following table. The $t_{d(SK)}$ skew delay is the maximum static skew between any of the 17 transmit data channel signals at the receiver end. The $t_{d(SK)}$ skew delay does not contain any jitter components. Jitter is normally averaged away when this parameter is measured. The $t_{jr(D)}$ parameter is measured on any of the data channel signals while keeping TDClk fixed. This means triggering on the TDClk while measuring on a data channel signal. Waveforms should be collected for a duration larger than the time interval between SPI-4.2 training sequences. The timing reference point is shown in Figure 56. Alfa for training sequences must be greater than 3.

Table 293. SPI-4.2 Transmit Data Channel Interface Timing (Dynamic Alignment)[1]

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	TDClk frequency		300	407	MHz
	TDClk duty cycle	0 V differential.	45	55	%
t_r, t_f [2]	Rise time and fall time for all signals	20% – 80% of V_S.	100	0.36 t_{ui}	ps
$t_{d(SK)}$	Static skew at receiver between any pair of TD[15:0], TCtrl	See Figure 57.		1.00	UI
$t_{jr(D)}$	Peak-peak jitter at receiver relative to TDClk [3]	See Figure 58.		0.20 [4]	UI

1. AC requirements are defined at the package edge under the condition that the routing guidelines for the SPI-4.2 interface are followed when routing from ball to the package edge and that the differential impedance is 100 Ω. For information about the routing guidelines, see "SPI-4.2 Interface," page 297.
2. Rise time and fall time exclude reflections.
3. Measured with an oscilloscope using waveform persistence that is larger than the repetition interval of training sequences.
4. For more information, see "SPI-4.2 Jitter Considerations," page 293.

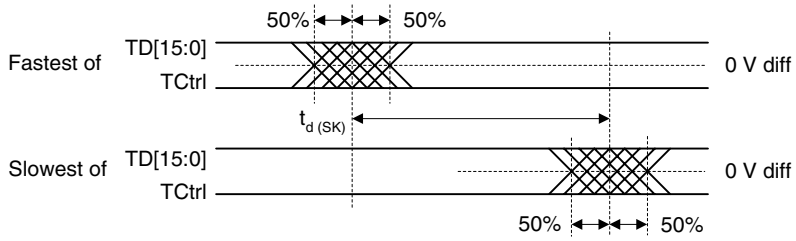


Figure 57. SPI-4.2 Transmit Data Channel Static Skew Definition (Dynamic Alignment)

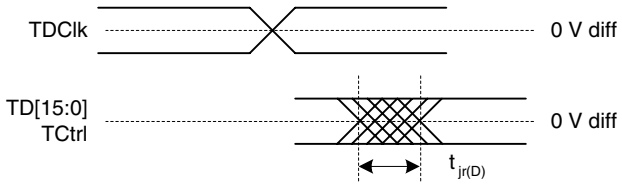


Figure 58. SPI-4.2 Transmit Data Channel Jitter Definition (Dynamic Alignment)

5.2.5.3 SPI-4.2 Transmit Status Channel, Egress

The following figure shows the SPI-4.2 transmit status waveforms and required measurement points for the different signals.

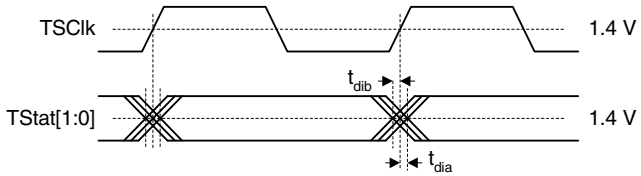


Figure 59. SPI-4.2 Transmit Status Channel Interface Timing Definitions

All SPI-4.2 transmit status signals comply with the conditions in the following table when measured using the test circuit shown in Figure 60.

Table 294. SPI-4.2 Transmit Status Channel Interface Timing

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	TSClk frequency[1]		18.75	102	MHz
	TSClk duty cycle	@ 1.4 V	40	60	%
t_{dia}	Status invalid after clock			2.5	ns
t_{dib}	Status invalid before clock			1	ns

1. The TSClk frequency can be set to either TDClk / 4 or 75 MHz.

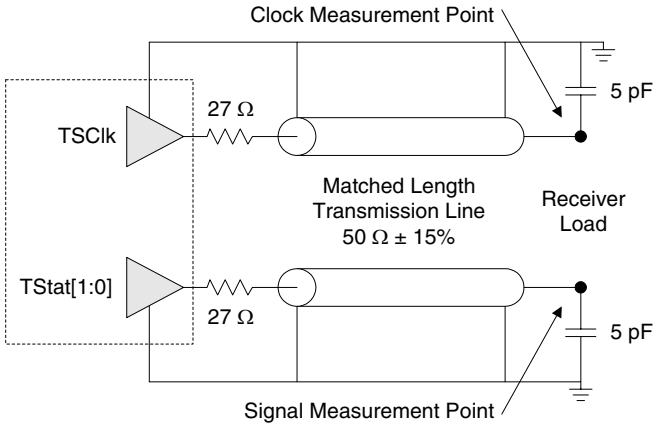


Figure 60. SPI-4.2 Transmit Status Channel Test Circuit

5.2.5.4 SPI-4.2 Receive Data and Status Channel (Device Output), Ingress

Figure 61 shows the SPI-4.2 receive data waveforms and required measurement points for the different signals.

The data invalid time period before clocking a SPI-4.2 signal relative to the clock edge is defined as the length of time between when the differential signal enters the switching region and when the clock passes the threshold voltage (0 V differential).

The data invalid time after clocking a SPI-4.2 signal relative to the clock edge is defined as the length of time between when the clock passes the threshold value (0 V differential) and when the differential signal exits and remains out of the switching region.

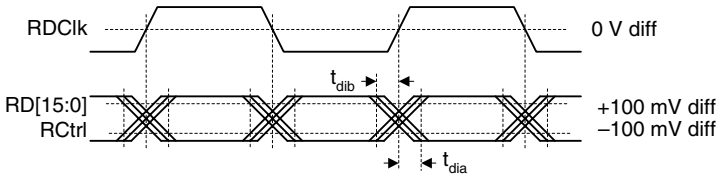


Figure 61. SPI-4.2 Receive Data Channel Interface Timing Definitions (Device Output)

All signals of the SPI-4.2 drivers (receive data channel) comply with the conditions in the following table when measured with the test circuit shown in Figure 62, and referenced at the package edge, not the ball, as shown in Figure 55, page 236.

Table 295. SPI-4.2 Receive Data Channel Interface Timing (Device Output)[1]

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	RDClk frequency		75 [2]	407 [2]	MHz
	RDClk duty cycle	@ 0 V differential	48	52	%
$t_{j(Clk)}$	RDClk phase jitter p-p[3]			0.10	UI
$t_{j(R)}$	RD and RCtrl phase jitter p-p [3]	PLL1 differential clock		0.12	UI
		PLL1 single-ended clock		0.24	UI
t_{dia}	Data invalid after clock			280	ps
t_{dib}	Data invalid before clock			280	ps
t_r, t_f	Rise time and fall time for all signals [4]	20% – 80% of V_S	100	500 [5]	ps

1. AC requirements are defined at the package edge under the condition that the routing guidelines for the SPI-4.2 interface are followed when routing from ball to the package edge and that the differential impedance is 100 Ω. For information about the routing guidelines, see "SPI-4.2 Interface," page 297.
2. Three ranges: 75 MHz to 112.5 MHz, 150 MHz to 225 MHz, and 300 MHz to 407 MHz.
3. For more information, see "SPI-4.2 Jitter Considerations," page 293.
4. Rise time and fall time exclude reflections
5. The maximum rise time and fall time is made frequency independent compared to the standard specified $0.3t_{ui}$.

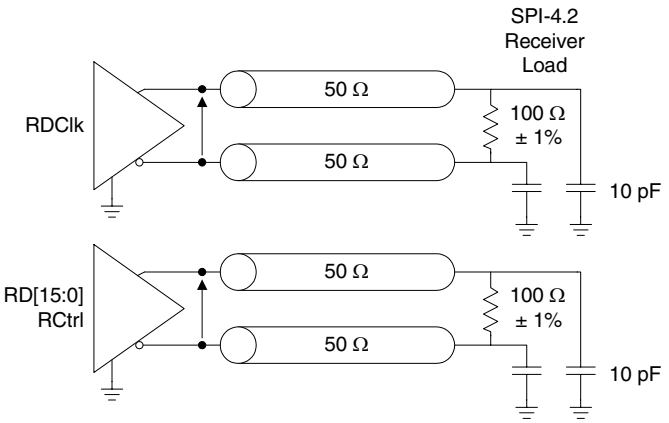


Figure 62. SPI-4.2 Receive Channel Test Circuit

Figure 63 shows the SPI-4.2 transmit status waveforms and required measurement points for the different signals.

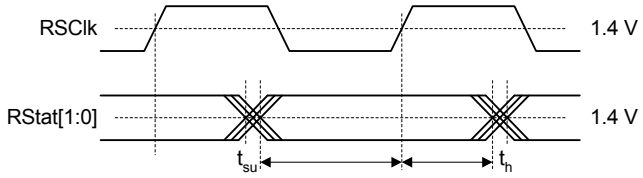


Figure 63. SPI-4.2 Receive Status Channel Interface Timing Definitions

All signals applied to the SPI-4.2 receiver status inputs should comply with the requirements in the following table at the pins of the Schaumburg device.

Table 296. SPI-4.2 Receive Status Channel Interface Timing

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	RSClk frequency		0	112	MHz
	RSClk duty cycle	@ 1.4 V	40	60	%
t_{su}	Status setup time[1]		2.0		ns
t_h	Status hold time[1]		0.5		ns

1. The minimum values are attained under conditions where RSClk and RStat[1:0] have a maximum rise time and fall time of 3 ns measured from 0.8 V to 2.0 V.

5.2.5.5 SPI-4.2 Receive Channel 90° Mode (Device Output)

Figure 64 shows the SPI-4.2 90° mode receive data waveforms and required measurement points for the different signals.

The setup time to clock of a SPI-4.2 signal relative to the clock edge is defined as the length of time between when the signal leaves the switching region (and remains out of the switching region) and when the clock passes the threshold voltage 0 V differential.

The hold time after clock of a SPI-4.2 signal relative to the clock edge is defined as the length of time between when the clock passes the threshold value 0 V differential. and when the signal enters the switching region.

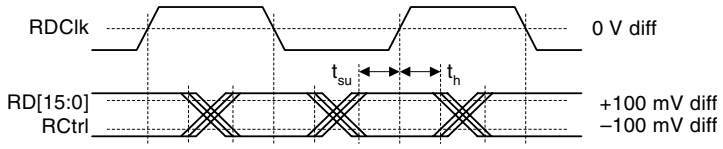


Figure 64. SPI-4.2 Receive Data Channel Interface Timing Definitions (Device Output), 90° Mode

All SPI-4.2 drivers (receive data channel) signals comply with the conditions in the following table when measured with the test circuit shown in Figure 62.

Table 297. SPI-4.2 Receive Data Channel Interface Timing (Device Output), 90° Mode[1]

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	RDClk frequency		75 [2]	407 [2]	MHz
	RDClk duty cycle	@ 0 V differential	48	52	%
$t_{j(Clk)}$	RDClk phase jitter p-p [3]			0.10	UI
$t_{j(R)}$	RD and RCtrl phase jitter p-p [3]			0.24	UI
t_{su}	RDx setup to RDClk [4]		0.24 × T–330 [5]		ps
t_h	RDx hold after clock [4]		0.24 × T–330 [5]		ps
t_r, t_f	Rise time and fall time for all signals[6]	20% – 80% of V_S	100	500 [7]	ps

1. AC requirements are defined at the package edge under the condition that the routing guidelines for the SPI-4.2 interface are followed when routing from ball to the package edge and that the differential impedance is 100 Ω. For information about the routing guidelines, see "SPI-4.2 Interface," page 297.
2. Three ranges: 75 MHz to 112.5 MHz, 150 MHz to 225 MHz, and 300 MHz to 407 MHz.
3. For more information, see "SPI-4.2 Jitter Considerations," page 293.
4. T = 1/f.
5. Minimum t_{su} and minimum t_h will not occur at the same time.
6. Rise time and fall times exclude reflections.
7. The maximum rise time and fall time is made frequency independent compared to the standard specified $0.3t_{ui}$.

5.2.6 Parallel CPU Interface (PI)

The parallel CPU interface can operate in either two-wire or three-wire mode. In both modes, the use of PI_nOE is optional.

5.2.6.1 PI Two-Wire Mode

This mode requires that PI_nRd is always low. If PI_nOE is not used, then it must be pulled low.

Figure 65 shows the read-timing diagram for the two-wire mode. Figure 66 shows the write-timing diagram for the two-wire mode.

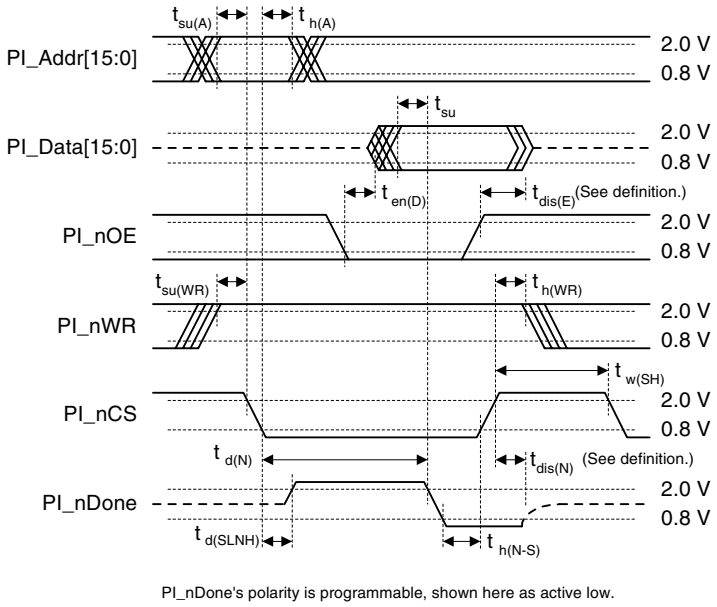


Figure 65. Two-Wire Mode Read Timing Diagram With and Without OE

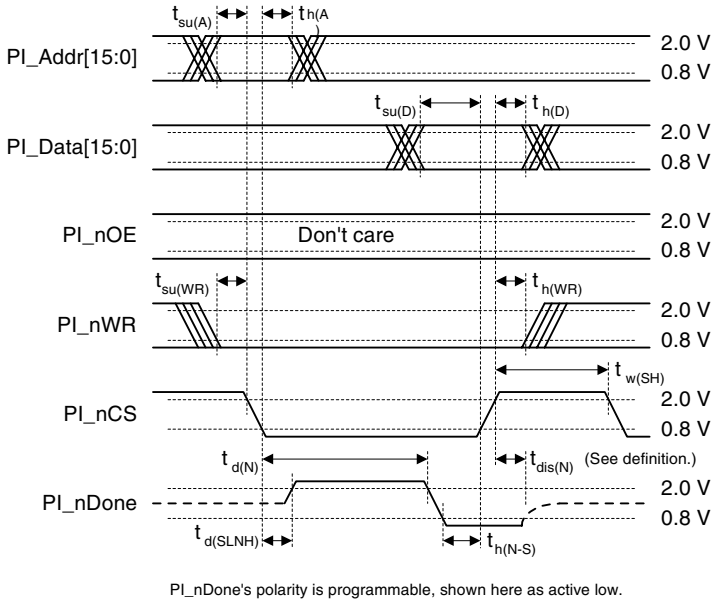


Figure 66. Two-Wire Mode Write Timing Diagram With and Without OE

All PI signals comply with the specifications in the following table. The PI receive signal requirements must be provided at the receive pins on the Schaumburg device.

Table 298. PI Two-Wire Mode Timing Specifications

Symbol	Parameter	Condition	Minimum	Maximum	Unit
$t_{su(A)}$	Addr setup to nCS falling		–2		ns
$t_{h(A)}$	Addr hold from nCS low		18		ns
$t_{su(WR)}$	nWR setup to nCS falling		2		ns
$t_{h(WR)}$	nWR hold from nCS high		2		ns
$t_{d(SLNH)}$	Delay from low nCS to nDone high [1]	C_L = 30 pF.		8	ns
$t_{d(N)}$	Delay from low nCS to nDone falling [1]	C_L = 30 pF.		43	ns
$t_{h(N-S)}$	nCS hold from nDone low [1]	C_L = 30 pF.	0		ns
$t_{dis(N)}$	nDone disable time from nCS high [1, 2]	See Figure 67.		8	ns
$t_{w(SH)}$	Width of nCS high		18 [3]		ns
$t_{su(D)}$	Write data setup to nCS rising		11		ns
$t_{h(D)}$	Write data hold from nCS rising		3		ns
$t_{en(D)}$	nOE and nCS low to read data enabled [4]	C_L = 30 pF.		10	ns
t_{su}	Read data setup to nDone falling on read [1]	C_L = 30 pF.	3		ns
$t_{dis(E)}$	Read data disable time from either nCS or nOE high [2, 4]	See Figure 67.		10	ns

1. nDone's polarity is programmable. It is illustrated as active low in Figure 65 and Figure 66.
2. The pin begins to float when a 300 mV change occurs from the loaded V_{OH}/V_{OL} level.
3. The number is the required device select inactive period between the two 16-bit accesses of a full 32-bit access for the required device select. The time between two different 32-bit write accesses is 130 ns. The time between a 32-bit write access and the first read access is 250 ns. The time from an initial read access to the first of the two accesses to LOCAL_DATA::DATA is based on the timing of PI_nDRdy, resulting in a worst-case time of 1.00 µs.
4. Internal data output enable requires both nOE and nCS active.

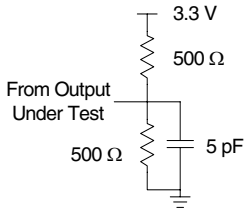


Figure 67. Test Circuit for PI Signal Disable Test

5.2.6.2 PI Three-Wire Mode

If PI_nOE is not used, it must be pulled low.

The following figure shows the read-timing diagram for the three-wire mode.

