# PM7375

LASAR-155™

# PROGRAMMER'S GUIDE

**Preliminary Information**

**Issue 1: March, 1996**

## TABLE OF CONTENTS

# List of Figures

# 1. INTRODUCTION

The objective of the LASAR-155™ Programmer's Guide is to describe the primary functions provided by the LASAR-155 from a software engineer's perspective. This document does not cover all the functions provided by the LASAR-155. Please contact PMC-Sierra for specific functions not covered in this document.

This document is an extension to the LASAR-155 Longform Datasheet. The two documents should be used together to design an ATM Network Interface Card (NIC). In case of a discrepancy between the Programmer's Guide and the Longform Datasheet, the Longform Datasheet shall always be considered correct.

## 1.1 Target Audience

The document has been prepared for readers with some prior knowledge of ATM technology. Introductory materials on ATM technology are readily available from several sources.

Although the examples provided in this document are described in 'C' language syntax, they are not meant as compile-ready code segments.

## 1.2 Numbering Conventions

Unless specified otherwise, the following numbering conventions are used:

| | |
|---|---|
| binary | 001B, 111B |
| decimal | 129, 6, 12 |
| hexadecimal | 0x1FE, 09FH |

## 1.3 Register Overview

Unless specified otherwise, LASAR-155 registers are described using the convention: **REGISTER_NAME(PCI Offset)**.

### Normal Mode Registers

Normal mode registers are used to configure and monitor the operation of the LASAR-155. Normal mode registers can be one of three types:

1) Microprocessor Only Access registers (MO) - these registers can only be accessed through the microprocessor interface and cannot be accessed through the PCI port.
2) PCI Host Only Access registers (PO) - these registers can only be accessed through the PCI Host interface and cannot be accessed directly using the microprocessor interface. However, indirect access via the microprocessor interface is provided.

3) Selectable Master registers (SM) - these registers can be accessed by either the PCI Host or the microprocessor at any given moment as configured by input, MPENB. Input MPENB must be set when the device is powered up.

The PCI Host can detect the presence or absence of the microprocessor by reading the MPEN bit of the **PCID Interrupt Status (0x304)**.

| MPEN | Function | R |
|---|---|---|
| 0 | Microprocessor is not installed. | |
| 1 | Microprocessor is installed. | |

**Notes on Normal Mode Register Bits**

1) Writing values into unused register bits has no effect. However, to ensure software compatibility with future version of the product, unused register bits must be written with logic zero. Reading back unused bits can produce either logic one or a logic zero; hence unused register bits should be masked off by software when read.
2) All configuration bits that can be written into can also be read back. This allows the processor controlling the LASAR-155 to determine the programming state of the block.
3) Writable normal mode register bits are cleared to logic zero upon reset unless otherwise noted.
4) Writing into read-only normal mode register bit locations does not affect LASAR-155 operation unless otherwise noted.
5) Certain register bits are reserved. These bits are associated with megacell functions that are unused in this application. To ensure that the LASAR-155 operates as intended, reserved register bits must only be written with their default values. Similarly, writing to reserved registers should be avoided.

## 1.4 References

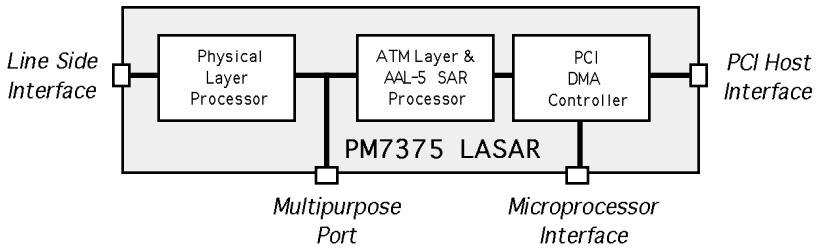

- ATM Forum - ATM User-Network Interface Specification, V3.1, September, 1994.
- ATM Forum - "An ATM PHY Data path Interface - Level 1", V2.0, February 1994
- ITU-TS Recommendation G.709 - "Synchronous Multiplexing Structure", Helsinki, March 1993.
- ITU-TS Recommendation I.363 - "B-ISDN ATM Adaptation Layer (AAL) Specification", Helsinki, March 1993.
- ITU-TS Recommendation I.432 DRAFT - "B-ISDN User-Network Interface-Physical Interface Specification", Helsinki, March 1993.
- ITU-TS Recommendation I.610 - "B-ISDN Operation and Maintenance Principles and Functions", Helsinki, March 1993.
- Bell Communications Research - SONET Transport Systems: Common Generic Criteria, GR-253-CORE, Issue 1, December 1994.

  Bell Communications Research - Broadband-ISDN User to Network Interface and Network Node Interface Physical Layer Generic Criteria, TR-NWT-001112, Issue 1, June 1993.
- Bell Communications Research - Asynchronous Transfer Mode (ATM) and ATM Adaptation Layer (AAL) Protocols Generic Requirements, TA-NWT-001113, Issue 2, July 1993.
- Bell Communications Research - Generic Requirements for Operations of Broadband Switching Systems, TA-NWT-001248, Issue 2, October 1993.
- American National Standard for Telecommunications - B-ISDN ATM Adaptation Layer Type 5, ANSI T1.635-1993.
- T1X1.3/93-006R3, Draft American National Standard for Telecommunications, Synchronous Optical Network (SONET): Jitter at Network Interfaces
- IEEE 1149.1 - Standard Test Access Port and Boundary Scan Architecture, May 21, 1990.
- PCI Special Interest Group, PCI Local Bus Specification, April 30, 1993, Version 2.0.
- PMC-940212, ATM_SCI_PHY, "SATURN Compliant Interface For ATM Devices", February 1994, Issue 1.
- PMC-931127, Local ATM Segmentation and Reassembly & Physical Layer Interface Longform Datasheet, Issue 3, March, 1996.

## 2 LASAR-155 OVERVIEW

The PM7375 LASAR-155 is a highly integrated semiconductor device that is ideally suited for ATM Network Interface Cards. The LASAR-155 combines a Physical Layer Processor, ATM Layer and AAL-5 SAR Processor and a PCI DMA Controller on a single chip to simplify the design, programming and manufacturing of ATM adapters. The major functional blocks of the LASAR-155 are illustrated in Figure 2.1.

Figure 2.1: LASAR-155 Major Functional Blocks.




The Physical Layer (PHY) Processor incorporates the industry standard PMC PM5346 S/UNI-LITE. It provides SONET and SDH interfaces at either SONET STS-3c/STM-1 (155.52 Mbps) or SONET STS-1 (51.84 Mbps) rates and supports control registers and signal pins that are compatible with those of S/UNI-LITE's. On the line side, the PHY Processor supports a direct interface to the line drivers for fiber and UTP-5 cabling and provides on-chip clock recovery and clock synthesis units that are compliant with Bellcore TR-NWT-000253 Issue 2 and ITU-T G.958 jitter requirements. The configuration and monitoring of the PHY Processor is described in "Section 3. Physical Layer Processing Overview".

The AAL-5 SAR Processor supports the simultaneous segmentation and reassembly of 128 open virtual circuits (VCs). The connection parameters for the 128 transmit and 128 receive VCs are maintained in an internal table to simplify the design of the NIC and to sustain a high data throughput rate. For traffic shaping, the LASAR-155 implements a leaky bucket peak cell rate (PCR) enforcement using eight programmable rate queues arranged as a set of four high priority queues and a set of four low priority queues. The PCR can be specified on a per VC basis by associating a transmit VC with one of the eight rate queues and by selecting whether to use 100%, 50%, or 25% of the peak rate provided by the queue. The LASAR-155 also supports the enforcement of sustainable cell rate (SCR) using a token generation mechanism. The configuration and monitoring of the ATM and AAL layer processing is described in "Section 9. ATM & AAL Layer Overview".

The LASAR-155 provides a 32 bit and 33 MHz Peripheral Component Interconnect (PCI) interface that is compliant with the PCI Local Bus Specifications Version 2.0. PCI's high bandwidth and low bus latency features allow packets to be segmented and reassembled in the host memory thereby eliminating the need for expensive local packet memory. The LASAR's PCI DMA Controller supports both bus-master and bus-slave modes. In bus-master mode, the PCI DMA Controller can access the host

memory independent of the host processor. In bus-slave mode, the host processor can access LASAR's registers and communicate with the optional microprocessor via a mailbox mechanism provided by the PCI DMA Controller. The LASAR-155 provides an efficient DMA controller to manage the transfer of packets between LASAR's SAR engine and the host memory. The configuration and monitoring of the PCI DMA Controller is described in "Section 17. PCI DMA Controller Overview".

The LASAR-155 conforms to ATM Forum User-Network Interface (UNI) Specification Version 3.1, Bellcore Standard TA-NWT-001113 and ITU-T Recommendations I.432 and I.363.

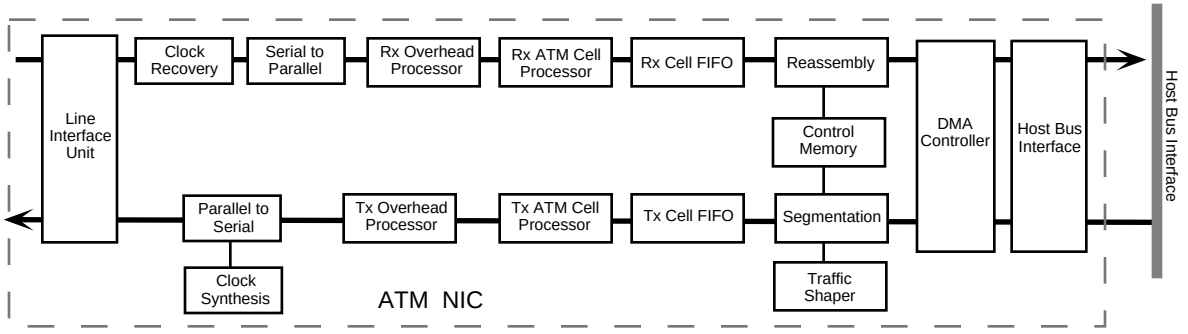
## 2.1 Application Examples

The LASAR-155 can be configured for many application needs. This section highlights how the LASAR-155 can be used in an ATM adapter, in conjunction with an external physical layer processor and in multimedia applications.

### ATM Network Interface Card

An ATM NIC implements PHY, ATM, and AAL protocol layers such that an end system can communicate with an ATM switch. Figure 2.2 illustrates a generic ATM NIC architecture and identifies the functional blocks that must be implemented and integrated to form the basis of the User-Network Interface.

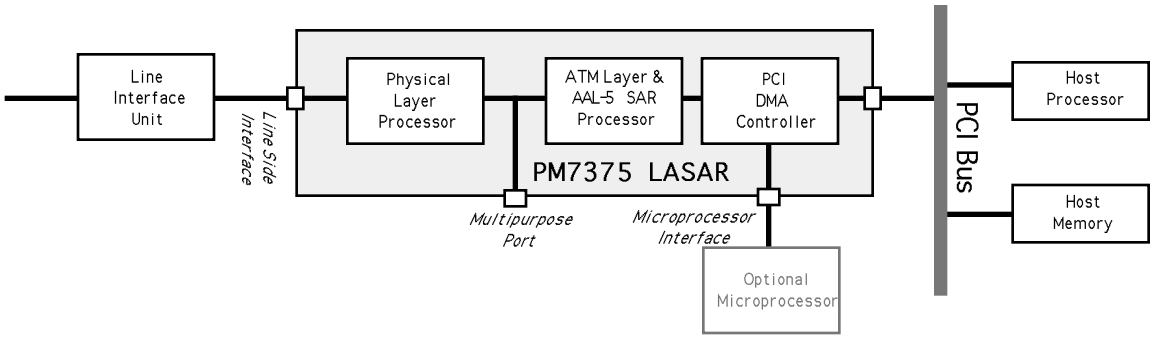
Figure 2.2: A generic ATM NIC architecture.




The LASAR-155 combines all the necessary functions on a single device. On the line side, the LASAR-155 can connect directly to an OC-3 optical driver or to a UTP-5 line interface unit (LIU). On the system side, the LASAR-155 can interface directly to a 32 bit, 33 MHz PCI bus. Figure 2.3 illustrates the design of a SONET STS-1/3c[1] ATM NIC for the PCI bus.

[1] Unless specified otherwise, the term "SONET" is used to refer to both SONET and SDH and "STS-1/3c" is used to refer to STS-1 and STS-3C/STM-1 in the remainder of this document.

Figure 2.3: SONET STS-3c/STS-1 ATM NIC example using the LASAR-155.



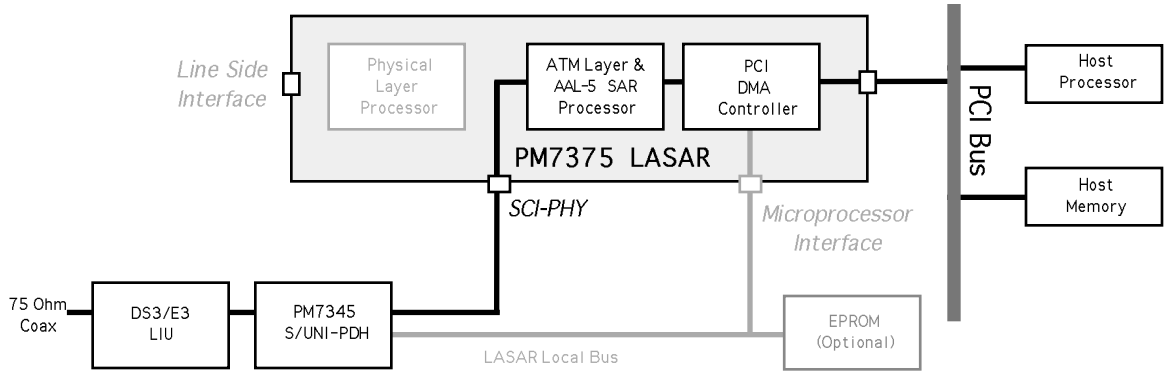

The initial configuration, ongoing control and monitoring of the LASAR-155 can be performed by a PCI Host, an optional microprocessor, or the combination of both. The high bandwidth and low bus latency features offered by the PCI bus allow the segmentation and reassembly of packets in the host memory. In the transmit direction, the LASAR-155 provides an 8 cell Service Data Unit (SDU) FIFO and in the receive direction a 96 cell SDU FIFO to compensate for the bus latency during packet transfer across the PCI bus. The transfer of packets between the LASAR-155 and the host memory is facilitated by the PCI DMA Controller independent of the PCI Host.

## Interface to External Physical Layer Processor

The LASAR's Multipurpose Port segments the ATM Layer & AAL-5 SAR Processor and PCI DMA Controller from the internal Physical Layer Processor. For applications that require transmission technologies other than SONET STS-1/3c, the Multipurpose Port can be configured as an 8 bit SCI-PHY™ compliant interface to connect to an external physical layer processor.

Figure 2.4 provides an example where the LASAR-155 is connected to the PM7345 S/UNI-PDH to provide a DS3/E3 User-Network Interface. The LASAR-155 can access external devices when the Microprocessor Interface is configured for master mode operation. In this mode, the PCI Host configures, controls, and monitors the LASAR-155 and the external devices attached to the Microprocessor Interface.

Figure 2.4: DS3/E3 ATM UNI Example.

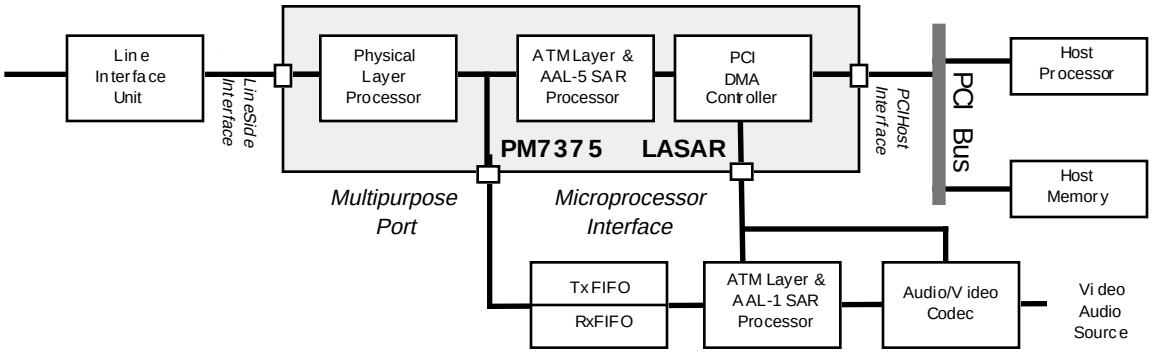



While providing the ability to extend to external physical layer processors, the LASAR-155 allows the designer to reuse the same device driver that interfaces to upper layer programs such as Signaling , LAN Emulation, and Network Management.

## Multimedia Applications

The LASAR-155 can be combined with an external AAL-1 SAR processor for certain video and multimedia applications. The external AAL-1 SAR processor can share the SONET framer with the internal AAL-5 SAR processor. The Multipurpose Port can be connected to an external FIFO to support the insertion and extraction of CBR cells that carry encoded video and/or audio signals. The Microprocessor Interface can be used to control the AAL-1 processor and the Audio/Video Codec as illustrated in Figure 2.5.

Figure 2.5: Multimedia Terminal Example.




In the receive direction, the LASAR-155 supports cell copying and cell filtering operations. Cell copying is the action of directing cells from the Physical Layer Processor through the Multipurpose Port to the external FIFO. Cell filtering is the action of dropping cells destined for the PCI Host. The LASAR-155 can be configured to detect and direct CBR cells to an FIFO where they can be processed by the AAL-1 SAR and the Audio/Video Codec.

In the transmit direction, the Multipurpose Port allows the insertion of CBR cells into an aggregate cell stream destined for the Physical Layer Processor. The AAL-1 SAR Processor is responsible for implementing a traffic shaper for each CBR VC. The LASAR-155 can optionally shape the aggregate cell traffic inserted into the Multipurpose Port. The aggregate PCR is configurable to control the cell traffic from 32 Kbps to the full cell rate of 149 Mbps.

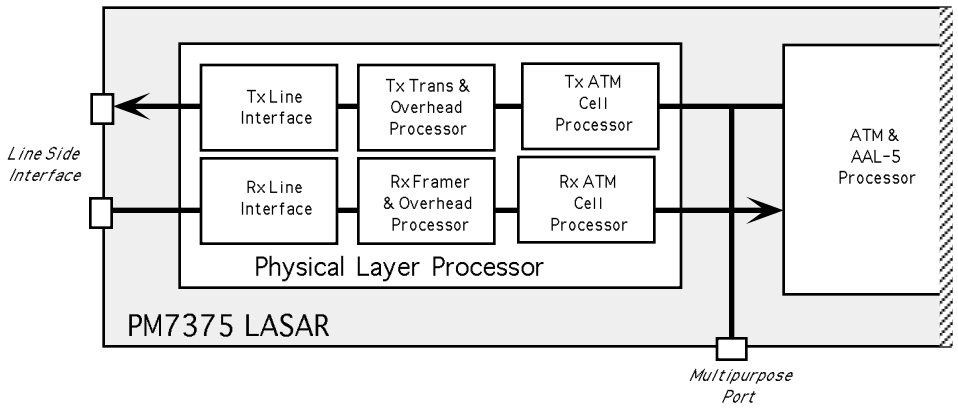
# 3. PHYSICAL LAYER PROCESSING OVERVIEW

This section describes the functional blocks of the Physical Layer Processor and provides an overview of the basic SONET frame format.

## 3.1 Physical Layer Processor Overview

The Physical Layer Processor performs the core of SONET STS-1/3c physical layer functions as illustrated in Figure 3.1:

Figure 3.1: Physical Layer Processor Functional Blocks




### Rx Line Interface

The Rx Line Interface block consists of a clock recovery unit (CRU) and a serial to parallel converter (SIPO). The CRU recovers the clock from the incoming bit serial data stream and is compliant with SONET jitter requirements. The SIPO converts the received bit serial SONET stream to a byte serial stream, searches for the SONET framing pattern (A1, A2) in the incoming stream and performs serial to parallel conversion on octet boundaries.

The configuration and monitoring of Rx Line Interface is described in "Section 4. Line Interface Controls".

### Rx Framer and Overhead Processor

The Rx Framer and Overhead Processor frames to an incoming SONET STS-1/3c stream and performs all the SONET section, line and path overhead processing. It performs framing (A1, A2), descrambling, detects alarm conditions, and monitors section, line and path bit interleaved parity (B1, B2, B3), accumulating error counts at each level for performance monitoring purposes. Line and path far end block error indications (Z2, G1) are also accumulated. The LASAR interprets the received payload pointers (H1, H2) and extracts the synchronous payload envelope (SPE) which carries the received ATM cell payloads.

The configuration and monitoring of Rx Framer and Overhead Processor is described in "Section 5. Receive SONET Overhead Processing".

### Rx ATM Cell Processor

The Rx ATM Cell Processor receives the SPE from the Rx Framer and Overhead Processor and performs the following functions.

- ATM cell delineation
- Cell filtering based on idle/unassigned cell detection
- cell filtering based on the Header Error Code (HEC) error detection and single bit correction
- Generic Flow Control (GFC) field extraction
- ATM layer alarm detection (OCD, LCD) and performs
- ATM cell payload descrambling
- Accumulate the number of received assigned cells

The configuration and monitoring of Rx ATM Cell Processor is described in "Section 10. Receive ATM Layer Processing".

### Tx ATM Cell Processor

In the transmit direction, the Transmit ATM Cell Processor inserts ATM cells into SONET SPEs. The GFC bits may optionally be inserted using a dedicated serial port. The HEC is automatically calculated and inserted. The cell payload is optionally scrambled. Generated transmit cells are automatically inserted into SONET STS-1/3c SPEs. In the absence of transmit cells, the LASAR automatically inserts idle/unassigned cells into the SPE.

The configuration and monitoring of Tx ATM Cell Processor is described in "Section 11. Transmit ATM Layer Processing".

### Tx Transmit and Overhead Processor

The Tx Transmit and Overhead Processor generates the payload pointer (H1, H2) and inserts the SPE which carries the ATM cell payload. In addition, it formats SONET section, line, and path overhead appropriately. It performs framing pattern insertion (A1, A2), scrambling, alarm signal insertion, and creates section, line and path bit interleaved parity (B1, B2, B3) as required to allow performance monitoring at the far end. Line and path far end block error indications (Z2, G1) are also inserted.

The configuration and monitoring of Rx Transmit and Overhead Processor is described in "Section 6. Transmit SONET Overhead Processing".

## Tx Line Interface

The Tx Line Interface performs clock synthesis and performs parallel to serial conversion. The transmit clock may be synthesized from either a 19.44 MHz or a 6.48 MHz reference. The phase lock loop filter transfer function is optimized to enable the PLL to track the reference, yet attenuate high frequency jitter on the reference clock.

The configuration and monitoring of Tx Line Interface is described in "Section 4. Line Interface Controls".

# 4 LINE INTERFACE CONTROLS

This section describes the configuration and monitoring of the Rx Line Interface and Tx Line Interface units.

## 4.1 Line Rate Selection

The line rate is selected by Rate[1:0] bits of **LASAR-155 Master Configuration (0x004)**. The device will not operate correctly if a Reserved mode is selected.

| RATE[1:0] | Function | R/W |
|---|---|---|
| 00B | Reserved | |
| 01B | Reserved | |
| 10B | Selects SONET STS-1 rate. | |
| 11B | Selects SONET STS-3c rate. | default |

## 4.2 SONET Hierarchy Selection

The STS1 bit of **LASAR-155 Master Configuration (0x004)** selects device operation using either a SONET STS-1 or a STS-3c Transmission Convergence (TC) sublayer type.

| STS | Function | |
|---|---|---|
| 0 | Expect SONET STS-3c TC sublayer type. | default |
| 1 | Expect SONET STS-1 TC sublayer type. | |

## 4.3 Clock Recovery Unit

The Rx Line Interface block consists of a clock recovery unit (CRU) and a serial to parallel converter (SIPO). The CRU recovers the clock from the incoming bit serial data stream using a reference frequency.

The RREFSEL bit of **LASAR-155 Clock Recovery Control and Status (0x01C)** determines the expected frequency of input signals RRCLK+/-.

| RREFSEL | Function | R/W |
|---|---|---|
| 0 | Expect a 19.44 MHz reference frequency for SONET STS-3c operation. | default |
| 1 | Expect a 6.48 MHz reference frequency for SONET STS-1 operation. | |

The RXDINV bit of **LASAR-155 Master Configuration (0x004)** selects the active polarity of the receive differential data inputs (RXD+, RXD-).

| RXDINV | Function | R/W |
|---|---|---|
| 0 | RXD+ is active high and RXD- is active low. | default |
| 1 | RXD+ is active low and RXD- is active high. | |

The current state of the clock recovery unit can be determined by reading the RDOOLV bit of **LASAR-155 Clock Recovery Control and Status (0x01C).** A RDOOLV change of state can trigger an interrupt on RDOOLI bit of **LASAR-155 Master Interrupt Status (0x008)** if it is enabled by the RDOOLE bit of **LASAR-155 Master Interrupt Enable (0x00C)**.

| RDOOLV | Function | R |
|---|---|---|
| 0 | The receive PLL is recovering the clock successfully. | |
| 1 | Indicate a receive data out of lock status. The divided down recovered clock frequency is not within 488 ppm of the RRCLK+/- frequency or if no transitions have occurred on the RXD+/- inputs for more than 80 bit periods. | |

## 4.4 Clock Synthesis Unit

The transmit clock may be synthesized from a either a 19.44 MHz or 6.48 MHz reference. The TREFSEL bit of **LASAR-155 Clock Synthesis Control and Status (0x018)** determines the expected frequency of TRCLK+/-.

| TREFSEL | Function | R/W |
|---|---|---|
| 0 | The correct line clock frequency is synthesized if the reference frequency is 19.44 MHz (SONET STS-3c). | default |
| 1 | The correct line clock frequency is synthesized if the reference frequency is 6.48 MHz (SONET STS-1). | |

The state of the clock synthesis phase locked loop can be determined by reading the TROOLV bit of **LASAR-155 Clock Synthesis Control and Status (0x018).** A TROOLV change of state can trigger an interrupt on the TROOLI bit of **LASAR-155 Master Interrupt Status (0x008)** if it is enabled by the TROOLE bit of **LASAR-155 Master Interrupt Enable (0x00C)**.

| TROOLV | Function | R |
|---|---|---|
| 0 | The correct line clock frequency is synthesized. | |
| 1 | Indicate a transmit reference out of lock status. The divided down synthesized clock frequency is not within 488 ppm of the TRCLK+/- frequency. | |

## 4.5 Recommended Procedure

The following procedures can be used to select either a SONET STS-3c or SONET STS-1 rate. The SONET rate is selected at system startup and should not be changed for the duration of the system operation.

### SONET STS-3c Rate

To select the SONET STS-3c rate, the reference frequency for TRCLK+/- and RRCLK+/- must be 19.44 MHz. The appropriate register bits must be configured as described below. (Note that this is the default configuration after a reset operation.)

| Bit | Register Name & Offset | Value |
|---|---|---|
| STS1 | LASAR-155 Master Configuration (0x004) . | 0 |
| RATE[1:0] | LASAR-155 Master Configuration (0x004) . | 11B |
| RREFSEL | LASAR-155 Clock Recovery Control and Status (0x01C). | 0 |
| TREFSEL | LASAR-155 Clock Synthesis Control and Status (0x018). | 0 |

- Check the correct polarity is selected for RXD+/- inputs.
- Enable the appropriate interrupts for error handling.

### SONET STS-1 Rate

To select the SONET STS-1 rate, the reference frequency for TRCLK+/- and RRCLK+/- must be 6.48 MHz. The appropriate register bits must be configured as:

| Bit | Register Name & Offset | Value |
|---|---|---|
| STS1 | LASAR-155 Master Configuration (0x004). | 1 |
| RATE[1:0] | LASAR-155 Master Configuration (0x004). | 10B |
| RREFSEL | LASAR-155 Clock Recovery Control and Status (0x01C). | 1 |
| TREFSEL | LASAR-155 Clock Synthesis Control and Status (0x018). | 1 |

- Select the correct polarity for RXD+/- inputs.
- Enable the appropriate interrupts for error handling.

# 5 RECEIVE SONET OVERHEAD PROCESSING

The Rx Framer and Overhead Processor aligns to an incoming SONET STS-1/3c stream and performs all the SONET section, line and path overhead processing. Section, line and path processing are performed using the Receive Section Overhead, Receive Line Overhead and the Receive Path Overhead Processors.

## 5.1 Receive Section Overhead Processor

The Receive Section Overhead Processor (RSOP) performs the following functions:

- Frame synchronization
- Section level alarm detection (LOS, OOF, LOF)
- Performance monitoring

### Framing Algorithm Selection

The ALGO2 bit of **RSOP Control/Interrupt Enable (0x040)** controls the framing algorithm used to determine and maintain the frame alignment.

| ALGO2 | Function | R/W |
|---|---|---|
| 0 | The framer uses the first of the framing algorithms where all the A1 framing bytes and all the A2 framing bytes are examined. | default |
| 1 | The framer uses the second of the framing algorithms where only the first A1 framing byte and the first 4 bits of the first A2 framing byte (12 bits in total) are examined. All other framing bits are ignored. | |

### Out of Frame

While in-frame, the framing bytes in each frame are compared against the expected pattern. Out-of-frame is declared when four consecutive frames containing one or more framing pattern errors have been received.