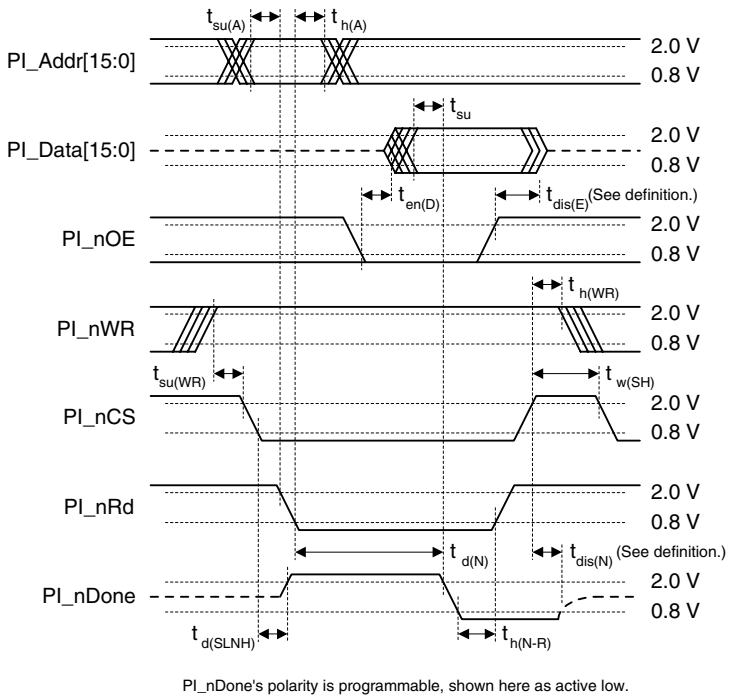
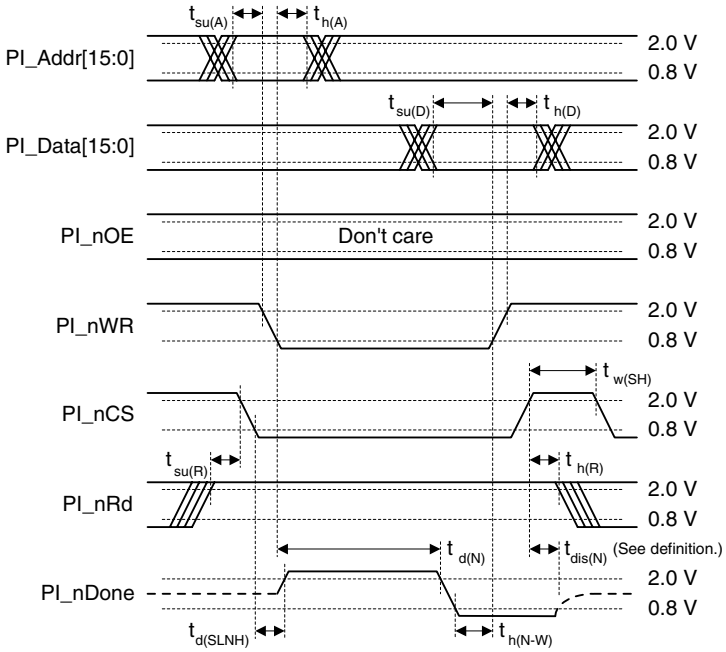


Figure 68. Three-Wire Mode Read Timing Diagram

Table 299. PI Three-Wire Mode Read Timing Specifications

Symbol	Parameter	Condition	Minimum	Maximum	Unit
$t_{su(A)}$	Addr setup to nRd falling[1]		2		ns
$t_{h(A)}$	Addr hold from nRd low[1]		18		ns
$t_{su(WR)}$	nWR setup to nCS falling		2		ns
$t_{h(WR)}$	nWR hold from nCS high		2		ns
$t_{d(SLNH)}$	Delay from low nCS to nDone high [2]	C_L = 30 pF.		8	ns
$t_{d(N)}$	Delay from low nRd to nDone falling [1, 2]	C_L = 30 pF.		43	ns
$t_{h(N-R)}$	nRd and nCS hold from nDone low [1, 2]		0		ns
$t_{dis(N)}$	nDone disable time from nCS high [2, 3]	See Figure 67.		8	ns
$t_{w(SH)}$	Width of nCS high		18 [4]		ns
$t_{en(D)}$	nOE, nCS and nRd low to read data enabled [5]	C_L = 30 pF.		10	ns
t_{su}	Read data setup to nDone falling on read [2]	C_L = 30 pF.	3		ns
$t_{dis(E)}$	Read data disable time from either nCS, nOE or nRd high [3, 5]	See Figure 67.		10	ns

1. nRd and nCS are ORed together internally. If nCS goes low after or high before nRd, then the respective timing should be referred to nCS instead.
2. nDone's polarity is programmable, here it is illustrated as active low.
3. The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.
4. The number is the required chip select inactive period between the two 16-bit accesses of a full 32-bit access. The time between two different 32-bit write accesses is 130 ns. The time between a 32-bit write access and the first read access is 250 ns. The time from an initial read access to the first of the two accesses to LOCAL_DATA is based on the timing of PI_nDRdy, resulting in a worst-case time of 1.00 µs.
5. Internal data output enable requires that nOE, nCS and nRd all are active.



PI_nDone's polarity is programmable, shown here as active low.

Figure 69. Three-Wire Mode Write Timing Diagram

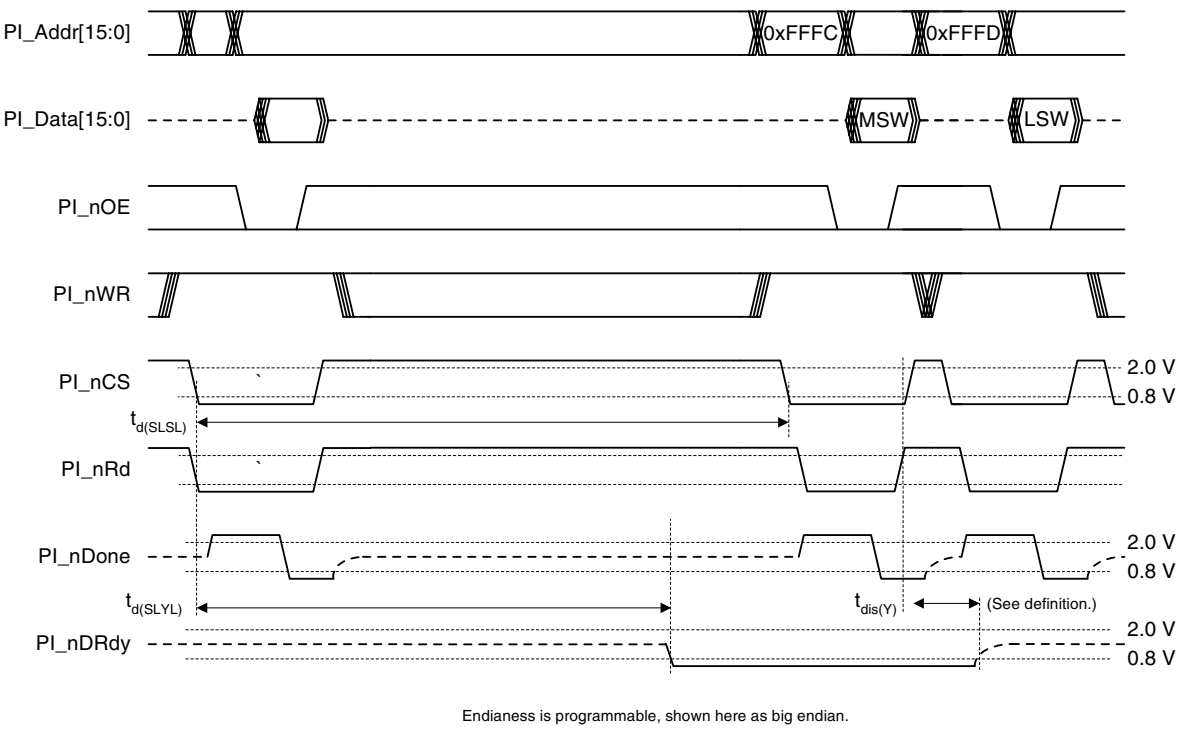
Table 300. PI Three-Wire Mode Write Timing Specifications

Symbol	Parameter	Condition	Minimum	Maximum	Unit
$t_{su(A)}$	Addr setup to nWR falling[1]		2		ns
$t_{h(A)}$	Addr hold from nWR low [1]		18		ns
$t_{su(R)}$	nRd setup to nCS falling		2		ns
$t_{h(R)}$	nRd hold from nCS high		2		ns
$t_{d(SLNH)}$	Delay from low nCS to nDone high [2]	C_L = 30 pF.		8	ns
$t_{d(N)}$	Delay from low nWR to nDone falling [1, 2]	C_L = 30 pF.		43	ns
$t_{h(N-W)}$	nWR and nCS hold from nDone low [1]		0		ns
$t_{dis(N)}$	nDone disable time from nCS high [2, 3]	See Figure 67.		8	ns
$t_{w(SH)}$	Width of nCS high		18 [4]		ns
$t_{su(D)}$	Write data setup to nWR rising [1]		11		ns
$t_{h(D)}$	Write data hold from nWR high [1]		3		ns

1. nWR and nCS are ORed together internally. If nCS goes low after or high before nWR, then the respective timing should be referred to nCS instead.
2. nDone's polarity is programmable, here it is illustrated as active low.
3. The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.
4. The number is the required device select inactive period between the two 16-bit accesses of a full 32-bit access. The time between two different 32-bit write accesses is 130 ns. The time between a 32-bit write access and the first read access is 250 ns. The time from an initial read access to the first of the two accesses to LOCAL_DATA::DATA is based on the timing of PI_nDRdy, resulting in a worst-case time of 1.00 µs.

5.2.6.3 PI Full Read Access

Figure 70 shows a full posted read access using three-wire mode. The shown address and data correspond to big endian mode.



Endianess is programmable, shown here as big endian.

Figure 70. Full Posted Read Access

Table 301. PI nDRdy Timing Specification

Symbol	Parameter	Condition	Minimum	Maximum	Unit
$t_{d(SLYL)}$	Delay from start-of-read to nDRdy low [1]			1000	ns
$t_{d(SLSL)}$	Delay from start-of-read register to start-of-read local register		1000		ns
$t_{dis(Y)}$	nDRdy disable time from end-of-read of a local register [2, 3, 4]	See Figure 67.		20	ns

1. Start-of-read is the first time not(nCS or nRd or not nWR) is true.
2. The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.
3. End-of-read is the first time (nCS or nRd or not nWR) is true.
4. nDRdy is disabled by the end-of-read of the MSW and LSW of LOCAL_DATA, and of the LSW of LOCAL_STATUS. Reading the MSW of LOCAL_STATUS does not influence nDRdy.

5.2.7 Serial CPU Interface (SI)

All SI AC timing requirements are specified relative to the input low and high threshold levels.

The SI timing parameters and required measurement points are defined in the following two figures.

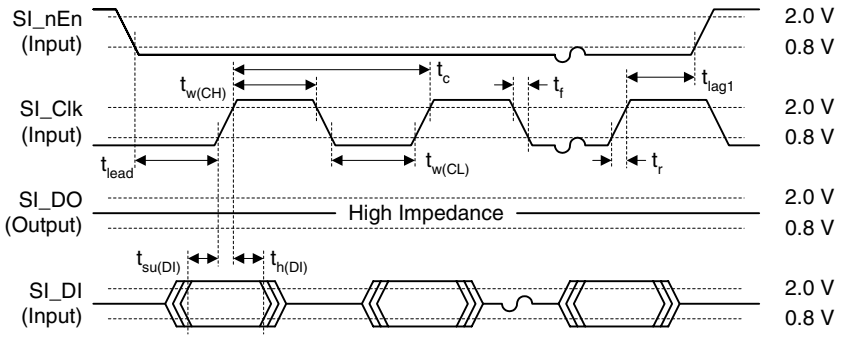


Figure 71. SI Input Data Waveform

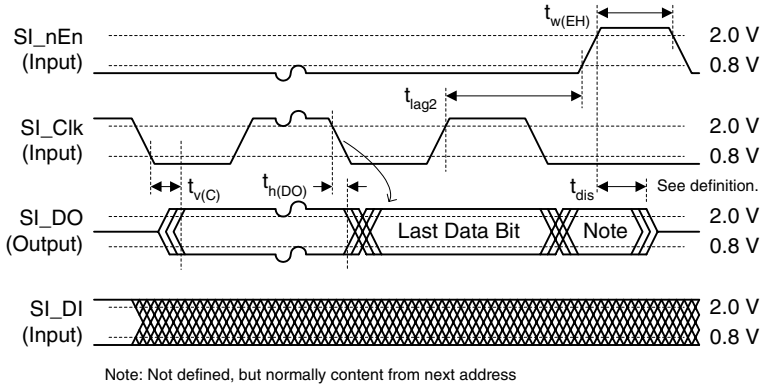


Figure 72. SI Output Data Waveform

All SI signals comply with the specifications in the following table. The SI receive signal requirements are requested at the receive pins on the Schaumburg device.

Table 302. AC Specifications for SI

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	Clock frequency			24[1]	MHz
$t_{w(CH)}$	Clock time high		16		ns
$t_{w(CL)}$	Clock time low		16		ns
t_r, t_f	Clock rise time and fall time	Between V_{IL} and V_{IH}.		10	ns
$t_{su(DI)}$	DI setup to clock		4		ns
$t_{h(DI)}$	DI hold from clock		2		ns

Table 302. AC Specifications for SI *(continued)*

Symbol	Parameter	Condition	Minimum	Maximum	Unit
t_{lead}	Enable active before first clock		10		ns
t_{lag1}	Enable inactive after clock (input cycle) [2]		27		ns
t_{lag2}	Enable inactive after clock (output cycle)		See note[3].		ns
$t_{w(EH)}$	Enable inactive width		10		ns
$t_{v(C)}$	DO valid after clock	C_L = 30 pF.		12	ns
$t_{h(DO)}$	DO hold from clock	C_L = 0 pF.	0		ns
t_{dis}	DO disable time [4]	See Figure 73.		15[4]	ns

1. If the SI clock frequency exceeds 0.5 MHz, dummy bytes must be inserted.
2. t_{lag1} is only defined for write operations to the chip, not read.
3. The last rising edge on the clock is necessary for the master to be able to read in the data. The lag time depend on the necessary hold time on the master data input.
4. The pin begins to float when a 300 mV change from the loaded VOH/VOL level occurs.

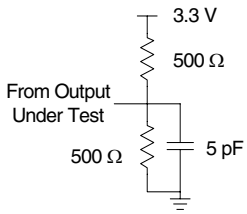


Figure 73. Test Circuit for SI_DO Disable Test

5.2.8 JTAG Interface

All AC specifications for the JTAG interface have been designed to meet the requirements of IEEE Std 1149.1-2001.

All JTAG AC timing requirements are specified relative to the input low and high threshold level.

The following figure shows the JTAG transmit and receive waveforms and required measurement points for the different signals.

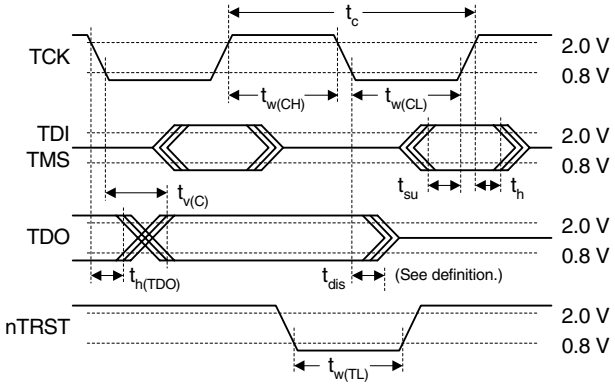


Figure 74. JTAG Interface Timing Definitions

All JTAG signals comply with the specifications in the following table, and the JTAG receive signal requirements are requested at the device pins.

Table 303. AC Specifications for JTAG

Symbol	Parameter	Condition	Minimum	Maximum	Unit
f	TCK frequency			10	MHz
t_c	TCK cycle time		100		ns
$t_{w(CH)}$	TCK time high		40		ns
$t_{w(CL)}$	TCK time low		40		ns
t_{su}	Setup to TCK		10		ns
t_h	Hold from TCK		10		ns
$t_{v(C)}$	TDO valid after clock	C_L = 10 pF.		28	ns
$t_{h(TDO)}$	TDO hold from clock	C_L = 0 pF.	0		ns
t_{dis}	TDO disable time [1]	See Figure 75.		30	ns
$t_{w(TL)}$	nTRST time low		30		ns

1. The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.

JTAG_nTRST signal is asynchronous to the clock and consequently does not have a setup and hold time requirement.

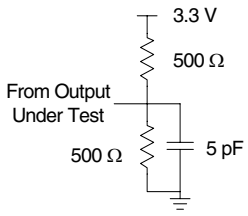


Figure 75. Test Circuit for TDO Disable Test

5.3 Power Consumption

5.3.1 Maximum Operating Current

Table 304 shows the absolute maximum operating current for the Schaumburg device. These currents are found at the worst-case silicon and worst-case operating conditions.

Table 304. Maximum Operating Current

Symbol	Parameter	Condition	Maximum	Unit
I_{DD}	Maximum active operating current for core	V_{DD} = maximum	2.6	A
I_{DD_IO25}	Maximum active operating current for V_{DD_IO25}	V_{DD_IO25} = maximum	1.05	A
I_{DD_OUT33}	Maximum active operating current for V_{DD_OUT33}	V_{DD_OUT33} = maximum	0.3	A
I_{DD_PLLx}	Maximum active operating current for both PLLs	V_{DD_PLLx} = maximum	0.08	A
Note: All power consumption values are at 100% Tx activity.				

5.3.2 Typical Current Consumption

Table 305 shows values for current consumption of the Schaumburg device in typical situations. The numbers are measured under typical process, temperature, and supply voltage conditions.

In Table 305, all power consumption values are at 100% Tx activity on the specified ports.

Table 305. Typical Current Consumption

Configuration	Current Consumption			Power
	V_{DD} (1.8 V)	V_{DD_IO25} + V_{DD_PLLx} (2.5 V)	V_{DD_OUT33} (3.3 V)	
12 × 1 G RGMII + SPI-4.2[1]	2.0 A	0.75 A	0.02 A	5.5 W
8 × 1 G RGMII + SPI-4.2[1]	1.7 A	0.60 A	0.02 A	4.7 W

5.4 Stress Ratings

Table 306. Stress Ratings

Symbol	Parameter	Minimum	Maximum	Unit
T_S	Storage temperature	–65	150	°C
	Junction temperature		125	°C
V_{ESD}	Electrostatic discharge voltage, human body model	–1000	1000	V

Stresses listed under Absolute Maximum Ratings may be applied to devices one at a time without causing permanent damage. Functionality at or above the values listed is not implied. Exposure to these values for extended periods may affect device reliability.

ELECTROSTATIC DISCHARGE

This device can be damaged by ESD. Vitesse recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures may adversely affect reliability of the device.

5.5 Operating Conditions

Table 307. Recommended Operating Conditions

Symbol	Parameter	Minimum	Typical	Maximum	Unit
V_{DD}	Power supply voltage	1.70	1.80	1.90	V
V_{DD_IO25}	Power supply voltage	2.37	2.50	2.63	V
V_{DD_OUT33}	Power supply voltage	3.13	3.30	3.47	V
V_{DD_PLLx}	Power supply voltage	2.37	2.50	2.63	V
T	Operating temperature[1]	0		100	°C

1. Lower limit of specification is ambient temperature, and upper limit is case temperature.

The following table lists the recommended clock frequencies. For more information on clock frequencies, see "PLL Timing," page 231.

Table 308. Recommended Clock Frequencies

	Parameter	Minimum	Typical	Maximum	Unit
PLL0_Clk	System reference clock input frequency		25/125		MHz
PLL1_Clk	SPI-4.2 PLL input frequency	25 [1]		125 [1]	MHz

1. 25 to 37.5 MHz or 66.67 to 125 MHz, depending upon the setting of PLL_CLK_SPEED::SPI4_XTAL_SEL.

5.6 Power Sequencing

The following power sequencing parameters must be observed whenever the Schaumburg device is in operation including power up and power down.

$V_{DD\_OUT33} \leq V_{DD\_IO25} + 1.35\ V$

$V_{DD\_PLLx} \leq V_{DD\_IO25} + 0.5\ V$

The core power supply (V_{DD}) does not have any power sequencing requirement.

6 PIN DESCRIPTIONS

Many of the signal names used in the Schaumburg device are prefixed by the name of the interface to which they correspond to make it easier to distinguish the various signals from each other and to help relate a signal to the interface. Additionally, an active low signal is denoted with a lower case "n" as a prefix on the signal name (not the interface name): for example, PI_nOE.

Differential signals use a P or N suffix to distinguish their positive or negative parts; for example, SPI_TDClkN and SPI_TDClkP.

Table 309 defines the pin types and their abbreviations as used in the Schaumburg device.

Table 309. Pin Types

Pin Type	Definition
I	Input only
O	Output only
I/O	Bidirectional
OD	Open-drain output
OZ	3-state output
A	Analog
Power	Power
GND	Ground

Pin Type	Definition
NC	No connection, do not connect
5V	5 V tolerant
3V	3.3 V tolerant
2V	2.5 V tolerant
TD	Built-in differential termination
TS	Built-in series termination
Diff	Differential signal

6.1 Signals by Function

This section provides descriptions of the signals used on the Schaumburg device, grouped according to their function. For information about the signals grouped according to the pin (ball) number, see Table 324, "Signal List by Ball Number," page 269. For information about the signals grouped according to signal names. see Table 325, "Signal List by Signal Name," page 278.

6.1.1 RGMII Ports

The Schaumburg device has 12 tri-speed RGMII PHY interfaces.

All the RGMII outputs have a built-in series termination that has been optimized for 50 Ω transmission line use.

Table 310. Tri-Speed Ports

Signal Name	Type	Description
RGMII[11:0]_Rx_Clk	I	Receive clock. Synchronizes the receive data and control.
RGMII[11:0]_RD0 RGMII[11:0]_RD1 RGMII[11:0]_RD2 RGMII[11:0]_RD3	I	Receive data. 10/100 Mbit: Contains bit [3:0] on the rising edge of the Rx_Clk. Data is sampled on the rising edge only. 1000 Mbit: Contains bit [3:0] on the rising edge of the Rx_Clk and bit [7:4] on the falling edge.
RGMII[11:0]_Rx_Ctrl	I	Receive control. Data valid and receive error input. On the rising edge of the Rx_Clk, this input serves as data valid, signaling that valid data from the PHY is available on RD[3:0]. On the falling edge of Rx_Clk, it contains a logical derivative of data valid and receive error from the PHY. For more information, see the RGMII standard.
RGMII[11:0]_Tx_Clk	O, TS	Transmit clock. This clock is continuously driven from the MAC, and transmit data and control are synchronized to it.
RGMII[11:0]_TD0 RGMII[11:0]_TD1 RGMII[11:0]_TD2 RGMII[11:0]_TD3	O, TS	Transmit data. 10/100 Mbit: Contains bit [3:0] on the rising edge of the Tx_Clk. No data change occurs on the falling edge of Tx_Clk. 1000 Mbit: Contains bit [3:0] on the rising edge of the Tx_Clk and bit [7:4] on the falling edge.
RGMII[11:0]_Tx_Ctrl	O, TS	Transmit control. On the rising edge of Tx_Clk, this signal serves as transmit enable, indicating valid data on TD[3:0]. On the falling edge, this signal contains a logical derivative based on transmit enable and error from the MAC (for more information, see the RGMII standard).

6.1.2 MII Management Interface

Two independent MII Management interfaces are incorporated into the Schaumburg device.

Table 311. MII Management Interface

Signal Name	Type	Description
MDC_0	O	Management data clock, interface 0. MDC is sourced by the Station Management entity (Schaumburg) to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal.
MDIO_0	I/O, 3V	Management data input/output, interface 0. MDIO is a bi-directional signal between the PHY and Schaumburg, used to transfer control and status information. Control information is driven by Schaumburg synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by Schaumburg. MDIO is tri-stated between cycles.
MDC_1	O	Management data clock, interface 1. For a description, see MDC_0.
MDIO_1	I/O, 3V	Management data input, interface 1. For a description, see MDIO_0.

6.1.3 SPI-4.2 Interface

You can swap the bit order of the 16 bit SPI-4.2 transmit and receive buses on the Schaumburg device. When the bit order is swapped on a bus, the pin that is normally bit 0 on that bus becomes bit 15. Separate configuration bits are available for the transmit and receive buses. This makes the task of laying out PCBs much easier. For more information, see "SPI4_MISC (Address 0x00), Block 5, Subblock 0," page 185.

Table 312. SPI-4.2

Signal Name	Type	Description
SPI_En	I, 3V	SPI-4.2 interface enable or disable. 0 = SPI-4.2 is disabled and in power down. 1 = SPI-4.2 interface can be enabled or disabled using the SPI4_MISC register. This signal can be read from the SPI4_STATUS register.

Table 313. SPI-4.2 Receive Channel (Ingress)

Signal Name	Type	Description
SPI_RCtrlN SPI_RCtrlP	O, Diff	Receive control. SPI_RCtrlN or SPI_CtrlP is high only when a control word is present on RD[15:0].
SPI_RD0N SPI_RD0P SPI_RD1N SPI_RD1P SPI_RD2N SPI_RD2P SPI_RD3N SPI_RD3P SPI_RD4N SPI_RD4P SPI_RD5N SPI_RD5P SPI_RD6N SPI_RD6P SPI_RD7N SPI_RD7P SPI_RD8N SPI_RD8P SPI_RD9N SPI_RD9P SPI_RD10N SPI_RD10P SPI_RD11N SPI_RD11P SPI_RD12N SPI_RD12P SPI_RD13N SPI_RD13P SPI_RD14N SPI_RD14P SPI_RD15N SPI_RD15P	O, Diff	Receive data. Carries ingress payload data and in-band control from the Schaumburg device to the host. The order of the bits can be reversed. This makes the task of laying out PCBs much easier.
SPI_RDClkN SPI_RDClkP	O, Diff	Receive data clock. Clock associated with RD and RCtrl. Data and control lines are shifted on both the rising and falling edges of this clock.
SPI_RSClk	I, 3V	Receive status clock. Clock associated with RStat. Must be pulled to GND when SPI-4.2 interface is disabled.
SPI_RStat0 SPI_RStat1	I, 3V	Receive FIFO status. Carries round-robin FIFO status information, along with associated error detection and framing. Must be pulled to GND when SPI-4.2 interface is disabled.

Table 314. SPI-4.2 Transmit Channel (Egress)

Signal Name	Type	Description
SPI_TCtrlN SPI_TCtrlP	I, Diff, TD	Transmit control. SPI_TCtrlN or SPI_TCtrlP is high only when a control word is present on TD[15:0]. Leave floating if SPI-4.2 interface is disabled.
SPI_TD0N SPI_TD0P SPI_TD1N SPI_TD1P SPI_TD2N SPI_TD2P SPI_TD3N SPI_TD3P SPI_TD4N SPI_TD4P SPI_TD5N SPI_TD5P SPI_TD6N SPI_TD6P SPI_TD7N SPI_TD7P SPI_TD8N SPI_TD8P SPI_TD9N SPI_TD9P SPI_TD10N SPI_TD10P SPI_TD11N SPI_TD11P SPI_TD12N SPI_TD12P SPI_TD13N SPI_TD13P SPI_TD14N SPI_TD14P SPI_TD15N SPI_TD15P	I, Diff, TD	Transmit data. Carries egress payload data and in-band control words from the host to the Schaumburg device. The order of the bits can be reversed. This makes the task of laying out PCBs much easier. Leave floating if SPI-4.2 interface is disabled.
SPI_TDClkN SPI_TDClkP	I, Diff, TD	Transmit data clock. Clock associated with TD and SPI_TCtrlX. Data and control lines are sampled on both the rising and falling edges of this clock. Leave floating if SPI-4.2 interface is disabled.
SPI_TSClk	O	Transmit status clock. Clock associated with SPI_TStat*x*.
SPI_TStat0 SPI_TStat1	O	Transmit FIFO status. Carries round-robin FIFO status information, along with associated error detection and framing.

6.1.4 Parallel CPU Interface (PI)

Table 315. Parallel CPU Interface (PI)

Signal Name	Type	Description
PI_Addr0 PI_Addr1 PI_Addr2 PI_Addr3 PI_Addr4 PI_Addr5 PI_Addr6 PI_Addr7 PI_Addr8 PI_Addr9 PI_Addr10 PI_Addr11 PI_Addr12 PI_Addr13 PI_Addr14 PI_Addr15	I, 3V	LSB. Parallel CPU interface address bus. Selects the Block, Subblock, and Register Address. Note that the addressing is in 16-bit units, and *not* 8-bit units as on most CPUs. For information about the address mapping, see "CPU Interfaces," page 90. MSB.
PI_Data0 PI_Data1 PI_Data2 PI_Data3 PI_Data4 PI_Data5 PI_Data6 PI_Data7 PI_Data8 PI_Data9 PI_Data10 PI_Data11 PI_Data12 PI_Data13 PI_Data14 PI_Data15	I/O, 3V	LSB. Parallel CPU interface data bus. Driven by CPU at write, by the Schaumburg device at read. MSB.
PI_nDRdy	OD, 3V	Data ready. An active low signal that indicates when a posted read access to an internal register is performed. This occurs at a maximum of one microsecond after the posted read operation is initiated. When low, the data is ready for reading from LOCAL_DATA::DATA.
PI_nCS	I, 3V	Chip select. An active low signal that informs the parallel interface that it has been selected for a read or write operation. When the PI_nCS pin is driven high, no read or write operations can occur.

Table 315. Parallel CPU Interface (PI) *(continued)*

Signal Name	Type	Description
PI_nDone	OZ, 3V	Cycle done. A programmable active low or active high signal that informs the CPU that the current access can be terminated. Alternatively, an appropriate number of wait states must be inserted from the CPU when accessing Schaumburg.
PI_nOE	I, 3V	Output enable. An active low input signal that informs the parallel interface that it should drive data out on the PI_Data pins. This signal makes it possible to use a multiplexed address and data bus. PI_nOE can permanently pulled low if separate address and data buses exist. Even if PI_nOE is low, the PI_Data pins are only driven when PI_nCS is low and a read operation is in progress.
PI_nRd	I, 3V	Read. In three-wire mode, this is an active low signal that informs the parallel interface that it has been selected for a read operation. In two-wire mode, this should be pulled low.
PI_nWR	I, 3V	Write (three-wire mode only). An active low signal that informs the parallel interface that a write operation is to take place if the PI_nCS pin is driven to low. Read/Write (two-wire mode only). 0 = Write operation. 1 = Read operation.

6.1.5 Serial CPU Interface (SI)

Table 316. Serial CPU Interface (SI)

Signal Name	Type	Description
SI_DO	OZ, 3V	Serial output to the master
SI_DI	I, 3V	Serial input from the master
SI_Clk	I, 3V	SI clock from the master
SI_nEn	I, 3V	0 = Enable SI 1 = Disable SI

6.1.6 PLL

Table 317. System PLL

Signal Name	Type	Description
PLL0_Cap0 PLL0_Cap1	A	PLL loop filter capacitor. Connect a 0.1 µF capacitor between the pins.
PLL0_ClkN PLL0_ClkP	I, 3V, Diff	System reference clock, either 25 MHz or 125 MHz. LVDS, LVPECL, or single-ended LVTTL inputs. In LVDS or LVPECL mode, connect both pins to achieve a differential input. In LVTTL mode, if PLL0_ClkN is left open, PLL0_ClkP can be used as a LVTTL input. If an inverted clock is required, then use PLL0_ClkN and float PLL0_ClkP.
PLL0_Clk125_En	I, 3V	0 = 25 MHz system reference clock. 1 = 125 MHz system reference clock.
PLL0_En	I, 3V	Pull high.
VDD_PLL0	Power	PLL power supply = 2.5 V.
VSS_PLL0	GND	PLL ground.

Table 318. SPI-4.2 PLL

Signal Name	Type	Description
PLL1_Cap0 PLL1_Cap1	A	PLL loop filter capacitor. Connect a 0.1 µF capacitor between the pins.
PLL1_ClkN PLL1_ClkP	I, 3V, Diff	SPI-4.2 ingress clock, 25 – 37.5 MHz (reduced jitter performance), 66.67 –125 MHz. LVDS, LVPECL, or single-ended LVTTL inputs. In LVDS or LVPECL mode, connect both pins to achieve a differential input. In LVTTL mode, if PLL1_ClkN is left open, PLL1_ClkP can be used as a LVTTL input. If an inverted clock is required, then use PLL1_ClkN and float PLL1_ClkP.
PLL1_En	I, 3V	Pull high.
VDD_PLL1	Power	PLL power supply = 2.5 V.
VSS_PLL1	GND	PLL ground.

6.1.7 General-Purpose I/O (GPIO)

Table 319. General-Purpose I/O

Signal Name	Type	Description
GPIO0 GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO7	I/O	General-purpose I/O bits that can be individually defined as either input or outputs. As outputs, they are controllable by CPU registers. As inputs, they are readable from CPU registers. They are configured as inputs at power-up and can be controlled by GPIO_CTRL::PD7_0. GPIO[1:0] can be configured to output system PLL lock status (GPIO0) and SPI-4.2 PLL lock status (GPIO1).

6.1.8 JTAG Interface

Table 320. JTAG Interface

Signal Name	Type	Description
JTAG_nTRST	I, 5V	JTAG test reset, active low. For normal operation, this signal should be pulled low.
JTAG_TCK	I, 5V	JTAG clock.
JTAG_TDI	I, 5V	JTAG test data in.
JTAG_TDO	OZ	JTAG test data out.
JTAG_TMS	I, 5V	JTAG test mode select.

6.1.9 Miscellaneous

Table 321. Miscellaneous Signals

Signal Name	Type	Description
nReset	I, 2V	Global chip reset, active low.
Test_Mode	I, 2V	Internal test. Pull low.
Test_Enable	I, 2V	Internal test. Pull low.

6.1.10 Power Supplies

Table 322. Power Supplies

Signal Name	Type	Description
VDD	Power	Core power supply voltage = 1.8 V.
VDD_IO25	Power	IO power supply voltage (SPI-4.2 data, RGMII, MIIM) = 2.5 V.
VDD_OUT33	Power	Output power supply voltage (SPI-4.2 status, JTAG, GPIO, PI, SI) = 3.3 V.
VSS	GND	Ground, except for PLLs.

6.1.11 Reserved Balls

Table 323. Reserved Balls

Signal Name	Type	Description
Reserved_51	NC	Reserved. Leave floating.
Reserved_52	NC	Reserved. Leave floating.
Reserved_53	I, 2V	Reserved. Pull low.
Reserved_54	I, 2V	Reserved. Pull low.
Reserved_50	I, 3V	Reserved. Pull low.
Reserved_24, Reserved_35, Reserved_36, Reserved_37, Reserved_38, Reserved_39, Reserved_40, Reserved_47	I, 3V	Reserved. Pull high or low.
Reserved_53, Reserved_54, Reserved_56, Reserved_68, Reserved_80, Reserved_92, Reserved_104, Reserved_116, Reserved_128, Reserved_140, Reserved_152, Reserved_164, Reserved_176, Reserved_188	I, 2V	Reserved. Pull low.

Table 323. Reserved Balls *(continued)*

Signal Name	Type	Description
Reserved_55, Reserved_57, Reserved_58, Reserved_59, Reserved_60, Reserved_67, Reserved_69, Reserved_70, Reserved_71, Reserved_72, Reserved_79, Reserved_81, Reserved_82, Reserved_83, Reserved_84, Reserved_91, Reserved_93, Reserved_94, Reserved_95, Reserved_96, Reserved_103, Reserved_105, Reserved_106, Reserved_107, Reserved_108, Reserved_115, Reserved_117, Reserved_118, Reserved_119, Reserved_120, Reserved_127, Reserved_129, Reserved_130, Reserved_131, Reserved_132, Reserved_139, Reserved_141, Reserved_142, Reserved_143, Reserved_144, Reserved_151, Reserved_153, Reserved_154, Reserved_155, Reserved_156, Reserved_163, Reserved_165, Reserved_166, Reserved_167, Reserved_168, Reserved_175, Reserved_177, Reserved_178, Reserved_179, Reserved_180, Reserved_187, Reserved_189, Reserved_190, Reserved_191, Reserved_192	I, 2V	Reserved. Pull high or low.