The frame alignment can be determined by reading the OOFV bit of the **RSOP Status/Interrupt Status (0x044)**. A change in OOFV state can trigger an interrupt on OOFI bit of **RSOP Status/Interrupt Status (0x044)** if it is enabled by the OOFE bit of **RSOP Control/Interrupt Enable (0x040)**.

| OOFV | Function | R |
|---|---|---|
| 0 | In-frame state. | |
| 1 | An out-of-frame state. | |

### Loss of Frame

A loss of frame (LOF) is declared when an out-of-frame (OOF) condition persists for 3 ms. LOF is cleared when an in-frame condition persists for a period of 3 ms.

The LOF state can be determined by reading the LOFV bit of **RSOP Status/Interrupt Status (0x044)**. A change in LOF state can trigger an interrupt on the LOFI bit of **RSOP Status/Interrupt Status (0x044)** if it is enabled by the LOFE bit of **RSOP Control/Interrupt Enable (0x040)**.

| LOFV | Function | R |
|---|---|---|
| 0 | In-frame state. | |
| 1 | A loss-of-frame state. | |

## Loss of Signal

A loss of signal (LOS) condition is declared when 20 ± 3 µs of all zeros pattern is detected. LOS is cleared when two valid framing words are detected and during the intervening time, no LOS condition is detected.

The LOS state can be determined by reading the LOSV bit of **RSOP Status/Interrupt Status (0x044)**. The change of LOS state can trigger an interrupt on the LOSI bit of **RSOP Status Interrupt Status (0x044)** if it is enabled by the LOSE bit of the **RSOP Control/Interrupt Enable (0x040)**.

| LOSV | Function | R |
|---|---|---|
| 0 | Signal is valid. | |
| 1 | RSOP has declared a loss of signal. | |

## BIP-8 Errors

Section BIP-8 (B1) calculation and verification is automatically performed with errors accumulated in **RSOP Section BIP-8 LSB (0x048)** and **RSOP Section BIP-8 MSB (0x04C)** registers.

A section layer (B1) bit error can generate an interrupt on the BIPEI bit of **RSOP Status/Interrupt Status (0x044)** if it is enabled by the BIPEE bit of **RSOP Control/Interrupt Enable (0x040)**

## 5.2 Receive Line Overhead Processor

The Receive Line Overhead Processor (RLOP) performs the following functions:

- Line level alarm detection (line FERF, line AIS)
- Performance monitoring.

### Line Far End Receive Failure

Line Far End Receive Failure (FERF) is declared when a 110B pattern is detected in bits 6, 7, and 8 of the K2 byte, for five consecutive frames. FERF is removed when any pattern other than 110B is detected in bits 6, 7, and 8 of the K2 byte for five consecutive frames.

The FERF state can be determined by reading the FERFV bit of **RLOP Control/Status (0x060)**. The change of FERF state can trigger an interrupt on the FERFI bit of **RLOP Interrupt Enable/Status (0x064)** if it is enabled by the FERFE bit of **RLOP Interrupt Enable/Status (0x064)**.

| FERFV | Function | R |
|---|---|---|
| 0 | No FERF is reported. | |
| 1 | Line Far End Receive Failure is declared. | |

### Line Alarm Indication Signal (AIS)

Line Alarm Indication Signal (AIS) is declared when a 111B pattern is detected in bits 6,7,8 of the K2 byte, for five consecutive frames. LAIS is removed when any pattern other than 111B is detected in bits 6, 7, and 8 of the K2 byte for five consecutive frames.

The Line AIS state can be determined by reading the LAISV bit of **RLOP Control/Status (0x060)**. The change of LAISV state can trigger an interrupt on the LAISI bit of **RLOP Interrupt Enable/Status (0x064)** if it is enabled by the LAISE bit of **RLOP Interrupt Enable/Status (0x064)**.

| LAISV | Function | R |
|---|---|---|
| 0 | No Line AIS is reported. | |
| 1 | Line AIS is reported. | |

### Line BIP-8/24

Line BIP-8/24 calculation and verification is automatically performed with errors accumulated into **RLOP Line BIP-8/24 LSB (0x068)**, **RLOP Line BIP-8/24 (0x06C)** and **RLOP Line BIP-8/24 MSB (0x070)** registers. Refer to "Section 15. Counter Overview" for additional information.

A line BIP-8/24 error can generate an interrupt on the BIPEI bit of **RLOP Interrupt Enable/Status (0x064)** if it is enabled by the BIPEE bit of **RLOP Interrupt Enable/Status (0x064)**.

### Far End Block Error

The number of line FEBE errors (Z2) is maintained in **RLOP Line FEBE LSB (0x074)**, **RLOP Line FEBE (0x078)** and **RLOP Line FEBE MSB (0x07C)** registers. Refer to "Section 15. Counter Overview" for additional information.

A line FEBE error can generate an interrupt on the FEBEI bit of **RLOP Interrupt Enable/Status (0x064)** if it is enabled by the FEBEE bit of **RLOP Interrupt Enable/Status (0x064)**.

## 5.3 Receive Path Overhead Processor

The Receive Path Overhead Processor (RPOP) performs the following functions:

- Pointer interpretation
- Extraction of path overhead
- Extraction of the synchronous payload envelope
- Path level alarm detection (LOP, path AIS, RDI)
- Performance monitoring.

### Loss of Pointer

The RPOP interprets the incoming pointer (H1, H2) in accordance with both Bellcore and ETSI standards.

Loss of pointer (LOP) in the incoming STS-1/3c stream is declared as a result of eight consecutive invalid pointers or eight consecutive NDF enabled indications. LOP is removed when the same valid pointer with normal NDF is detected for three consecutive frames.

The LOP state can be determined by reading the LOP bit of **RPOP Status/Control (0x0C0)**. A change in the LOP state can trigger an interrupt on the LOPI bit of **RPOP Interrupt Status (0x0C4)** if it is enabled by the LOPE bit of **RPOP Interrupt Enable (0x0CC)**.

| LOP | Function | R |
|---|---|---|
| 0 | Not in LOP state. | |
| 1 | In LOP state. | |

### Path Alarm Indication Signal

Path Alarm Indication Signal (PAIS) in the incoming STS-1/3c stream is declared after three consecutive AIS indications. PAIS is removed when the same valid pointer with normal NDF is detected for three consecutive frames or when a valid pointer with NDF enabled is detected.

The path AIS state can be determined by reading the PAIS bit of **RPOP Status/Control (0x0C0)**. A change in the PAIS state will trigger an interrupt on the PAISI bit of **RPOP Interrupt Status (0x0C4)** if it is enabled by the PAISE bit of **RPOP Interrupt Enable (0x0CC)**.

| PAIS | Function | R |
|---|---|---|
| 0 | No PAIS is reported. | |
| 1 | PAIS is reported. | |

### Path Remote Defect Indication

Path Remote Defect Indication (RDI) is raise when bit 5 of the path G1 byte is set high for five consecutive frames. Path RDI is cleared when bit 5 is low for five consecutive frames.

The PRDI state can be determined by reading the PRDI bit of **RPOP Status/Control (0x0C0)**. A change in the PRDI state can trigger an interrupt on the PRDII bit of **RPOP Interrupt Status (0x0C4)** if it is enabled by the PRDIE bit of **RPOP Interrupt Enable (0x0CC)**.

| PRDI | Function | R |
|---|---|---|
| 0 | No PRDI is reported. | |
| 1 | PRDI is reported. | |

### Path BIP-8 Error (B3)

The number of path BIP-8 errors (B3) is maintained in **RPOP Path BIP-8 LSB (0x0E0)** and **RPOP Path BIP-8 MSB (0x0E4)** registers. Refer to "Section 15. Counter Overview" for additional information.

A path BIP-8 error can generate an interrupt on the BIPEI bit of **RPOP Interrupt Status (0x0C4)** if it is enabled by the BIPEE bit of **RPOP Interrupt Enable (0x0CC)**.

### Path FEBE (G1)

The number of path FEBE errors (G1) is maintained in **RPOP Path FEBE LSB (0x0E8)** and **RPOP Path FEBE MSB (0x0EC)** registers. Refer to "Section 15. Counter Overview" for additional information.

A Path FEBE error can generate an interrupt on the FEBEI bit of **RPOP Interrupt Status (0x0C4)** if it is enabled by the FEBEE bit of **RPOP Interrupt Enable (0x0CC)**.

# 6 TRANSMIT SONET OVERHEAD PROCESSING

The Transmit Transmitter and Overhead Processor block inserts all path, line and section overhead bytes in the outgoing SONET STS-1/3c stream. Section, line and path overhead processing is performed using the Transmit Section Overhead, Transmit Line Overhead and the Transmit Path Overhead Processors as described below.

## 6.1 Transmit Section Overhead Processor

The Transmit Section Overhead Processor (TSOP) performs the following functions:

- Frame pattern insertion (A1, A2)
- Section BIP-8 (B1) insertion
- Section level alarm signal insertion

### BIP-8 Calculation

The section BIP-8 code is based on a bit interleaved parity calculation using even parity calculated over all SONET frame bytes. The calculated BIP-8 code is inserted into the B1 byte of the SONET frame.

## 6.2 Transmit Line Overhead Processor

The Transmit Line Overhead Processor (TLOP) performs the following functions:

- Line level alarm signal insertion
- Line BIP-8/24 insertion (B2)

### Line Alarm Generation

The AUTOFEBE bit of **LASAR-155 Master Configuration (0x004)** determines whether line and path far end block errors are sent upon detection of an incoming line and path BIP error event.

| AUTOFEBE | Function | R/W |
|---|---|---|
| 0 | Line or Path BIP errors do not generate FEBE events. | |
| 1 | One line or path FEBE is inserted for each line or path BIP error event. | default |

The AUTOLRDI bit of **LASAR-155 Master Configuration (0x004)** determines whether the line Remote Defect Indication (RDI) is sent immediately upon the detection of an incoming alarm.

| AUTOLRDI | Function | R/W |
|---|---|---|
| 0 | Line RDI is not generated. | |
| 1 | Line RDI is inserted immediately upon declaration of loss of signal (LOS), loss of frame (LOF) or line AIS. | default |

### BIP-8/24 Calculation

The line BIP-8/24 code is based on a bit interleaved parity calculation using even parity calculated over all SONET frame bytes except the section overhead bytes. The calculated BIP-8/24 code is inserted into the B2 byte(s) of the SONET frame.

## 6.3 Transmit Path Overhead Processor

The Transmit Path Overhead Processor (TPOP) performs the following functions:

- Transport frame alignment generation
- Pointer generation (H1, H2)
- Path overhead insertion
- Synchronous payload envelope insertion
- Path BIP-8 (B3) insertion

### Path Alarm Generation

The AUTOPRDI bit of **LASAR-155 Master Configuration (0x004)** determines whether path Remote Defect Indication (RDI) is sent immediately upon the detection of an incoming alarm.

| AUTOPRDI | Function | R/W |
|---|---|---|
| 0 | STS path RDI event are not generated. | |
| 1 | STS path RDI is inserted immediately upon declaration of loss of signal (LOS), loss of frame (LOF), line AIS, loss of pointer (LOP) or STS path AIS. | default |

The AUTOFEBE bit of **LASAR-155 Master Configuration (0x004)** determines whether line and path far end block errors are sent upon detection of an incoming line and path BIP error event.

| AUTOFEBE | Function | R/W |
|---|---|---|
| 0 | Line or Path BIP errors do not generate FEBE events. | |
| 1 | One line or path FEBE is inserted for each line or path BIP error event. | default |

### Pointer Adjustment

The FIXPTR bit of **LASAR-155 Master Control (0x014)** controls whether the transmit payload pointer adjustment is enabled or disabled.

| TPOP | Function | R/W |
|---|---|---|
| 0 | Pointer adjustment is enabled. The payload pointer is controlled by the contents of the **TPOP Pointer Control (0x104)** register. Please refer to the LASAR-155 Longform Datasheet for additional information. | |
| 1 | Transmit payload is set at 522, the byte after the C1 byte. | default |

## Path BIP-8 (B3) Calculation

The path BIP-8 code is based on a bit interleaved parity calculation using even parity calculated over all SONET synchronous payload envelope (SPE) bytes. The calculated BIP-8 code is inserted into the B3 byte of the following frame.

# 7 RECEIVE TRANSMISSION CONVERGENCE SUBLAYER PROCESSING

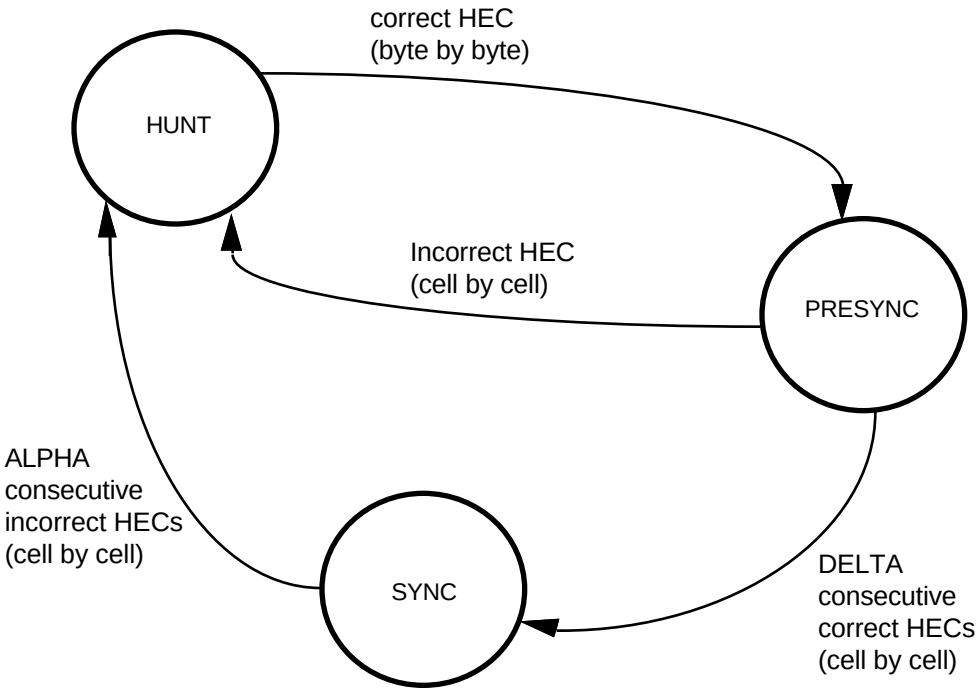
The Receive ATM Cell Processor (RACP) performs the following functions:

- ATM Cell Delineation
- Idle/Unassigned Cell Filtering
- HEC Verification
- GFC Extraction

## 7.1 ATM Cell Delineation

Cell Delineation is the process of framing to ATM cell boundaries using the header error code (HEC) field found in the cell header. The HEC is a CRC-8 calculation over the first 4 octets of the ATM cell header. Cell delineation is performed using the Cell Delineation State Diagram as illustrated below. ALPHA is chosen to be 7 and DELTA is chosen to be 6. These values result in a maximum average time to delineate of 31 μs for SONET STS-3c and 93 μs for SONET STS-1.

Figure 7.1: Cell Delineation State Diagram



## Out of Cell Delineation

The out of cell delineation state can be determined by reading the OCDV bit of **RACP Control/Status (0x140)**. A change in the cell delineation state machine can trigger an interrupt on the OCDI bit of **RACP Interrupt Enable/Status (0x144)**. The OCDI bit is set high when the RACP block transitions from the PRESYNC state to the SYNC state or from the SYNC state to the HUNT state. This interrupt is enabled by the OCDE bit of **RACP Interrupt Enable/Status (0x144)**.

| OCDV | Function | R |
|---|---|---|
| 0 | The cell delineation state machine is in 'SYNC' state and cells are passed to the RALP block. The number of cells passed is maintained in an internal counter. | |
| 1 | The cell delineation state machine is in the 'HUNT' or 'PRESYNC' states. The RACP is hunting for the cell boundaries in the Synchronous Payload Envelope. | |

## Loss of Cell Delineation

The loss of cell delineation state can be determined by reading the LCDV bit of **LASAR-155 Master Control (0x014)**. A change in the loss of cell delineation state can trigger an interrupt on the LCDI interrupt bit of **LASAR-155 Master Interrupt Status (0x008)**. This interrupt is enabled by the LCDE bit of **LASAR-155 Master Interrupt Enable (0x00C)**.

| LCDV | Function | R |
|---|---|---|
| 0 | The SYNC state has been maintained for at least 4 ms. | |
| 1 | An out of cell delineation (OCD) state has persisted for 4 ms without any lower level alarms (LOS, LOP, Path AIS, Line AIS) occurring. | |

## Payload Descrambling

The self synchronous descrambler operates on the 48 byte cell payload using the $x^{43}+1$ polynomial. The descrambler is disabled for the duration of the header and HEC fields, and may optionally be disabled by the DDSCR bit of **RACP Control/Status (0x140)**.

| DDSCR | Function | R/W |
|---|---|---|
| 0 | The cell payload descrambling is enabled. | default |
| 1 | The cell payload descrambling is disabled. | |

## 7.2 Idle/Unassigned Cell Filtering

A cell can be dropped if it matches a predefined cell header pattern. This function is enabled by the PASS bit of the **RACP Control/Status (0x140)**.

| PASS | Function | R/W |
|---|---|---|
| 0 | All cells matching the cell filter criteria are dropped. | default |
| 1 | The match header pattern registers are ignored and filtering of cells with VPI and VCI fields set to 0 is not performed. | |

The matching pattern for the CLP, PTI and GFC fields of ATM cell header is specified in **RACP Match Header Pattern (0x148)**. The corresponding comparison mask is defined in **RACP Match Header Mask (0x14C)**.

## 7.3 HEC Verification

The Header Error Check (HEC) field of the ATM cell header detects and corrects errors in the header. The HEC is a CRC-8 calculation over the first 4 octets of the ATM cell header using the polynomial, $x^8+x^2+x+1$. The HEC error correction algorithm can be enabled or disabled by the DISCOR bit of **RACP Control/Status (0x140)**.

| DISCOR | Function | R/W |
|---|---|---|
| 0 | The HEC error correction algorithm is enabled. Single bit errors detected in the cell header are corrected. | default |
| 1 | The HEC error correction algorithm is disabled. Any error detected in the cell header is treated as an uncorrectable HEC error. | |

If the HEC error correction algorithm is enabled, the HECADD bit of **RACP Control/Status (0x140)** controls whether to add the coset polynomial, $x^6+x^4+x^2+1$ to the received HEC octet before comparing with the calculated result.

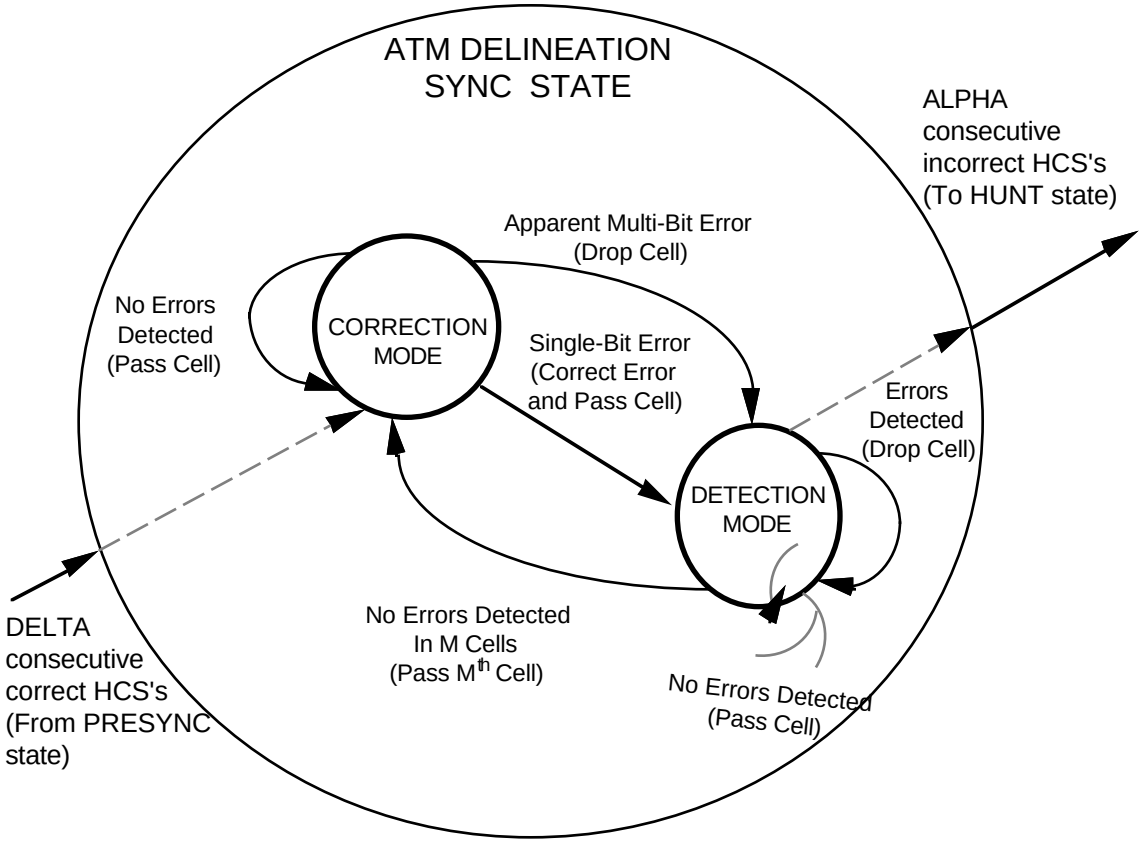
| HECADD | Function | R/W |
|---|---|---|
| 0 | The coset polynomial is not added to the received HEC octet before comparison. | |
| 1 | The coset polynomial is added to the received HEC octet before comparison. | default |

While the Cell Delineation State Machine is in the SYNC state, the HEC verification circuit implements the state machine shown in Figure 7.2. In normal operation, the HEC verification state machine remains in the 'Correction Mode' state. Incoming cells containing no HEC errors are passed for further ATM/AAL layer processing. In addition, incoming cells with single-bit errors are corrected and the resulting cells are passed for further ATM/AAL layer processing. Upon detection of a single- bit error or a multi-bit error, the state machine transitions to the 'Detection Mode' state. In this state, programmable HEC error filtering is provided. The detection of any HEC error causes the corresponding cell to be dropped. The state machine transitions back to

the 'Correction Mode' state when M error free cells are received. The M$^{th}$ cell is not discarded. The cell acceptance threshold (M) is specified in HECFTR[1:0] bits of **RACP Configuration (0x164)**.

| HECFTR[:0] | Function | R/W |
|---|---|---|
| 00B | One ATM cell with correct HEC before resumption of cell acceptance. This cell is accepted. | default |
| 01B | Two ATM cell with correct HEC before resumption of cell acceptance. The last cell is accepted. | |
| 10B | Four ATM cell with correct HEC before resumption of cell acceptance. The last cell is accepted. | |
| 11B | Eight ATM cell with correct HEC before resumption of cell acceptance. The last cell is accepted. | |

Figure 7.2: HEC Verification State Diagram




The number of correctable HEC errors are maintained in the **RACP Correctable HEC Error Count (0x150)**. The number of uncorrectable HEC errors are maintained in the **RACP Uncorrectable HEC Error Count (0x154)**. Counters are enabled only when the cell delineation state machine is in the SYNC state.

The detection of a correctable HEC error can generate an interrupt on the CHECI bit of **RACP Interrupt Enable/Status (0x144)** if it is enabled by the HECE bit of **RACP Interrupt Enable/Status (0x144)**. The detection of a uncorrectable HEC error can generate an interrupt on the UHECI bit of **RACP Interrupt Enable/Status (0x144)** if it is enabled by the HECE bit of **RACP Interrupt Enable/Status (0x144)**.

Dropping of uncorrectable HEC error is controlled by the HECPASS bit of **RACP Control/Status (0x140)**. Regardless of the programming of this bit, cells are always dropped while the cell delineation state machine is in the 'HUNT' or 'PRESYNC' states.

| HECPASS | Function | R/W |
|---|---|---|
| 0 | Cell containing an uncorrectable HEC error are dropped. | default |
| 1 | Cells are passed to the RALP block regardless of errors detected. | |

## 7.4 GFC Extraction

The received GFC bits can be serialized out on output RGFC/RLD pin. This feature is enabled by the UNI_POTS bit in the **LASAR-155 Master Configuration (0x004)** register.

| UNI_POTS | Function | R/W |
|---|---|---|
| 0 | The RGFC/RLD, RCP/RLDCLK, TGC/TLD and TCP/TLDCLK signals are configured to source and sink the GFC bits in the cell header. | default |
| 1 | The RGFC/RLD, RCP/RLDCLK, TGC/TLD and TCP/TLDCLK signals are configured to source and sink the SONET Line Data Communication Channel (DCC). | |

The state of RGFC with respect to GFC bits is controlled by the RGFCE[3:0] bits of **RACP Configuration (0x164)**. RGFC[3] corresponds to the most significant GFC bit. The serial link is forced low if cell delineation is lost.

| RGFC[N] | Function | R/W |
|---|---|---|
| 0 | The RGFC remains in its current state. | |
| 1 | The RGFC output changes in the appropriate bit location to the state of the associated GFC bit in the current cell. | default |

## 7.5 Receive Cell Counter

The number of ATM cells received and passed to RALP block are maintained in the following registers. Refer to "Section 15. Counter Overview" for additional information.

- **RACP Receive Cell Counter (LSB) (0x158)**
- **RACP Receive Cell Counter (0x15C)**
- **RACP Receive Cell Counter (MSB) (0x160)**

## 7.6 Receive ATM Cell FIFO

The FIFORST bit of **RACP Control/Status (0x140)** is used to reset the internal four cell receive FIFO.

| FIFORST | Function | R/W |
|---|---|---|
| 0 | The FIFO operates normally. | default |
| 1 | The FIFO is immediately emptied and ignores writes. | |

# 8 TRANSMIT TRANSMISSION CONVERGENCE SUBLAYER PROCESSING

The Transmit ATM Cell Processor (TACP) performs the following functions:

- Rate adaptation via idle/unassigned cell insertion
- HEC generation and insertion
- GFC insertion
- ATM cell scrambling
- Transmit Cell Counting

## 8.1 Idle/Unassigned Cell Generation

Cell rate decoupling is accomplished by transmitting idle/unassigned cells when no outstanding data cells need to be transmitted. The all zeros pattern is transmitted in the VCI and VPI fields of the idle cell.

The CLP, PTI and GFC fields of the idle/unassigned cell header are defined in the **TACP Idle/Unassigned Cell Header Pattern (0x184)** and the cell payload pattern can be specified in **TACP Idle/Unassigned Cell Payload Octet Pattern (0x188)**.

## 8.2 HEC Generation

The HEC calculation over the first four header octets is performed using the polynomial, $x^8+x^2+x+1$. The HECB bit of **TACP Control/Status (0x180)** enables the internal generation and insertion of the HEC octet into the transmit cell stream.

| HECB | Function | R/W |
|---|---|---|
| 0 | HEC is generated and inserted internally. | default |
| 1 | The HEC octet is set to 00H. | |

The coset polynomial, $x^6+x^4+x^2+1$ is optionally added (modulo 2) to the residue by the HECADD bit of **TACP Control/Status (0x180)**.

| HECADD | Function | R/W |
|---|---|---|
| 0 | The coset polynomial is not added. | |
| 1 | The coset polynomial is added. | default. |

## Cell Payload Scrambling

Scrambling is performed using the self synchronous scrambler, $x^{43}+1$. Scrambling is performed only over cell payloads. Cell headers are not scrambled. Scrambling is controlled by the DSCR bit of **TACP Control/Status (0x180)**.

| DSCR | Function | R/W |
|---|---|---|
| 0 | Payload scrambling is enabled. | default |
| 1 | Payload scrambling is disabled. | |

## 8.3 GFC Insertion

The GFC bits in the ATM cell header can be generated from one of the following sources:

- If enabled, the GFC bits can be specified by the TGFC serial input. This feature is controlled by the UNI_POTS bit in the **LASAR-155 Master Configuration (0x004)** register.

| UNI_POTS | Function | R/W |
|---|---|---|
| 0 | The RGFC/RLD, RCP/RLDCLK, TGFC/TLD and TCP/TLDCLK signals are configured to source and sink the GFC bits in the cell header. | default |
| 1 | The RGFC/RLD, RCP/RLDCLK, TGFC/TLD and TCP/TLDCLK signals are configured to source and sink the SONET Line Data Communication Channel (DCC). | |

- For assigned cells, the GFC bits are defined in GFC[3:0] bits of **COPS VPI (0x28C)** on a per VC basis.
- For Idle/Unassigned cells, the GFC bits are defined in the GFC bits of the **TACP Idle/Unassigned Cell Header Pattern (0x184)**.

The value of the GFC field is determined by the TGFCE[3:0] bits of **TACP Configuration (0x19C)**.

| TGFC[N] | Function | R/W |
|---|---|---|
| 0 | For assigned cells, the GFC is defined by the GFC[3:0] bits of **COPS VPI (0x28C)**.<br>For Idle/Unassigned cells, the GFC bit value is supplied by the GFC bit of the **TACP Idle/Unassigned Cell Header Pattern (0x184)**. | default |
| 1 | The $N^{th}$ bit of the 4 bit serial sequence is inserted into the GFC bit. TGFC[3] corresponds to the most significant bit of the GFC header. TGFC[0] corresponds to the least significant bit of the GFC header. | |

## 8.4 Transmit Cell Counter

The number of cells transmitted by the TALP in the last accumulation interval, excluding Idle/Unassigned cells, is maintained in the following registers. Refer to "Section 15. Counter Overview" for additional information.

- **TACP Transmit Cell Counter (LSB) (0x190)**
- **TACP Transmit Cell Counter (0x194)**
- **TACP Transmit Cell Counter (MSB) (0x198)**

## 8.5 Transmit ATM Cell FIFO

The FIFORST bit of **TACP Control/Status (0x180)** is used to reset the internal four cell transmit FIFO.

| FIFORST | Function | R/W |
|---|---|---|
| 0 | The FIFO operates normally. | default |
| 1 | FIFO is immediately emptied and ignores writes. | |

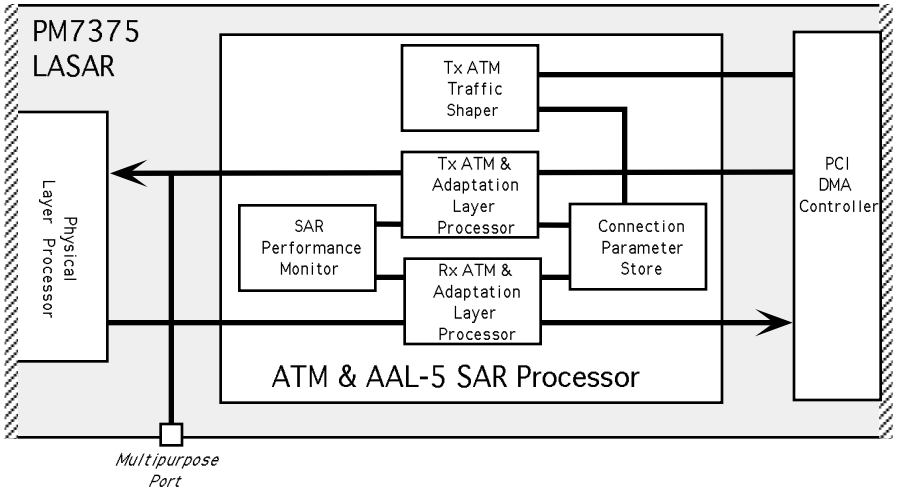
The depth of the transmit FIFO can be configured by the FIFODP[1:0] bits of **TACP FIFO Configuration (0x18C)**. FIFO depth control may be important in systems where the cell latency through the TACP must be controlled.

| FIFODP[1:0] | Function | R/W |
|---|---|---|
| 00B | Transmit ATM Cell FIFO depth is 4 cells. | default |
| 01B | Transmit ATM Cell FIFO depth is 3 cells. | |
| 10B | Transmit ATM Cell FIFO depth is 2 cells. | |
| 11B | Transmit ATM Cell FIFO depth is 1 cells. | |

# 9 ATM & AAL LAYER OVERVIEW

The ATM & AAL-5 SAR Processor implements core functions of ATM and AAL layers. It is a service user of the Physical Layer Processor and a service provider to the higher layer entities via the PCI DMA Controller Examples of higher layer entities include Q.2931/Q.SAAL signaling, LAN Emulation, native ATM applications, and network management agents. The functional blocks of ATM & AAL-5 SAR Processor are illustrated in Figure 9.1.

Figure 9.1: ATM & AAL-5 SAR Processor Functional Blocks.




## Rx ATM and Adaptation Layer Processor

The Rx ATM and Adaptation Layer Processor performs ATM Layer and AAL-5 reassembly functions. ATM Layer processing includes open VC verification, cell filtering, cell copying and CRC-10 verification. Cell filtering is the action of dropping cells intended for the PCI Host. Cell copying is the action of directing cells to the Multipurpose Port when it is configured in a cell source mode. Cell copying to the Multipurpose Port can be based on the VPI/VCI code and/or PTI codepoints.

The AAL-5 processing includes the reassembly and verification of AAL-5 protocol data units (PDUs). The Rx ATM & Adaptation Layer Processor can optionally perform VC aging. VC aging protects the LASAR from slow or dead connections that may be holding valuable receive buffer resources for an unacceptable duration. Non-activity termination allows for the automatic termination of a PDU under reassembly when the packet exceeds a user specified time-out period.

The configuration and monitoring of the Rx ATM and Adaptation Layer Processor is described in "Section 10. Receive ATM Layer Processing" and "Section 12. Receive AAL Processing".

## Tx ATM Traffic Shaper

The Tx ATM Traffic Shaper provides peak cell rate (PCR) enforcement using eight peak rate queues arranged as a group of four high priority queues and a group of four low priority queues. As part of the provisioning process, a VC must be associated with one of the eight rate queues. Once a VC is provisioned, packets supplied by the PCI DMA Controller are attached to the associated rate queue. High priority queues must be completely serviced before the low priority queues are serviced. If the high priority queues consume all the available link bandwidth, the low priority queues are allowed to starve. An indication is provided when any queue is experiencing a starvation condition.

The Tx ATM Traffic Shaper allows packet transmission at either the PCR or the sustainable cell rate (SCR). The SCR is specified as a fraction of the PCR (that is, SCR = PCR/n where n =1 to 8). The PCR transmission is defined on a per VC basis by associating a VC with one of the eight rate queues and by selecting whether to use 100%, 50%, or 25% of the PCR provided by the queue.

The SCR transmission is managed on a per VC basis through a token generation mechanism. The Tx ATM Traffic Shaper provides each VC with a token bucket and transmission is allowed only if the token bucket is not empty. Each transmitted cell consumes one token. When the VC is idle, the bucket is replenished at the SCR up to the capacity of the bucket. When the bucket is full, newly generated tokens are discarded. Cell transmission can be maintained at PCR when the bucket is not empty. When the bucket is empty, transmission continues at the SCR which is the rate of token generation.

The configuration and monitoring of the Tx ATM Traffic Shaper is described in "Section 14. Traffic Shaping".

## Tx ATM & Adaptation Layer Processor

The Tx ATM & Adaptation Layer Processor performs ATM layer and AAL-5 segmentation functions. The AAL-5 processing is performed in conjunction with the PCI DMA Controller and the Tx Traffic Shaper. As the Tx Traffic Shaper schedules when cells from a packet under segmentation can be sent, the PCI DMA Controller retrieves the payload from the host memory and passes it to the Tx ATM & Adaptation Layer Processor which calculates the CRC-32 fields and forms the AAL-5 protocol data unit (PDU).

The ATM layer processing includes generating the GFC, VPI, VCI, PTI, CLP fields and optionally generating the CRC-10 field for each cell transmitted.

The configuration and monitoring of the Tx ATM and Adaptation Layer Processor is described in "Section 11. Transmit ATM Layer Processing" and "Section 13. Transmit AAL Processing".

## SAR Performance Monitor

The SAR Performance Monitor accumulates the following statistics:

- ATM Cell unprovisioned VPI/VCI errors
- ATM Cell CRC-10 errors
- Receive CPAAL5_PDU Invalid Common Part Indicator errors
- Receive CPAAL5_PDU Invalid SDU Length errors
- Receive CPAAL5_PDU CRC-32 errors
- Receive CPAAL5_PDU Oversize SDU errors
- Receive CPAAL5_PDU Abort errors
- Receive CPAAL5_PDU Count
- Receive CPAAL5_PDU Time-outs Count
- Receive buffer errors
- Transmit CPAAL5_PDU Oversize SDU errors
- Transmit CPAAL5_PDU Count

The configuration and monitoring of the SAR Performance Monitor is described in "Section 15. Counter Overview".

## Connection Parameter Store

The Connection Parameter Store provides the internal VC parameter storage for both the 128 transmit and 128 receive VCs. The VC parameters are referenced by the Tx Traffic Shaper, Tx ATM & Adaptation Layer Processor, and Rx ATM & Adaptation Layer Processor.

The procedure to access the Connection Parameter Store is described in "Section 16. VC Parameter Table Access".

# 10 RECEIVE ATM LAYER PROCESSING

The Receive ATM Layer Processor (RALP) performs the following functions:

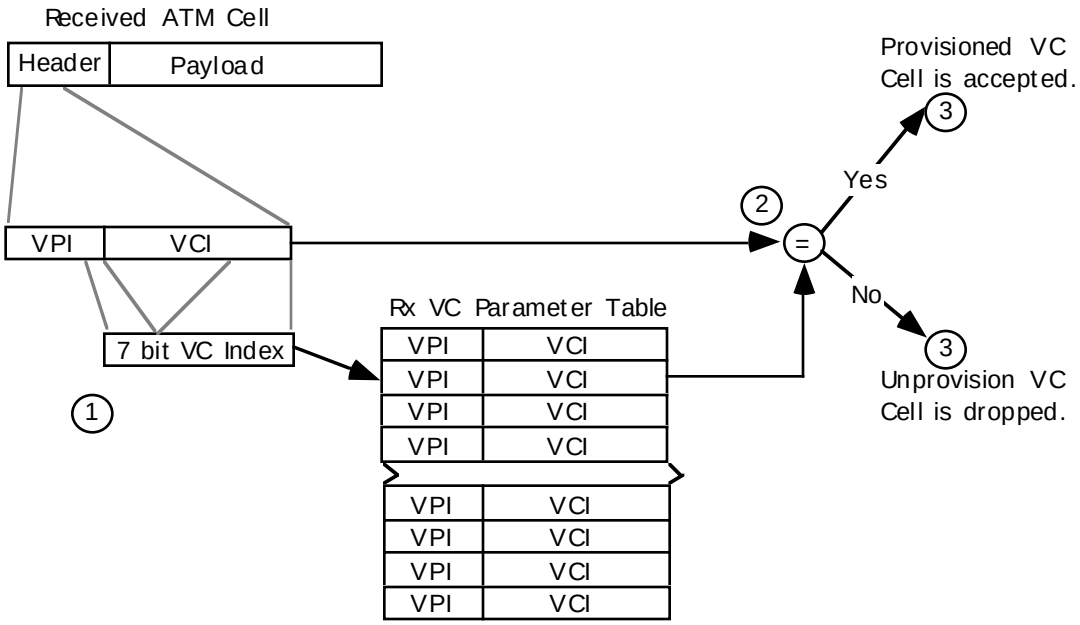
- Open VC verification
- Cell filtering and cell copying

Cell filtering is the action of dropping cells to the PCI Host. Cell copying is the action of copying cells to the Multipurpose Port. Cell filtering and cell copying are mutually exclusive operations.

## 10.1 Open VC Verification

For every incoming cell, the RALP must verify that the VPI and VCI header fields correspond to a provisioned VC. When a cell arrives the LASAR-155 performs the steps illustrated in Figure 10.1 to determine whether this is an open VC or not. Note that the full VCI/VPI code is compared.

Figure 10.1: Open VC Verification Process




1. A 7 bit VC index is generated from the VPI/VCI code.
2. The VPI/VCI code in the Rx VC parameter table associated with the 7 bit VC index is compared with the VPI/VCI code in the ATM cell header.
3. If the comparison yields an exact match, this cell is received on a provisioned VC and is processed. If no match, this cell is received on a unprovisioned VC and is dropped. The reception of a unprovisioned cell can generate an

interrupt on the UVPI/VCII bit of **RALP Interrupt Status (0x204)** if it is enabled by the UVPI/VCIE bit of **RALP Interrupt Enable (0x208)**.

## 10.2 Cell Filtering Cell Copying

The LASAR-155 can filter or route ATM cells based on the PTI field of the header. ATM cells can be passed to the PCI Host, Multipurpose Port or both. ATM cells can only be copied to the Multipurpose Port if it is configured in non-bypass mode as controlled by the RXPHYBP pin set to logic zero.

### Undefined Cell

An ATM cell with the PTI field equals to 111B has not been defined and is considered invalid. The reception of an ATM cell with PTI=111B can generate an interrupt on PTIDI bit of **RALP Interrupt Status (0x204)** if it is enabled by the PTIDE bit of **RALP Interrupt Enable (0x208)**.

The PTI_111_RD bit of **RALP Control (0x200)** controls whether to discard cells with PTI header fields equal to 111B or to pass them on to the PCI Host.

| PTI_111_RD | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=111B are passed to the PCI Host. | |
| 1 | Cells with PTI=111B are dropped. | default |

If the Multipurpose Port is configured in non-bypass mode (RXPHYBP=0), the PTI_111_ED bit of **RALP Control (0x200)** controls whether to discard cells with PTI header fields equal to 111B or to copy the cells to the Multipurpose Port.

| PTI_111_ED | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=111B are passed to the Multipurpose Port. | |
| 1 | Cells with PTI=111B are dropped. | default |

### Resource Management Cell

An ATM cell with the PTI field equals to 110B is a Resource Management (RM) cell. The reception of an ATM cell with PTI=110B can generate an interrupt on PTIDI bit of **RALP Interrupt Status (0x204)** if it is enabled by the PTIDE bit of **RALP Interrupt Enable (0x208)**.

The PTI_110_RD bit of **RALP Control (0x200)** controls whether to discard cells with PTI header fields equal to 110B or to pass them on to the PCI Host.

| PTI_110_RD | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=110B are passed to the PCI Host. | |
| 1 | Cells with PTI=110B are dropped. | default |

If the Multipurpose Port is configured in non-bypass mode (RXPHYBP=0) , the PTI_110_ED bit of **RALP Control (0x200)** controls whether to discard cells with PTI header fields equal to 110B or to copy the cells to the Multipurpose Port.

| PTI_110_ED | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=110B are passed to the Multipurpose Port. | |
| 1 | Cells with PTI=110B are dropped. | default |

**F5 OAM Cell**

An ATM cell with the PTI field equal to 100B is a segment OAM F5 cell. An ATM cell with the PTI field equals to 101B is an end-to-end OAM F5 cell. The reception of F5 OAM cells can generate an interrupt on the F5OAMDI bit of **RALP Interrupt Status (0x204)** if it is enabled by the F5OAMDE bit of **RALP Interrupt Enable (0x208)**.

The PTI_10X_RD bit of **RALP Control (0x200)** controls whether to discard or pass cells with PTI header fields equal to 100B or 101B.

| PTI_10X_RD | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=100B or 101B are passed to the PCI Host. | |
| 1 | Cells with PTI=100B or 101B are dropped. | default |

If the Multipurpose Port is configured in non-bypass mode (RXPHYBP=0), the PTI_10X_ED bit of **RALP Control (0x200)** controls whether to discard or pass cells with PTI header fields equal to 100B or 101B.

| PTI_10X_ED | Function | R/W |
|---|---|---|
| 0 | Cells with PTI=100B or 101B are passed to the Multipurpose Port. | |
| 1 | Cells with PTI=100B or 101B are dropped. | default |

**User Cell Routing**

The VC_PACKET_QUEUE_EN bit of Receive VC Parameter Table, which is mapped to the CTL[1] bit of **COPS VC Control and Status (0x294)**, determines whether user cells are copied to the PCI Host. User cells are those with PTI=0XXB.

| VC_PACKET_QUEUE_EN | Function | R/W |
|---|---|---|
| 0 | User cells are copied to the PCI Host. | default |
| 1 | User cells are not copied to the PCI Host. | |

The CEX_EN bit of Receive VC Parameter Table, which is mapped to the CTL[0] bit of **COPS VC Control and Status (0x294)**, determines whether user cells are copied to the Multipurpose Port or not. User cells that are copied to the Multipurpose Port must be processed by an external entity.

| CEX_EN | Function | R/W |
|---|---|---|
| 0 | User cells are not copied to the Multipurpose Port. | default |
| 1 | User cells are copied to the Multipurpose Port. | |

# 11 TRANSMIT ATM LAYER PROCESSING

Transmit ATM Layer processing includes:

- VPI/VCI generation
- CLP insertion
- PTI insertion.

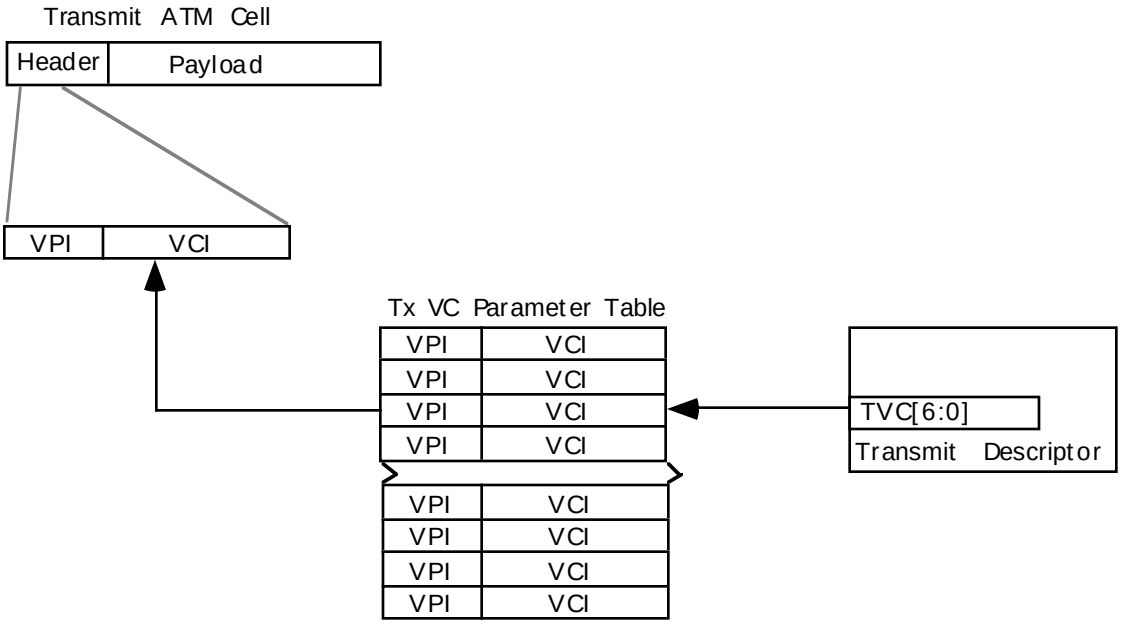
## 11.1 VPI/VCI Generation

During the provisioning of a transmit VC, the VPI[7:0] bits of **COPS VPI (0x28C)** are used to set the ATM VPI cell header field and the VCI[15:0] bits of **COPS VCI (0x290)** are used to set the ATM VCI cell header field.

When the PCI Host is ready to send a packet, it specifies the transmit VC in the TVC[6:0] bits of the Transmit Descriptor. This is the 7 bit index to the transmit VC parameter table. The LASAR-155 translates the 7 bit index to the complete VPI/VCI code and inserts it into all the ATM cells in that packet.

If the PCI Host tries to send a packet on a unprovisioned VC, an interrupt can be generated on the TXVC_IDLI bit of **PCID Interrupt Status (0x304)** if it is enabled by the TXVC_IDLE bit of **PCID Interrupt Enable (0x308)**.

Figure 11.1: VCI/VPI Assignment



## 11.2 CLP Insertion

Cell Header Select (CHS) bit of the Transmit Descriptor selects the source of the cell header fields used to transport the current TD's buffer data. This selection can be made on a packet by packet basis.

| CHS | Function |
|---|---|
| 0 | The default value is specified in the CLP bit of **COPS VPI (0x28C)** on a per VC basis. |
| 1 | The CLP bit in the ATM cell header is defined in the CLP bit of the current TD. |

## 11.3 PTI Insertion

Cell Header Select (CHS) bit of the Transmit Descriptor selects the source of the cell header fields used to transport the current TD's buffer data. This selection can be made on a packet by packet basis.

| CHS | Function |
|---|---|
| 0 | The default value as is specified in the PTI[2:0] bits of **COPS VPI (0x28C)** on a per VC basis. |
| 1 | The CLP bit in the ATM cell header is defined in the Congestion (CG) bit and Cell Type CT[1:0] bits of the current TD. |

# 12 RECEIVE AAL PROCESSING

This section describes the functions and controls of receive ATM Adaptation Layer.

## 12.1 Reassembly Enable

In order to receive a packet on a provisioned VC, both the global reassembly control and per VC reassembly control must be enabled. The global reassembly enable is controlled by the REAS_EN bit of **RALP Control (0x200)**.

| REAS_EN | Function | R/W |
|---|---|---|
| 0 | Reassembly is disabled. Any active reassemblies can be terminated either immediately or gracefully as controlled by the REAS_DM bit of **RALP Control (0x200)**. | default |
| 1 | The reassembly is enabled. Receive VCs must be provisioned before packets can be reassembled. | |

The per VC reassembly enable is controlled by the VC_REAS_EN bit of the Receive VC Parameter Table which is mapped to the CNTL[4] bit of **COPS VC Control and Status (0x294)**.

The global reassembly status can be determined by reading the RESEM_ACTV bit of **RALP Interrupt Status (0x204)**.

| RESEM_ACTV | Function | R |
|---|---|---|
| 0 | No reassemblies are active. | |
| 1 | One or more reassemblies are active. | |

The per VC reassembly status can be determined by reading the ACTV_RSMB bit of the Receive VC Parameter Table which is mapped to the STATUS[1] bit of **COPS VC Control and Status (0x294)**.

## 12.2 Reassembly Termination

Active VCs can be shutdown on a per VC basis or on a global basis. The reassembly termination mode is controlled by the REAS_DM bit of **RALP Control (0x200)**.

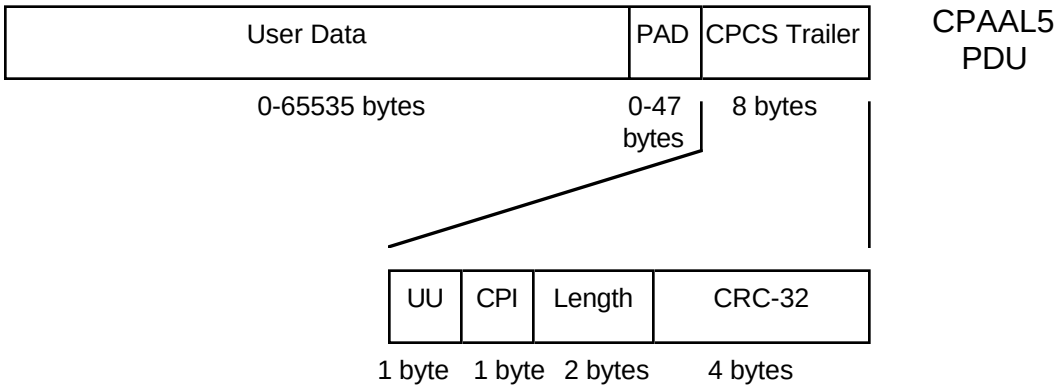
| REAS_DM | Function | R/W |
|---|---|---|
| 0 | Active reassemblies are allowed to be completed after reassembly has been disabled. | default |
| 1 | Active reassemblies are terminated immediately. Any receive buffers still in use are frozen, but can be released using the Packet Aging Circuitry. | |

## 12.3 AAL-5 PDU Reassembly

The AAL-5 packet consists of multiple fields as described in Figure 12.1. The processing of these fields by the LASAR-155 is described in the following subsections.

Before reading this section, the reader should be familiar with the LASAR-155 data structures. Refer to "Section 19. Descriptors" for additional information.

Figure 12.1: CPAAL5_PDU Fields




If user cells are copied to the PCI Host, the QUEUE_SEL[1:0] bits of the Receive VC Parameter Table, which is mapped to CTL[3:2] bits of **COPS VC Control and Status (0x294)**, should be set to 11B to select the Receive Packet Queue (RPQ).

| QUEUE_SEL[1:0] | Function |
|---|---|
| 00B | Cells are directed to the Receive Management Queue (RMQ). CRC-10 checking is applied to the cell payload. |
| 01B | Cells are directed to the RMQ, CRC-32 checking is applied to the cell payload. |
| 10B | Cells are directed to the RMQ, no CRC checking is applied to the cell payload. |
| 11B | All user cells (PTI=0XXB) are directed to the Receive Packet Queue (RPQ). No CRC checking is applied to user cells. User cells are reassembled into CPAAL5_PDU which are in turn automatically verified using the CRC-32.<br><br>Management cells (PTI=1XXB) are directed to the RMQ. CRC-10 checking is automatically applied to management cells (PTI=100B, 101B, 110B). No CRC is applied to cells with PTI=111B. |

### User Data

The LASAR-155 reassembles the User Data in the host memory. One or more Receive Packet Descriptors (RPDs) are used to describe the packet in the host memory as described in "Section 17. PCI DMA Controller Overview".

The ENDIAN bit of **PCID Control (0x300)** determines whether Big or Little Endian byte ordering is used when reading from or writing to the data buffers in the host memory.

| ENDIAN | Function | R/W |
|---|---|---|
| 0 | Big Endian format is selected. | |
| 1 | Little Endian format is selected. | default |

### PAD Field

The PAD field is used to align the CPAAL5_PDU to a 48 byte boundary. The PAD field is dropped by the LASAR and is not passed to the PCI Host.

### User-to-User Field

The User-to-User (UU) field contains system specific information that can be used by end systems to communicate with one another. The UU field is copied to the CPAAL5_PDU UU[7:0] bits of the first RPD.

### Common Part Indicator Field

The use of Common Part Indicator (CPI) field is currently not specified and is normally coded to 00H. The received CPI field is copied to the CPAAL5_PDU CPI[7:0] bits of the first RPD.

The reception of a non-zero CPI byte can generate an interrupt on the NZCPII bit of **RALP Interrupt Status (0x204)** if it is enabled by the NZCPIE bit of **RALP Interrupt Enable (0x208)**.

### Length Field

The Length field is used to indicate the length of the User Data field. When set to zero, it indicates a forward abort condition.

The received Length field is copied to the CPAAL5_PDU LENGTH[15:0] bits of the first RPD. A receive length mismatch condition can generate an interrupt on the SDULI bit of **RALP Interrupt Status (0x204)** if it is enabled by the SDULE bit of **RALP Interrupt Enable (0x208)**. A forward abort condition (when Length=0) does not result in a SDULI interrupt. Instead, the PDUABI bit of **RALP Interrupt Status (0x204)** indicates the detection of a forward abort indication if it is enabled by the PDUABE bit of the **RALP Interrupt Enable (0x208)**.

During packet reassembly, LASAR-155 can check the length of receiving User Data to ensure that a user programmable maximum value is not exceeded. This function is described in the Maximum SDU Length section.

## CRC-32 Field

The four byte CRC-32 field provides a CRC check over the entire CPAAL5_PDU. The LASAR-155 automatically verifies the CRC-32 over the reassembled CPAAL5_PDU. The received CRC-32 field is copied to CPAAL5_PDU CRC-32[31:0] bits of the first RPD.

The detection of a CRC-32 error can generate an interrupt on the CRC32I bit of **RALP Interrupt Status (0x204)** if it is enabled by the CRC32E bit of **RALP Interrupt Enable (0x208)**.

## Packet Aging Circuitry

LASAR-155 provides an internal packet aging circuitry to terminate packet reassemblies that exceeded a global time-out period. When a packet reassembly is timed out, all successive bytes of the CPAAL5_PDU are dropped until the start of the next CPAAL5_PDU. This feature allows LASAR-155 to release buffer resources on VCs with no activity in the last global time-out period.

The packet aging circuitry is enabled globally by the PDU_TO_EN bit of **RALP Control (0x200).**

| PDU_TO_EN | Function | R/W |
|---|---|---|
| 0 | Packet aging is globally disabled. | |
| 1 | Packet aging is globally enabled. | default. |

On a per VC basis, the packet aging is enabled by the VC_PDU_TO_EN bit of the Rx VC Parameter Table which is mapped to CTL[4] bit of **COPS VC Control and Status (0x294)**.

| VC_PDU_TO_EN | Function | R/W |
|---|---|---|
| 0 | Per VC aging is disabled. | |
| 1 | Per VC aging is enabled if PDU_TO_EN bit of **RALP Control (0x200)** is enabled. That is, the VC_PDU_TO_EN bit is ANDed to the PDU_TO_EN bit to determine whether packet aging is enabled or not. | default. |

When the packet aging circuitry is enabled, a time-out event can generate an interrupt on the PDUTOI bit of **RALP Interrupt Status (0x204)** if it is enabled by the PDUTOE bit of **RALP Interrupt Enable (0x208)**. The number of receiving CPAAL5_PDUs that were time out during reassembly is maintained in **SAR PMON Receive PDU Timeout Errors (0x1E8)**.

The global time out period is specified in the PDU_TO[15:0] bits of **RALP CPAAL5_PDU Time-out (0x214)** in time-out clock periods. The time-out clock period, as expressed in number of SYSCLK periods, is specified in TO_CLK[15:0] bits of **RALP Time-out Timer Period (0x210)**. Refer to LASAR-155 Longform Datasheet for sample values.

The VC time-out interrogation period specifies how often the VCs are checked for time-out conditions. This is specified in VCTIP[7:0] bits of **RALP VC Time-out Interrogation Period (0x218)**. The VCTIP counter is clocked by SYSCLK divided by eight. VCTIP must be greater than 3.

| SYSCLK (MHz) | SYSCLK Period (nsec) | VCTIP Counter Granularity (nsec) | VCTIP Counter Range (usec) |
|---|---|---|---|
| 20 | 50 | 400 | 1.60 to 102 |
| 25 | 40 | 320 | 1.28 to 81.6 |
| 33 | 30 | 240 | 0.96 to 61.2 |

For SONET STS-3c the cell time is 2.83 usec. To check one VC per cell time at 33 MHz, the VCTIP[7:0] value can be determined by the following equation

```
VCTIP  = Cell Time / Granularity - 1
       = 2.83 usec / 240 nsec -1
       = 10.79
```

If the value of 10B is used, the VCs will be interrogated faster than once per cell time. Therefore, the user should program in the value 11B to ensure that the maximum interrogation rate of once per VC is met.