Table 323. Reserved Balls *(continued)*

Signal Name	Type	Description
Reserved_0, Reserved_1, Reserved_2, Reserved_3, Reserved_4, Reserved_5, Reserved_6, Reserved_7, Reserved_8, Reserved_9, Reserved_10, Reserved_11, Reserved_12, Reserved_13, Reserved_14, Reserved_15, Reserved_16, Reserved_17, Reserved_18, Reserved_19, Reserved_20, Reserved_21, Reserved_22, Reserved_23, Reserved_25, Reserved_26, Reserved_27, Reserved_28, Reserved_29, Reserved_30, Reserved_31, Reserved_32, Reserved_33, Reserved_34, Reserved_41, Reserved_42, Reserved_43, Reserved_44, Reserved_45, Reserved_46, Reserved_48, Reserved_49, Reserved_51, Reserved_52, Reserved_61, Reserved_62, Reserved_63, Reserved_64, Reserved_65, Reserved_66, Reserved_73, Reserved_74, Reserved_75, Reserved_76, Reserved_77, Reserved_78, Reserved_85, Reserved_86, Reserved_87, Reserved_88, Reserved_89, Reserved_90, Reserved_97, Reserved_98, Reserved_99, Reserved_100, Reserved_101, Reserved_102, Reserved_109, Reserved_110, Reserved_111, Reserved_112, Reserved_113, Reserved_114, Reserved_121, Reserved_122, Reserved_123, Reserved_124, Reserved_125, Reserved_126, Reserved_133, Reserved_134, Reserved_135, Reserved_136, Reserved_137, Reserved_138, Reserved_145, Reserved_146, Reserved_147, Reserved_148, Reserved_149, Reserved_150, Reserved_157, Reserved_158, Reserved_159, Reserved_160, Reserved_161, Reserved_162, Reserved_169, Reserved_170, Reserved_171, Reserved_172, Reserved_173, Reserved_174, Reserved_181, Reserved_182, Reserved_183, Reserved_184, Reserved_185, Reserved_186, Reserved_193, Reserved_194, Reserved_195, Reserved_196, Reserved_197, Reserved_198	NC	Reserved. Leave floating.

6.2 Signal List Summaries

The following sections provide summaries of the Schaumburg device signals.

6.2.1 Signals by Ball Number

Table 324 lists all the signals in ball number order.

Table 324. Signal List by Ball Number

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
A2	VSS_0	B3	VDD_IO25_0	C3	Test_Mode	D3	VSS_16
A3	Reserved_57	B4	Reserved_60	C4	Reserved_55	D4	Test_Enable
A4	Reserved_63	B5	Reserved_65	C5	Reserved_61	D5	VDD_IO25_1
A5	VSS_1	B6	Reserved_69	C6	Reserved_67	D6	Reserved_62
A6	Reserved_72	B7	Reserved_74	C7	Reserved_71	D7	VSS_17
A7	Reserved_79	B8	Reserved_81	C8	Reserved_77	D8	Reserved_73
A8	Reserved_84	B9	VDD_IO25_4	C9	Reserved_82	D9	VDD_IO25_5
A9	VSS_2	B10	Reserved_88	C10	Reserved_87	D10	Reserved_85
A10	Reserved_90	B11	Reserved_92	C11	Reserved_91	D11	VSS_18
A11	Reserved_95	B12	Reserved_98	C12	Reserved_96	D12	Reserved_94
A12	Reserved_100	B13	VSS_8	C13	Reserved_101	D13	VDD_IO25_11
A13	VDD_IO25_10	B14	Reserved_105	C14	Reserved_104	D14	Reserved_103
A14	Reserved_108	B15	Reserved_111	C15	Reserved_110	D15	VSS_19
A15	Reserved_112	B16	Reserved_117	C16	Reserved_116	D16	Reserved_115
A16	Reserved_118	B17	VDD_IO25_16	C17	Reserved_119	D17	VDD_IO25_17
A17	VSS_3	B18	Reserved_123	C18	Reserved_124	D18	Reserved_125
A18	Reserved_121	B19	Reserved_129	C19	Reserved_130	D19	VSS_20
A19	Reserved_126	B20	Reserved_133	C20	Reserved_134	D20	Reserved_136
A20	Reserved_131	B21	VSS_9	C21	Reserved_138	D21	VDD_IO25_23
A21	VDD_IO25_22	B22	Reserved_141	C22	Reserved_145	D22	Reserved_147
A22	Reserved_139	B23	Reserved_146	C23	Reserved_149	D23	VSS_21
A23	Reserved_142	B24	Reserved_151	C24	Reserved_152	D24	Reserved_156
A24	Reserved_148	B25	VDD_IO25_28	C25	Reserved_157	D25	VDD_IO25_29
A25	VSS_4	B26	Reserved_158	C26	Reserved_161	D26	Reserved_167
A26	Reserved_155	B27	Reserved_162	C27	Reserved_169	D27	VSS_22
A27	Reserved_160	B28	Reserved_170	C28	Reserved_172	D28	Reserved_178
A28	Reserved_165	B29	VSS_10	C29	Reserved_179	D29	VDD_IO25_34
A29	Reserved_171	B30	Reserved_180	C30	VDD_IO25_36	D30	Reserved_188
A30	Reserved_173	B31	VDD_IO25_38	C31	Reserved_183	D31	VSS_23
A31	Reserved_181	B32	Reserved_189	C32	VSS_14	D32	Reserved_192
A32	Reserved_182	B33	VSS_11	C33	Reserved_191	D33	VDD_IO25_41
A33	VSS_5	B34	VSS_12	C34	Reserved_197	D34	Reserved_4
B1	VSS_6	C1	VDD_IO25_101	D1	SI_DO	E1	VSS_24
B2	VSS_7	C2	VSS_13	D2	VSS_15	E2	JTAG_TDO

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
E3	VSS_25	F3	JTAG_nTRST	G3	SI_DI	H3	VDD_OUT33_0
E4	VSS_26	F4	JTAG_TDI	G4	VDD_OUT33_1	H4	VSS_50
E5	Reserved_54	F5	VSS_34	G5	Reserved_51	H5	JTAG_TMS
E6	Reserved_59	F6	nReset	G6	VSS_42	H6	VSS_51
E7	Reserved_64	F7	VDD_IO25_2	G7	Reserved_56	H7	Reserved_53
E8	VDD_IO25_3	F8	Reserved_66	G8	VSS_43	H8	Reserved_58
E9	Reserved_75	F9	VSS_35	G9	Reserved_68	H9	VDD_IO25_6
E10	VSS_27	F10	Reserved_78	G10	VDD_IO25_7	H10	Reserved_70
E11	Reserved_86	F11	VDD_IO25_8	G11	Reserved_80	H11	Reserved_76
E12	VDD_IO25_9	F12	Reserved_89	G12	VSS_44	H12	Reserved_83
E13	Reserved_97	F13	VSS_36	G13	Reserved_93	H13	VDD_IO25_12
E14	VSS_28	F14	Reserved_102	G14	VDD_IO25_13	H14	Reserved_99
E15	Reserved_109	F15	VDD_IO25_14	G15	Reserved_107	H15	Reserved_106
E16	VDD_IO25_15	F16	Reserved_114	G16	VSS_45	H16	Reserved_113
E17	Reserved_120	F17	VSS_37	G17	Reserved_122	H17	VDD_IO25_18
E18	VSS_29	F18	Reserved_127	G18	VDD_IO25_19	H18	Reserved_128
E19	Reserved_132	F19	VDD_IO25_20	G19	Reserved_137	H19	Reserved_135
E20	VDD_IO25_21	F20	Reserved_140	G20	VSS_46	H20	Reserved_143
E21	Reserved_144	F21	VSS_38	G21	Reserved_150	H21	VDD_IO25_24
E22	VSS_30	F22	Reserved_153	G22	VDD_IO25_25	H22	Reserved_159
E23	Reserved_154	F23	VDD_IO25_26	G23	Reserved_163	H23	Reserved_168
E24	VDD_IO25_27	F24	Reserved_164	G24	VSS_47	H24	Reserved_174
E25	Reserved_166	F25	VSS_39	G25	Reserved_175	H25	VDD_IO25_30
E26	VSS_31	F26	Reserved_176	G26	VDD_IO25_31	H26	Reserved_186
E27	Reserved_177	F27	VDD_IO25_32	G27	Reserved_185	H27	Reserved_190
E28	VDD_IO25_33	F28	Reserved_187	G28	VSS_48	H28	Reserved_196
E29	Reserved_184	F29	VSS_40	G29	Reserved_195	H29	VDD_IO25_35
E30	VSS_32	F30	Reserved_194	G30	VDD_IO25_37	H30	Reserved_0
E31	Reserved_193	F31	VDD_IO25_39	G31	VSS_49	H31	Reserved_2
E32	VDD_IO25_40	F32	Reserved_198	G32	Reserved_6	H32	VDD_OUT33_3
E33	VDD_OUT33_2	F33	Reserved_9	G33	Reserved_11	H33	Reserved_15
E34	VSS_33	F34	Reserved_13	G34	Reserved_17	H34	Reserved_20
F1	SI_nEn	G1	SPI_TStat1	H1	SPI_TSClk	J1	SPI_TCtrlP
F2	SI_Clk	G2	VSS_41	H2	SPI_TStat0	J2	SPI_TCtrlN

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
J3	SPI_TDClkP	K3	VDD_IO25_100	L3	SPI_TD12P	M3	VSS_88
J4	SPI_TDClkN	K4	VSS_64	L4	SPI_TD12N	M4	VDD_IO25_99
J5	JTAG_TCK	K5	SPI_TD15P	L5	SPI_TD13P	M5	SPI_TD10P
J6	Reserved_52	K6	SPI_TD15N	L6	SPI_TD13N	M6	SPI_TD10N
J7	VDD_0	K7	VDD_12	L7	VSS_76	M7	VSS_89
J8	VDD_1	K8	VDD_13	L8	VSS_77	M8	VSS_90
J9	VSS_52	K9	VSS_65	L9	VDD_24	M9	VDD_34
J10	VSS_53	K10	VSS_66	L10	VDD_25	M10	VDD_35
J11	VDD_2	K11	VDD_14	L11	VSS_78	M11	VSS_91
J12	VDD_3	K12	VDD_15	L12	VSS_79	M12	VSS_92
J13	VSS_54	K13	VSS_67	L13	VDD_26	M13	VDD_36
J14	VSS_55	K14	VSS_68	L14	VDD_27	M14	VDD_37
J15	VDD_4	K15	VDD_16	L15	VSS_80	M15	VSS_93
J16	VDD_5	K16	VDD_17	L16	VSS_81	M16	VSS_94
J17	VSS_56	K17	VSS_69	L17	VDD_28	M17	VDD_38
J18	VSS_57	K18	VSS_70	L18	VDD_29	M18	VDD_39
J19	VDD_6	K19	VDD_18	L19	VSS_82	M19	VSS_95
J20	VDD_7	K20	VDD_19	L20	VSS_83	M20	VSS_96
J21	VSS_58	K21	VSS_71	L21	VDD_30	M21	VDD_40
J22	VSS_59	K22	VSS_72	L22	VDD_31	M22	VDD_41
J23	VDD_8	K23	VDD_20	L23	VSS_84	M23	VSS_97
J24	VDD_9	K24	VDD_21	L24	VSS_85	M24	VSS_98
J25	VSS_60	K25	VSS_73	L25	VDD_32	M25	VDD_42
J26	VSS_61	K26	VSS_74	L26	VDD_33	M26	VDD_43
J27	VDD_10	K27	VDD_22	L27	VSS_86	M27	VSS_99
J28	VDD_11	K28	VDD_23	L28	VSS_87	M28	VSS_100
J29	VSS_62	K29	VDD_OUT33_5	L29	Reserved_3	M29	Reserved_8
J30	Reserved_1	K30	Reserved_5	L30	Reserved_10	M30	VSS_101
J31	Reserved_7	K31	VSS_75	L31	Reserved_16	M31	Reserved_21
J32	Reserved_14	K32	Reserved_18	L32	VDD_OUT33_6	M32	Reserved_24
J33	VSS_63	K33	Reserved_22	L33	Reserved_27	M33	VSS_102
J34	VDD_OUT33_4	K34	Reserved_23	L34	Reserved_29	M34	Reserved_30
K1	SPI_TD14P	L1	SPI_TD11P	M1	SPI_TD9P	N1	SPI_TD6P
K2	SPI_TD14N	L2	SPI_TD11N	M2	SPI_TD9N	N2	SPI_TD6N

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
N3	SPI_TD7P	P3	VDD_IO25_98	R3	SPI_TD2P	T3	SPI_RStat1
N4	SPI_TD7N	P4	VSS_113	R4	SPI_TD2N	T4	SPI_En
N5	SPI_TD8P	P5	SPI_TD5P	R5	SPI_TD3P	T5	SPI_TD0P
N6	SPI_TD8N	P6	SPI_TD5N	R6	SPI_TD3N	T6	SPI_TD0N
N7	VDD_44	P7	VDD_56	R7	VSS_125	T7	VSS_137
N8	VDD_45	P8	VDD_57	R8	VSS_126	T8	VSS_138
N9	VSS_103	P9	VSS_114	R9	VDD_68	T9	VDD_78
N10	VSS_104	P10	VSS_115	R10	VDD_69	T10	VDD_79
N11	VDD_46	P11	VDD_58	R11	VSS_127	T11	VSS_139
N12	VDD_47	P12	VDD_59	R12	VSS_128	T12	VSS_140
N13	VSS_105	P13	VSS_116	R13	VDD_70	T13	VDD_80
N14	VSS_106	P14	VSS_117	R14	VDD_71	T14	VDD_81
N15	VDD_48	P15	VDD_60	R15	VSS_129	T15	VSS_141
N16	VDD_49	P16	VDD_61	R16	VSS_130	T16	VSS_142
N17	VSS_107	P17	VSS_118	R17	VDD_72	T17	VDD_82
N18	VSS_108	P18	VSS_119	R18	VDD_73	T18	VDD_83
N19	VDD_50	P19	VDD_62	R19	VSS_131	T19	VSS_143
N20	VDD_51	P20	VDD_63	R20	VSS_132	T20	VSS_144
N21	VSS_109	P21	VSS_120	R21	VDD_74	T21	VDD_84
N22	VSS_110	P22	VSS_121	R22	VDD_75	T22	VDD_85
N23	VDD_52	P23	VDD_64	R23	VSS_133	T23	VSS_145
N24	VDD_53	P24	VDD_65	R24	VSS_134	T24	VSS_146
N25	VSS_111	P25	VSS_122	R25	VDD_76	T25	VDD_86
N26	VSS_112	P26	VSS_123	R26	VDD_77	T26	VDD_87
N27	VDD_54	P27	VDD_66	R27	VSS_135	T27	VSS_147
N28	VDD_55	P28	VDD_67	R28	VSS_136	T28	VSS_148
N29	Reserved_12	P29	VDD_IO25_42	R29	VDD_IO25_43	T29	VDD_IO25_44
N30	Reserved_19	P30	Reserved_25	R30	VDD_OUT33_9	T30	Reserved_38
N31	VDD_OUT33_7	P31	Reserved_26	R31	Reserved_34	T31	Reserved_39
N32	Reserved_28	P32	Reserved_32	R32	Reserved_35	T32	VSS_149
N33	Reserved_31	P33	VDD_OUT33_8	R33	Reserved_36	T33	Reserved_40
N34	Reserved_33	P34	VSS_124	R34	Reserved_37	T34	Reserved_41
P1	SPI_TD4P	R1	SPI_TD1P	T1	SPI_RStat0	U1	SPI_RCtrlN
P2	SPI_TD4N	R2	SPI_TD1N	T2	SPI_RSClk	U2	SPI_RCtrlP

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
U3	VSS_150	V3	SPI_RD15N	W3	VDD_IO25_94	Y3	SPI_RD11N
U4	VDD_IO25_97	V4	SPI_RD15P	W4	VSS_174	Y4	SPI_RD11P
U5	SPI_RDClkN	V5	VSS_163	W5	SPI_RD13N	Y5	VSS_188
U6	SPI_RDClkP	V6	VDD_IO25_95	W6	SPI_RD13P	Y6	VDD_IO25_93
U7	VDD_88	V7	VDD_100	W7	VSS_175	Y7	VSS_189
U8	VDD_89	V8	VDD_101	W8	VSS_176	Y8	VSS_190
U9	VSS_151	V9	VSS_164	W9	VDD_112	Y9	VDD_122
U10	VSS_152	V10	VSS_165	W10	VDD_113	Y10	VDD_123
U11	VDD_90	V11	VDD_102	W11	VSS_177	Y11	VSS_191
U12	VDD_91	V12	VDD_103	W12	VSS_178	Y12	VSS_192
U13	VSS_153	V13	VSS_166	W13	VDD_114	Y13	VDD_124
U14	VSS_154	V14	VSS_167	W14	VDD_115	Y14	VDD_125
U15	VDD_92	V15	VDD_104	W15	VSS_179	Y15	VSS_193
U16	VDD_93	V16	VDD_105	W16	VSS_180	Y16	VSS_194
U17	VSS_155	V17	VSS_168	W17	VDD_116	Y17	VDD_126
U18	VSS_156	V18	VSS_169	W18	VDD_117	Y18	VDD_127
U19	VDD_94	V19	VDD_106	W19	VSS_181	Y19	VSS_195
U20	VDD_95	V20	VDD_107	W20	VSS_182	Y20	VSS_196
U21	VSS_157	V21	VSS_170	W21	VDD_118	Y21	VDD_128
U22	VSS_158	V22	VSS_171	W22	VDD_119	Y22	VDD_129
U23	VDD_96	V23	VDD_108	W23	VSS_183	Y23	VSS_197
U24	VDD_97	V24	VDD_109	W24	VSS_184	Y24	VSS_198
U25	VSS_159	V25	VSS_172	W25	VDD_120	Y25	VDD_130
U26	VSS_160	V26	VSS_173	W26	VDD_121	Y26	VDD_131
U27	VDD_98	V27	VDD_110	W27	VSS_185	Y27	VSS_199
U28	VDD_99	V28	VDD_111	W28	VSS_186	Y28	VSS_200
U29	Reserved_43	V29	PI_nRd	W29	VDD_IO25_45	Y29	VDD_IO25_46
U30	VSS_161	V30	VDD_OUT33_11	W30	PI_Addr11	Y30	PI_Addr4
U31	Reserved_44	V31	PI_nWR	W31	PI_Addr14	Y31	PI_Addr8
U32	Reserved_45	V32	PI_nDone	W32	VDD_OUT33_12	Y32	PI_Addr12
U33	VDD_OUT33_10	V33	PI_nDRdy	W33	PI_nOE	Y33	PI_Addr15
U34	Reserved_42	V34	Reserved_46	W34	VSS_187	Y34	PI_nCS
V1	VDD_IO25_96	W1	SPI_RD14N	Y1	SPI_RD12N	AA1	SPI_RD10N
V2	VSS_162	W2	SPI_RD14P	Y2	SPI_RD12P	AA2	SPI_RD10P