**Maximum SDU Length Checking**

During packet reassembly, LASAR-155 can check the length of receiving User Data to ensure that a user programmable maximum value is not exceeded. The MRSDU_EN bit of **RALP Control (0x200)** controls whether to enforce the user programmed maximum receive CPAAL5_SDU length.

| MRSDU_EN | Function | R/W |
|---|---|---|
| 0 | The maximum CPAAL5_SDU length is not enforced. | default |
| 1 | The maximum CPAAL5_SDU length is enforced. | |

Packet reassemblies can be monitored on a per VC basis for maximum SDU Length infringement by polling the MAX_SDU_ABORT bit of the Rx VC Parameter Table which is mapped to STATUS[6] of **COPS VC Control and Status (0x294)**.

| MAX_SDU_ABORT | Function | R |
|---|---|---|
| 0 | The current packet was terminated due to the packet being larger than the specified maximum length. | |
| 1 | The current packet was terminated due to the packet being larger than the specified maximum length. | |

The maximum length of the receive CPAAL5 SDU is defined using the MRSDU[15:0] bits of **RALP Max Rx SDU Length (0x20C)**. This maximum length must either be an integer multiple of 48 or 0xFFFF bytes.

When the maximum length is enforced, CPAAL5_SDUs that exceed the maximum value are capped at the maximum length. All subsequent cells are ignored until the end of packet. The **SAR PMON Receive Oversize SDU Errors (0x1E4)** counter is incremented. An interrupt can also be generated on the OVSDUI bit of **RALP Interrupt Status (0x204)** if it is enabled by the OVSDUE bit of **RALP Interrupt Enable (0x208)**.

## 12.4 AAL 3/4 Cell Reassembly

To provision a VC to receive AAL3/4 cells, the QUEUE[1:0] bits of the Receive VC Parameter Table, which are mapped to CTL[3:2] of the **COPS VC Control and Status (0x294)** register, must be set to 00B. This option directs the AAL3/4 cells to the Receive Management Queues and applies the CRC-10 checking on the cell payload.

The detection of a CRC-10 error can generate an interrupt on the CRC10I bit of **RALP Interrupt Status (0x204)** if it is enabled by the CRC10E bit of **RALP Interrupt Enable (0x208)**. The number of CRC-10 errors is maintained in **SAR PMON Receive CRC-10 Errors (LSB) (0x1D0)** and **SAR PMON Receive CRC-10 Errors (MSB) (x01D4)** registers.

The payload of the AAL3/4 packet must be interpreted by the PCI Host in software.

## 12.5 Raw Cell Reassembly

Proprietary packet formats can be implemented by sending and receiving raw ATM cells. To provision a VC to receive raw ATM cells with or without CRC checking set the QUEUE[1:0] bits of the Receive VC Parameter Table, which are mapped to CTL[3:2] of the **COPS VC Control and Status (0x294)** register to the appropriate value depending on whether CRC checking is required or not.

## 12.6 Management Cell Reassembly

In the receive direction several control bits are provided in **RALP Control (0x200)** to determine whether management cells (PTI=100B, 101B, 110B, 111B) are passed to PCI Host, copied to the Multipurpose Port or both. This is described in "Section 10.3. PTI Handling".

To provision a VC to receive management cells set the QUEUE[1:0] bits of the Receive VC Parameter Table, which are mapped to CTL[3:2] of the **COPS VC Control and Status (0x294)** register, to 11B. In this configuration, management cells are routed to the Receive Management Ready Queue whereas user packets are reassembled to the Receive Packet Ready Queue.

The reception of a cell with a PTI field equal to 110B or 111B can generate an interrupt on the PTIDI bit of **RALP Interrupt Status (0x204)** if it is enabled by the PTIDE bit of the **RALP Interrupt Enable (0x208)**. The reception of a F5 OAM cell can generate an interrupt on the F5OAMDI bit of **RALP Interrupt Status (0x204)** if it is enabled by the F5OAMDE bit of the **RALP Interrupt Enable (0x208)**.

# 13 TRANSMIT AAL PROCESSING

This section describes the functions and controls of transmit ATM Adaptation Layer. The reader should be familiar with the fields in the Transmit Descriptor (TD). Refer to "Section 21. Descriptors" for additional information.

## 13.1 Segmentation Controls

A transmit VC must be provisioned before it can transmit packets. The parameters in the transmit VC table are described in "Section 16.8 Tx VC Parameter Table."

Before packets can be segmented, the Transmit Request Machine (TRM) must be enabled using the TRMEM bit of **PCID Control (0x300)**.

| TRMEN | Function | R/W |
|---|---|---|
| 0 | The TRM is held in an idle state and no segmentation is allowed. No processing of ready queues is performed. | default |
| 1 | The TRM is allowed to segment packets. | |

The VC_SEG_EN bit of Transmit VC Parameter Table, which is mapped to CTL[3] bit of **COPS VC Control and Status (0x294)**, controls cell segmentation on a per VC basis.

| VC_SEG_EN | Function | R/W |
|---|---|---|
| 0 | The VC is unprovisioned and cell segmentation is disabled. | default |
| 1 | The VC is provisioned and cell segmentation is enabled. | |

The STATUS[1:0] bits of **COPS VC Control and Status (0x294)** can be read to determine the state of segmentation.

| STATUS[1:0] | Function | R |
|---|---|---|
| 00B | The VC is disabled for segmentation. | |
| 01B | The VC is enabled for segmentation and no active segmentation is in progress. | |
| 10B | The VC is disabled for segmentation. | |
| 11B | The VC is enabled for segmentation and active segmentation is in progress. | |

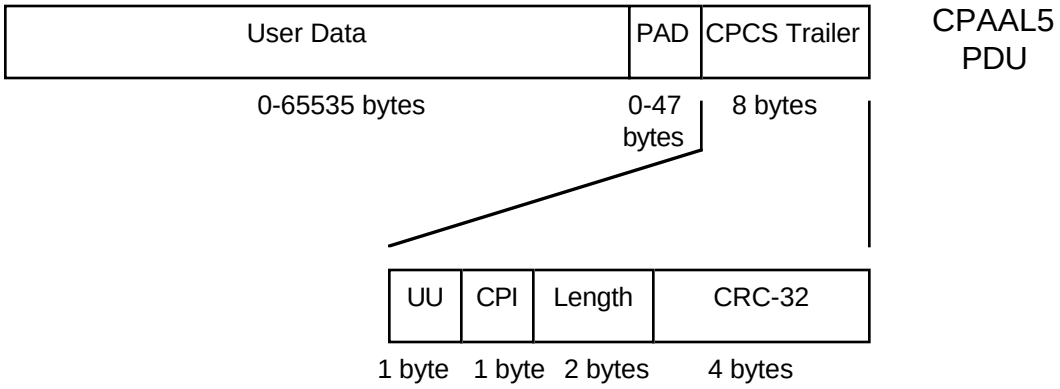
When unprovisioning a VC, the TXVC_UPIE bit of **PCID Control (0x300)** selects the VC unprovision process.

| TXVC_UPIE | Function | R/W |
|---|---|---|
| 0 | Any cells in the transmit FIFOs are flushed before a VC is unprovisioned. | default |
| 1 | The VC is unprovisioned immediately. | |

## 13.2 AAL-5 PDU Segmentation

To send an AAL-5 packet on a provisioned VC, the PCI Host must prepare the packet in the host memory using one or more Transmit Descriptors (TDs). The Cell Type (CT[1:0]) in all the TDs must be set to 00B to indicate that this is an AAL-5 packet.

Figure 13.1: CPAAL5_PDU Fields




### User Data

The LASAR-155 segments the User Data from the host memory as described by one or more TDs as described in the Host DMA section.

The ENDIAN bit of **PCID Control (0x300)** determines whether Big or Little Endian byte ordering is used when reading from or writing to the data buffers in the host memory.

| ENDIAN | Function | R/W |
|---|---|---|
| 0 | Big Endian format is selected. | |
| 1 | Little Endian format is selected. | default |

### PAD Field

The variable length PAD field is automatically generated by the LASAR-155.

### User-to-User Field

The TXPDU_UU_S bit of **PCID Control (0x300)** selects the source of the User-to-User (UU) byte:

| TXPDU_UU_S | Function | R/W |
|---|---|---|
| 0 | The CPAAL5_PDU UU[7:0] field from the TD is used. | default |
| 1 | 00H is inserted into the UU byte. | |

### Common Part Indicator Field

The TXPDU_CPI_S bit of **PCID Control (0x300)** selects the source of CPI byte:

| TXPDU_CPI_S | Function | R/W |
|---|---|---|
| 0 | The CPAAL5_PDU CPI[7:0] field from the TD is used. | default |
| 1 | 00H is inserted into the CPI byte. | |

### Length Field

The Packet Length[15:0] bits of a Transmit Descriptor are used by the PCI Host to indicate the packet size, in bytes. If multiple TDs are used, the Packet Length field for all the TDs must be the same.

### CRC-32 Field

The CRC[1:0] bits of the TD must be set to 00B to apply CRC-32 polynomial to the packet.

### Packet Transmission Abort

A packet transmission can be aborted by the following steps:

- The Abort (ABT) bit in the TD is set to 1.
- The More (M) bit in the TD is set to 0 to indicate that this is the last TD.
- The Packet Length[15:0] bits in the TD must be set to 0000H.

### Maximum SDU Length Checking

The MTSDU[15:0] bits of **PCID Max Tx SDU Length (0x3B0)** represents the user programmable maximum length of the transmit user data, in bytes. The default value is FFFFH.

When a packet with user data larger than the programmed maximum length, an interrupt can be generated on the MAX_SDUI bit of the **PCID Interrupt Status (0x304)** if it is enabled by the MAX_SDUE bit of **PCID Interrupt Enable (0x308)**. The TDs used by the packet are returned to the Transmit Descriptor Free Queue.

## 13.3 AAL 3/4 Cell Segmentation

The payload of the AAL3/4 packet must be prepared by the PCI Host in software. The TD that describes the AAL3/4 cell in the host memory must set CT[1:0] bits to 01B to select raw user cell and CRC[1:0] bits to 01B to select CRC-10 polynomial.

## 13.4 Raw Cell Segmentation

Raw cells sourced from the PCI Host must set CT[1:0] bits in the TD to 01B to select raw user cell and CRC[1:0] bits to select the appropriate CRC polynomial.

Raw cells can also be sourced from the Multipurpose Port if this function is enabled by the CINPORT_EN bit of **TALP Control (0x220)**.

| CINPORT_EN | Function | R/W |
|---|---|---|
| 0 | Cells cannot be sourced from the Multipurpose Port. | default |
| 1 | Cells can be sourced from the Multipurpose Port. | |

The CIN_CL_EN bit of **TALP Control (0x220)** controls whether CRC-10 is calculated and inserted to non F4 and F5 OAM cells.

| CIN_CL_EN | Function | R/W |
|---|---|---|
| 0 | CRC-10 is not calculated. | default |
| 1 | CRC-10 is calculated and inserted into non F4 and F5 OAM cells sourced from the Multipurpose Port. | |

## 13.5 Management Cell Segmentation

There are Virtual Path (VP) flows (F4) and Virtual Channel (VC) flows (F5) between connection endpoints that are defined as end-to-end OAM flows.

A virtual circuit, VCI=4, is dedicated to carry F4 OAM cells. For a VC flow (F5) a specific VCI cannot be used because all VCIs are available to users in the VC service. Therefore, the Payload Type (PTI) is used to differentiate between the end-to-end (PTI=110B) and segment (PTI=101B) flows in a VC. F5 OAM cells are transmitted on the same VPI/VCI as the user cells.

### F4 OAM Cell

F4 OAM cells are transmitted on VCI=4 and uses CRC-10 polynomial. F4 OAM cells sourced from the PCI Host must set the CT[1:0] bits in the TD to 01B and the CRC[1:0] bits to 01B to select CRC-10 polynomial.

F4 OAM cells can also be sourced from the Multipurpose Port if the CINPORT_EN bit of **TALP Control (0x220)** is set to one. The CIN_F4_EN bit of **TALP Control (0x220)** controls whether CRC-10 field is inserted into the F4 OAM cells sourced from the Multipurpose Port.

| CIN_F4_EN | Function | R/W |
|---|---|---|
| 0 | CRC-10 is not calculated. | default |
| 1 | CRC-10 is calculated and inserted into F4 OAM cells sourced from the Multipurpose Port. | |

### F5 OAM Cell

The TD that describes a segmented OAM F5 flow cell in the host memory must set the CT[1:0] bits to 10B and the CRC[1:0] bits to 01B to select CRC-10 polynomial. The TD that describes an end-to-end OAM F5 flow cell in the host memory must set the CT[1:0] bits to 11B and the CRC[1:0] bits to 01B to select CRC-10 polynomial.

F5 OAM cells can also be sourced from the Multipurpose Port if the CINPORT_EN bit of **TALP Control (0x220)** is set to one. The CIN_F5_EN bit of **TALP Control (0x220)** controls whether CRC-10 field is inserted into the F5 OAM cells sourced from the Multipurpose Port.

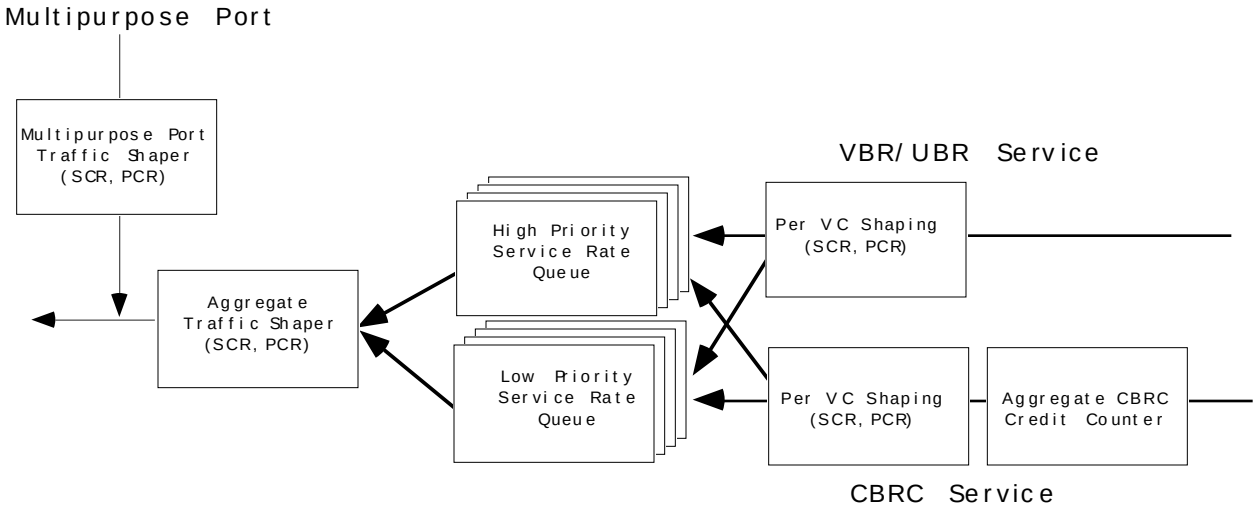
| CIN_F5_EN | Function | R/W |
|---|---|---|
| 0 | CRC-10 is not calculated. | default |
| 1 | CRC-10 is calculated and inserted into F5 OAM cells sourced from the Multipurpose Port. | |

# 14 TRAFFIC SHAPING

The LASAR-155 uses a numbers of traffic shapers to enforce the transmission of ATM cells in accordance with the negotiated traffic parameters.

## 14.1 Traffic Shapers Overview

Figure 14.1: Traffic Shaping Overview




### PCI Host Cell Traffic

ATM cells can be sourced from either the PCI Host or from the Multipurpose Port. Cells sourced from the PCI Host are subject to a number of traffic shapings. The traffic shapers must be initialized in the following order:

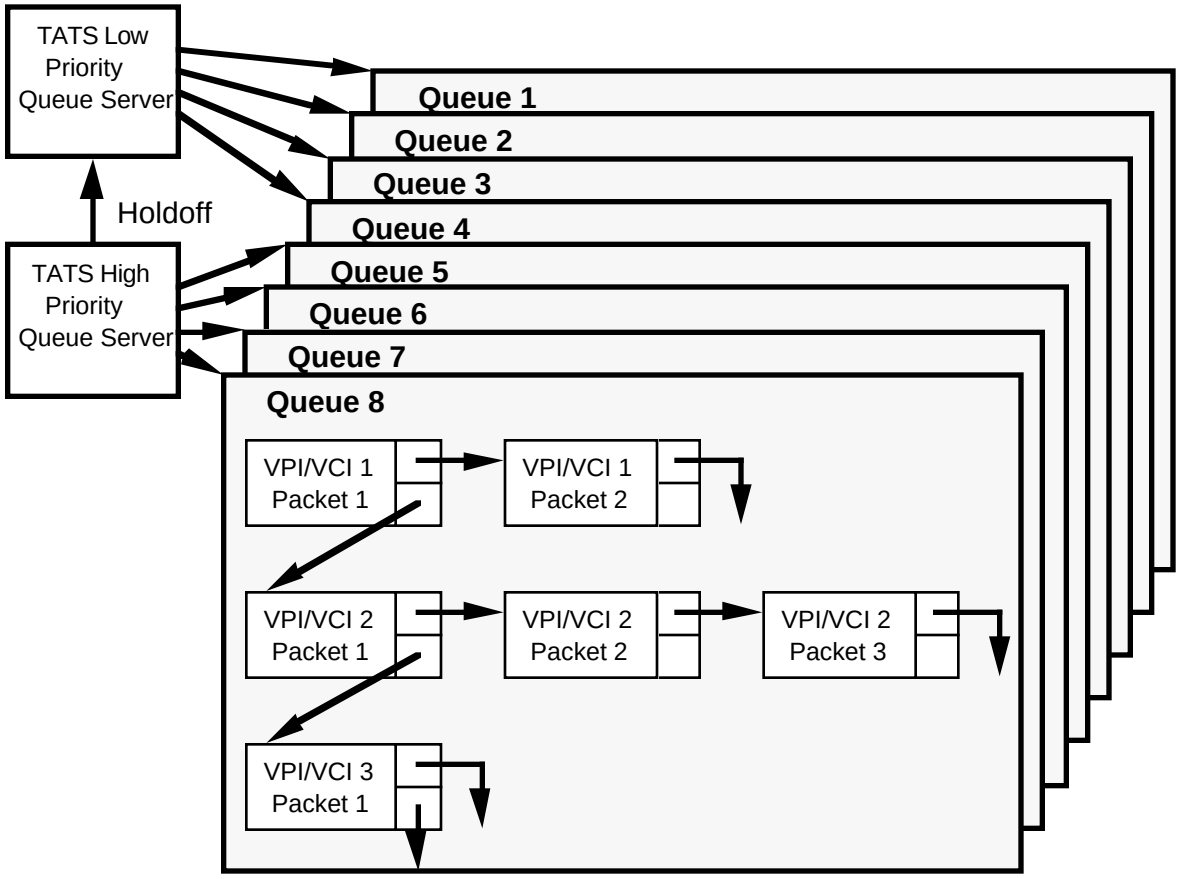
- Define the Peak Cell Rate for one or more of the Service Rate Queues (SRQ)
- If Credit Based Rate Control (CBRC) service is required, the parameters for the Aggregate CBRC Credit Counters must be initialized
- Define the PCR and Sustainable Cell Rate (SCR) for the Aggregate Traffic Shaper.
- For each VC, select a SRQ and specify the VC_PCR, VC_SCR and maximum burst size. Each VC must be configured to use either the VBR or CBRC service.[2]

[2] Unspecified Bit Rate (UBR) service can be selected by using the low priority SRQs.

## 14.2 Service Rate Queues

There are eight Service Rate Queues (SRQ). Four SRQs are grouped as high priority queues and the four remaining queues are grouped as low priority queues as illustrated in Figure 14.2.

Figure 14.2: Peak Cell Rate Queues Diagram




Servicing of high priority queues preempts servicing of low priority queues. Once LASAR-155's Transmit ATM Traffic Shaper (TATS) begins to service a high priority queue, the queue will be completely serviced before another queue is serviced. For both low and high priority queues, VCs on the same queue are serviced on a round-robin fashion. If the high priority queues consume all the available link bandwidth, the low priority queues are allowed to starve. The starvation condition can generate an interrupt on the SRQ*N*_STARVI (*N*=1 to 8) bit of **TATS Interrupt Status (0x244)** if it is enabled by the QSTARV_EN bit of **TATS Control/Interrupt Enable (0x240)**.

A peak cell rate is specified for each Service Rate Queue (SRQ PCR). Each VC specifies its own PCR (VC_PCR) by selecting whether to use 100%, 50% or 25% of the peak cell rate from one of the eight Service Rate Queues.

**SRQ Peak Cell Rate**

This section describes the procedure to initialize the PCR for SRQ 1. Initialization of PCR for SRQs 2 to 8 follows a similar procedure.

Transmission on the Service Rate Queue 1 is controlled by the SRQ1_EN bit of **TATS Service Rate Queue Enables (0x248)**. Before PCR can be initialized or changed, transmission on this SRQ must be disabled by setting SRQ1_EN to logic zero.

| SRQn_EN | Function | R/W |
|---|---|---|
| 0 | Transmission on all VCs associated with the corresponding Service Rate Queue is squelched. | default |
| 1 | Transmission on all VCs associated with the Service Rate Queue are allowed to transmit at the peak cell rate. | |

Peak Cell Rate (PCR) is measured in units of cells/second. Specifically, it specifies the minimum intercell spacing in seconds (i.e. the time interval from the first bit of one cell to the first bit of the next cell).

Alternatively, since each ATM cell carries 53 bytes, or 424 bits of data, PCR can be expressed in bits per second (bps) as:

```
PCR = 424 * (1/t) bps
```

When specifying a PCR, the transmission period, 1/t, is specified using a scaled down SYSCLK. The prescale value to be applied to the SYSCLK is specified in the SRQ1_PS[2:0] bits of **TATS Service Rate Queue 1 Parameter (0x24C)**.

| SRQ1_PS[2:0] | Function | R/W |
|---|---|---|
| 000B | None | default |
| 001B | 4 | |
| 010B | 16 | |
| 011B | 64 | |
| 100B | 256 | |
| 101B | 1024 | |
| 11XB | Reserved | |

The SRQ PCR is specified, in prescaled SYSCLK cycles, in SRQ1_CNT[7:0] bits of **TATS Service Rate Queue 1 Parameters (0x24C)**

```
PCR = 424 * (SYSCLK/Prescale) * (SRQ1_CNT[7:0]+1) bps
```

A wide range of PCRs can be specified by varying the values of SRQ1_CNT[7:0] and SRQ1_PS[2:0]. Examples of PCR configurations are provide in the LASAR-155 Longform Datasheet. Once SRQ PCR is selected, the transmission on the Service Rate Queue 1 must be enabled by setting the SRQ1_EN bit of **TATS Service Rate Queue Enables (0x248)** to logic one.

## 14.3 Per VC Traffic Parameters

In the provisioning of a transmit VC, a number of traffic shaping parameters must be specified in the Transmit VC Parameter Table. This section describes the steps to define VC_PCR, VC_SCR and maximum burst size.

### VC_PCR

Specifying the Peak Cell Rate for each VC involves the following steps:

- The VC must be associated with one of the eight Service Rate Queues. This is specified in the SRQ[3:0] bits of **COPS VC Control and Status (0x294)**. The SRQ must be initialized before it can be used by a VC.

| SRQ[3:0] | Function | R/W |
|---|---|---|
| 0000B | SRQ1 | default |
| 0001B | SRQ2 | |
| 0010B | SRQ3 | |
| 0011B | SRQ4 | |
| 0100B | SRQ5 | |
| 0101B | SRQ6 | |
| 0110B | SRQ7 | |
| 0111B | SRQ8 | |
| 1XXXB | Reserved | |

- The VC_PCR is specified as a fraction of SRQ's PCR in SUB_SRQ_R[3:0] bits of **COPS VC Control and Status (0x294)**.

| SUB_SRQ_R[3:0] | Function | R/W |
|---|---|---|
| 0000B | VC_PCR = SRQ PCR | default |
| 0001B | VC_PCR = 1/2 * SRQ PCR | |
| 0010B | VC_PCR = 1/4 * SRQ PCR | |
| 0011B | Reserved | |
| 01XXB | Reserved | |
| 1XXXB | Reserved | |

Note,

The LASAR-155 allows for the over-subscription of VC_PCRs. That is, the sum of VC_PCR for all the provisioned Transmit VC can be greater than the line rate.

### Per VC Sustainable Cell Rate

Sustainable Cell Rate (SCR) is the maximum average rate that a burst, on-off traffic source can be sent at. Each VC is provided with a token bucket. Tokens are added to the bucket at a Sustainable Cell Rate (SCR). SCR is specified, on a per VC basis, as a fraction of the VC_PCR by the UTIL[3:0] bits of **COPS VC Parameters (0x298)**.

| UTIL[3:0] | Function | R/W |
|---|---|---|
| 0000B | VC_PCR | default |
| 0001B | 1/2 * VC_PCR | |
| 0010B | 1/3 * VC_PCR | |
| 0011B | 1/4 * VC_PCR | |
| 0100B | 1/5 * VC_PCR | |
| 0101B | 1/6 * VC_PCR | |
| 0110B | 1/7 * VC_PCR | |
| 0111B | 1/8 * VC_PCR | |
| 1XXXB | Reserved | |

One token is consumed for each cell transmitted. If the token bucket is not empty, cell transmission is allowed up to VC_PCR. If the token bucket is empty, cell transmission is allowed at the replenish rate, which is SCR.

### Maximum Burst Size

Maximum Burst Size (MBS) is the maximum number of cells that can be sent at the peak cell rate. The token bucket size defines the Maximum Burst Size (MBS) for a VC and is defined in the BUCKET_DEF[11:0] bits of **COPS VC Parameters (0x298)**. For the transmission to occur the BUCKET_DEF must be at least one. If BUCKET_DEF is zero, no transmission occurs.

The equation below describes how MBS can be calculated using BUCKET_DEF and UTIL. The result should be truncated to an integer.

```
MBS = (BUCKET_DEF) * UTIL
```

The ranges of MSB for selected UTIL values are provided below:

| UTIL | Min MBS | BUCKET_DEF | Max MBS | BUCKET_DEF |
|---|---|---|---|---|
| 1/8 | 1 | 8 | 511 | 4088 |
| 1/7 | 1 | 7 | 585 | 4095 |
| 1/6 | 1 | 6 | 682 | 4092 |
| 1/5 | 1 | 5 | 819 | 4095 |
| 1/4 | 1 | 4 | 1023 | 4092 |
| 1/3 | 1 | 3 | 1365 | 4095 |
| 1/2 | 1 | 2 | 2047 | 4094 |
| 1/1 | unbound | 1 | unbound | 1 |

## 14.4 Aggregate Cell Traffic Shaper

The aggregate cell traffic from the PCI Host can be subject to an aggregate traffic shaping. This is controlled by the AGG_PR_EN bit of **TALP Control (0x220)**.

| AGG_PR_EN | Function | R/W |
|---|---|---|
| 0 | Aggregate traffic shaping is disabled. | default |
| 1 | Aggregate traffic shaping is enabled. | |

Aggregate PCR is specified in the same manner as that described in the Service Rate Queue. The prescale to be applied to the SYSCLK is specified in the APR_PS[2:0] bits of **TALP Aggregate Peak Cell Rate (0x22C)**. The Aggregate PCR is specified in the APR_CNT[7:0] bits of **TALP Aggregate Peak Cell Rate (0x22C)**.

Aggregate SCR is specified as a fraction of the aggregate PCR. ASR_CNT[2:0] of **TALP Aggregate Bucket Capacity (0x230)** specify the value of the aggregate SCR as a fraction of the aggregate PCR.

| ASR_CNT[2:0] | Function | R/W |
|---|---|---|
| 000B | Aggregate SCR (ASCR) is the same as Aggregate PCR. | default |
| 001B | ASCR = APCR/2 | |
| 010B | ASCR = APCR/3 | |
| 011B | ASCR = APCR/4 | |
| 100B | ASCR = APCR/5 | |
| 101B | ASCR = APCR/6 | |
| 110B | ASCR = APCR/7 | |
| 111B | ASCR = APCR/8 | |

The maximum burst size for the aggregate traffic is controlled by the size of the aggregate token bucket which is specified in AB_CAP[7:0] of **TALP Aggregate Bucket Capacity (0x230)**.

## 14.5 Credit Based Rate Control

VCs are classified as either Variable Bit Rate (VBR) or Credit Based Rate Control (CBRC). Regardless of classification, both types of VCs are subject to both PCR and SCR traffic shaping. However, in addition to PCR and SCR shaping, an aggregate credit counter is used to allow the network to back pressure transmission on all of CBRC VCs. The aggregate CBRC traffic enforcement is controlled by the AGR_CBRC_EN bit of **TATS Control/Interrupt Enable (0x240)** .

| AGR_CBRC_EN | Function | R/W |
|---|---|---|
| 0 | The aggregate CBRC traffic enforcement is turned off. CBRC traffic is treated in the same manner as VBR traffic. | default |
| 1 | The aggregate CBRC VC traffic enforcement using the credit basket scheme is enforced. | |

The CBRC credit basket size is controlled by the CBRC_CNT[7:0] bits of **TATS Control/Interrupt Enable (0x240)**. The default selects the maximum credit basket size of 255.

CBRC credits are replenished by the network using the second most significant bit of the GFC field in the received cell header. When the CBRC credit basket is not empty, each CBRC cell transmission consumes a single credit regardless of the VC. If the CBRC credit basket is empty, transmission is suspended on all the CBRC VCs.

## 14.6 Multipurpose Port Traffic Shaper

When the Multipurpose Port is configured in non-bypass mode, either CBR cells and/or OAM cells can be inserted into the cell stream. This is controlled by the CINPORT_EN bit of **TALP Control (0x220)**.

| CINPORT_EN | Function | R/W |
|---|---|---|
| 0 | Cells cannot be sourced on the Multipurpose Port. | default |
| 1 | Cells can be sourced on the Multipurpose port. | |

When the port indicates that a cell is ready to be transmitted, the TALP block either inserts the cell into the aggregate cell stream at the earliest opportunity or waits until no cells are being sourced from the PCI Host before inserting the cell into the aggregate cell stream. This is controlled by the CINPORT_PR bit of **TALP Control (0x220)**.

| CINPORT_PR | Function | R/W |
|---|---|---|
| 0 | The cells sourced from the PCI Host take priority over cells sourced from the Multipurpose Port. | default |
| 1 | Cells sourced from the Multipurpose Port take priority over cells sourced from the PCI Host. | |

The CIN_PR_EN bit of TALP Control (0x220) enables control of cell transmission rate of the traffic sourced from the Multipurpose Port.

| CIN_PR_EN | Function | R/W |
|---|---|---|
| 0 | Traffic enforcement on Multipurpose Port is disabled. | default |
| 1 | Traffic enforcement on Multipurpose Port is enabled. | |

Multipurpose Port's PCR is specified in the same manner as that described in the Service Rate Queue section. The prescale to be applied to the SYSCLK is specified in CIN_PS[2:0] of **TALP Multipurpose Port Peak Cell Rate (0x234)**. The Aggregate PCR is specified in CINPR_CNT[7:0] bits of **TALP Multipurpose Port Peak Cell Rate (0x234)**.

Multipurpose Port's SCR is specified as a fraction of the PCR. CINSR_CNT[2:0] bits of **TALP Multipurpose Port Bucket Capacity (0x238)** specify the value of the SCR with respect to the PCR.

| CINSR_CNT | Function | R/W |
|---|---|---|
| 000B | SCR is the same as PRC | default |
| 001B | SCR = PCR/2 | |
| 010B | SCR = PCR/3 | |
| 011B | SCR = PCR/4 | |
| 100B | SCR = PCR/5 | |
| 101B | SCR = PCR/6 | |
| 110B | SCR = PCR/7 | |
| 111B | SCR = PCR/8 | |

The maximum burst size for the Multipurpose Port traffic is controlled by the size of the token bucket which is specified in CINB_CAP[7:0] bits of **TALP Multipurpose Port Bucket Capacity (0x238)**.

**Multipurpose Port Cell Traffic**

The Multipurpose Port allows CBR cells and OAM cells generated from an external source to be inserted into the cell stream. When this feature is enabled, the priority of the Multipurpose Port cell traffic and PCI Host cell traffic is determined by the CINPORT_PR bit of the **TALP Control (0x220)**.

| CINPORT_PR | Function | R/W |
|---|---|---|
| 0 | The cells sourced from the PCI Host take priority over cells sourced from the Multipurpose Port. | default |
| 1 | Cells sourced from the Multipurpose Port take priority over cells sourced from the PCI Host. | |

The cell traffic from the Multipurpose Port is shaped by the Multipurpose Port Traffic Shaper.

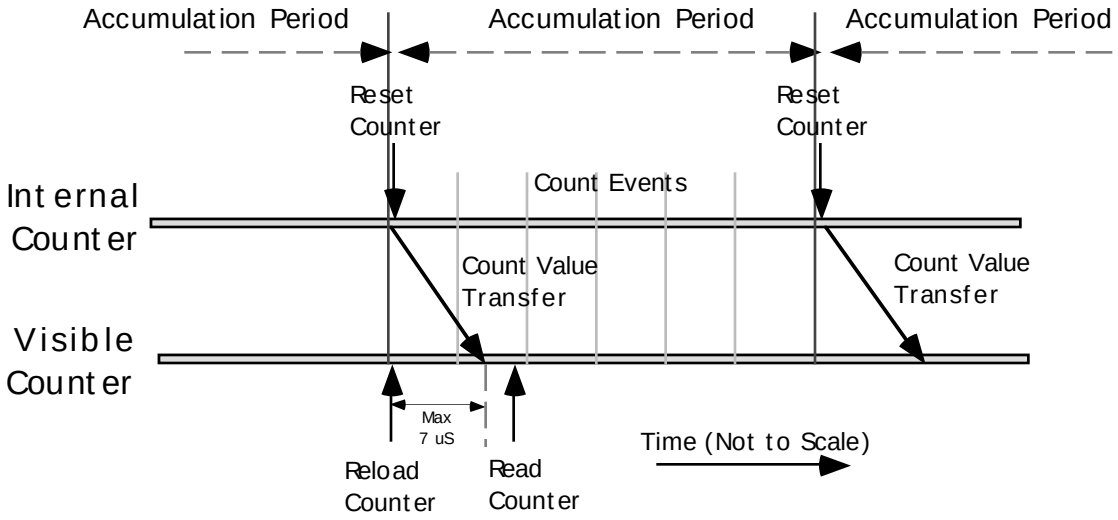
# 15 COUNTERS OVERVIEW

This section describes the counters that LASAR-155 maintain to accumulate a number of statistics including error and successful events at SONET, ATM and AAL Layers. The counters can be used by upper layer software such as a network management tool to report errors, isolate network problems and calculate network performance.

## 15.1 Counter Overview

LASAR-155 maintains a number of saturating counters that record error and successful events. Periodically the device driver must poll these counters before they are overflowed. Figure 15.1 illustrates the events when the LASAR-155 counters are polled.

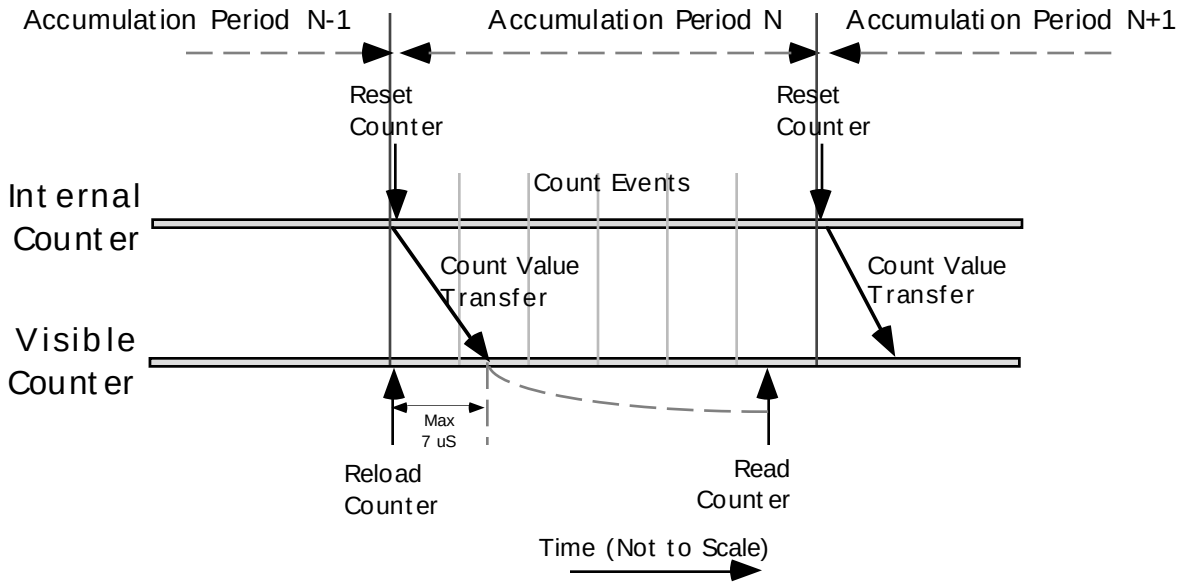
Figure 15.1: Events in Polling a Counter




The device driver initiates a counter reload request by writing to the **LASAR-155 Master Reset and Identity / Load Meters (0x000)** register as described in the following sections. As noted in Figure 15.1, there is a delay of approximately 7 µs to complete the transfers from internal counters to the visible counters.

An alternative polling method reads the counters first before initiating a reload command. As the counters are always read one accumulation period behind, the counter transfer latency can be ignored (for accumulation periods longer than 7 µs).

Figure 15.2: Alternative Polling Procedure




**Reload Counters**

Writing to the **LASAR-155 Master Reset and Identity / Load Meters (0x000)** register initiates transfers for all counters.

**Counter Polling Frequency**

SONET counters and ATM cell counters are sized such that they must be polled at least once per second. Packet registers are sized such that they must be polled at least once per second given that every AAL5 packet is in error and the average packet size is 8 cells. If packet characteristics are different and the exact AAL5 packet count is required, the user must poll once every 125 ms.

When the RACP block declares a loss of cell delineation (LCD), no receive statistics are accumulated except for time-out events.

## 15.2 LASAR-155 Counters

This section lists the counters maintained by LASAR-155 and identifies the registers where the count values are stored.

**SONET Error Counters**

- The number of section BIP-8 errors (B1) is maintained in **RSOP Section BIP-8 LSB (0x048)** and **RSOP Section BIP-8 MSB (0x04C)** registers.
- The number of section BIP-8/24 errors (B2) is maintained in **RLOP Line BIP-8/24 LSB (0x068)**, **RLOP Line BIP-8/24 (0x06C)** and **RLOP Line BIP-8/24 MSB (0x070)** registers.

- The number of line FEBE errors (Z2) is maintained in **RLOP Line FEBE LSB (0x074), RLOP Line FEBE (0x078)** and **RLOP Line FEBE MSB (0x07C)** registers.
- The number of path BIP-8 errors (B3) is maintained in **RPOP Path BIP-8 LSB (0x0E0)** and **RPOP Path BIP-8 MSB (0x0E4)** registers.
- The number of path FEBE errors (G1) is maintained in **RPOP Path FEBE LSB (0x0E8)** and **RPOP Path FEBE MSB (0x0EC)** registers.

**ATM Cell Counters**

- The number of ATM cells received is maintained in **RACP Receive Cell Counter (LSB) (0x158), RACP Receive Cell Counter (0x15C)** and **RACP Receive Cell Counter (MSB) (0x160)**.
- The number of ATM cells that are inserted into the SPE (excluding Idle/unassigned cells) is maintained in **TACP Transmit Cell Counter (LSB) (0x190), TACP Transmit Cell Count (0x194)** and **TACP Transmit Cell Count (MSB) (0x198)**.

**ATM Cell Error Counters**

- The number of ATM cells received on unprovisioned VCs is stored in **SAR PMON Receive Unprovisioned VPI/VCI Errors (LSB) (0x1C8)** and **SAR PMON Receive Unprovision VPI/VCI Errors (MSB) (0x1CC)** registers.
- The number of ATM cells with CRC-10 errors is collected in **SAR PMON Receive CRC-10 Errors (LSB) (0x1D0)** and **SAR PMON Receive CRC-10 Errors (MSB) (0x1D4)** registers.
- The number of ATM cells with correctable HEC errors is stored in **RACP Correctable HEC Error Count (0x150)**.
- The number of ATM cells with uncorrectable HEC error is stored in **RACP Uncorrectable HEC Error Count (0x154)**.

**AAL5 Packet Counters**

- The number of CPAAL5_PDU successfully reassembled is maintained in **SAR PMON Receive PDU Count (0x1F4)**.
- The number of CPAAL5_PDU successfully segmented is maintained in **SAR PMON Transmit PDU Count (0x1FC)**.

**AAL5 Packet Error Counters**

- The number of CPAAL5_PDUs received with non-zero CPI fields is collected in **SAR PMON Receive Non Zero Common Part Indicator Errors (0x1D8)**.

- The number of CPAAL5_PDUs received with a non-zero LENGTH field and a LENGTH field which did not match the actual received SDU length is collected in **SAR PMON Receive SDU Length Errors (0x1DC)**.
- The number of CPAAL5_PDUs received with CRC-32 errors received is maintained in **SAR PMON Receive CRC-32 Errors (0x1E0)**.
- The number of CPAAL5_PDUs received with SDU exceeding the user programmable maximum length, as specified in **RALP Max Rx SDU Length (0x20C),** is maintained in **SAR PMON Receive Oversize SDU Errors (0x1E4)**. The maximum SDU length is enforced only if the MRSDU_EN bit in the **RALP Control (0x200)** is enabled.
- The number of received CPAAL5_PDUs that were time out during reassembly is maintained in **SAR PMON Receive PDU Timeout Errors (0x1E8)**.
- The number of receiving CPAAL5_PDUs that were aborted by the sender (LENGTH=0) is maintained in **SAR PMON Receive PDU Abort Errors (0x1EC)**.
- The number of oversized transmit CPAAL5_PDUs that were aborted is maintained in **SAR PMON Transmit Oversize SDU Errors (0x1F8)**. The maximum transmit SDU Length is specified in **PCID Max Tx SDU Length (0x3B0)**.

**Receive Buffer Error Counters**

- The number of times the LASAR-155 ran out of Receive Management Descriptors is maintained in RMBE[7:0] of **SAR PMON Receive Buffer Errors (0x1F0)**.
- The number of times the LASAR-155 ran out of Receive Packet Descriptors is maintained in RPBE[7:0] of **SAR PMON Receive Buffer Errors (0x1F0)**.

# 16 VC PARAMETER TABLE ACCESS

The LASAR-155 provides an internal control memory space to store the parameters associated with the 128 transmit and 128 receive VCs. This section describes the procedure to access the transmit and receive parameter tables.

## 16.1 Table Initialization

At system start up time, the control memory for the VC parameter tables must be allocated and initialized. The initialization procedure is described in "Section 23. Master Reset Procedure".

Note:
After a reset, the TRMEN bit of **PCID Control (0x300)** must be set to logic one in order to access the VC Parameter Table to provision and unprovision VCs.

## 16.2 Table Index

Since only 128 VCs are allowed, a subset of the VPI and VCI field bits are concatenated to form a 7 bit unique index. Selection of which VPI and VCI bits are used in the formation of the VPI/VCI index is controlled by the NVCI[3:0] and the NVPI[3:0] bits of **COPS Control (0x280)**. The same indexing scheme is used to access both the transmit and receive VC parameter tables.

| NVCI[3:0] | Function | R/W |
|---|---|---|
| 00XXB | Reserved | |
| 0100B | Reserved | |
| 0101B | 5 VCI bits | |
| 0110B | 6 VCI bits | |
| 0111B | 7 VCI bits | |
| 1XXXB | Reserved | |

| NVPI[3:0] | Function | R/W |
|---|---|---|
| 0000B | 0 VPI bits | default |
| 0001B | 1 VPI bits | |
| 0010B | 2 VPI bits | |
| 0011B | Reserved | |
| 0100B | Reserved | |
| 01XXB | Reserved | |
| 1XXXB | Reserved | |

The valid combinations are:

| NVPI[3:0] | NVPI[3:0] | 7 bit Index |
|---|---|---|
| 0101B | 0010B | 5 VCI + 2 VPI |
| 0110B | 0001B | 6 VCI + 1 VPI |
| 0111B | 0000B | 7 VCI + 0 VPI[3] |

### 16.3 Table Selection

The RX/TXB bit of **COPS Parameter Access Control (0x284)** selects either the receive or transmit VC parameter table.

| RX/TXB | Function | R/W |
|---|---|---|
| 0 | The transmit VC parameter table is accessed. | default |
| 1 | The receive VC parameter table is accessed. | |

### 16.4 Read/Write Operation Selection

The RD/WRB bit of **COPS Parameter Access Control (0x284)** selects whether to perform read or write operation

| RD/WRB | Function | R/W |
|---|---|---|
| 0 | Write operation. | default |
| 1 | Read operation. | |

### 16.5 Read Operation

The following procedure must be performed in order to read the parameters associated with an open VC:

- Poll the BUSY bit of the **COPS Parameter Access Control (0x284)** to ensure that no table access is in progress.

| BUSY | Function | R |
|---|---|---|
| 0 | No read or write operation is in progress. | |
| 1 | A read or write operation is in progress. | |

- Specify a 7 bit table index in the VCNUM[6:0] bits of the **COPS VC Number (0xA2)**
- To read from the transmit VC table, write a 0x0002 to the **COPS Parameter Access Control (0x284)** register[4]. This sets the RD/WRB bit to 1 to select the read operation and RX/TXB bit to 0 to select the transmit table.
- To read from the receive VC table, write a 0x0003 to the **COPS Parameter Access Control (0x284)** register. This sets the RD/WRB bit to 1 to select the read operation and RX/TXB bit to 1 to select the receive table.

[3]In LAN applications where the VPI is typically set to zero, the least significant 7 bits of the VCI can be used to formulate the internal VC index.

[4]Refer to the LASAR-155 databook for the bit positions of RD/WRB and RX/TXB bits in the COPS Parameter Access Control register.

- A write to the **COPS Parameter Access Control (0x284)** initiates the loading of the parameters for the table entry associated with the 7 bit index. The read operation is completed when the BUSY bit of the **COPS Parameter Access Control (0x284)** is returned to logic zero.
- The parameter values can be accessed from **COPS VC Number (0x288), COPS VPI (0x28C), COPS VCI (0x290), COPS VC Control and Status (0x294), COPS VC Parameters (0x298)**. Note that these registers are interpreted differently for transmit and receive VC parameter tables. Please refer to the LASAR-155 Longform Datasheet for additional information.

## 16.6 Write Operation

The following steps describe a write operation to either the transmit or receive VC parameter table.

- Poll the BUSY bit of the **COPS Parameter Access Control (0x284)** to ensure that no table access is in progress.
- Specify the 7 bit table index in the VCNUM[6:0] bits of the **COPS VC Number (0xA2)**
- Enter the new parameter values in **COPS VC Number (0x288), COPS VPI (0x28C), COPS VCI (0x290), COPS VC Control and Status (0x294), COPS VC Parameters (0x298)**. Note that these registers are interpreted differently for transmit and receive VC parameter tables. Please refer to the LASAR-155 Longform Datasheet for additional information.
- To write to the transmit VC table, write a 0x0000 to the **COPS Parameter Access Control (0x284)** register[5]. This sets the RD/WRB bit to 0 to select the write operation and RX/TXB bit to 0 to select the transmit table.
- To write to the receive VC table, write a 0x0001 to the **COPS Parameter Access Control (0x284)** register. This sets the RD/WRB bit to 0 to select the write operation and RX/TXB bit to 1 to select the receive table.
- A write to the **COPS Parameter Access Control (0x284)** initiates the transfer of the parameters from the COPS registers to the for the table entry associated with the 7 bit index. The write operation is completed when the BUSY bit of the **COPS Parameter Access Control (0x284)** is returned to logic zero.

## 16.7 Modify Operation

Use the read/modify/write cycle to change one or more fields in the transmit or receive VC parameter table.

[5]Refer to the LASAR-155 databook for the bit positions of RD/WRB and RX/TXB bits in the COPS Parameter Access Control register.

## 16.8 Tx VC Parameter Table

This section describes the fields of the Rx VC Parameter Table.

| Field | Description | Map to Register |
|---|---|---|
| VCNUM[6:0] | Index to the table. | COPS VC Number (0x288) |
| VPI[7:0] | Value inserted in the Virtual Path Identifier field of cell header. | COPS VPI (0x28C) |
| VCI[[15:0] | Value inserted in the Virtual Channel Identifier (VCI) in the cell header. | COPS VCI (0x290) |
| GFC[3:0] | The default value in the GFC field of cell header. | COPS VPI (0x28C) |
| PTI[2:0] | The default value of the PTI field in the cell header. Can be overridden by the Congestion (CG) bit and the Cell Type (CT[1:0]) bits of the Transmit Descriptor. | COPS VPI (0x28C) |
| CLP | The default value of the CLP field in the cell header. The default value can be overridden by the CLP bit of the Transmit Descriptor. | COPS VPI (0x28C) |
| SRQ[3:0] | Identify the Service Rate Queue to be associated with. | COPS VC Control and Status (0x294) |
| SUB_SRQ_R[3:0] | Specify the VC's PCR as a scaled version of the SRQ's PCR. | COPS VC Control and Status (0x294) |
| STATUS[1] | Indicates that the current VC is actively segmenting a packet. | COPS VC Control and Status (0x294) |
| STATUS[0] | Indicates segmentation is enabled on this VC. | COPS VC Control and Status (0x294) |
| VC_SEG_EN | Control bit to provision and unprovision transmit VCs. | CTL[3] bit of COPS VC Control and Status (0x294) |
| VC_CBRC_EN | Control bit to select whether a VC is configured for VBR or CBRC service. | CTL[2] bit of COPS VC Control and Status (0x294) |
| BUCKET_DEF[11:0] | Specify the size of the token bucket to determine the Maximum Burst Size (MBS) for the VC. | COPS VC Parameter (0x298) |
| UTIL[3:0] | Specify the Sustainable Cell Rate (SCR) as a fraction of VC_PCR. | COPS VC Parameters (0x298) |

## 16.9 Rx VC Parameter Table

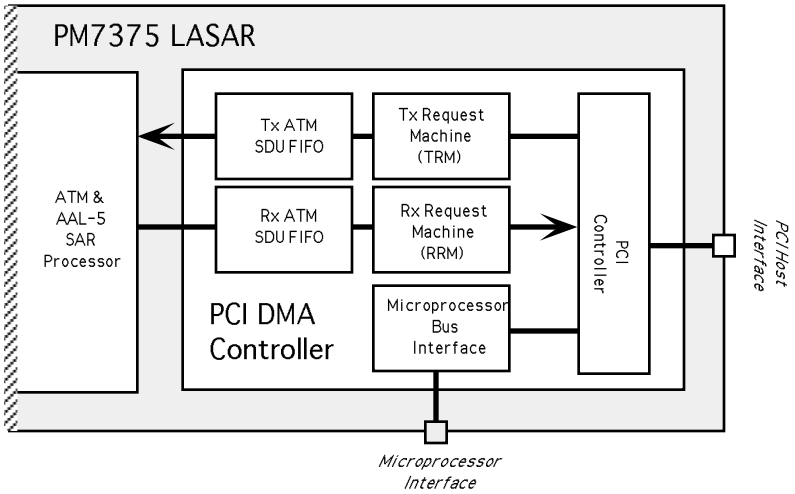
This section describes the contents of the Rx VC Parameter Table.

| Field | Description | Map to Register |
|---|---|---|
| VCNUM[6:0] | Index to the table. | COPS VC Number (0x288) |
| VPI[7:0] | Value compared to the Virtual Path Identifier field of cell header. | COPS VPI (0x28C) |
| VCI[[15:0] | Value compared to the Virtual Channel Identifier (VCI) in the cell header. | COPS VCI (0x290) |
| TO_ABORT | Status bit to indicate the reassembly of the current packet was terminated due to a time-out condition. | STATUS[7]bit of COPS VC Control and Status (0x294) |
| MAX_SDU_ABORT | Status bit to indicate the reassembly of the current packet was terminated due the packet length larger than the user specified SDU length. | STATUS[6] bit of COPS VC Control and Status (0x294) |
| CLP_RCVD | Status bit to indicate the packet under reassembly includes a cell with its CLP bit equal to one. | Status[3] bit of COPS VC Control and Status (0x294) |
| CG_RCVD | Status bit to indicate the packet under reassembly includes a cell that experienced congestion. | STATUS[2] bit of COPS VC Control and Status (0x294) |
| ACTV_RSMB | Status bit to indicate a packet is actively being reassembled on the VC. | STATUS[1] bit of COPS VC Control and Status (0x294) |
| VC_REAS_EN | Control bit used to enable and disable cell processing and packet reassembly. | CTL[7] bit of COPS VC Control and Status (0x294) |
| VC_PDU_TO_EN | Control bit to enable packet aging on a per VC basis. | CTL[4] bit of COPS VC Control and Status (0x294) |
| QUEUE_SEL[1:0] | Control bit to indicate whether cells on a VC should be directed to the Receive Packet Queue (PRQ) or the Receive Management Queue (RMQ). | CTL[3:2] bits of COPS VC Control and Status (0x294) |
| VC_PACKET_QUEUE_EN | Control bit to determine whether the user cells are copied to the Receive Packet Queue (RPQ) or not. | CTL[1] bit of COPS VC Control and Status (0x294) |
| CEX_EN | Control bit to select whether the user cells are copied to the Multipurpose Port or not. | CTL[0] bit of COPS VC Control and Status (0x294) |

## 17 PCI DMA CONTROLLER OVERVIEW

The PCI DMA Controller (PCID) facilitates the communication between the PCI Host and the SAR engine as well as between the PCI Host and the microprocessor. Figure 17.1 illustrates the functional blocks of the PCID:

Figure 17.1: PCI DMA Controller Block Diagram




The PCI Controller provides a 32-bit, 33 MHz PCI Local Bus interface that is compliant with Version 2.0 of the PCI Specifications. The PCI Controller supports both bus-master and bus-slave access modes. As a bus master, the PCI Controller uses burst DMA cycles to read or write data on the PCI bus independent of the PCI Host. As a bus slave, the PCI Controller allows the PCI Host to access the LASAR's internal registers, to communicate with the optional microprocessor, or to access external devices when the microprocessor is not present.

Two DMA channels are provided: A Transmit DMA Channel managed by the Tx Request Machine (TRM) and a Receive DMA Channel managed by the Rx Request Machine (RRM). To allow for the short but expected bus latency, up to 8 cells can be prefetched into the Tx ATM SDU FIFO in the transmit direction. A 96 cell Rx ATM SDU FIFO is provided between the Rx ATM & AAL Processor and the RRM to allow for a 270 µs PCI bus latency.

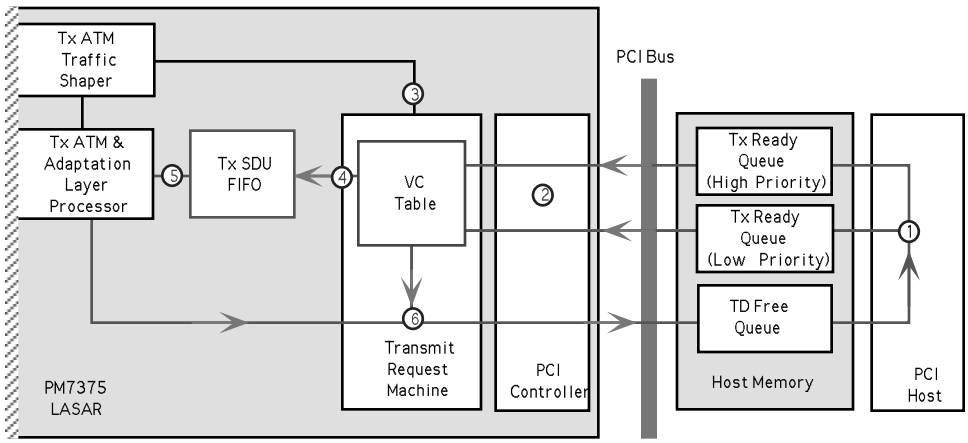
The microprocessor interface is provided for device configuration, ongoing control and monitoring by an external local microprocessor. For applications where local microprocessor control is not required and all device operations are performed by the PCI Host, the microprocessor interface allows the PCI Host to access the LASAR local bus. In the absence of the microprocessor, LASAR local bus can support up to two devices without using any external logic.

## Transmit DMA Channel

The Transmit DMA Channel transfers cell payloads of packet prepared by the PCI Host to the Tx ATM & Adaptation Layer Processor where they are combined with headers to form ATM cells. The Transmit DMA Channel is serviced at the source end by the PCI Host and at the destination end by the TRM.

The PCI Host uses a Transmit Descriptor (TD) to describe a packet or portion of a packet to TRM. The packet exchange mechanism uses three Descriptor Reference Queues as illustrated in Figure 17.2. The TD Free Queue contains TD References (TDRs) that points to unused TDs that the PCI Host can use to prepare packets in the host memory. The Ready Queues contain TDRs that point to TDs that describe packets in the host memory.

Figure 17.2: Transmit DMA Channel




The PCI Host is responsible for creating the data structures used in the DMA channel. All data structures and packet data reside in the host memory. After the VCs are provisioned, the operation of the Transmit DMA Channel is based on the process described below:

1 When the PCI Host is ready to send a packet, it can either use a single transmit buffer or multiple transmit buffers. If a single transmit buffer is used, it must be able to store the entire packet. If multiple transmit buffers are used, each transmit buffer can be smaller than the packet size. However, the sum of the transmit buffers must be equal to or larger than the packet size. For each transmit buffer, the PCI Host uses a free TDRs from the TD Free Queue, update the control attributes, and write the physical address of the transmit buffer. When multiple buffers are used, the PCI Host must chain the TDs together to form a linked list. The PCI Host can place the TDRs into either the High Priority or Low Priority Tx Ready Queue.

2 The TRM checks the content of the Tx Ready Queues via the PCI Controller interface. The TRM services the High Priority Tx Ready Queue and only when it is empty does it service the Low Priority Tx Ready Queue. The TRM

maintains an internal VC Table that keep track of prepared packets in the host memory on a per VC basis.

3 When the line bandwidth is available for a particular VC, the Tx Traffic Shaper places a request to the TRM to read cells from the host memory to the Tx ATM and Adaptation Layer Processor.

4 The TRM burst reads one cell at a time from the host memory into the Transmit SDU FIFO. Up to 8 cells can be buffered in the Tx SDU FIFO to allow for the bus latency.