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
AA3	VSS_201	AB3	SPI_RD7N	AC3	VDD_IO25_90	AD3	SPI_RD4N
AA4	VDD_IO25_92	AB4	SPI_RD7P	AC4	VSS_225	AD4	SPI_RD4P
AA5	SPI_RD9N	AB5	VSS_213	AC5	SPI_RD5N	AD5	VSS_239
AA6	SPI_RD9P	AB6	VDD_IO25_91	AC6	SPI_RD5P	AD6	VDD_IO25_88
AA7	VDD_132	AB7	VDD_144	AC7	VSS_226	AD7	VSS_240
AA8	VDD_133	AB8	VDD_145	AC8	VSS_227	AD8	VSS_241
AA9	VSS_202	AB9	VSS_214	AC9	VDD_156	AD9	VDD_166
AA10	VSS_203	AB10	VSS_215	AC10	VDD_157	AD10	VDD_167
AA11	VDD_134	AB11	VDD_146	AC11	VSS_228	AD11	VSS_242
AA12	VDD_135	AB12	VDD_147	AC12	VSS_229	AD12	VSS_243
AA13	VSS_204	AB13	VSS_216	AC13	VDD_158	AD13	VDD_168
AA14	VSS_205	AB14	VSS_217	AC14	VDD_159	AD14	VDD_169
AA15	VDD_136	AB15	VDD_148	AC15	VSS_230	AD15	VSS_244
AA16	VDD_137	AB16	VDD_149	AC16	VSS_231	AD16	VSS_245
AA17	VSS_206	AB17	VSS_218	AC17	VDD_160	AD17	VDD_170
AA18	VSS_207	AB18	VSS_219	AC18	VDD_161	AD18	VDD_171
AA19	VDD_138	AB19	VDD_150	AC19	VSS_232	AD19	VSS_246
AA20	VDD_139	AB20	VDD_151	AC20	VSS_233	AD20	VSS_247
AA21	VSS_208	AB21	VSS_220	AC21	VDD_162	AD21	VDD_172
AA22	VSS_209	AB22	VSS_221	AC22	VDD_163	AD22	VDD_173
AA23	VDD_140	AB23	VDD_152	AC23	VSS_234	AD23	VSS_248
AA24	VDD_141	AB24	VDD_153	AC24	VSS_235	AD24	VSS_249
AA25	VSS_210	AB25	VSS_222	AC25	VDD_164	AD25	VDD_174
AA26	VSS_211	AB26	VSS_223	AC26	VDD_165	AD26	VDD_175
AA27	VDD_142	AB27	VDD_154	AC27	VSS_236	AD27	VSS_250
AA28	VDD_143	AB28	VDD_155	AC28	VSS_237	AD28	VSS_251
AA29	VDD_IO25_47	AB29	PI_Data6	AC29	GPIO7	AD29	GPIO3
AA30	PI_Data14	AB30	VSS_224	AC30	PI_Data7	AD30	PI_Data2
AA31	PI_Addr5	AB31	PI_Data15	AC31	VDD_OUT33_14	AD31	VSS_252
AA32	VSS_212	AB32	PI_Addr3	AC32	PI_Addr0	AD32	Reserved_48
AA33	PI_Addr10	AB33	VDD_OUT33_13	AC33	PI_Addr6	AD33	VSS_253
AA34	PI_Addr13	AB34	PI_Addr9	AC34	PI_Addr7	AD34	PI_Addr2
AB1	SPI_RD8N	AC1	SPI_RD6N	AD1	VDD_IO25_89	AE1	SPI_RD3N
AB2	SPI_RD8P	AC2	SPI_RD6P	AD2	VSS_238	AE2	SPI_RD3P

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
AE3	SPI_RD2N	AF3	VSS_264	AG3	VDD_IO25_87	AH3	VSS_283
AE4	SPI_RD2P	AF4	VSS_265	AG4	PLL0_Cap0	AH4	PLL0_Cap1
AE5	PLL0_AGND	AF5	PLL1_Cap1	AG5	VSS_278	AH5	VSS_284
AE6	PLL0_AVDD	AF6	PLL1_Cap0	AG6	VSS_279	AH6	RGMII11_TD0
AE7	VDD_176	AF7	VDD_188	AG7	RGMII11_TD1	AH7	RGMII11_Tx_Ctrl
AE8	VDD_177	AF8	VDD_189	AG8	RGMII11_RD0	AH8	VDD_IO25_82
AE9	VSS_254	AF9	VSS_266	AG9	VDD_IO25_79	AH9	RGMII10_Tx_Ctrl
AE10	VSS_255	AF10	VSS_267	AG10	RGMII10_TxClk	AH10	VSS_285
AE11	VDD_178	AF11	VDD_190	AG11	RGMII9_TD0	AH11	RGMII9_Tx_Ctrl
AE12	VDD_179	AF12	VDD_191	AG12	RGMII9_RD0	AH12	VDD_IO25_76
AE13	VSS_256	AF13	VSS_268	AG13	VDD_IO25_73	AH13	RGMII8_Tx_Ctrl
AE14	VSS_257	AF14	VSS_269	AG14	RGMII8_Rx_Ctrl	AH14	VSS_286
AE15	VDD_180	AF15	VDD_192	AG15	RGMII7_TxClk	AH15	RGMII7_RD0
AE16	VDD_181	AF16	VDD_193	AG16	RGMII6_TD1	AH16	VDD_IO25_70
AE17	VSS_258	AF17	VSS_270	AG17	VDD_IO25_67	AH17	RGMII6_RD3
AE18	VSS_259	AF18	VSS_271	AG18	RGMII5_TD3	AH18	VSS_287
AE19	VDD_182	AF19	VDD_194	AG19	RGMII5_RxClk	AH19	RGMII4_TD0
AE20	VDD_183	AF20	VDD_195	AG20	RGMII4_RD0	AH20	VDD_IO25_64
AE21	VSS_260	AF21	VSS_272	AG21	VDD_IO25_61	AH21	RGMII3_TD2
AE22	VSS_261	AF22	VSS_273	AG22	RGMII3_RxClk	AH22	VSS_288
AE23	VDD_184	AF23	VDD_196	AG23	RGMII2_RD1	AH23	RGMII2_TD2
AE24	VDD_185	AF24	VDD_197	AG24	RGMII1_TD1	AH24	VDD_IO25_58
AE25	VSS_262	AF25	VSS_274	AG25	VDD_IO25_55	AH25	RGMII1_TD2
AE26	VSS_263	AF26	VSS_275	AG26	RGMII0_TD1	AH26	VSS_289
AE27	VDD_186	AF27	VDD_198	AG27	RGMII0_RD1	AH27	RGMII0_TD0
AE28	VDD_187	AF28	VDD_199	AG28	MDC0	AH28	VDD_IO25_52
AE29	VDD_OUT33_15	AF29	VSS_276	AG29	VDD_IO25_48	AH29	RGMII0_RxClk
AE30	GPIO5	AF30	GPIO0	AG30	Reserved_50	AH30	MDIO1
AE31	PI_Data5	AF31	PI_Data1	AG31	GPIO4	AH31	VDD_OUT33_17
AE32	PI_Data12	AF32	PI_Data9	AG32	VSS_280	AH32	PI_Data0
AE33	Reserved_47	AF33	VSS_277	AG33	PI_Data11	AH33	PI_Data8
AE34	PI_Addr1	AF34	VDD_OUT33_16	AG34	Reserved_49	AH34	PI_Data13
AF1	SPI_RD1N	AG1	SPI_RD0N	AH1	VSS_281	AJ1	PLL0_ClkP
AF2	SPI_RD1P	AG2	SPI_RD0P	AH2	VSS_282	AJ2	PLL0_ClkN

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
AJ3	PLL0_Clk125_En	AK3	PLL1_AVDD	AL3	VSS_308	AM3	RGMII11_TD2
AJ4	PLL0_En	AK4	VSS_299	AL4	VSS_309	AM4	RGMII11_TxClk
AJ5	VSS_290	AK5	VSS_300	AL5	RGMII11_RD1	AM5	VDD_IO25_85
AJ6	RGMII11_TD3	AK6	VDD_IO25_84	AL6	RGMII11_Rx_Ctrl	AM6	RGMII10_TD3
AJ7	VDD_IO25_83	AK7	RGMII10_TD0	AL7	VSS_310	AM7	RGMII10_RD1
AJ8	RGMII10_TD2	AK8	VSS_301	AL8	RGMII10_RD3	AM8	RGMII9_TD1
AJ9	VSS_291	AK9	RGMII10_RxClk	AL9	VDD_IO25_80	AM9	RGMII9_RD1
AJ10	RGMII9_TD2	AK10	VDD_IO25_78	AL10	RGMII9_RD3	AM10	RGMII9_RxClk
AJ11	VDD_IO25_77	AK11	RGMII9_Rx_Ctrl	AL11	VSS_311	AM11	RGMII8_TD3
AJ12	RGMII8_TD1	AK12	VSS_302	AL12	RGMII8_RD0	AM12	RGMII8_RD2
AJ13	VSS_292	AK13	RGMII8_RD3	AL13	VDD_IO25_74	AM13	RGMII7_TD2
AJ14	RGMII7_TD1	AK14	VDD_IO25_72	AL14	RGMII7_TD3	AM14	RGMII7_Tx_Ctrl
AJ15	VDD_IO25_71	AK15	RGMII7_RD3	AL15	VSS_312	AM15	RGMII7_Rx_Ctrl
AJ16	RGMII6_TD2	AK16	VSS_303	AL16	RGMII6_TD3	AM16	RGMII6_Tx_Ctrl
AJ17	VSS_293	AK17	RGMII6_RD2	AL17	VDD_IO25_68	AM17	RGMII6_RD1
AJ18	RGMII5_TD2	AK18	VDD_IO25_66	AL18	RGMII5_TD1	AM18	RGMII5_TD0
AJ19	VDD_IO25_65	AK19	RGMII5_RD2	AL19	VSS_313	AM19	RGMII5_RD0
AJ20	RGMII4_Tx_Ctrl	AK20	VSS_304	AL20	RGMII4_TD1	AM20	RGMII5_Rx_Ctrl
AJ21	VSS_294	AK21	RGMII4_RD1	AL21	VDD_IO25_62	AM21	RGMII4_TD2
AJ22	RGMII3_Tx_Ctrl	AK22	VDD_IO25_60	AL22	RGMII4_Rx_Ctrl	AM22	RGMII4_RD2
AJ23	VDD_IO25_59	AK23	RGMII3_RD0	AL23	VSS_314	AM23	RGMII3_TD1
AJ24	RGMII2_Tx_Ctrl	AK24	VSS_305	AL24	RGMII3_RD2	AM24	RGMII3_TxClk
AJ25	VSS_295	AK25	RGMII2_RD0	AL25	VDD_IO25_56	AM25	RGMII3_RD3
AJ26	RGMII1_Tx_Ctrl	AK26	VDD_IO25_54	AL26	RGMII2_RD2	AM26	RGMII2_TD1
AJ27	VDD_IO25_53	AK27	RGMII1_TxClk	AL27	VSS_315	AM27	RGMII2_RD3
AJ28	RGMII0_TD2	AK28	VSS_306	AL28	RGMII1_RD0	AM28	RGMII1_TD0
AJ29	VSS_296	AK29	RGMII0_TD3	AL29	VDD_IO25_51	AM29	RGMII1_RD1
AJ30	MDIO0	AK30	VSS_307	AL30	RGMII0_Tx_Ctrl	AM30	RGMII1_Rx_Ctrl
AJ31	VSS_297	AK31	RGMII0_Rx_Ctrl	AL31	VSS_316	AM31	RGMII0_TxClk
AJ32	GPIO1	AK32	MDC1	AL32	RGMII0_RD3	AM32	VSS_318
AJ33	VDD_OUT33_18	AK33	GPIO6	AL33	VDD_IO25_49	AM33	RGMII0_RD2
AJ34	PI_Data10	AK34	PI_Data4	AL34	PI_Data3	AM34	GPIO2
AK1	VSS_298	AL1	PLL1_ClkP	AM1	PLL1_En	AN1	VSS_319
AK2	PLL1_AGND	AL2	PLL1_ClkN	AM2	VSS_317	AN2	VSS_320

Table 324. Signal List by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
AN3	VDD_IO25_86	AN19	RGMII5_TxClk	AP2	VSS_326	AP18	RGMII6_Rx_Ctrl
AN4	RGMII11_RD3	AN20	RGMII5_RD3	AP3	RGMII11_RD2	AP19	RGMII5_Tx_Ctrl
AN5	RGMII10_TD1	AN21	VSS_322	AP4	RGMII11_RxClk	AP20	RGMII5_RD1
AN6	RGMII10_RD0	AN22	RGMII4_TxClk	AP5	VSS_327	AP21	VDD_IO25_63
AN7	RGMII10_Rx_Ctrl	AN23	RGMII4_RxClk	AP6	RGMII10_RD2	AP22	RGMII4_TD3
AN8	RGMII9_TxClk	AN24	RGMII3_TD3	AP7	RGMII9_TD3	AP23	RGMII4_RD3
AN9	VDD_IO25_81	AN25	VDD_IO25_57	AP8	RGMII9_RD2	AP24	RGMII3_TD0
AN10	RGMII8_TD0	AN26	RGMII3_Rx_Ctrl	AP9	VSS_328	AP25	VSS_330
AN11	RGMII8_TxClk	AN27	RGMII2_TD3	AP10	RGMII8_TD2	AP26	RGMII3_RD1
AN12	RGMII8_RxClk	AN28	RGMII2_Rx_Ctrl	AP11	RGMII8_RD1	AP27	RGMII2_TD0
AN13	VSS_321	AN29	VSS_323	AP12	RGMII7_TD0	AP28	RGMII2_TxClk
AN14	RGMII7_RD1	AN30	RGMII1_RD2	AP13	VDD_IO25_75	AP29	RGMII2_RxClk
AN15	RGMII7_RxClk	AN31	VDD_IO25_50	AP14	RGMII7_RD2	AP30	RGMII1_TD3
AN16	RGMII6_TxClk	AN32	RGMII0_RD0	AP15	RGMII6_TD0	AP31	RGMII1_RD3
AN17	VDD_IO25_69	AN33	VSS_324	AP16	RGMII6_RD0	AP32	RGMII1_RxClk
AN18	RGMII6_RxClk	AN34	VSS_325	AP17	VSS_329	AP33	VSS_331

6.2.2 Signals by Signal Name

Table 325 lists all the signals in signal name order.