5 The Tx ATM and Adaptation Layer Processor retrieves cells from the Transmit SDU FIFO and converts them into AAL-5 PDUs.

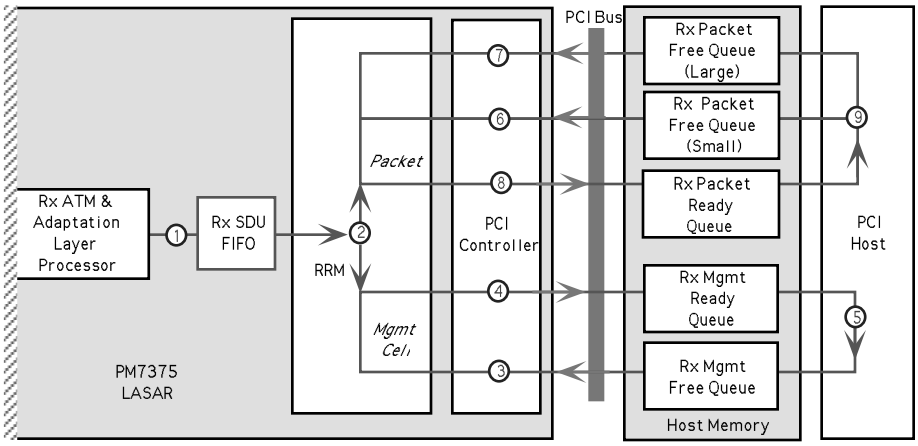
6 When a TDR used to describe the packet is processed, it is returned to the TD Free Queue to be reused. The TRM attaches the status of segmentation with the returning TDR to the PCI Host.

## Receive DMA Channel

The Receive DMA Channel transfers packets received by the Rx ATM and Adaptation Layer Processor to the host memory. The receive DMA channel is serviced at the source end by the Receive Request Machine (RRM) and at the destination end by the PCI Host.

The RRM uses Receive Packet Descriptors (RPD), two Receive Packet Descriptor Reference Free queues, and a Receive Packet Descriptor Reference Ready queue on the packet reassembly path. The RRM uses Receive Management Descriptors (RMD), a Receive Management Descriptor Reference Free Queue and a Receive Management Descriptor Reference Ready Queue on the management cell path. All data structures, reassembled packets, and management cells reside in the host memory. The PCI Host is responsible for the initialization of the data structures used in the Receive DMA Channel.

Figure 17.3: Receive DMA Channel




The operation of the Receive DMA Channel is described below:

1 The RRM receives cells associated with an open VC from the Rx ATM & Adaptation Layer Processor. Received cells are buffered in a 96 cell Receive SDU FIFO to allow for up to 270 µs of latency during bus request/grant cycles.

2 The RRM retrieves a cell from the SDU FIFO and determines whether it is a management cell or part of a packet under reassembly. Management cells and packet cells are processed separately.

**Management Cell Processing**

3 The RRM first retrieves a RMD Reference (RMDR) from the Rx Management Free Queue to describe a management cell in the host memory.

4 The RMDR points to a RMD which points to a buffer in the host memory where the management cell is burst written to. When the transfer of the management cell is completed, the RMDR is moved to the Rx Management Ready Queue and the PCI Host is alerted.

5 Once alerted, the PCI Host retrieves the RMDR from the Rx Management Ready Queue and follows the pointers to the management cell. When the management cell is processed, the PCI Host returns the RMDR back to the Rx Management Free Queue where it can be reused.

**Packet Cell Processing**

To ensure the optimal use of the host memory, the RRM uses two different buffer sizes to store packets in the host memory. The RRM always uses a small buffer to save the head of a packet. If the packet cannot fit completely inside the small buffer, the remainder of the packet is saved using one or more large buffers. The size of the small and large buffers are configurable at runtime and can be adjusted to suit different applications. The steps involved in the receive packet cell processing are described below:

6 When the start of a packet is detected, the RRM retrieves a RPDR from the Small Rx Packet Free Queue. The attributes of the packet, such as VPI/VCI, are copied to the RPD. The content of the packet is burst written to the small buffer referenced to by the RPD.

7 If a packet cannot fit completely inside a small buffer, the RRM retrieves a RPDR from the Large Rx Packet Free Queue.

8 The remainder of the packet is burst written to the large buffer referenced to by the RPD. The new RPD is joined to the tail of the previous RPD to form a linked list. This process is repeated until the entire packet is saved to the host memory.

9 The RRM alerts the PCI Host when the complete packet has been reassembled and all the RPDRs used to describe the packet are in the Rx Packet Ready Queue. The PCI Host retrieves the packet by traversing the RPD linked list and access the data in the buffers. Once processed, the PCI Host returns the first RPD to the Small Receive Packet Free Queue and all following RPDs to the Large Receive Packet Free Queue where they can be reused.

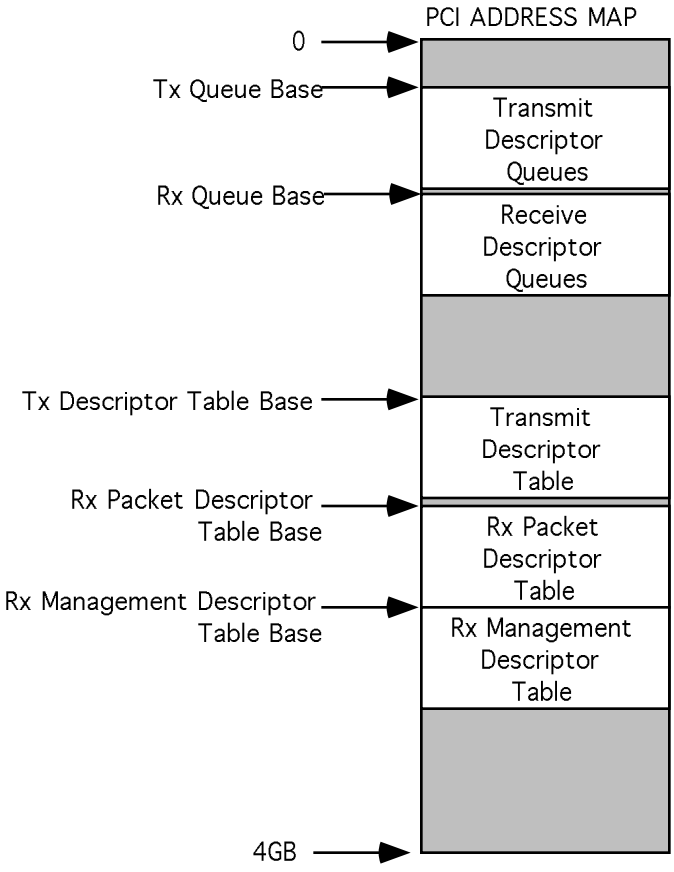
# 18 LASAR-155 DATA STRUCTURE OVERVIEW

A number of data structures are used to communicate packet information between the PCI Host and the LASAR-155.

## 18.1 Data Structures Overview

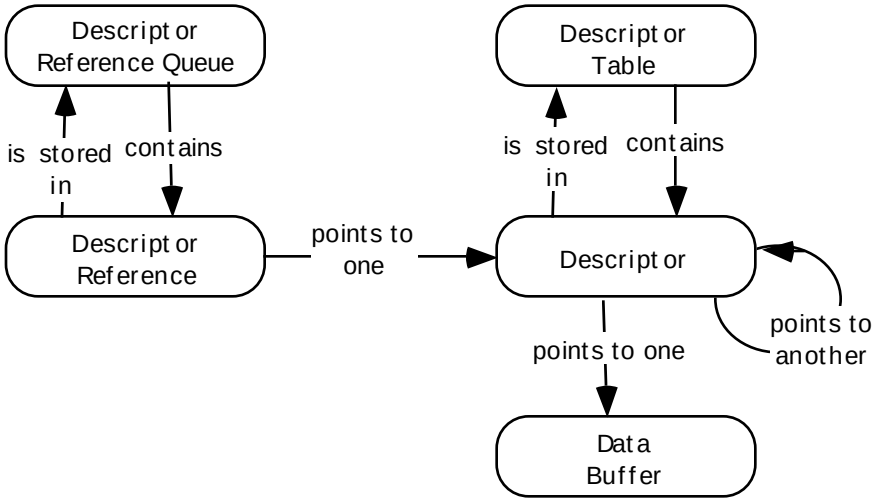
This section provides an overview of the data structures used in the DMA channels. The data structures are Descriptor, Descriptor Table, Descriptor Reference, and Descriptor Reference Queue. The data structures are located in the host memory using a number of base pointers as illustrated in Figure 18.1. The data structures must be allocated and initialized by the PCI Host.

Figure 18.1: Data Structures Memory Map



The general relationships among the data structures are illustrated in Figure 18.2.

Figure 18.2: Data Structure Relationships




In this figure, the direction of the arrow refers to the direction of the relationship. For example, each Descriptor Reference can point to one Descriptor. The relationship between a Descriptor and another Descriptor is non-recursive: Each Descriptor contains a Descriptor Reference field that can be used to point to another Descriptor. Therefore a linked-list of Descriptor can be created using this mechanism.

## 18.2 Descriptor

A Descriptor is a 32 byte data structure that contains control fields, status fields, a pointer to a data buffer and a Descriptor Reference to point to another Descriptor.

Transmit Descriptors (TD) are used in the transmit direction to describe packets ready for segmentation. Receive Packet Descriptors (RPD) are used in the receive direction to describe reassembled packets in the host memory. Receive Management Descriptors (RMD) are used in the receive direction to describe management cells in the host memory.

Descriptors are array elements in the Descriptor Table. For example, Transmit Descriptors are located in the Transmit Descriptor Table whose base address is specified in the Tx Descriptor Table Base. Each element in the descriptor table can be referenced using an offset index with respect to the base address. The offset index is also referred to as the Descriptor Reference. The mechanism to address a Descriptor element in the Descriptor Table is described in the Descriptor Reference section.

## 18.3 Descriptor Table

A Descriptor Table contains one or more Descriptors of the same type. Three Descriptor Tables are used: TD Table, RPD Table and RMD Table. Each Descriptor

Table can be located anywhere within a 32 bit address space using a base address pointer. The base pointer must be aligned on a 32 byte boundary. Therefore, the least significant 5 bits of the base address must be written to logic zero.

The size of the Descriptor Table is determined by the device driver at system initialization. As each descriptor is a 32 byte data structure, the descriptor table size must be a multiple of 32 bytes. The size of each Descriptor Table is implied by the amount of memory allocated.

The base pointers for the Descriptor Tables are listed in the table below:

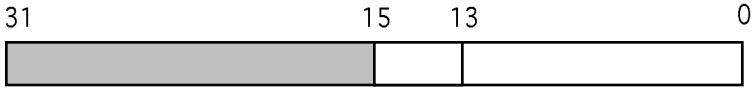
| Register Name | Register Offset | Function | Maximum Descriptors |
|---|---|---|---|
| PCID Receive Packet Descriptor Table Base | 0x314 | RPDTB[31:0] | 16,384 |
| PCID Rx Management Descriptor Table Base | 0x318 | RMDTB[31:0] | 16,384 |
| PCID Tx Descriptor Table Base | 0x378 | TDTB[31:0] | 16,384 |

## 18.4 Descriptor Reference

A descriptor reference is a 4 byte data structure with the following sub-fields:

- The lower 14 bits are used as an index to a descriptor table.
- Bits 13 and 14 are used by LASAR-155 to communicate the status of segmentation or reassembly to the PCI Host. The status bits are valid only in certain descriptor reference queues as outlined in the table below.
- The upper 16 bits are unused and should be masked out by the device driver.

Figure 18.3: Descriptor Reference Structure




| Descriptor Reference Queue | Status Valid | Index Name |
|---|---|---|
| TDR Ready High Queue | No | TDR[13:0] |
| TDR Ready Low Queue | No | TDR[13:0] |
| TDR Free Queue | Yes | TDR[13:0] |
| RPDR Large Free Queue | No | RPDR[13:0] |
| RPDR Small Free Queue | No | RPDR[13:0] |
| RPDR Ready Queue | Yes | RPDR[13:0] |
| RMDR Free Queue | No | RMDR[13:0] |
| RMDR Ready Queue | Yes | RMDR[13:0] |

Each descriptor can be accessed using the index field of the descriptor reference and the base address of the descriptor table as illustrated by the pseudo code below:

```
/* Need to mask out the upper 18 bits of the desc reference to extract
 * the index field. */
#define      DR_INDEX_MASK     0x00003FFF

index = desc_ref & DR_INDEX_MASK;

/* In general, the physical address of an element in
 * the descriptor table can be determined by */
desc_addr =  desc_table_base_addr + (index * sizeof(descriptor));
```

Alternatively, since LASAR-155 descriptors are 32 bytes in length, the above calculation can be simplified to:

```
desc_addr =  desc_table_base_addr + (index << 5);
```

## 18.5 Descriptor Reference Queue

Descriptor Reference Queues are FIFOs used to maintain Descriptor References.

The descriptor reference queues[6] used in the transmit direction are stored in a contiguous location in the host memory. The base address for the transmit descriptor reference queues is specified in register **PCID Tx Queue Base (0x37C)**. The base address must be DWORD aligned.

The descriptor reference queues[7] used in the receive direction are located in a contiguous location in the host memory. The base address for the receive descriptor reference queues is specified in register **PCID Rx Queue Base (0x31C)**[8]. The base address must be DWORD aligned.

[6]The transmit descriptor reference queues include TDR Ready High Queue, TDR Ready Low Queue, and TDR Free Queue.

[7]The receive descriptor reference queues include RPDR Ready Queue, RPDR Small Free Queue, RPDR Large Free Queue, RMDR Ready Queue, and RMDR Free Queue.

[8]Once initialized, it is unlikely that the device driver will need to change the PCID Tx & Rx Queue Base addresses. Therefore it is recommended that these values be saved in non-LASAR-155 register space to minimize the number of PCI accesses required.

## 18.6 Descriptor Reference Queue Operations

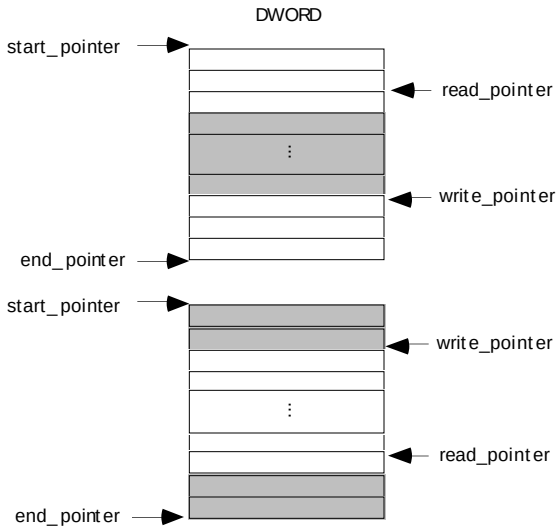
Four pointers are used to manage the Descriptor Reference Queue as described in the following table and illustrated in Figure 18.4.

| ITEM | DESCRIPTION |
|---|---|
| start_pointer | The start_pointer marks the first location where an descriptor reference can be stored in the host memory. This pointer should remain static after initialization.[9] |
| end_pointer | The end_pointer marks the end of the queue. This pointer should remain static after initialization. **The address referenced by the end-pointer is not part of the queue.**[10] |
| write_pointer | The write_pointer marks the next location where a descriptor reference can be inserted at. **The write pointer is post-incremented after an element is written to a queue**. |
| read_pointer | The read pointer, when pre-incremented by one, marks the location of the next descriptor reference that can be retrieved from. **The read pointer is pre-incremented before an element is retrieved from a queue.** |

[9]Once initialized, it is unlikely that the device driver will need to change the start and end pointers. Therefore it is recommended that these values be saved in non-LASAR-155 register space to minimize the number of PCI accesses required.

[10]Once initialized, it is unlikely that the device driver will need to change the start and end pointers. Therefore it is recommended that these values be saved in non-LASAR-155 register space to minimize the number of PCI accesses required.

Figure 18.4: Descriptor Reference Queue Overview




The descriptor queue reference pointers are mapped onto LASAR-155 registers. The table below summarizes the register where each pointer is mapped to.

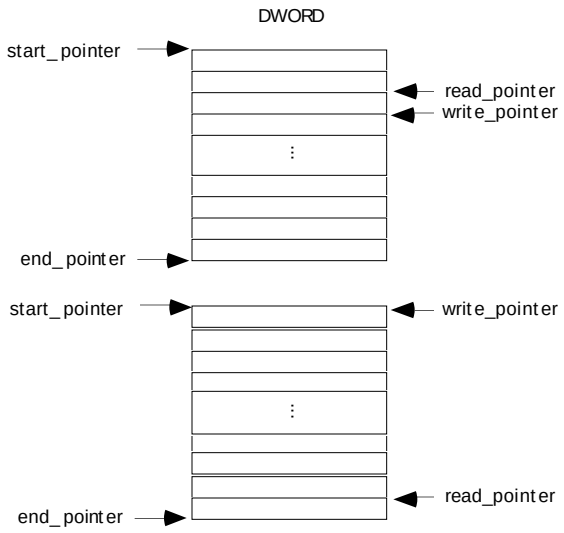
| Descriptor Reference Queue | Pointer Type | Register Name | Register Offset |
|---|---|---|---|
| RDPR Large Buffer Free Queue | start | PCID Rx Packet Descriptor Reference Large Buffer Free Queue Start | 0x320 |
| | write | PCID Rx Packet Descriptor Reference Large Buffer Free Queue Write | 0x324 |
| | read | PCID Rx Packet Descriptor Reference Large Buffer Free Queue Read | 0x328 |
| | end | PCID Rx Packet Descriptor Reference Large Buffer Free Queue End | 0x32C |
| RPDR Small Buffer Free Queue | start | PCID Rx Packet Descriptor Reference Small Buffer Free Queue Start | 0x330 |
| | write | PCID Rx Packet Descriptor Reference Small Buffer Free Queue Write | 0x334 |
| | read | PCID Rx Packet Descriptor Reference Small Buffer Free Queue Read | 0x338 |
| | end | PCID Rx Packet Descriptor Reference Small Buffer Free Queue End | 0x33C |
| RPDR Ready Queue | start | PCID Rx Packet Descriptor Reference Ready Queue Start | 0x340 |
| | write | PCID Rx Packet Descriptor Reference Ready Queue Write | 0x344 |
| | read | PCID Rx Packet Descriptor Reference Ready Queue Read | 0x348 |

| | | | |
|---|---|---|---|
| | end | PCID Rx Packet Descriptor Reference Ready Queue End | 0x34C |
| RMDR Free Queue | start | PCID Rx Management Descriptor Reference Free Queue Start | 0x350 |
| | write | PCID Rx Management Descriptor Reference Free Queue Write | 0x354 |
| | read | PCID Rx Management Descriptor Reference Free Queue Read | 0x358 |
| | end | PCID Rx Management Descriptor Reference Free Queue End | 0x35C |
| RMDR Ready Queue | start | PCID Rx Management Descriptor Reference Ready Queue Start | 0x360 |
| | write | PCID Rx Management Descriptor Reference Ready Queue Write | 0x364 |
| | read | PCID Rx Management Descriptor Reference Ready Queue Read | 0x368 |
| | end | PCID Rx Management Descriptor Reference Ready Queue End | 0x36C |
| TDR Free Queue | start | PCID Tx Descriptor Reference Free Queue Start | 0x380 |
| | write | PCID Tx Descriptor Reference Free Queue Write | 0x384 |
| | read | PCID Tx Descriptor Reference Free Queue Read | 0x388 |
| | end | PCID Tx Descriptor Reference Free Queue End | 0x38C |
| TDR High Priority Ready Queue | start | PCID Tx Descriptor Reference High Priority Ready Queue Start | 0x390 |
| | write | PCID Tx Descriptor Reference High Priority Ready Queue Write | 0x394 |
| | read | PCID Tx Descriptor Reference High Priority Ready Queue Read | 0x398 |
| | end | PCID Tx Descriptor Reference High Priority Ready Queue End | 0x39C |
| TDR Low Priority Ready Queue | start | PCID Tx Descriptor Reference Low Priority Ready Queue Start | 0x3A0 |
| | write | PCID Tx Descriptor Reference Low Priority Ready Queue Write | 0x3A4 |
| | read | PCID Tx Descriptor Reference Low Priority Ready Queue Read | 0x3A8 |
| | end | PCID Tx Descriptor Reference Low Priority Ready Queue End | 0x3AC |

## Empty Queue States

The empty queue states are illustrated in Figure 18.5.

Figure 18.5: Empty Queue States



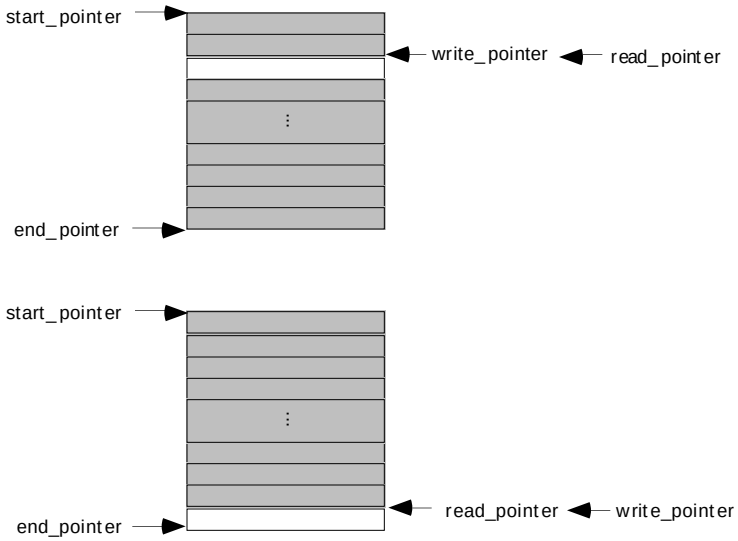

The following pseudo code determines whether the queue is empty or not.

```
/*  If the read pointer is at the bottom of the queue, reposition it at
the top. */
#define      DWORD       4
#define      END_QUEUE   (end_pointer - DWORD)

if (read_pointer == END_QUEUE)
   test_pointer = start_pointer
else
    /* Read pointer is not at the end of the queue, pre-increment it. */
    test_pointer = read_pointer + DWORD;

if (test_pointer == write_pointer)
   /*  queue is empty */
else
   /*  queue is not empty */
```

## Full Queue States

The full queue states are illustrated in Figure 18.6. The queue is full when the read_pointer is the same as the write_pointer.

Figure 18.6: Full Queue States




The following pseudo code determines whether the queue is full or not.

```
if (write_pointer == read_pointer)
   /*  queue is full */
else
   /*  queue is not full */
```

## To Add a new Element

The following pseudo code serves as an example to add a new element into the queue.

```
/* Note, the base address must be first initialized and aligned on a DWORD
 * boundary.  The write_pointer, start_pointer, and end_pointer must
 * be initialized. */
#define      DWORD       4
#define      END_QUEUE   (end_pointer - DWORD)

/* Check to make sure the queue is not full.  Refer to Full Queue States
for additional information. */

/* Need to mask out the upper 16 bits of the write pointer */
index = write_pointer & 0x0000FFFF;

/* The physical address of the descriptor reference can be determined from
the 'index' offset with respect to the base address. Also, as each
descriptor reference element is 32 bits in size, the index must be
multiplied by 4 bytes or shifted left 2 bits. */
*(base_addr + index << 2) = new_desc_ref;

/* Need to post increment the write_pointer. */
if (write_pointer == END_QUEUE)
  /* At the end of the queue, reposition the write_pointer at the top
   * of the queue. */
  write_pointer = start_pointer;
else
  write_pointer += DWORD;
```

## To Read an Element

The following pseudo code serves as an example to read a new element from the queue.

```
#define      DWORD       4
#define      END_QUEUE   (end_pointer - DWORD)

/* Note, the base address must be first initialized and aligned on a DWORD
 * boundary.  The read_pointer, start_pointer, and end_pointer must
 * be initialized. */

/* Check to make sure the queue is not empty.  Refer to Empty Queue States
for additional information. */

/* Need to preincrement the read_pointer. */
if (read_pointer  == END_QUEUE)
  /* At the end of the queue, reposition the read_pointer at the top
     of the queue. */
  read_pointer = start_pointer;
else
  /* pre-increment the read pointer. */
  read_pointer += DWORD;

/* Need to mask out the upper 16 bits of the read pointer */
index = read_pointer & 0x0000FFFF;

/* The physical address of the descriptor reference can be determined from
the 'index' offset with respect to the base address. Also, as each
descriptor reference element is 32 bits in size, the index must be
multiplied by 4 bytes or shifted left 2 bits. */
desc_ref = *(base_addr + index << 2);
```

## 18.7 LASAR-155 Data Structures Initialization Procedure

This section describes the procedure to initialize the LASAR-155 data structures in the host memory. This procedure should only be executed after PCI configuration, Master Reset Procedure and LASAR-155 initialization have been successfully completed.

| Note: |
| --- |
| The TRMEN bit of **PCID Control (0x300)** must be set to logic zero when the PCID data structures are being initialized. |

### Allocate Memory and Assign Base Addresses:

- Determine the maximum number of Transmit Descriptors required, allocate the memory and assign the base address to **PCID Tx Descriptor Table Base (0x378)**. The TD table must be aligned on a 32 byte boundary.
- Determine the maximum number of Receive Packet Descriptors required, allocate the memory and assign the base address to **PCID Rx Packet Descriptor Table Base (0x314)**. The RPD table must be aligned on a 32 byte boundary.
- Determine the maximum number of Receive Management Descriptors required, allocate the memory and assign the base address to **PCID Rx Management Descriptor Table Base (0x318)**. The RMD table must be aligned on a 32 byte boundary.
- Determine the size of the Tx Descriptor Reference Queues, allocate the memory and assign the base address to **PCID Tx Queue Base (0x37C)**. This TDR Queues must be aligned on a DWORD boundary.
- Determine the collective size of the Rx Packet Descriptor Reference Queues and Rx Packet Management Reference Queues, allocate the memory and assign the base address to **PCID Rx Queue Base (0x31C)**. The RPDR Queues must be aligned on a DWORD boundary.

### Initialize Tx Descriptor Reference Queues

- Locate the Tx Descriptor Reference Free Queue in the memory previously allocated for the Tx Descriptor Reference Queues.[11] This involves assigning the start and end pointers.
- The designer can pre-allocate the data buffer at startup-time or allocate as required during run-time. If the data buffer is pre-allocated, the **Transmit Buffer Size[15:0]** and **Data Buffer Start Address [31:0]** must be updated appropriately. The transmit data buffer can be aligned on a BYTE boundary.

[11]The Tx Descriptor Reference Free Queue should be large enough to hold the number of TDs in the TD Table. Otherwise, not all TDs can be utilized.

- The initialized TD can either be inserted into the TDR Free Queue or managed using a system specific data management mechanism.
- Locate the TDR High Priority Queue in the memory space previously allocated for the Tx Descriptor Reference Queues and update the start, end, write, and read pointers. Initially, this queue should be empty.
- Locate the TDR Low Priority Queue in the memory space previously allocated for the Tx Descriptor Reference Queues and update the start, end write and read pointers. Initially, this queue should be empty.

**Initialize Rx Packet Descriptor Reference Queues**

- Locate the Rx Packet Descriptor Reference Large Buffer Free Queue in the memory space previously allocated for the Rx Packet Descriptor Reference Queues and update the start, end, write and read pointers. Each RPD must be initialized.
- The receive data buffer must be available to LASAR-155 during run-time to store incoming packets. The Receive Buffers must be DWORD aligned.
- Update the **Data Buffer Start Address[31:0]** and **Bytes in Buffer[15:0]** fields with the appropriate values.
- Insert the RPD into RPDR Large Buffer Free Queue.
- Locate the Rx Packet Descriptor Reference Small Buffer Free Queue in the memory space previously allocated for the Rx Packet Descriptor Reference Queues and update the start, end, write and read pointers. Each RPD must be initialized using the procedure described for the large buffer free queue.
- The RPDR Small Buffer Free Queue and RDPR Large Buffer Free Queue must be not empty.
- Locate the Rx Packet Descriptor Reference Ready Queue in the memory space previously allocated for the Rx Packet Descriptor Reference Queues and update the start, end write and read pointers. Initially, the Ready Queue should be empty.

**Initialize Rx Management Descriptor Reference Queues**

- Position the Rx Management Descriptor Reference Free Queue in the memory space previously allocated for the Rx Management Descriptor Reference Queues by updating the start, and end pointers. Initialize the queue to full. Initialize and insert the RMD into the free queue:
- Determine and allocate the management buffer size in the host memory. The management buffer must be large enough to hold at least one ATM cell payload (i.e. 48 or more bytes) and aligned on DWORD boundary. A management buffer is required for each RMD.