Table 325. Signal List by Signal Name

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
GPIO0	AF30	PI_Data0	AH32	PLL1_ClkN	AL2	Reserved_31	N33
GPIO1	AJ32	PI_Data1	AF31	PLL1_ClkP	AL1	Reserved_32	P32
GPIO2	AM34	PI_Data2	AD30	PLL1_En	AM1	Reserved_33	N34
GPIO3	AD29	PI_Data3	AL34	Reserved_0	H30	Reserved_34	R31
GPIO4	AG31	PI_Data4	AK34	Reserved_1	J30	Reserved_35	R32
GPIO5	AE30	PI_Data5	AE31	Reserved_2	H31	Reserved_36	R33
GPIO6	AK33	PI_Data6	AB29	Reserved_3	L29	Reserved_37	R34
GPIO7	AC29	PI_Data7	AC30	Reserved_4	D34	Reserved_38	T30
JTAG_nTRST	F3	PI_Data8	AH33	Reserved_5	K30	Reserved_39	T31
JTAG_TCK	J5	PI_Data9	AF32	Reserved_6	G32	Reserved_40	T33
JTAG_TDI	F4	PI_Data10	AJ34	Reserved_7	J31	Reserved_41	T34
JTAG_TDO	E2	PI_Data11	AG33	Reserved_8	M29	Reserved_42	U34
JTAG_TMS	H5	PI_Data12	AE32	Reserved_9	F33	Reserved_43	U29
MDC0	AG28	PI_Data13	AH34	Reserved_10	L30	Reserved_44	U31
MDC1	AK32	PI_Data14	AA30	Reserved_11	G33	Reserved_45	U32
MDIO0	AJ30	PI_Data15	AB31	Reserved_12	N29	Reserved_46	V34
MDIO1	AH30	PI_nCS	Y34	Reserved_13	F34	Reserved_47	AE33
nReset	F6	PI_nDone	V32	Reserved_14	J32	Reserved_48	AD32
PI_Addr0	AC32	PI_nDRdy	V33	Reserved_15	H33	Reserved_49	AG34
PI_Addr1	AE34	PI_nOE	W33	Reserved_16	L31	Reserved_50	AG30
PI_Addr2	AD34	PI_nRd	V29	Reserved_17	G34	Reserved_51	G5
PI_Addr3	AB32	PI_nWR	V31	Reserved_18	K32	Reserved_52	J6
PI_Addr4	Y30	PLL0_AGND	AE5	Reserved_19	N30	Reserved_53	H7
PI_Addr5	AA31	PLL0_AVDD	AE6	Reserved_20	H34	Reserved_54	E5
PI_Addr6	AC33	PLL0_Cap0	AG4	Reserved_21	M31	Reserved_55	C4
PI_Addr7	AC34	PLL0_Cap1	AH4	Reserved_22	K33	Reserved_56	G7
PI_Addr8	Y31	PLL0_Clk125_En	AJ3	Reserved_23	K34	Reserved_57	A3
PI_Addr9	AB34	PLL0_ClkN	AJ2	Reserved_24	M32	Reserved_58	H8
PI_Addr10	AA33	PLL0_ClkP	AJ1	Reserved_25	P30	Reserved_59	E6
PI_Addr11	W30	PLL0_En	AJ4	Reserved_26	P31	Reserved_60	B4
PI_Addr12	Y32	PLL1_AGND	AK2	Reserved_27	L33	Reserved_61	C5
PI_Addr13	AA34	PLL1_AVDD	AK3	Reserved_28	N32	Reserved_62	D6
PI_Addr14	W31	PLL1_Cap0	AF6	Reserved_29	L34	Reserved_63	A4
PI_Addr15	Y33	PLL1_Cap1	AF5	Reserved_30	M34	Reserved_64	E7

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
Reserved_65	B5	Reserved_99	H14	Reserved_133	B20	Reserved_167	D26
Reserved_66	F8	Reserved_100	A12	Reserved_134	C20	Reserved_168	H23
Reserved_67	C6	Reserved_101	C13	Reserved_135	H19	Reserved_169	C27
Reserved_68	G9	Reserved_102	F14	Reserved_136	D20	Reserved_170	B28
Reserved_69	B6	Reserved_103	D14	Reserved_137	G19	Reserved_171	A29
Reserved_70	H10	Reserved_104	C14	Reserved_138	C21	Reserved_172	C28
Reserved_71	C7	Reserved_105	B14	Reserved_139	A22	Reserved_173	A30
Reserved_72	A6	Reserved_106	H15	Reserved_140	F20	Reserved_174	H24
Reserved_73	D8	Reserved_107	G15	Reserved_141	B22	Reserved_175	G25
Reserved_74	B7	Reserved_108	A14	Reserved_142	A23	Reserved_176	F26
Reserved_75	E9	Reserved_109	E15	Reserved_143	H20	Reserved_177	E27
Reserved_76	H11	Reserved_110	C15	Reserved_144	E21	Reserved_178	D28
Reserved_77	C8	Reserved_111	B15	Reserved_145	C22	Reserved_179	C29
Reserved_78	F10	Reserved_112	A15	Reserved_146	B23	Reserved_180	B30
Reserved_79	A7	Reserved_113	H16	Reserved_147	D22	Reserved_181	A31
Reserved_80	G11	Reserved_114	F16	Reserved_148	A24	Reserved_182	A32
Reserved_81	B8	Reserved_115	D16	Reserved_149	C23	Reserved_183	C31
Reserved_82	C9	Reserved_116	C16	Reserved_150	G21	Reserved_184	E29
Reserved_83	H12	Reserved_117	B16	Reserved_151	B24	Reserved_185	G27
Reserved_84	A8	Reserved_118	A16	Reserved_152	C24	Reserved_186	H26
Reserved_85	D10	Reserved_119	C17	Reserved_153	F22	Reserved_187	F28
Reserved_86	E11	Reserved_120	E17	Reserved_154	E23	Reserved_188	D30
Reserved_87	C10	Reserved_121	A18	Reserved_155	A26	Reserved_189	B32
Reserved_88	B10	Reserved_122	G17	Reserved_156	D24	Reserved_190	H27
Reserved_89	F12	Reserved_123	B18	Reserved_157	C25	Reserved_191	C33
Reserved_90	A10	Reserved_124	C18	Reserved_158	B26	Reserved_192	D32
Reserved_91	C11	Reserved_125	D18	Reserved_159	H22	Reserved_193	E31
Reserved_92	B11	Reserved_126	A19	Reserved_160	A27	Reserved_194	F30
Reserved_93	G13	Reserved_127	F18	Reserved_161	C26	Reserved_195	G29
Reserved_94	D12	Reserved_128	H18	Reserved_162	B27	Reserved_196	H28
Reserved_95	A11	Reserved_129	B19	Reserved_163	G23	Reserved_197	C34
Reserved_96	C12	Reserved_130	C19	Reserved_164	F24	Reserved_198	F32
Reserved_97	E13	Reserved_131	A20	Reserved_165	A28	RGMII0_RD0	AN32
Reserved_98	B12	Reserved_132	E19	Reserved_166	E25	RGMII0_RD1	AG27

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
RGMII0_RD2	AM33	RGMII3_RD0	AK23	RGMII5_Tx_Ctrl	AP19	RGMII8_TD2	AP10
RGMII0_RD3	AL32	RGMII3_RD1	AP26	RGMII5_TxClk	AN19	RGMII8_TD3	AM11
RGMII0_Rx_Ctrl	AK31	RGMII3_RD2	AL24	RGMII6_RD0	AP16	RGMII8_Tx_Ctrl	AH13
RGMII0_RxClk	AH29	RGMII3_RD3	AM25	RGMII6_RD1	AM17	RGMII8_TxClk	AN11
RGMII0_TD0	AH27	RGMII3_Rx_Ctrl	AN26	RGMII6_RD2	AK17	RGMII9_RD0	AG12
RGMII0_TD1	AG26	RGMII3_RxClk	AG22	RGMII6_RD3	AH17	RGMII9_RD1	AM9
RGMII0_TD2	AJ28	RGMII3_TD0	AP24	RGMII6_Rx_Ctrl	AP18	RGMII9_RD2	AP8
RGMII0_TD3	AK29	RGMII3_TD1	AM23	RGMII6_RxClk	AN18	RGMII9_RD3	AL10
RGMII0_Tx_Ctrl	AL30	RGMII3_TD2	AH21	RGMII6_TD0	AP15	RGMII9_Rx_Ctrl	AK11
RGMII0_TxClk	AM31	RGMII3_TD3	AN24	RGMII6_TD1	AG16	RGMII9_RxClk	AM10
RGMII1_RD0	AL28	RGMII3_Tx_Ctrl	AJ22	RGMII6_TD2	AJ16	RGMII9_TD0	AG11
RGMII1_RD1	AM29	RGMII3_TxClk	AM24	RGMII6_TD3	AL16	RGMII9_TD1	AM8
RGMII1_RD2	AN30	RGMII4_RD0	AG20	RGMII6_Tx_Ctrl	AM16	RGMII9_TD2	AJ10
RGMII1_RD3	AP31	RGMII4_RD1	AK21	RGMII6_TxClk	AN16	RGMII9_TD3	AP7
RGMII1_Rx_Ctrl	AM30	RGMII4_RD2	AM22	RGMII7_RD0	AH15	RGMII9_Tx_Ctrl	AH11
RGMII1_RxClk	AP32	RGMII4_RD3	AP23	RGMII7_RD1	AN14	RGMII9_TxClk	AN8
RGMII1_TD0	AM28	RGMII4_Rx_Ctrl	AL22	RGMII7_RD2	AP14	RGMII10_RD0	AN6
RGMII1_TD1	AG24	RGMII4_RxClk	AN23	RGMII7_RD3	AK15	RGMII10_RD1	AM7
RGMII1_TD2	AH25	RGMII4_TD0	AH19	RGMII7_Rx_Ctrl	AM15	RGMII10_RD2	AP6
RGMII1_TD3	AP30	RGMII4_TD1	AL20	RGMII7_RxClk	AN15	RGMII10_RD3	AL8
RGMII1_Tx_Ctrl	AJ26	RGMII4_TD2	AM21	RGMII7_TD0	AP12	RGMII10_Rx_Ctrl	AN7
RGMII1_TxClk	AK27	RGMII4_TD3	AP22	RGMII7_TD1	AJ14	RGMII10_RxClk	AK9
RGMII2_RD0	AK25	RGMII4_Tx_Ctrl	AJ20	RGMII7_TD2	AM13	RGMII10_TD0	AK7
RGMII2_RD1	AG23	RGMII4_TxClk	AN22	RGMII7_TD3	AL14	RGMII10_TD1	AN5
RGMII2_RD2	AL26	RGMII5_RD0	AM19	RGMII7_Tx_Ctrl	AM14	RGMII10_TD2	AJ8
RGMII2_RD3	AM27	RGMII5_RD1	AP20	RGMII7_TxClk	AG15	RGMII10_TD3	AM6
RGMII2_Rx_Ctrl	AN28	RGMII5_RD2	AK19	RGMII8_RD0	AL12	RGMII10_Tx_Ctrl	AH9
RGMII2_RxClk	AP29	RGMII5_RD3	AN20	RGMII8_RD1	AP11	RGMII10_TxClk	AG10
RGMII2_TD0	AP27	RGMII5_Rx_Ctrl	AM20	RGMII8_RD2	AM12	RGMII11_RD0	AG8
RGMII2_TD1	AM26	RGMII5_RxClk	AG19	RGMII8_RD3	AK13	RGMII11_RD1	AL5
RGMII2_TD2	AH23	RGMII5_TD0	AM18	RGMII8_Rx_Ctrl	AG14	RGMII11_RD2	AP3
RGMII2_TD3	AN27	RGMII5_TD1	AL18	RGMII8_RxClk	AN12	RGMII11_RD3	AN4
RGMII2_Tx_Ctrl	AJ24	RGMII5_TD2	AJ18	RGMII8_TD0	AN10	RGMII11_Rx_Ctrl	AL6
RGMII2_TxClk	AP28	RGMII5_TD3	AG18	RGMII8_TD1	AJ12	RGMII11_RxClk	AP4

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
RGMII11_TD0	AH6	SPI_RD10P	AA2	SPI_TD8N	N6	VDD_11	J28
RGMII11_TD1	AG7	SPI_RD11N	Y3	SPI_TD8P	N5	VDD_12	K7
RGMII11_TD2	AM3	SPI_RD11P	Y4	SPI_TD9N	M2	VDD_13	K8
RGMII11_TD3	AJ6	SPI_RD12N	Y1	SPI_TD9P	M1	VDD_14	K11
RGMII11_Tx_Ctrl	AH7	SPI_RD12P	Y2	SPI_TD10N	M6	VDD_15	K12
RGMII11_TxClk	AM4	SPI_RD13N	W5	SPI_TD10P	M5	VDD_16	K15
SI_Clk	F2	SPI_RD13P	W6	SPI_TD11N	L2	VDD_17	K16
SI_DI	G3	SPI_RD14N	W1	SPI_TD11P	L1	VDD_18	K19
SI_DO	D1	SPI_RD14P	W2	SPI_TD12N	L4	VDD_19	K20
SI_nEn	F1	SPI_RD15N	V3	SPI_TD12P	L3	VDD_20	K23
SPI_En	T4	SPI_RD15P	V4	SPI_TD13N	L6	VDD_21	K24
SPI_RCtrlN	U1	SPI_RDClkN	U5	SPI_TD13P	L5	VDD_22	K27
SPI_RCtrlP	U2	SPI_RDClkP	U6	SPI_TD14N	K2	VDD_23	K28
SPI_RD0N	AG1	SPI_RSClk	T2	SPI_TD14P	K1	VDD_24	L9
SPI_RD0P	AG2	SPI_RStat0	T1	SPI_TD15N	K6	VDD_25	L10
SPI_RD1N	AF1	SPI_RStat1	T3	SPI_TD15P	K5	VDD_26	L13
SPI_RD1P	AF2	SPI_TCtrlN	J2	SPI_TDClkN	J4	VDD_27	L14
SPI_RD2N	AE3	SPI_TCtrlP	J1	SPI_TDClkP	J3	VDD_28	L17
SPI_RD2P	AE4	SPI_TD0N	T6	SPI_TSClk	H1	VDD_29	L18
SPI_RD3N	AE1	SPI_TD0P	T5	SPI_TStat0	H2	VDD_30	L21
SPI_RD3P	AE2	SPI_TD1N	R2	SPI_TStat1	G1	VDD_31	L22
SPI_RD4N	AD3	SPI_TD1P	R1	Test_Enable	D4	VDD_32	L25
SPI_RD4P	AD4	SPI_TD2N	R4	Test_Mode	C3	VDD_33	L26
SPI_RD5N	AC5	SPI_TD2P	R3	VDD_0	J7	VDD_34	M9
SPI_RD5P	AC6	SPI_TD3N	R6	VDD_1	J8	VDD_35	M10
SPI_RD6N	AC1	SPI_TD3P	R5	VDD_2	J11	VDD_36	M13
SPI_RD6P	AC2	SPI_TD4N	P2	VDD_3	J12	VDD_37	M14
SPI_RD7N	AB3	SPI_TD4P	P1	VDD_4	J15	VDD_38	M17
SPI_RD7P	AB4	SPI_TD5N	P6	VDD_5	J16	VDD_39	M18
SPI_RD8N	AB1	SPI_TD5P	P5	VDD_6	J19	VDD_40	M21
SPI_RD8P	AB2	SPI_TD6N	N2	VDD_7	J20	VDD_41	M22
SPI_RD9N	AA5	SPI_TD6P	N1	VDD_8	J23	VDD_42	M25
SPI_RD9P	AA6	SPI_TD7N	N4	VDD_9	J24	VDD_43	M26
SPI_RD10N	AA1	SPI_TD7P	N3	VDD_10	J27	VDD_44	N7

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VDD_45	N8	VDD_79	T10	VDD_113	W10	VDD_147	AB12
VDD_46	N11	VDD_80	T13	VDD_114	W13	VDD_148	AB15
VDD_47	N12	VDD_81	T14	VDD_115	W14	VDD_149	AB16
VDD_48	N15	VDD_82	T17	VDD_116	W17	VDD_150	AB19
VDD_49	N16	VDD_83	T18	VDD_117	W18	VDD_151	AB20
VDD_50	N19	VDD_84	T21	VDD_118	W21	VDD_152	AB23
VDD_51	N20	VDD_85	T22	VDD_119	W22	VDD_153	AB24
VDD_52	N23	VDD_86	T25	VDD_120	W25	VDD_154	AB27
VDD_53	N24	VDD_87	T26	VDD_121	W26	VDD_155	AB28
VDD_54	N27	VDD_88	U7	VDD_122	Y9	VDD_156	AC9
VDD_55	N28	VDD_89	U8	VDD_123	Y10	VDD_157	AC10
VDD_56	P7	VDD_90	U11	VDD_124	Y13	VDD_158	AC13
VDD_57	P8	VDD_91	U12	VDD_125	Y14	VDD_159	AC14
VDD_58	P11	VDD_92	U15	VDD_126	Y17	VDD_160	AC17
VDD_59	P12	VDD_93	U16	VDD_127	Y18	VDD_161	AC18
VDD_60	P15	VDD_94	U19	VDD_128	Y21	VDD_162	AC21
VDD_61	P16	VDD_95	U20	VDD_129	Y22	VDD_163	AC22
VDD_62	P19	VDD_96	U23	VDD_130	Y25	VDD_164	AC25
VDD_63	P20	VDD_97	U24	VDD_131	Y26	VDD_165	AC26
VDD_64	P23	VDD_98	U27	VDD_132	AA7	VDD_166	AD9
VDD_65	P24	VDD_99	U28	VDD_133	AA8	VDD_167	AD10
VDD_66	P27	VDD_100	V7	VDD_134	AA11	VDD_168	AD13
VDD_67	P28	VDD_101	V8	VDD_135	AA12	VDD_169	AD14
VDD_68	R9	VDD_102	V11	VDD_136	AA15	VDD_170	AD17
VDD_69	R10	VDD_103	V12	VDD_137	AA16	VDD_171	AD18
VDD_70	R13	VDD_104	V15	VDD_138	AA19	VDD_172	AD21
VDD_71	R14	VDD_105	V16	VDD_139	AA20	VDD_173	AD22
VDD_72	R17	VDD_106	V19	VDD_140	AA23	VDD_174	AD25
VDD_73	R18	VDD_107	V20	VDD_141	AA24	VDD_175	AD26
VDD_74	R21	VDD_108	V23	VDD_142	AA27	VDD_176	AE7
VDD_75	R22	VDD_109	V24	VDD_143	AA28	VDD_177	AE8
VDD_76	R25	VDD_110	V27	VDD_144	AB7	VDD_178	AE11
VDD_77	R26	VDD_111	V28	VDD_145	AB8	VDD_179	AE12
VDD_78	T9	VDD_112	W9	VDD_146	AB11	VDD_180	AE15

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VDD_181	AE16	VDD_IO25_15	E16	VDD_IO25_49	AL33	VDD_IO25_83	AJ7
VDD_182	AE19	VDD_IO25_16	B17	VDD_IO25_50	AN31	VDD_IO25_84	AK6
VDD_183	AE20	VDD_IO25_17	D17	VDD_IO25_51	AL29	VDD_IO25_85	AM5
VDD_184	AE23	VDD_IO25_18	H17	VDD_IO25_52	AH28	VDD_IO25_86	AN3
VDD_185	AE24	VDD_IO25_19	G18	VDD_IO25_53	AJ27	VDD_IO25_87	AG3
VDD_186	AE27	VDD_IO25_20	F19	VDD_IO25_54	AK26	VDD_IO25_88	AD6
VDD_187	AE28	VDD_IO25_21	E20	VDD_IO25_55	AG25	VDD_IO25_89	AD1
VDD_188	AF7	VDD_IO25_22	A21	VDD_IO25_56	AL25	VDD_IO25_90	AC3
VDD_189	AF8	VDD_IO25_23	D21	VDD_IO25_57	AN25	VDD_IO25_91	AB6
VDD_190	AF11	VDD_IO25_24	H21	VDD_IO25_58	AH24	VDD_IO25_92	AA4
VDD_191	AF12	VDD_IO25_25	G22	VDD_IO25_59	AJ23	VDD_IO25_93	Y6
VDD_192	AF15	VDD_IO25_26	F23	VDD_IO25_60	AK22	VDD_IO25_94	W3
VDD_193	AF16	VDD_IO25_27	E24	VDD_IO25_61	AG21	VDD_IO25_95	V6
VDD_194	AF19	VDD_IO25_28	B25	VDD_IO25_62	AL21	VDD_IO25_96	V1
VDD_195	AF20	VDD_IO25_29	D25	VDD_IO25_63	AP21	VDD_IO25_97	U4
VDD_196	AF23	VDD_IO25_30	H25	VDD_IO25_64	AH20	VDD_IO25_98	P3
VDD_197	AF24	VDD_IO25_31	G26	VDD_IO25_65	AJ19	VDD_IO25_99	M4
VDD_198	AF27	VDD_IO25_32	F27	VDD_IO25_66	AK18	VDD_IO25_100	K3
VDD_199	AF28	VDD_IO25_33	E28	VDD_IO25_67	AG17	VDD_IO25_101	C1
VDD_IO25_0	B3	VDD_IO25_34	D29	VDD_IO25_68	AL17	VDD_OUT33_0	H3
VDD_IO25_1	D5	VDD_IO25_35	H29	VDD_IO25_69	AN17	VDD_OUT33_1	G4
VDD_IO25_2	F7	VDD_IO25_36	C30	VDD_IO25_70	AH16	VDD_OUT33_2	E33
VDD_IO25_3	E8	VDD_IO25_37	G30	VDD_IO25_71	AJ15	VDD_OUT33_3	H32
VDD_IO25_4	B9	VDD_IO25_38	B31	VDD_IO25_72	AK14	VDD_OUT33_4	J34
VDD_IO25_5	D9	VDD_IO25_39	F31	VDD_IO25_73	AG13	VDD_OUT33_5	K29
VDD_IO25_6	H9	VDD_IO25_40	E32	VDD_IO25_74	AL13	VDD_OUT33_6	L32
VDD_IO25_7	G10	VDD_IO25_41	D33	VDD_IO25_75	AP13	VDD_OUT33_7	N31
VDD_IO25_8	F11	VDD_IO25_42	P29	VDD_IO25_76	AH12	VDD_OUT33_8	P33
VDD_IO25_9	E12	VDD_IO25_43	R29	VDD_IO25_77	AJ11	VDD_OUT33_9	R30
VDD_IO25_10	A13	VDD_IO25_44	T29	VDD_IO25_78	AK10	VDD_OUT33_10	U33
VDD_IO25_11	D13	VDD_IO25_45	W29	VDD_IO25_79	AG9	VDD_OUT33_11	V30
VDD_IO25_12	H13	VDD_IO25_46	Y29	VDD_IO25_80	AL9	VDD_OUT33_12	W32
VDD_IO25_13	G14	VDD_IO25_47	AA29	VDD_IO25_81	AN9	VDD_OUT33_13	AB33
VDD_IO25_14	F15	VDD_IO25_48	AG29	VDD_IO25_82	AH8	VDD_OUT33_14	AC31

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VDD_OUT33_15	AE29	VSS_30	E22	VSS_64	K4	VSS_98	M24
VDD_OUT33_16	AF34	VSS_31	E26	VSS_65	K9	VSS_99	M27
VDD_OUT33_17	AH31	VSS_32	E30	VSS_66	K10	VSS_100	M28
VDD_OUT33_18	AJ33	VSS_33	E34	VSS_67	K13	VSS_101	M30
VSS_0	A2	VSS_34	F5	VSS_68	K14	VSS_102	M33
VSS_1	A5	VSS_35	F9	VSS_69	K17	VSS_103	N9
VSS_2	A9	VSS_36	F13	VSS_70	K18	VSS_104	N10
VSS_3	A17	VSS_37	F17	VSS_71	K21	VSS_105	N13
VSS_4	A25	VSS_38	F21	VSS_72	K22	VSS_106	N14
VSS_5	A33	VSS_39	F25	VSS_73	K25	VSS_107	N17
VSS_6	B1	VSS_40	F29	VSS_74	K26	VSS_108	N18
VSS_7	B2	VSS_41	G2	VSS_75	K31	VSS_109	N21
VSS_8	B13	VSS_42	G6	VSS_76	L7	VSS_110	N22
VSS_9	B21	VSS_43	G8	VSS_77	L8	VSS_111	N25
VSS_10	B29	VSS_44	G12	VSS_78	L11	VSS_112	N26
VSS_11	B33	VSS_45	G16	VSS_79	L12	VSS_113	P4
VSS_12	B34	VSS_46	G20	VSS_80	L15	VSS_114	P9
VSS_13	C2	VSS_47	G24	VSS_81	L16	VSS_115	P10
VSS_14	C32	VSS_48	G28	VSS_82	L19	VSS_116	P13
VSS_15	D2	VSS_49	G31	VSS_83	L20	VSS_117	P14
VSS_16	D3	VSS_50	H4	VSS_84	L23	VSS_118	P17
VSS_17	D7	VSS_51	H6	VSS_85	L24	VSS_119	P18
VSS_18	D11	VSS_52	J9	VSS_86	L27	VSS_120	P21
VSS_19	D15	VSS_53	J10	VSS_87	L28	VSS_121	P22
VSS_20	D19	VSS_54	J13	VSS_88	M3	VSS_122	P25
VSS_21	D23	VSS_55	J14	VSS_89	M7	VSS_123	P26
VSS_22	D27	VSS_56	J17	VSS_90	M8	VSS_124	P34
VSS_23	D31	VSS_57	J18	VSS_91	M11	VSS_125	R7
VSS_24	E1	VSS_58	J21	VSS_92	M12	VSS_126	R8
VSS_25	E3	VSS_59	J22	VSS_93	M15	VSS_127	R11
VSS_26	E4	VSS_60	J25	VSS_94	M16	VSS_128	R12
VSS_27	E10	VSS_61	J26	VSS_95	M19	VSS_129	R15
VSS_28	E14	VSS_62	J29	VSS_96	M20	VSS_130	R16
VSS_29	E18	VSS_63	J33	VSS_97	M23	VSS_131	R19

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VSS_132	R20	VSS_166	V13	VSS_200	Y28	VSS_234	AC23
VSS_133	R23	VSS_167	V14	VSS_201	AA3	VSS_235	AC24
VSS_134	R24	VSS_168	V17	VSS_202	AA9	VSS_236	AC27
VSS_135	R27	VSS_169	V18	VSS_203	AA10	VSS_237	AC28
VSS_136	R28	VSS_170	V21	VSS_204	AA13	VSS_238	AD2
VSS_137	T7	VSS_171	V22	VSS_205	AA14	VSS_239	AD5
VSS_138	T8	VSS_172	V25	VSS_206	AA17	VSS_240	AD7
VSS_139	T11	VSS_173	V26	VSS_207	AA18	VSS_241	AD8
VSS_140	T12	VSS_174	W4	VSS_208	AA21	VSS_242	AD11
VSS_141	T15	VSS_175	W7	VSS_209	AA22	VSS_243	AD12
VSS_142	T16	VSS_176	W8	VSS_210	AA25	VSS_244	AD15
VSS_143	T19	VSS_177	W11	VSS_211	AA26	VSS_245	AD16
VSS_144	T20	VSS_178	W12	VSS_212	AA32	VSS_246	AD19
VSS_145	T23	VSS_179	W15	VSS_213	AB5	VSS_247	AD20
VSS_146	T24	VSS_180	W16	VSS_214	AB9	VSS_248	AD23
VSS_147	T27	VSS_181	W19	VSS_215	AB10	VSS_249	AD24
VSS_148	T28	VSS_182	W20	VSS_216	AB13	VSS_250	AD27
VSS_149	T32	VSS_183	W23	VSS_217	AB14	VSS_251	AD28
VSS_150	U3	VSS_184	W24	VSS_218	AB17	VSS_252	AD31
VSS_151	U9	VSS_185	W27	VSS_219	AB18	VSS_253	AD33
VSS_152	U10	VSS_186	W28	VSS_220	AB21	VSS_254	AE9
VSS_153	U13	VSS_187	W34	VSS_221	AB22	VSS_255	AE10
VSS_154	U14	VSS_188	Y5	VSS_222	AB25	VSS_256	AE13
VSS_155	U17	VSS_189	Y7	VSS_223	AB26	VSS_257	AE14
VSS_156	U18	VSS_190	Y8	VSS_224	AB30	VSS_258	AE17
VSS_157	U21	VSS_191	Y11	VSS_225	AC4	VSS_259	AE18
VSS_158	U22	VSS_192	Y12	VSS_226	AC7	VSS_260	AE21
VSS_159	U25	VSS_193	Y15	VSS_227	AC8	VSS_261	AE22
VSS_160	U26	VSS_194	Y16	VSS_228	AC11	VSS_262	AE25
VSS_161	U30	VSS_195	Y19	VSS_229	AC12	VSS_263	AE26
VSS_162	V2	VSS_196	Y20	VSS_230	AC15	VSS_264	AF3
VSS_163	V5	VSS_197	Y23	VSS_231	AC16	VSS_265	AF4
VSS_164	V9	VSS_198	Y24	VSS_232	AC19	VSS_266	AF9
VSS_165	V10	VSS_199	Y27	VSS_233	AC20	VSS_267	AF10

Table 325. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VSS_268	AF13	VSS_284	AH5	VSS_300	AK5	VSS_316	AL31
VSS_269	AF14	VSS_285	AH10	VSS_301	AK8	VSS_317	AM2
VSS_270	AF17	VSS_286	AH14	VSS_302	AK12	VSS_318	AM32
VSS_271	AF18	VSS_287	AH18	VSS_303	AK16	VSS_319	AN1
VSS_272	AF21	VSS_288	AH22	VSS_304	AK20	VSS_320	AN2
VSS_273	AF22	VSS_289	AH26	VSS_305	AK24	VSS_321	AN13
VSS_274	AF25	VSS_290	AJ5	VSS_306	AK28	VSS_322	AN21
VSS_275	AF26	VSS_291	AJ9	VSS_307	AK30	VSS_323	AN29
VSS_276	AF29	VSS_292	AJ13	VSS_308	AL3	VSS_324	AN33
VSS_277	AF33	VSS_293	AJ17	VSS_309	AL4	VSS_325	AN34
VSS_278	AG5	VSS_294	AJ21	VSS_310	AL7	VSS_326	AP2
VSS_279	AG6	VSS_295	AJ25	VSS_311	AL11	VSS_327	AP5
VSS_280	AG32	VSS_296	AJ29	VSS_312	AL15	VSS_328	AP9
VSS_281	AH1	VSS_297	AJ31	VSS_313	AL19	VSS_329	AP17
VSS_282	AH2	VSS_298	AK1	VSS_314	AL23	VSS_330	AP25
VSS_283	AH3	VSS_299	AK4	VSS_315	AL27	VSS_331	AP33

7 PACKAGE INFORMATION

The VSC7326 device is available in two package types. VSC7326VV is a 1152-pin, thermally enhanced, plastic ball grid array (PBGA). The device is also available in a lead-free package, VSC7326XVV.

Lead-free products from Vitesse comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

This section provides the package drawing, thermal specifications, and moisture sensitivity rating for the VSC7326 device.

7.1 Package Drawing

The following illustration shows the top, bottom, and side view of the package drawing. The A1 ball corner location is at the top of the package. All measurements are in millimeters.

All Measurements are in mm
Body Size: 35.00 × 35.00 Thermally Enhanced PBGA Package
No. of Balls: 1152
Ball Pitch: 1.00 mm
Ball footprint: 34 × 34
A1 Ball Corner Location Indication on Top of Package

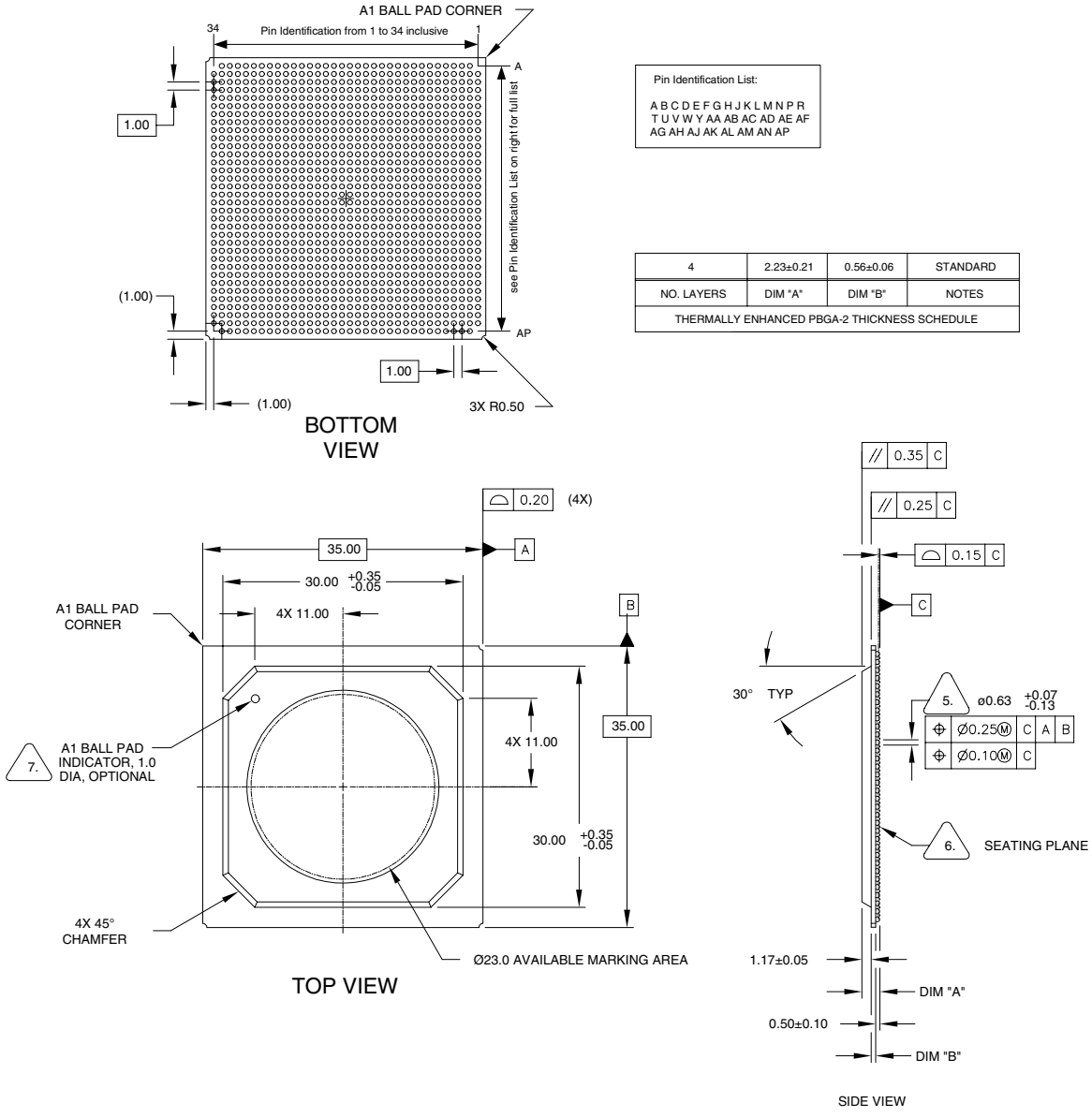


Figure 76. Package Drawing

7.2 Thermal Specifications

Thermal specifications for this device are based on the JEDEC standard EIA/JESD51-2 and have been modeled using a four-layer test board with two signal layers, a power plane, and a ground plane (2s2p PCB). For more information, see the JEDEC standard.

Table 326. Thermal Resistances

Part Number	θ_{JC}	θ_{JA} (°C/W) vs. Airflow (ft/min)		
		0	100	200
VSC7326VV	2.1	11.3	9.0	7.7
VSC7326XVV	2.1	11.3	9.0	7.7

To achieve results similar to the modeled thermal resistance measurements, the guidelines for board design described in the JEDEC standard EIA/JESD51 series must be applied. For information about specific applications, see the following:

EIA/JESD51-5, *Extension of Thermal Test Board Standards for Packages with Direct Thermal Attachment Mechanisms*

EIA/JESD51-7, *High Effective Thermal Conductivity Test Board for Leaded Surface Mount Packages*

EIA/JESD51-9, *Test Boards for Area Array Surface Mount Package Thermal Measurements*

EIA/JESD51-10, *Test Boards for Through-Hole Perimeter Leaded Package Thermal Measurements*

EIA/JESD51-11, *Test Boards for Through-Hole Area Array Leaded Package Thermal Measurements*

7.3 Moisture Sensitivity

This device is rated moisture sensitivity level 4 as specified in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

8 DESIGN GUIDELINES

8.1 Power Supplies

The following guidelines apply to designing power supplies for use with the Schaumburg device.

- Make an unbroken ground plane (GND).
- Use an unbroken plane for each major supply voltage.
- Use the power and ground plane combination as an effective power supply bypass capacitor. The capacitance is proportional to the area of the two planes and inversely proportional to the separation between the planes. Typical values with a 0.25 mm (0.01 in) separation are 100 pF/in^2. This capacitance is more effective than a capacitor of equivalent value because the planes have no inductance or Equivalent Series Resistance (ESR).
- Do not cut up the power or ground planes in an effort to steer current paths. This usually produces more noise, not less. Furthermore, place vias and clearances in a configuration that maintains the integrity of the plane. Groups of vias spaced close together often overlap clearances. This can form a large hole in the plane. As a result, return currents are forced around the holes. This increases the loop area and therefore EMI emissions. Signals should never be placed on a ground plane because the resulting slot in the GND plane forces return currents around the slot.
- Vias used to connect power planes to the supply and ground balls should be at least 0.25 mm (0.010 in) in diameter, preferably with no thermal relief, and preferably plated closed with copper or solder. Use separate (or even multiple) vias for each supply and ground ball.

8.1.1 Power Supply Decoupling

Each power supply voltage should have both bulk and high-frequency decoupling capacitors. Recommended capacitors are as follows:

- For bulk decoupling, use 10 µF high-capacity and low-ESR capacitors distributed across the board.
- For high frequency decoupling, use 0.01 µF to 0.1 µF high frequency (for example, X7R) ceramic capacitors placed on the side of the PCB closest to the plane being decoupled and as close as possible to the power ball. A larger value in the same housing is better.
- Use surface-mounted components for lower lead inductance and pad capacitance. Smaller form factor components are best (that is, 0603 is better than 0805).