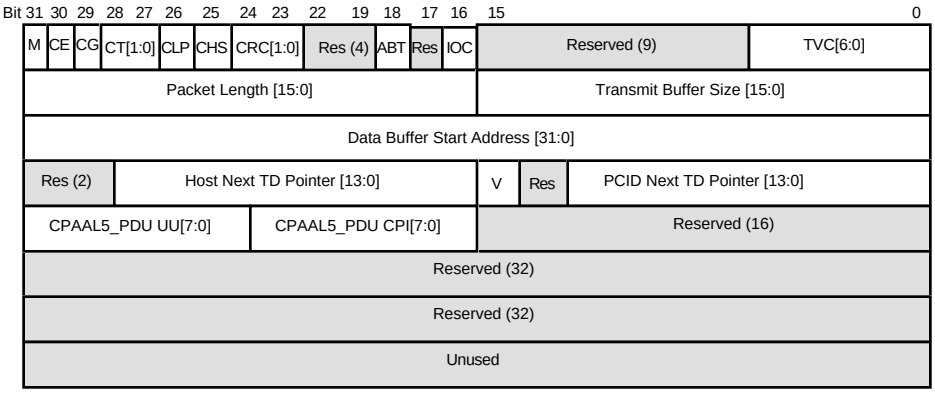
- Update the **Receive Buffer Size[31:0]** and **Data Buffer Start Address[31:0]** fields with the appropriate values.
- Insert the RMD into RMDR Free Queue.
- Position the Rx Management Descriptor Reference Ready Queue in the memory space previously allocated for the Rx Management Descriptor Reference Queues by updating the start, end pointers. Initialize the RMDR Ready Queue to empty by updating the write and read pointers.

# 19 DESCRIPTORS

## 19.1 Transmit Descriptor

The thirty-two byte Transmit Descriptor Data Structure is used by the PCI Host to describe a packet or portion of a packet to the PCID's TRM.

Figure19.1: Transmit Descriptor




Note,

In a descriptor, a field is labeled as either "Reserved" or "Unused". A "Reserved" field is used by the LASAR-155 and should not be accessed by the device driver. A "Unused" field is not used by the LASAR-155 and can be used by the device driver to store system specific data, if required.

| Field | Description |
|---|---|
| M | The More (M) bit is used by the PCI Host to support packets that require multiple Transmit Descriptors (TDs). If M is set to logic one, the LASAR-155 assumes that the current TD is just one of several TDs for the current packet. If M is set to logic zero, the LASAR-155 assumes that this TD describes the entire packet for the single TD packet case or describes the last packet for the multiple TD packet case.<br><br>Note, the More bit cannot be set to logic one when the Chain End bit is set to logic one. |

| | |
|---|---|
| CE | The Chain End (CE) bit is used by the PCI Host to indicate the end of a linked list of TDs presented to the LASAR-155. The linked list can contain one or more packets as delineated using the M bit. When CE is set to logic one, the current TD is the last TD of a linked list of TDs. When CE is set to logic zero, the current TD is not the last TD of a linked list. When the current TD is not the last of the linked list, the Host Next TD Pointer[13:0] field is valid; otherwise the field is not valid.<br><br>Note, the Chain End bit cannot be set to logic one when the More bit is set to logic one. |
| CG | The Congestion (CG) bit is used by the PCI Host to indicate whether the cells used to transmit the buffer data described by the current TD should be used to indicate congestion using their Payload Type Indicators (PTIs). Note, this bit is not considered unless the CHS bit is set to logic one.<br>When CG is set to logic one, all cells involved in the transport of the current buffer will contain a PTI field which indicates congestion. If CG is set to logic zero, congestion is not indicated. The CG bit cannot be set to logic one at the same time the CT[1:0] bits are set to 10 or 11. |
| CT[1:0] | The Cell Type (CT[1:0]) bits are used by the PCI Host to determine the cell type as defined below.<br><br>00B - AAL5 user cell<br><br>01B - Raw user cell<br><br>10B - segment OAM F5 flow cell<br><br>11B - End to end OAM F5 flow cell<br><br>When CT specifies non AAL5 user cells (i.e. 01B, 10B or 11B), the current TD's Packet Length must be less than or equal to forty eight octets. The exact number depends on the CRC type indicated. |

| | |
|---|---|
| CLP | The Cell Loss Priority (CLP) bit is used by the PCI Host to control the CLP field in the cell headers used to transport the current TD's buffer data. Note, this bit is not considered unless the CHS bit is set to logic one.<br><br>When CLP is set to logic one, all cells involved in the transport of the current buffer will contain a CLP field set to logic one. If CLP is set to logic zero, all cells involved in the transmit of the current TD's buffer data will contain a CLP field set to logic zero. |
| CHS | The Cell Header Select (CHS) bit is used by the PCI Host to select the source of the cell header fields used to transport the current TD's buffer data. When CHS is logic one, the CG, CT[1:0] and CLP bits in the current TD are used to form the Payload Type Indicator (PTI) field and the Cell Loss Priority (CLP) field in the cell headers. When the CHS is logic zero, the default PTI and CLP fields programmed into the VC Parameter Table are used.<br><br>Note, if CT[1:0] selects AAL5 user cells (00), then CHS must be set to logic one. |
| CRC[1:0] | The CRC select (CRC[1:0]) bits select whether the CRC-32 polynomial, CRC-10 polynomial or no CRC should be applied to the packet/cell described in the current TD.<br><br>00B - CRC-32<br>01B - CRC-10<br>10B - No CRC<br>11B - No CRC<br><br>For multiple TD packets, for all TDs of the packet, the CRC bits must be consistent. |

| | |
|---|---|
| ABT | The Abort (ABT) bit is used by the PCI Host to abort a packet transmission. When ABT is logic one, the packet described by the current TD is aborted and its AAL-5 CPCS PDU field is set to zero. If ABT is set in the current TD, the M bit must be logic zero.<br><br>Note, the packet length field in a TD with the ABT bit set must be set to logic zero. |
| IOC | The Interrupt On Complete (IOC) bit is used by the PCI Host to instruct the LASAR-155 to interrupt it when it has completed transmission of the current TD's buffer. When IOC is logic one, the LASAR-155 will interrupt the PCI Host using the PCIINTB output if it is enabled using a register bit. In addition, it will flush the LASAR-155's internal six deep TD buffer and place all the TDs onto the Tx Descriptor Reference Free Queue.<br><br>If IOC is logic zero, the LASAR-155 will not interrupt the PCI Host. The TD will be place in the internal six deep TD buffer (if enabled). |
| TVC[6:0] | The Transmit Virtual Connection Code (TVC[6:0]) bits are used by the PCI Host to indicate which VC a TD should be associated with. The TVC bits are constructed from N (where N=0,1,2) bits of the Virtual Path Identifier (VPI) and M (where M=7,6,5) bits of the Virtual Channel Identifier. Selection of M and N are made using a COPS register. In addition, for all TDs of the same link list structure the PCI Host adds to the Tx Descriptor Reference Ready Queue, the TVC fields must be the same. |
| Packet Length [15:0] | The Packet Length bits are used by the PCI Host to indicate the total number of bytes in the packet to be transmitted. For multiple TD packets, the Packet Length fields of each TD must be the same. In all but the last TD, all bytes of each TD's buffer are expected to contain packet data.<br><br>The Packet Length [15:0] field can be modified by the LASAR-155. The original value set by the PCI Host may not be present when the TD is processed by the LASAR-155. |

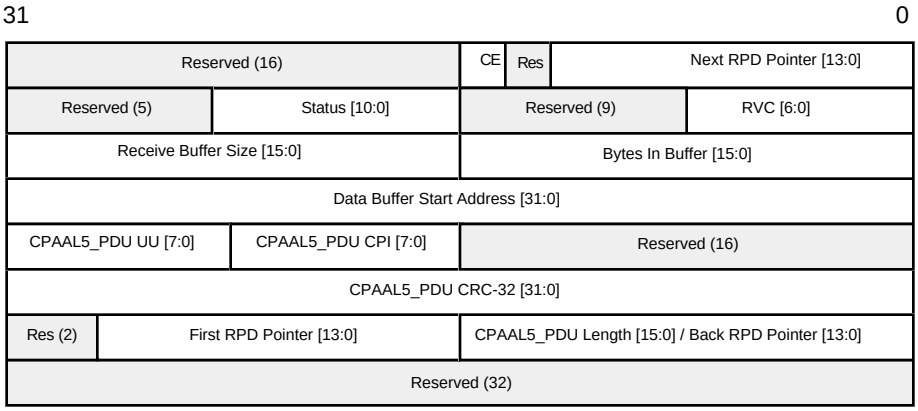
| | |
|---|---|
| Transmit Buffer Size [15:0] | The Transmit Buffer Size bits are used to indicate the size in octets of the current TD's data buffer. This field must be configured by the PCI Host.<br><br>The minimum size supported is 4 bytes. |
| Data Buffer Start Address[31:0] | The Data Buffer Start Address bits are used as a pointer to the packet buffer of the current TD in PCI Host memory.<br><br>No DWORD alignment is assumed. |
| Host Next TD Pointer [13:0] | The Host Next TD Pointer is used to store TDRs which permits the PCI Host to support linked lists of TDs. As mentioned above, linked lists are terminated using the CE bit. Linked lists of TDs are used by the PCI Host to pass the LASAR-155 multiple TD packets or to pass the LASAR-155 multiple packets associated with the same VC. |
| V | The V bit is used to indicate that the PCID Next TD Pointer field is valid. When V is set to logic one the PCID Next TD Pointer[13:0] field is valid. When V is set to logic zero, the PCID Next TD Pointer[13:0] field is invalid. This field is used by the host to rebuild the PCID packet links in the event of a segmentation failure condition. This field must be initialized to zero by the PCI Host. |
| PCID Next TD<br><br>Pointer [13:0] | The PCID Next TD Pointer bits are used to store TDRs which permits the TRM to create linked lists, of TDs passed to it via the TDRR Queues. The TDs are linked with other TDs with the same VC. In the case of an abort condition the TRM will only place the first TD of a VC on the TDRRF Queue. It is the responsibility of the PCI Host to follow the PCID and Host links in order to recover all the buffers. |
| CPAAL5_PDU UU[7:0] | The Common Part Convergence Sublayer AAL Type 5 Protocol Data Unit User to User (CPAAL5_PDU UU) octet is used by the LASAR-155 to insert into the UU field of the associated CPAAL5_PDU. Insertion can be enabled or disabled using a register bit. If disabled, the LASAR-155 inserts 00H into the UU field of the associated CPAAL5_PDU. |

| | |
|---|---|
| CPAAL5_PDU CPI[7:0] | The Common Part Convergence Sublayer AAL Type 5 Protocol Data Unit Common Part Indicator (CPAAL5_PDU CPI) octet is used by the LASAR-155 to insert into the CPI field of the associated CPAAL5_PDU. Insertion can be enabled or disabled using a register bit. If disabled, the LASAR-155 inserts 00H into the CPI field of the associated CPAAL5_PDU. |

## 19.2 Receive Packet Descriptor

The thirty-two byte Receive Packet Descriptor (RPD) Data Structure is used by the LASAR-155 to describe a reassembled packet to the PCI Host. RPD's can be linked together to form a linked list to accommodate large packets. The data structure and the fields are described below:

Figure 19.2: Receive Packet Descriptor




| Field | Description |
|---|---|
| CE | The Chain End (CE) bit is used by the LASAR-155 to indicate the end of a linked list of RPDs presented to the PCI Host. When CE is set to logic 1, the current RPD is the last RPD of a linked list of RPDs. When CE is set to logic 0, the current RPD is not the last RPD of a linked list. When the current RPD is not the last of the linked list (CE=0), the Next RPD Pointer[13:0] field is valid; otherwise the field is not valid. |
| Next RPD Pointer [13:0] | The Next RPD Pointer is used to store RPDRs which permits the LASAR-155 to support linked lists of RPDs. As mentioned above, linked lists are terminated using the CE bit. If CE is set to logic 1, this field is not valid.<br><br>Linked lists of RPDs are used by the LASAR-155 to pass the PCI Host multiple RPD packets. |

| | |
|---|---|
| Status [10:0] | The Status field is used by the LASAR-155 to indicate the status of the received packet. Below are the Status field bit definitions. When the bit is logic 1, the packet has experienced the indicated condition.<br>Status[0] - cell of packet experienced congestion. One or more cells of received packet contained a cell with PTI value 010 or 011.<br>Status[1] - cell of packet has, CLP=1.<br>Status[2] - CPAAL5_SDU Oversize error. Packet received exceeds maximum size programmed in the **RALP Max Rx SDU Length register (0x20C)**.<br>Status[3] - Unused.<br>Status[4] - CPAAL5_PDU UU field non zero.<br>Status[5] - CPAAL5_PDU CPI field non zero.<br>Status[6] - CPAAL5_PDU LENGTH field zero.<br>Status[7] - CPAAL5_PDU CRC-32 error.<br>Status[8] - CPAAL5_SDU length error. Packet received contains a length field that does not equal the received length.<br>Status[9] - Reassembly Time-out<br>Status[10] - Unused.<br>Note: for multiple RPD packets, only the first RPD's Status field is valid. All other RPD Status fields of the linked list are invalid and should be ignored. |
| RVC[6:0] | The Receive Virtual Connection Code (RVC[6:0]) bits are used by the LASAR-155 to indicate which VC a RPD is associated with. The RVC bits are constructed from N (where N=0,1,2) bits of the Virtual Path Identifier (VPI) and M (where M=7,6,5) bits of the Virtual Channel Identifier. Selection of M and N are made using a COPS register.<br>Note: for multiple RPD packets, only the first RPD's RVC field is valid. All other RPD RVC fields of the linked list are invalid and should be ignored. |

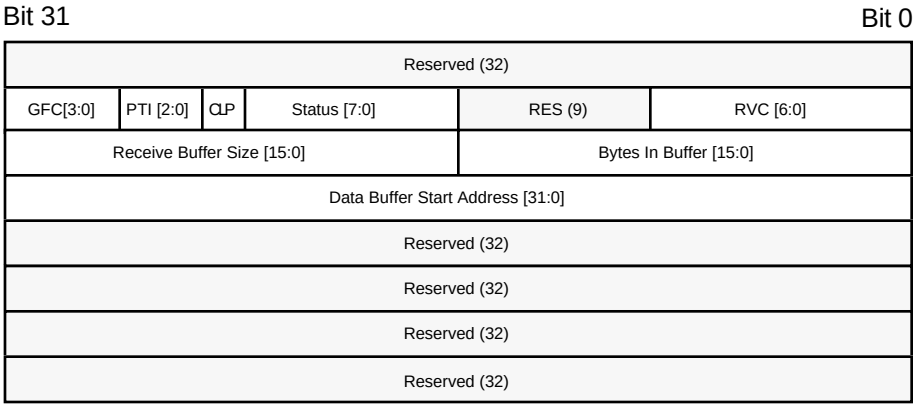
| | |
|---|---|
| Receive Buffer Size [15:0] | The Receive Buffer Size bits indicate the size in bytes of the current RPD's data buffer. This field must be configured by the PCI Host during initialization.<br><br>Note, Receive Buffer Sizes must be an integer multiple of four. Non integer multiples are not supported. In addition, the minimum buffer size supported is thirty-two bytes. |
| Bytes in Buffer [15:0] | The Bytes in Buffer bits indicate the number of bytes actually used in the current RPD's packet buffer to store packet data. |
| Data Buffer Start Address[31:0] | The Data Buffer Start Address bits are used as a pointer to the packet buffer of the current RPD into PCI Host memory.<br><br>Note, Receive Buffers must be DWORD aligned. For example, Data Buffer Start Address[1:0] is expected to be set to 00 binary. |
| CPAAL5_PDU UU [7:0] | The CPAAL5_PDU UU byte is the received User to User byte of the CPCS AAL Type 5 Protocol Data Unit associated with the current packet.<br><br>Note: for multiple RPD packets, only the first RPD's CPAAL5_PDU UU field is valid. All other RPD CPAAL5_PDU UU fields of the linked list are invalid and should be ignored. |
| CPAAL5_PDU CPI [7:0] | The CPAAL5_PDU CPI byte is the received Common Part Indicator byte of the CPCS AAL Type 5 Protocol Data Unit associated with the current packet.<br><br>Note: for multiple RPD packets, only the first RPD's CPAAL5_PDU CPI field is valid. All other RPD CPAAL5_PDU CPI fields of the linked list are invalid and should be ignored. |
| CPAAL5_PDU CRC-32 [31:0] | The CPAAL5_PDU CRC-32 word is the received CRC-32 field of the CPCS AAL Type 5 Protocol Data Unit associated with the current packet.<br><br>Note: for multiple RPD packets, only the first RPD's CPAAL5_PDU CRC-32 field is valid. All other RPD CPAAL5_PDU CRC-32 fields of the linked list are invalid and should be ignored. |

| | |
|---|---|
| First RPD Pointer [15:0] | The First RPD Pointer is used to store a RPDR pointer to the first RPD of a linked list representing a multiple RPD packet.<br><br>Note, this field is invalid for the first RPD of a linked list. |
| CPAAL5_PDU LENGTH [15:0] | The CPAAL5_PDU LENGTH word is the received LENGTH field of the CPCS AAL Type 5 Protocol Data Unit associated with the current packet.<br><br>Note: for multiple RPD packets, only the first RPD's CPAAL5_PDU LENGTH field is valid. All other RPD CPAAL5_PDU LENGTH fields of the linked list are invalid and should be ignored. |

## 19.3 Receive Management Descriptor

The thirty-two byte Receive Management Descriptor (RMD) Data Structure is used by the LASAR-155 to describe a cell to the PCI Host. The data structure and the fields are described below:

Figure 19.3: Receive Management Descriptor




| Field | Description |
|---|---|
| GFC[3:0] | The GFC[3:0] bits are the received GFC bits in the associated cell header. |
| PTI[2:0] | The PTI[2:0] bits are the received PTI bits in the associated cell header. |
| CLP | The CLP bit is the received CLP bit in the associated cell header. |

| | |
|---|---|
| Status [7:0] | The Status field is used by the LASAR-155 to indicate the status of the received packet. Below are the Status field bit definitions. When the bit is logic one, the packet has experienced the indicated condition.<br>Status[0] - cell experienced congestion. Cell received with PTI value 010 or 011.<br>Status[1] - cell received with CLP=1.<br>Status[2] - Unused.<br>Status[3] - cell CRC-10 error.<br>Status[4] - Unused.<br>Status[5] - Unused.<br>Status[6] - Unused.<br>Status[7] - cell CRC-32 error. |
| RVC[6:0] | The Receive Virtual Connection Code (RVC[6:0]) bits are used by the LASAR-155 to indicate which VC a RMD is associated with. The RVC bits are constructed from N (where N=0,1,2) bits of the Virtual Path Identifier (VPI) and M (where M=7,6,5) bits of the Virtual Channel Identifier. Selection of M and N are made using a COPS register. |
| Receive Buffer Size [15:0] | The Receive Buffer Size bits indicate the size in bytes of the current RMD's data buffer. This field must be configured by the PCI Host during initialization and must be large enough to accommodate one cell (i.e. 48 or more bytes). |
| Bytes in Buffer [15:0] | The Bytes in Buffer bits indicate the number of bytes actually used in the current RMD's buffer to store cell data. For RMDs, this value should be between 1 to 48 bytes. |
| Data Buffer Start Address[31:0] | The Data Buffer Start Address bits are used as a pointer to the buffer of the current RMD into PCI Host memory. |

# 20 PCID CONTROLS

## 20.1 FIFO Controls

### Transmit SDU FIFO

A Transmit ATM SDU FIFO (TAS FIFO) is provided to store cells queued for transmission. The depth of the TAS FIFO can be adjusted using the TXSDU_FD[2:0] bits of **PCID Control (0x300)**.

| TXSDU_FD [2:0] | Function | R/W |
|---|---|---|
| 000B | TAS FIFO is 1 cell deep. | |
| 001B | TAS FIFO is 2 cells deep. | |
| 010B | TAS FIFO is 3 cells deep. | |
| 011B | TAS FIFO is 4 cells deep. | |
| 100B | TAS FIFO is 5 cells deep. | |
| 101B | TAS FIFO is 6 cells deep. | |
| 110B | TAS FIFO is 7 cells deep. | |
| 111B | TAS FIFO is 8 cells deep. | default |

The detection of a FIFO underrun in the TAS FIFO can generate an interrupt on TXSDUF_UDRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the TXSDUF_UDRE bit of **PCID Interrupt Enable (0x308)**.

### Receive SDU FIFO

A 96 cell Receive ATM SDU FIFO is provided to allow for up to 270 µs of bus latency. The detection of a FIFO overrun in this FIFO can generate an interrupt on RXSDUF_OVRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the RXSDUF_OVRE bit of **PCID Interrupt Enable (0x308)**.

### Transmit Request FIFO

A Transmit Request FIFO (TR FIFO) is used to buffer requests for transmit data from the TATS block to the PCID block. The depth of the TR FIFO can be adjusted using the TXREQ_FD[1:0] bits of **PCID Control (0x300)**.

| TXREQ_FD [2:0] | Function | R/W |
|---|---|---|
| 00B | TR FIFO is 1 request deep. | |
| 01B | TR FIFO is 4 requests deep. | |
| 10B | TR FIFO is 8 requests deep. | default |
| 11B | Reserved | |

The detection of a FIFO overrun in the TR FIFO can generate an interrupt on TXREQF_OVRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the TXREQF_OVRE bit of **PCID Interrupt Enable (0x308)**.

### Transmit Free Queue FIFO

The TXFQ_E bit of **PCID Control (0x300)** controls the operation of the internal Transmit Descriptor Free Queue FIFO memory. This FIFO is used to temporarily store Free transmit descriptors inside the PCID device in order to reduce the number of PCI bus accesses.

| TXFQ_E | Function | R/W |
|---|---|---|
| 0 | All TDRs are written directly to the TDRF queue. | default |
| 1 | The internal TDR Free Queue FIFO memory is enabled and free TDRs are stored in the FIFO until either the FIFO is full or an interrupt is generated via the IOC bit of the Transmit Descriptor. | |

## 20.2 Transmit and Receive Request Arbitration

The LASAR-155 supports two arbitration modes to service simultaneous transmit and receive requests. The arbitration mode is controlled by the ARBMD bit of **PCID Control (0x300)**.

| ARBMD | Function | R/W |
|---|---|---|
| 0 | The receive management queue DMA requests have priority over the receive packet queue DMA requests which in turn have priority over the transmit DMA requests. | default |
| 1 | DMA requests are serviced in a fair round-robin basis. | |

## 20.3 Packet Completion Notification

### Transmit Notification

When transmitting a packet, interrupts can be controlled on a per Transmit Descriptor basis. The Interrupt On Complete (IOC) bit in the Transmit Descriptor controls whether an interrupt is generated when the current TD's buffer has been transmitted.

The interrupt is generated on the IOCI bit of **PCID Interrupt Status (0x304)** if it is enabled by the IOCE bit of **PCID Interrupt Enable (0x308)**.

### Receive Packet Notification

The RPQ_INTR[2:0] bits of **PCID Control (0x300)** specify the number of Receive Packet Descriptors (RPD) that are queued by the LASAR-155 onto the Receive Packet Ready Queue before the LASAR-155 interrupts the PCI Host.

| RPQ_INTR[2:0] | Function | R/W |
|---|---|---|
| 000B | One interrupt for every 1 RPD received. | default |
| 001B | One interrupt for every 4 RPDs received. | |
| 010B | One interrupt for every 8 RPDs received. | |
| 011B | One interrupt for every 16 RPDs received. | |
| 100B | One interrupt for every 32 RPDs received. | |
| 101B | Reserved | |
| 110B | Reserved | |
| 111B | Reserved | |

When the interrupt enable bit, RPQ_RDYE of **PCID Interrupt Enable (0x308)**, is enabled and the number of RPD written onto the Receive Packet Ready Queue is equal to the number specified in RPQ_INTR[2:0], the RPQ_RDYI bit of **PCID Interrupt Status (0x304)** is set to logic one and an interrupt is generated.

### Receive Management Cell Notification

The RMQ_INTR[2:0] bits of **PCID Control (0x300)** specify the number of Receive Management Descriptors (RPD) that are queued by the LASAR-155 onto the Receive Management Ready Queue before the LASAR-155 interrupts the PCI Host.

| RMQ_INTR [2:0] | Function | R/W |
|---|---|---|
| 000B | One interrupt for every 1 RMD received. | default |
| 001B | One interrupt for every 4 RMDs received. | |
| 010B | One interrupt for every 8 RMDs received. | |
| 011B | One interrupt for every 16 RMDs received. | |
| 100B | One interrupt for every 32 RMDs received. | |
| 101B | Reserved | |
| 110B | Reserved | |
| 111B | Reserved | |

When the interrupt enable bit, RMQ_RDYE of **PCID Interrupt Enable (0x308)**, is enabled and the number of RMD written onto the Receive Packet Ready Queue is equal to the number specified in RMQ_INTR[2:0], the RMQ_RDYI bit of **PCID Interrupt Status (0x304)** is set to logic one and an interrupt is generated.

## 20.4 System and Parity Errors

The SOE_E bit of **PCID Control (0x300)** determines the action the PCI controller will take when a system or parity error is detected. Setting this bit low when an error has occurred will reactivate the PCI REQB signal.

| SOE_E | Function | R/W |
|---|---|---|
| 0 | The PCI controller will continue to allow master transactions on the PCI bus. | default |
| 1 | PCI controller will disconnect the PCI REQB signal from the PCi bus. This will prevent the PCID from becoming a master device on the PCI bus when either the SERR or DPR bits in the Command/Status register in the PCI Configuration Space are set high. | |

The detection of a parity error can generate an interrupt on PERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the PERRE bit of **PCID Interrupt Enable (0x308)**. Parity errors can be detected by the LASAR-155 when it is an initiator or reported by a target using the PERRB signal.

The detection of a system error can generate an interrupt on SERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the SERRE bit of **PCID Interrupt Enable (0x308)**. System errors include address parity errors, data parity errors on Special Cycle commands and all on parity errors.

## 20.5 Descriptor Queue Errors

### Receive Management Descriptor Queue Errors

When the LASAR-155 requires a RMD and the RMD Free Queue is empty, an interrupt can be asserted on the RMQ_ERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the RMQ_ERRE bit of **PCID Interrupt Enable (0x308)**.

When the LASAR-155 has a RMD to return to the PCI Host and the RMD Ready Queue is full, an interrupt can be asserted on the RMDR_ERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the RMDR_ERRE bit of **PCID Interrupt Enable (0x308)**.

### Receive Packet Descriptor Queue Errors

When the LASAR-155 requires a RPD and the RPD Free Queue is empty, an interrupt can be asserted on the RPQ_ERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by RMQ_ERRE bit of **PCID Interrupt Enable (0x308)**.

When the LASAR-155 has a RPD to return to the PCI Host and the RPD Ready Queue is full, an interrupt can be asserted on the RPDR_ERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by RMDR_ERRE bit of **PCID Interrupt Enable (0x308)**.

### Transmit Descriptor Queue Error

When the LASAR-155 has a TD to return to the PCI Host and the TD Free Queue is full, an interrupt can be asserted on the TDFQ_ERRI bit of **PCID Interrupt Status (0x304)** if it is enabled by the TDFQ_ERRE bit of **PCID Interrupt Enable (0x308)**.

# 21 LASAR-155 INTERRUPT OVERVIEW

This section provides an overview of the LASAR-155 interrupt architecture. Please refer to the LASAR-155 Longform Datasheet for detailed information on the individual interrupts.

## 21.1 LASAR-155 Interrupt Types

The LASAR-155 supports three different interrupt types. The interrupt types can be distinguished by their master as outlined below:

1) Microprocessor Only: These interrupt registers can only be accessed through the microprocessor interface and cannot be accessed through the PCI port. These interrupts are asserted on the INTB pin.
2) PCI Host Only : These interrupt registers can only be accessed through the PCI Host Interface. These interrupts are asserted on the PCIINTB pin.
3) Selectable Master: These interrupt registers can be accessed by either the PCI Host or the microprocessor as defined by the MPENB pin. The PCI Host can detect the presence or absence of the microprocessor by reading the MPEN bit of the **PCID Interrupt Status (0x304)**.

| MPEN | Function | R |
|---|---|---|
| 0 | Microprocessor is not installed. | |
| 1 | Microprocessor is installed. | |

### Selectable Master Interrupts, Master Mode

In the master mode (MPENB=1) the LASAR-155 is controlled by the PCI Host and all active interrupts are asserted on the PCIINTB pin. The routing of the interrupt within the PCI Host environment is system specific and can be determined from the interrupt control register as described in "Section 22 PCI Configuration Space Overview".

In master mode, the INTB pin is configured as an input to accept an interrupt request from an external device. When enabled, the external interrupt request can be relayed to the PCI Host on PCIINTB pin. The enable procedure is described below:

- Set the EXTINTBE bit of **LASAR-155 Master Interrupt Enable (0x00C)** to logic one and
- Set the EXTIE bit of the **PCID Mailbox / Microprocessor Interrupt Status/Enable (0x30C)** to logic one

### Selectable Master Interrupts, Slave Mode

In the slave mode (MPENB=0) LASAR-155 is controlled by the microprocessor and all active Selectable Master Interrupts are asserted on the INTB pin[12]. Interrupt from an external device (e.g. S/UNI PDH) cannot be relayed to the PCI Host.

[12]When MPENB=0, the INTB pin is configured as an on drain output.

## 21.2 Interrupt Conventions

This section describes the graphical conventions used to illustrate the interrupt structures.