8.2 PLL

8.2.1 PLL Reference Clock

The differential LVDS inputs PLL0_ClkN, PLL0_ClkP and PLL1_ClkN, PLL1_ClkP do not have an internal differential termination resistor. When using the inputs in differential mode, an external termination resistor network is required. PLLx_ClkN and PLLx_ClkP have internal biasing resistors of 20 kΩ to VDD_PLLx, and 27 kΩ to VSS_PLLx, resulting in a bias voltage of approximately 1.4 V.

When used as an LVTTL input, the unused input must be left floating. The threshold voltage will be approximately 1.4 V.

8.2.2 PLL Supply Filtering

Analog PLL supplies should be filtered with a ferrite bead in series with the decoupling capacitor. The capacitor should be located between the ball and the vias to both planes.

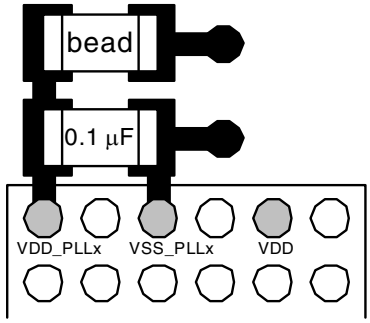


Figure 77. Filtered Analog Power Decoupling

8.2.3 PLL Filter Capacitors

The on-chip PLLs use an external 0.1 µF capacitor connected between the Cap0 and Cap1 terminals to control the loop filter. This capacitor must be a multilayer ceramic dielectric, or better, with a working voltage rating of at least 5 V, and a good temperature coefficient (NPO is preferred but X7R may be acceptable). These capacitors are used to minimize the impact of common-mode noise (especially power supply noise) on the Clock Multiplier Unit. Higher value capacitors provide better robustness in systems. An NPO capacitor is preferred over an X7R capacitor because the power supply noise sensitivity varies with temperature.

For best noise immunity, the designer may choose to use a three-capacitor circuit with one differential capacitor between Cap0 and Cap1 (C1), a capacitor from Cap0 to GND (C2), and a capacitor from Cap1 to GND (C3). Larger values are better but 0.1 µF is adequate. However, if the designer cannot use a three-capacitor circuit, a single differential capacitor, C1, is adequate. These components must be isolated from noisy traces.

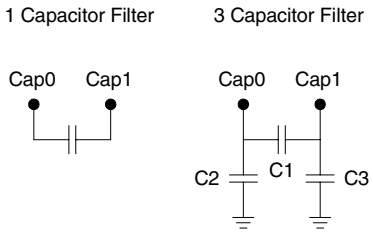


Figure 78. PLL Filter Capacitor

8.2.4 SPI-4.2 Jitter Considerations

Using a low frequency (25 MHz to 37.5 MHz) clock for the SPI-4.2 PLL clock input (PLL1_Clk) can result in SPI-4.2 clock output jitter that violates SPI-4.2 specifications in dynamic mode. If this is a concern, you should feed PLL1_Clk with a differential clock source, with a frequency of at least 100 MHz, which may not be generated by a PLL-based reference oscillator.

For more information about SPI-4.2 jitter considerations when operating in dynamic mode, see your Vitesse application engineer.

8.3 Interfaces

8.3.1 General Recommendations

High-speed signals require excellent frequency and phase response up to the third harmonic. The best design would provide excellent frequency and phase response up to the seventh harmonic. The following recommendations can improve signal quality and minimize transmission distances:

- Keep traces as short as possible. Initial component placement should be considered very carefully.
- The impedance of the traces must match the impedance of the termination resistors, connectors, and cable. This reduces reflections due to impedance mismatches.
- Impedance matching termination resistors should be located as close to the input pin of the receiver as possible to minimize stub length. On the receiver, place the termination resistors at the end of the trace about 0.25 in. beyond the package pin to provide high quality, feed-through termination. However, bunching passive components can result in impedance discontinuities due to the capacitance of the pads.
- Differential impedance must be maintained in a 100 Ω differential application. Routing two 50 Ω traces is not adequate. The two traces must be separated by enough distance to maintain differential impedance. A good rule of thumb is to keep trace separation at least 2.5 times the trace width. If tighter spacing is desired, you must calculate the differential impedance rather than assume the impedance is twice the single-ended impedance.
- When routing differential pairs, keep the two trace lengths identical. Differences in trace lengths translate directly into signal skew. Note that the differential impedance may be affected when separations occur.

- Keep differential pair traces on the same layer of the PCB to minimize impedance discontinuities.
- Do not group all the passive components together. The pads of the components add capacitance to the traces. At the frequencies encountered, this can result in unwanted reductions in impedance. Use surface-mounted 0603 components to reduce this effect.
- Eliminate or reduce stub lengths.
- Reduce, or eliminate, vias to minimize impedance discontinuities. Remember that vias and their clearance holes in the power or ground planes can cause impedance discontinuities in nearby signals. Keep vias away from traces at a distance of at least 2.5 times the trace width. Optimally, keep them away from traces at a distance of ten times the trace width if possible.
- Use rounded corners rather than 90^{o} or 45^{o} corners on wide traces.
- Keep critical signal traces away from other signals, which might capacitively couple noise into the signals. A good rule of thumb is to keep the traces apart by ten times the width of the trace.
- Do not route digital signals from other circuits across the area of the high-speed transmitter and receiver signals.
- Using grounded guard traces is typically not effective for improving signal quality. A ground plane is more effective. However, a common use of guard traces is to route them during the layout, but remove them prior to design completion. This has the benefit of enforcing “keep-out areas” around sensitive high-speed signals so that vias and other traces are not accidentally placed incorrectly.
- Mismatch between signals in a differential pair leads to common mode currents, which are usually less than one percent of the differential currents in a well-designed system. These common-mode currents represent a primary source of EMI emissions. Routing of differential traces to match length reduces common mode currents. For example, a 5 mm (0.2 in) length mismatch between differential signals having rise and fall times of 200 ps results in the common mode current being up to 18% of the differential current.

All applications must use a termination impedance. The recommended value is between 90 Ω and 110 Ω for differential signals. The actual value must be selected to match the media characteristic impedance (±10%) at the application frequency. The termination impedance may be integrated into the receiver-integrated circuit, but it still must meet the requirements. The Schaumburg device has a built-in differential termination in the inputs. If the termination impedance is not integrated into the receiver circuit, it must be located at the load end of the balanced interconnecting media.

Notes:

1. Due to the high application frequency, you must be careful to choose proper components (such as the termination resistors) when designing the layout of a printed circuit board. The use of surface-mounted components is highly recommended to minimize parasitic inductance and lead length of the termination resistor. Wire-wound resistors are not recommended.
2. Matching the impedance of the PCB traces, connectors, and balanced interconnect media is highly recommended. Impedance variations along the entire interconnect path must be minimized because they degrade the signal path and may cause reflections of the signal.

8.3.1.1 RGMII

The RGMII interface consists of a 4-bit data bus, a clock, and a control line. It is a source-synchronous Double Data Rate (DDR) interface operating at 125 MHz.

The RGMII signals can be routed on any PCB trace layer with the following constraints:

- The Tx_Ctrl and TD[3:0] output signals should have matched electrical lengths. It is recommended that the length not exceed 40 cm (16 in.).
- The Tx_Clk signal requires an additional 1.5 ns of trace delay on the PCB unless the attached PHY can skew the received clock by the necessary amount.
- The Rx_Ctrl, Rx_Clk, and RD[3:0] input signals should have matched electrical lengths. The PCB termination and clock delays should be implemented as required by the attached PHY.
- If a port is not used, then pull Rx_Ctrl low, and pull RD[3:0] and Rx_Clk pull high or low.
- All RGMII outputs are designed to drive one load and they have built-in series termination optimized for 50 Ω transmission lines. If any additional termination is added to the RGMII output signals, it is recommended that all signals within each port be terminated equally. All RGMII traces should be impedance controlled. The recommended value is 50 Ω to a ground plane, but this is not a strict requirement and the board designer can experiment with higher values if needed.
- To reduce the crosstalk between signal and clock (Tx_Clk and Rx_Clk) PCB lines, it is recommended that the spacing on each side of the clock lines be larger than twice the track width.

8.3.2 MII Management

Schaumburg includes two identical management buses. It is recommended that they are equally loaded and the layout is done very carefully.

It is recommended to route the clock signal (MDC_x) from Schaumburg to PHY 1, then PHY 2, then PHY n, and then terminated with 100 Ω to GND and 90 Ω to VDD_IO25. It is also recommended to route the PCB trace with as high impedance as possible (at least 68 Ω).

MDIO_x should have a 1.5 kΩ pull-up resistor to 2.5 V, and the trace should follow the MDC_x trace as shown in the following figure.

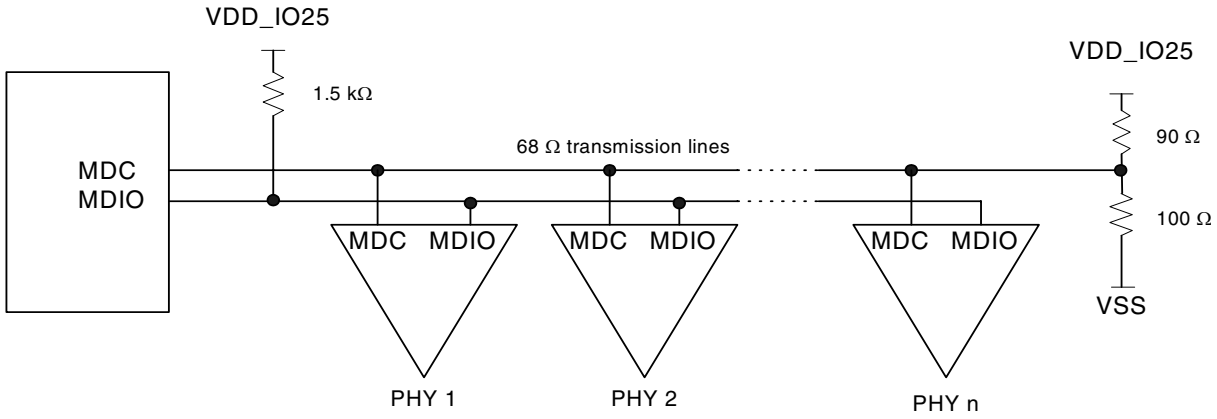


Figure 79. MDC and MDIO Layout Scheme

Be careful to follow the layout guidelines described in the Termination Considerations section. For more information, see "Termination Considerations," page 296.

If it is desired to route the PCB-trace with 50 Ω, it is recommend that a clock driver be used. This driver should be placed close to the Schaumburg. It must be verified against the appropriate specifications in this document that this driver is capable of driving 35 Ω traces. This is because the impedance is lowered when the trace is loaded capacitive-wise by the PHYs. Routing guidelines are the same as above, and the termination should be equal to the loaded trace impedance (typically 35 Ω).

It also is possible to use individual clock drivers for each PHY, routed in a star configuration.

8.3.3 Termination Considerations

8.3.3.1 MAC Interfaces

All signals in the MAC interface and the reference Clk input (both on Schaumburg and the PHYs), require termination as described in the RGMII section. For more information, see "RGMII," page 295.

Summary:

- Place series termination resistors, if needed, as close to the output pins as possible.
- Keep output traces, between Schaumburg and one PHY, approximately the same length. Notice the special clock track requirement in RGMII mode if the PHY does not include the clock delay.
- Keep input traces, between one PHY and Schaumburg, approximately the same length. Notice the special clock track requirement in RGMII mode, if the PHY does not include the clock delay.

8.3.3.2 Other Outputs from Schaumburg

All other outputs do not have a built-in series termination, and they should be terminated appropriately on the board if required.

8.3.3.3 Other Inputs to Schaumburg

For more information about recommendations on the termination of input signals to Schaumburg, see the datasheet for the specific devices.

8.3.4 SPI-4.2 Interface

The SPI-4 phase 2 interface consists of a 16-bit data path with associated clock and control signals and a 2-bit FIFO status channel with an associated clock. The data path signals are source-synchronous Double Data Rate (DDR) differential LVDS operating between 75 MHz and 407 MHz. The data path signals support dynamic deskew for clock frequencies greater than 300 MHz The status channel signals are source-synchronous single-ended LVTTL operating at a maximum of 112 MHz.

The SPI-4.2 signals can be routed on any PCB trace layer with the following constraints:

- Keep traces within a differential pair on the same layer of the PCB to minimize impedance discontinuities and keep them coupled.
- The trace length differences that are due to signals being located on different ball rows are not compensated inside the package. Therefore, do not add external compensation on the board. That is, when routing from the ball to the edge of package (timing reference point), route directly from the ball to the package edge. As a result, compensation is obtained at the edge. That is, the negative part of a differential pair will be 1 mm longer than the positive part when routing from the ball to the package edge.
- If dynamic deskew is not used, the egress data path signals TCtrlX (X denotes both TCtrlP and TCtrlN, a differential pair) and TD[15:0]X must have matching electrical lengths to the timing reference point from the driver. You must ensure that TDClkX has a correct phase relationship to the other egress signals, either by adding trace delay relative to these or phase shifting the clock (by 90^{o}) at the source. If dynamic deskew is used, the maximum skew between differential pairs in the egress data path must be less than one bit time (1.25 ns for a 400 MHz clock frequency). To ease the designers task, the SPI-4.2 data channel input timing is specified with setup and hold times in Figure 56, page 236 and Table 292, page 236.
- The egress status channel signals TStat[1:0] must have matching electrical lengths. The signals should be source-terminated with 27 Ω when used with a 50 Ω transmission line. The Schaumburg device can phase shift the TSClk signal by 180^{o} (INVERTCLK bit in the SPI4_EGR_SETUP0 register). No trace delay is necessary to ensure a correct phase relationship to the other egress status signals. TSClk is either 75 MHz or TDClk / 4.
- If dynamic deskew is not used, the ingress data path signals RDClkX, RD[15:0]X, and RCtrlX should have matching electrical lengths from the timing reference point to the receiver. The Schaumburg device can phase shift the RDClkX signals by 90^{o} (RQC bit in the SPI4_MISC register) so no trace delay is necessary. This ensures correct phase relationship to the other ingress signals. If dynamic deskew is used, the maximum skew between differential pairs in the ingress data path must be less than one bit time (1.25 ns for a 400-MHz clock frequency).
- The ingress status channel signals RStat[1:0] must have matching electrical lengths. To ensure that RSClk has a correct phase relationship to the RStat signals, either add trace delay or phase shift (invert) the clock at its source. The SPI-4.2 status channel input timing is specified in Figure 63, page 240 and Table 296, page 240.

There is an internal termination resistor built into the SPI-4.2 differential inputs with a nominal value of 100 Ω.

If the SPI-4.2 interface is not used, the SPI_En signal should be pulled low, which disables the interface. RSClk and RStat[1:0] should be pulled high or low. All other signals can be left floating.

8.3.5 PI

The Parallel CPU Interface is 16-bit wide, and PI_Addr0 (LSB) selects a 16-bit half-word. If the PI is connected to a system bus that selects bytes with A0 (LSB), the PI_Addr0 should be connected to A1 of the system bus.

If the parallel CPU interface is not used, all input signals should be pulled high.

8.3.6 SI

The SI bus consists of the SI_Clk clock signal, the SI_DO and SI_DI data signals, and the SI_nEN device select signal.

When routing the SI_Clk signal, be sure to create clean edges. If the SI bus is connected to more than one slave device, it should be routed in a daisy-chain configuration with no stubs. The SI_Clk signal should be terminated correctly to avoid reflections and double clocking.

If it is not possible (or desirable) to route the bus in a daisy-chain configuration, the SI_Clk signal should be buffered and routed in a star topology from the buffers. Each buffered clock should be terminated at its source.

If the serial CPU interface is not used, all input signals should be pulled high.

8.3.7 JTAG

If the JTAG interface is not used:

- TDI, TMS must be pulled high.
- TDO can be left floating.
- nTRST must be pulled low to reset the JTAG controller. If JTAG chain is used, a pull-down resistor of 2 kΩ maximum should be used. If other JTAG resets must be pulled low in the JTAG chain, a smaller value of the resistor should be used. Consult specific devices for their internal pull resistors.
- TCK should either be pulled high or low.

8.3.8 Miscellaneous

The nReset input can easily be interfaced to a 3.3 V reset signal by using the circuit shown in Figure 80.

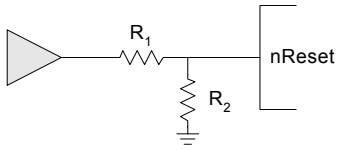


Figure 80. Interfacing nReset to 3.3 V

The resistors should be selected so that the V_{IH}/V_{IL} maximum and minimum conditions are fulfilled and the rise and fall time at the nReset ball is not too large. The recommended values when interfacing to a 3.3 V driver are R_1 = 2.2 kΩ and R_2 = 6.8 kΩ.

9 DESIGN CONSIDERATIONS

9.1 10/100M HDX Occasionally Transmits 36-Bit Jam Pattern

9.1.1 Issue

During UNH FEC MAC compliance testing, four test cases received comments:

- 4.1.1.1a, Collision Detect/Enforcement Test
- 4.1.1.2a, Collision Detection Timing Sensitivity Test
- 4.1.1.2b, Collision Detection Timing Sensitivity Test
- 4.1.1.3a, Late Collision Detection Test

In all four test cases, a 36-bit jam pattern was transmitted occasionally during a collision. The standard requires that the jam pattern be 32 bits. The extended jam pattern (36-bit instead of 32-bit) is transmitted if a collision occurs in an uneven nibble position. The complete UNH FEC MAC test report is available upon request.

9.1.2 Implications

This has no practical significance.

9.1.3 Workaround

No workaround exists.

9.1.4 Status

Currently, there are no plans to make adjustments.

10 ORDERING INFORMATION

The VSC7326 device is available in two package types. VSC7326VV is a 1152-pin, thermally enhanced, plastic ball grid array (PBGA). The device is also available in a lead-free package, VSC7326XVV.

Lead-free products from Vitesse comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

The following table lists the ordering information for the VSC7326 device.

Table 327. Ordering Information

Part Order Number	Description
VSC7326VV	35 × 35 mm, 1152-pin, thermally enhanced PBGA.
VSC7326XVV	Lead-free, 35 × 35 mm, 1152-pin, thermally enhanced PBGA.