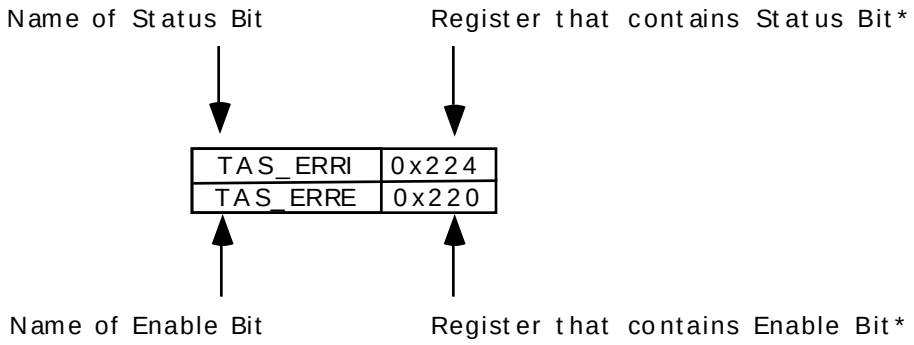
### Interrupt Status / Enable Bits

All LASAR-155 interrupts can be enabled or disabled using enable bits. When the enable bit is set to logic one, the corresponding interrupt can assert an interrupt on the interrupt pin. When the enable bit is set to logic zero, the corresponding interrupt will not assert an interrupt on the interrupt pin.

| Note: |
| --- |
| The interrupt status may change value even when the corresponding enable bit is set to logic zero. As such, it is necessary to AND the status bit and the enable bit in software when servicing the interrupt. |

Figure 21.1 illustrates the convention used in this document to describe the pair of interrupt status and enable bits.

Figure 21.1: Interrupt Status/Enable Convention



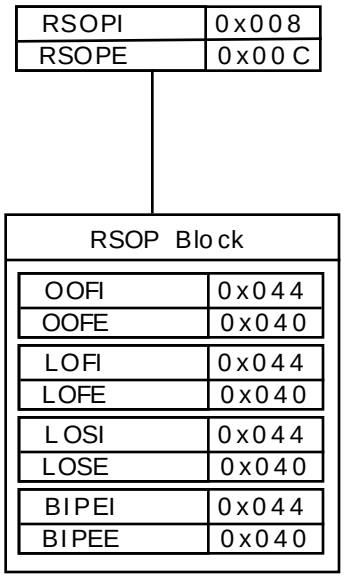

* For PCI Only & Selectable Master Interrupts:
refers to the PCI Offset.
For Microprocessor Only Interrupts,
refers to the microprocessor register address.

## Block Level Interrupt Hierarchy

To allow the device driver to quickly identify the source of an interrupt, the LASAR-155 implements a two level interrupt hierarchy. When configured, a block level interrupt status bit is used to indicate one or more active interrupt request from the block.

As an example, in Figure 21.2 there are four interrupt sources in the Receive Section Overhead Processor (RSOP). They are OOFI, LOFI, LOSI and BIPEI.

Figure 21.2: LASAR-155 Block Level Interrupt Example.




The RSOPI bit in **LASAR-155 Master Interrupt Status (0x008)** is the block level status indicator for the RSOP block. The RSOPI bit is high when one or more interrupt requests are active from the RSOP block. This relationship can be expressed as:

```
RSOPI = ((OOFI & OOFE) | (LOFI & LOFE) | (LOSI & LOSE) | (BIPEI & BIPEE))
```

All the block level interrupt status are maintained in **LASAR-155 Master Interrupt Status (0x008)** register. When servicing an interrupt, the device driver can use the content of this register to identify the LASAR-155 block(s) with active interrupt request(s).

## PCI Host Interrupt Service Overview

```
/*  Determine whether the microprocessor is attached or not by reading
 *  MPEN bit of PCID Interrupt Status (0x304). */

#define      PCID_INTERRUPT_STATUS   0x304
#define      MPEN_BIT_POSITION       0x00000001

/* register_base is the base address for the LASAR-155 registers.  It
 * must be initialized at system startup. */
micro_attached = *(register_base + PCID_INTERRUPT_STATUS) &
                    MPEN_BIT_POSITION;
if (micro_attached == MPEN_BIT_POSITION) {
    /* Microprocessor is attached.  The selectable master interrupts are
     * serviced by the microprocessor. */

    /*  Service the PCID only interrupts. */
}
else {
    /* Microprocessor is NOT attached.  The PCI Host must service both
     * the PCID and selectable master interrupts. */

    /* Service the PCID Only interrupts. */

    /* Service the selectable master interrupts. */
}
```

## 21.3 Microprocessor Only Interrupts:

The Microprocessor Only interrupts are listed in Figure 21.3. Note, the hexadecimal values associated with the status and enable bits refer to the address of the register.

Figure 21.3: Microprocessor Only Interrupts.

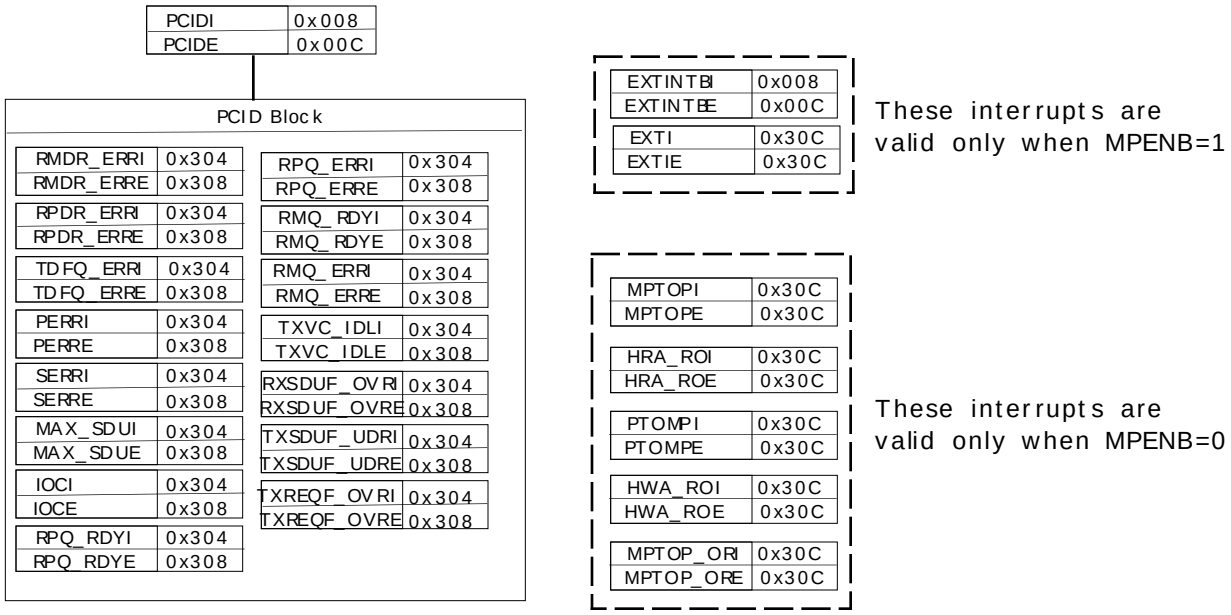
| | |
|---|---|
| MPTOPI | 0xC1 |
| MPTOPE | 0xC0 |
| HRA_ROI | 0xC1 |
| HRA_ROE | 0xC0 |
| PTOMPI | 0xC1 |
| PTOMPE | 0xC0 |
| MRA_ROI | 0xC1 |
| MRA_ROE | 0xC0 |
| MPTOP_ORI | 0xC1 |
| MPTOP_ORE | 0xC0 |

These interrupts are valid only when MPENB=0.

## 21.4 PCI Host Only Interrupts

The PCI Host Only Interrupts are illustrated in the Figure 21.4. These interrupts are asserted on PCIINTB and are visible only to the PCI Host. Note that both EXTINTBI and EXTI are the external interrupt status bits and are valid only in master mode (MPENB=1). The interrupts used in the PCI to Microprocessor mailbox mechanism (i.e. MPTOPI, HRA_ROI, PTOMPI, HWA_ROI, and MPTOP_ORI) are valid only in slave mode (MPENB=0).

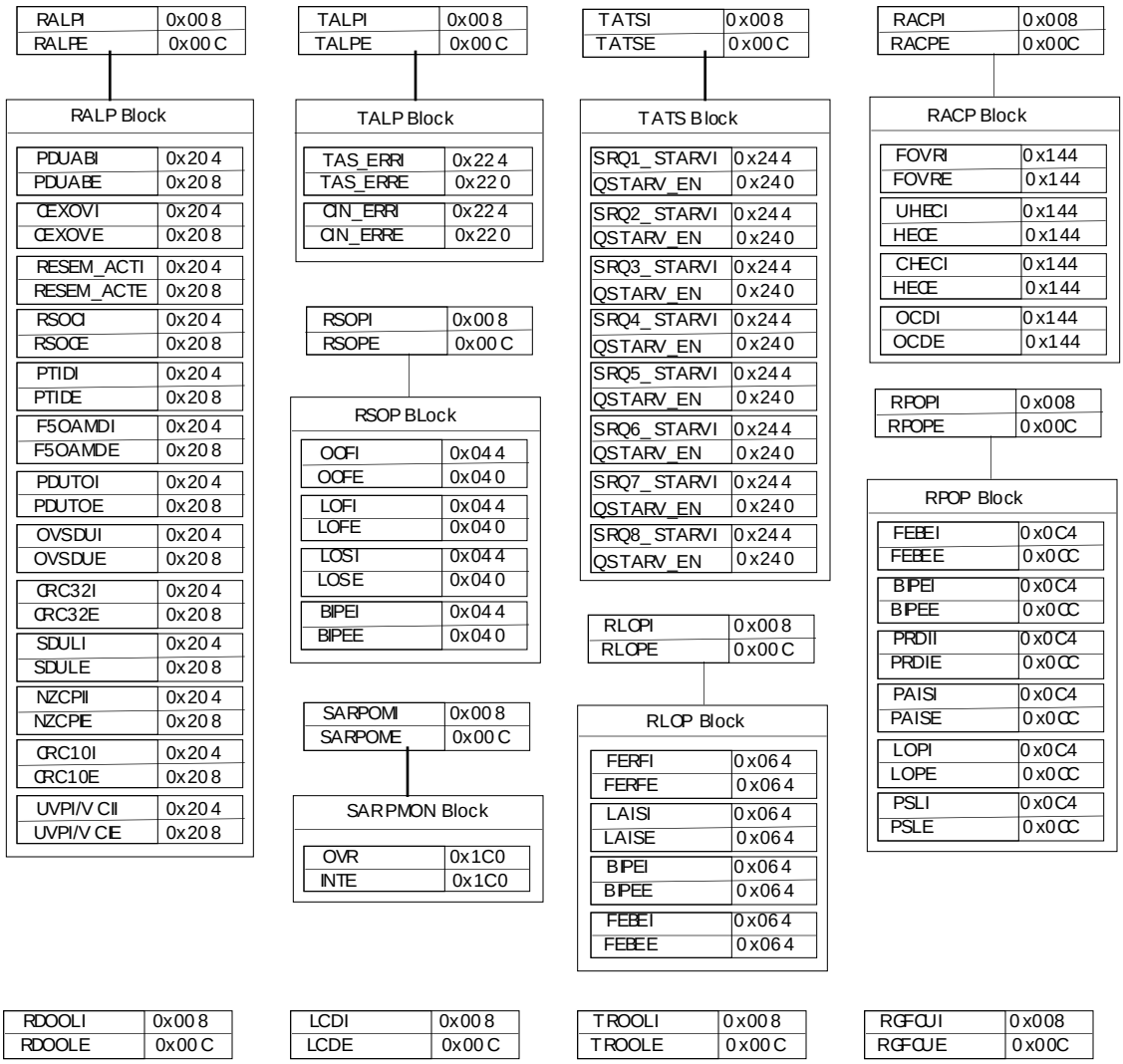
Figure 21.4: PCI Host Only Interrupts



## 21.5 Selectable Master Interrupts

Figure 21.5 summarizes the Selectable Master Interrupts. In master mode (MPENB=1), these interrupts are asserted on PCIINTB pin and are visible only to the PCI Host. In slave mode (MPENB=0) these interrupts are asserted on INTB pin and are visible only to the microprocessor.

Figure 21.5: Selectable Master Interrupts




## 21.6 Initialization

After a successful master reset procedure, all the interrupts are disabled. The device driver needs to activate the appropriate interrupts by setting the enable bits to logic one. Which interrupt is enabled or disabled is system specific.

# 22. PCI CONFIGURATION SPACE OVERVIEW

The purpose of the PCI Configuration Space is to provide device specific information in a common template such that a PCI Configuration Manager can identify each PCI device in the system as well as determine the individual functions provided by each device. In addition, the PCI Configuration Manager is responsible for determining the system resource each PCI device requests, obtaining the required resources from a System Resource Manager, and then assigning these resources to each PCI device.

The PCI Configuration Space consists of a block of 256 byte wide contiguous addresses into which all the PCI configuration registers are mapped. All PCI compliant devices must implement Configuration Space.

This chapter is limited to the definition of LASAR-155's Configuration Space for a wide variety of system types. System dependent issues for specific platforms, such as mapping various PCI address spaces into host CPU address spaces, access ordering rules, requirements for host-to-PCI bus bridges, etc., are not described in this chapter.

## 22.1 Configuration Space Organization[13]

This section provides an overview on the organization of Configuration Space registers and describes the record structure on the 256 byte space. This space is divided into a predefined header region and a device dependent region. The predefined header region consists of fields that uniquely identify the device and allow the device to be generically controlled.

## 22.2 Configuration Space Functions

PCI has the potential for greatly increasing the ease with which systems may be configured. To realize this potential, all PCI devices must provide certain functions that system configuration software can utilize.

LASAR-155's PCI configuration registers are described in **PCI CONFIGURATION REGISTER DESCRIPTION** section of [1]. This document describes the general functions of these registers, please refer to [1] for the detailed information.

| Note: |
| --- |
| LASAR-155's PCI registers can only be accessed when it is a PCI target and a configuration cycle is in progress as indicated by the IDSEL input. |
| The non-reserved bits in the PCI Registers are restored to their default values only on hardware resets. The PCI Register values are left unchanged on software resets. |

[13] Refer to PCI Local Bus Specification for additional information.

LASAR-155 implements "Type 00h Configuration Space Header" as depicted in Figure 3-1. The mandatory fields are listed below and are shown in bold text in Figure 3-1:

- Vendor ID
- Device ID
- Command
- Status
- Revision ID
- Class Code
- Header Type

Implementation of the other registers in a Type 00h header is optional. Fields marked with asterisks (*) are not implemented in the LASAR-155 Configuration Space. These fields will return 0 when read.

Figure 22.1: LASAR-155 TYPE 00h Configuration Space Header[14]

| Register | Address | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
|---|---|---|---|---|---|
| 0x00 | 0x00 | **Device ID** | | **Vendor ID** | |
| 0x01 | 0x04 | **Status** | | **Command** | |
| 0x02 | 0x08 | **Class Code** | | | **Revision ID** |
| 0x03 | 0x0C | BIST* | **Header Type** | Latency Timer | Cache Line Size* |
| 0x04 | 0x10 | Base Address 0 (LASAR-155 Registers) | | | |
| 0x05 | 0x14 | Base Address 1 (External Devices)[15] | | | |
| 0x06 | 0x18 | Base Address 2* | | | |
| 0x07 | 0x1C | Base Address 3* | | | |
| 0x08 | 0x20 | Base Address 4* | | | |
| 0x09 | 0x24 | Base Address 5* | | | |
| 0x0A | 0x28 | Cardbus CIS Pointer* | | | |
| 0x0B | 0x2C | Subsystem ID* | | Subsystem Vendor ID* | |
| 0x0C | 0x30 | Expansion ROM Base Address | | | |
| 0x0D | 0x34 | Reserved | | | |
| 0x0E | 0x38 | Reserved | | | |
| 0x0F | 0x3C | Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line |

[14] "Register" refers to LASAR-155's PCI register numbering scheme. "Address" refers to the offset, in bytes, from the base address of the PCI Configuration Space. Little-endian byte order scheme is used to access the individual bytes in the PCI Configuration Space.

[15] The Base Address 1 is implemented only if LASAR-155's Multipurpose Port is configured in master mode (MPENB=1). Access to this memory space is valid only if the Expansion ROM Access is disabled.

## Device Identification

Five fields in the PCI configuration header deal with device identification. All of these fields have read-only access.

| *Field* | Description | LASAR-155 Implementation |
|---|---|---|
| *Device ID* | This field identifies a particular device. | This field is specify in VNDRID[15:0] of Register 0x00. LASAR-155's identification code is 7375H. |
| *Revision ID* | This register specifies a device specific revision identifier. It is considered a vendor defined extension to the Device ID. | This field is specify in REVID[7:0] of Register 0x02. The revision identifier is 00H. |
| *Header Type* | This byte serves two purposes: It identifies the layout of the device dependent region beginning at offset 10H in the PCI configuration space. In addition, this field specifies whether the device is a single or multi-function device. | The header type is specify in HDTYPE[6:0] of Register 0x03 and has a value of 0000000B to indicate a "Type 0" format as shown in Figure 3-1. The multi-function bit is specify in MLTFNC of Register 0x03 and has a value of 0B to specify that the LASAR-155 is a single function device. |
| *Class Code* | The Class Code register identifies the generic function of the device using three sub-fields:<br>• The upper byte (at offset 0BH) is a base class code which broadly classifies the type of function the device performs.<br>• The middle byte (at offset 0AH) is a sub-class code that identifies more specifically the function of the device.<br>• The lower byte (at offset 09H) identifies a specific register-level programming interface such that device independent software can interact with the device. | • LASAR-155's base class code is specify in CCODE[23:16] of Register 0x02 and has a value of 02H to indicate a Network Controller.<br>• The sub-class code is specify in CCODE[15:8] of the same register and has a value of 03H to indicate that this is an ATM controller.<br>• There are no register-level programming interfaces defined for this base class. This is indicated by 00H in CCODE[7:0] of the same register. |

## Device Control

The Command register provides control over a device's ability to generate and respond to PCI cycles. When a 0 is written to this register, the device is logically disconnected from the PCI bus for all accesses except for configuration purpose.

The PCI Command register occupies the lower 16 bits of the Register 0x01. Please refer to [1] for additional information on the PCI Command register bits.

## Device Status

The PCI Status register records status information for various PCI bus related events. The PCI Status register is implemented as the upper 16 bits of the Register 0x01. Please refer to [1] for additional information on the PCI Status register.

Note:

Read to the PCI Status register behaves normally. However, write operations are slightly different in that bits can be reset, but not set. A particular bit can be reset by writing a one to that location.

## Latency Control

The control over PCI bus latency is specified by the following fields:

| *Field* | Description | LASAR-155 Implementation |
|---|---|---|
| *Latency Timer* | The field specifies, in units of PCI bus clocks, the value of the Latency Timer when this device is a bus master. | This field is specify in the LT[7:0] bits of Register 0x03. At reset, the value is 00H. |
| *Burst Period* | The burst period value is used to determine Latency Timer Value. This value is specified in 250 nanoseconds increments. | This field is specify in MINGNT[7:0] bits of Register 0x0F. The hardwired value is 08H which corresponds to 2 µS. |
| *Gain Access* | The gain access value is used to determine how often a bus master needs access to the PCI bus. This value is specified in 250 nanoseconds increments. | This field is specify in MAXLAT[7:0] bits of Register 0x0F. The hardwired value is 08H which corresponds to 2 µS. |

## Interrupt Control

The PCI Configuration Manager is responsible for routing LASAR-155's interrupt line (pin INTAB) to a system controller. This routing is dependent upon the specific hardware implementation and the assigned interrupt value must be entered into the Interrupt Line field. During runtime, the device driver may reference the Interrupt Line field to determine the interrupt line to which LASAR-155 is connected to.

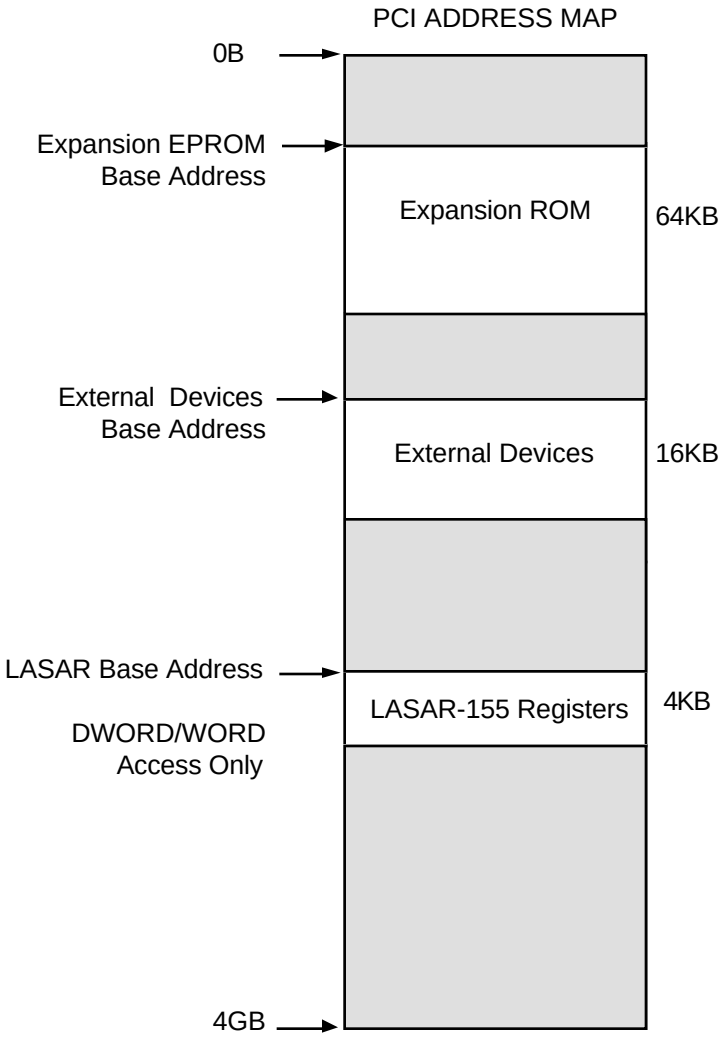
| *Field* | Description | LASAR-155 Implementation |
|---|---|---|
| *Interrupt Line* | This field identifies which interrupt request line of a system interrupt controller the PCI device is connected to. | This field is specify in INTLNE[7:0] bits of Register 0x0F. This value is system specific and must be initialized by the PCI Configuration Manager. |
| *Interrupt Pin* | This field specifies which interrupt pin a PCI device uses. | This field is specify in INTPIN[7:0] bits of Register 0x0F. INTPIN[7:0] is hardwired to 01H as LASAR-155 uses INTAB pin on the PCI bus. |

## 22.3 Memory Addressing

In the PCI Configuration Space (Type 00H), there are six Base Address Registers and one Expansion ROM Base Address register. Each Base Address Register allows a PCI device's I/O and memory functions to be dynamically mapped into either the memory or I/O address space. The Expansion ROM Base Address Register allows an Expansion ROM to be dynamically mapped into a system's memory space.

As a result of large amount of space required, LASAR-155 supports memory mapping only. LASAR-155's Address Map is described in [1] and duplicated here for convenience:

Figure 22.2: LASAR-155 Address Map.



## Base Address Registers

The LASAR-155 supports memory mapping only. PCI Base Address 0 is used to map LASAR-155's internal registers into the system address space. This is specify in Register 0x04. The LASAR-155 Registers require a contiguous 4K bytes of memory and can be mapped anywhere inside a 32-bit address space. The actual base address assigned to this memory block is the responsibility of the PCI Configuration Manager.

PCI Base Address 1 is used to map external devices on the LASAR-155 Local Bus into system address space. This is specify in Register 0x05. The External Devices requires a contiguous 16K bytes of memory and can be mapped anywhere inside a 32-bit address space. The memory space for External Devices is requested only when the Microprocessor Port is configured in master mode (MPENB=1). This memory space is accessible only if the Expansion ROM is not enabled[16].

PCI Base Address 2 to 5 are not used and will return 0 when read.

Note:

To assign a value to a memory base address register the software need to perform the following steps:

- Write a value of 1 to all bits within the Base Address register
- Confirm that bit 0 of the register contains a value of 0, indicating a memory request.
- Starting at bit location 4, search upwards for the first bit set to a value of 1. This bit is the binary weighted size of the total contiguous block of memory requested.
- Obtain an address space from the System Resource Manager that meets the required memory size. This memory address region must not conflict with any other memory space utilized within the system.
- Write the start address of the memory block assigned to this register.

Also, in order for LASAR-155 to respond to Memory Space accesses, the MCNTRL bit of Register 0x01 must be set to 1.

## Expansion ROM Base Addresses

The PCI device's expansion ROM can be mapped into a system's physical address space using the Expansion ROM Base Address Register. This register is in use only when the Microprocessor Port is configured in master mode (MPENB=1) and the Expansion ROM is installed (ROMP=1). If any one of the conditions is not met, this register will return 0 to indicate no memory is required. The Expansion ROM Base Address is specify in Register 0x0C and requires a 64K bytes of contiguous memory block mapped anywhere inside a 32 bit address space.

[16]Expansion ROM is disabled when bit ENABLE of PCI Register is set low. This is the default value after a hardware reset.

If both External Device and Expansion ROM are installed, the device driver should first enable the ROM access, retrieve the necessary information, and then disable the ROM access such that the External Device memory map can be accessed.

## 23. MASTER RESET PROCEDURE

This section describes the procedure to initialize LASAR-155. The software reset procedure should be executed immediately following a hardware reset or as required by the PCI Host. The steps to reset the device are described below:

- The RESET bit in **LASAR-155 Master Reset and Identity/Load Meters (0x000)** is activated by first set high, then set low.[17]
- Set the TRMEN bit of **PCID Controls (0x300)** to logic zero to disabled the Transmit Request Machine (TRM).
- Configure the internal control memory block, COPS, to select the entire memory space for 128 transmit and 128 receive VCs for reset. In the **COPS Control (0x280)** register, set NVCI[3:0] to 0111B to select 7 VCI and set NVPI[3:0] to 0000B to select 0 VPI bits[18].
- Set the INIT bit in the **LASAR-155 Master Control (0x014)** register to high to initiate the internal initialization procedures.
- Once the initialization procedure is started, the device driver should poll the INIT_STAT of **LASAR-155 Master Control (0x014)** register. The INIT_STAT bit remains high until the LASAR-155 has completed its initialization procedures. When initialization has completed, the LASAR-155 sets the INIT_STAT bit low.
- Set the INIT bit in the **LASAR-155 Master Control (0x014)** register to low to complete the initialization procedure. This bit must be manually be set low.
- The RROOLV bit of **LASAR-155 Clock Recovery Control and Status (0x01C)** should be polled after a power up reset to determine when the CRU PLL is operational.

| Note: |
| --- |
| Unlike the hardware reset input, RSTB, the software reset bit, RESET does not force the LASAR-155 digital out pins tri-state. In addition, all register except the PCID registers are reset to their default values. The PCID register values are preserved. Note, for proper operation after a soft reset, the TRMEN bit in the PCID Control register must be cleared before software reset is released. |

[17]The RESET bit allows the LASAR-155 to be reset under software control. If the RESET bit is logic one, the entire LASAR-155 except the PCI Interface is held in reset. This bit is not self-clearing. Therefore, a logic zero must be written to bring the LASAR-155 out of reset. Holding the LASAR-155 in a reset state places it into a low power, stand-by mode. A hardware reset clears the RESET bit, thus negating the software reset.

[18]The selection of 7 VCI and 0 VPI is arbitrary and does not have to correspond to the required VPI/VCI combination during run-time. In fact, any combination where the sum of VPI and VCI equals 7 will work.

# 24. DEVICE DIAGNOSTICS

## 24.1 Master Clock Monitoring

The **LASAR-155 Master Clock Monitor (0x010)** provides activity monitoring on the following clocks:

- TCLK output
- RCLK output
- TRCLK+ and TRCLK- inputs
- RRCLK+ and RRCLK- inputs
- PCICLK
- SYSCLK

When the monitored clock makes a low to high transition, the corresponding register bit is set high. This bit will remain high until this register is read, at which point, all the bits in this register are cleared. A lack of transitions is indicated by the corresponding register bit reading low. This register should be read at regular intervals to detect clock failures.

## 24.2 Diagnostic Loopbacks

The LASAR-155 provides the following diagnostic loop-backs. Simultaneous diagnostic loop-back and line loop-back is not supported.

### Diagnostic Loop-Back

The DLE bit of **LASAR-155 Master Control (0x014)** connects the transmit stream to the receive stream. In this mode, the activities of the transmit side can be monitored by the receive side.

### Line Loop-Back

The LLE bit of **LASAR-155 Master Control (0x014)** allows the RXD+ and RXD- signals to be connected internally to TSD+ and TXD-, respectively.

## 25 DEVICE IDENTIFICATION

Five fields are provided in the PCI Configuration Space for device identification purpose. Refer to PCI Configuration Space for additional information.

The following information are provided in LASAR-155 registers for device identification purpose.

- ID[3:0] of **LASAR-155 Master Reset and Identity / Load Meters (0x000)** contains the revision number.
- TYPE[3:0] of **LASAR-155 Master Reset and Identity / Load Meters (0x000)** contains the device type.

## NOTES

## NOTES

## NOTES

Seller will have no obligation or liability in respect of defects or damage caused by unauthorized use, mis-use, accident, external cause, installation error, or normal wear and tear. There are no warranties, representations or guarantees of any kind, either express or implied by law or custom, regarding the product or its performance, including those regarding quality, merchantability, fitness for purpose, condition, design, title, infringement of third-party rights, or conformance with sample. Seller shall not be responsible for any loss or damage of whatever nature resulting from the use of, or reliance upon, the information contained in this document. In no event will Seller be liable to Buyer or to any other party for loss of profits, loss of savings, or punitive, exemplary, incidental, consequential or special damages, even if Seller has knowledge of the possibility of such potential loss or damage and even if caused by Seller's negligence.

© 1996 PMC-Sierra, Inc.

PMC-960229p1 ref 931127p3

Issue date: March, 1996.

Printed in Canada