PM73122 / PM73123 / PM73124

AAL1GATOR PRODUCT FAMILY
AAL1GATOR-32 / -8 / -4

32 / 8 / 4 LINK CES/DBCES
AAL1 SAR

PROGRAMMER'S GUIDE

PROPRIETARY AND CONFIDENTIAL

PRELIMINARY

ISSUE 2: APRIL 2001

REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	January 2000	Created document.
2	April 2001	Corrected Configuration Steps Flow Chart. Clarified tributary mapping in the SBI Extract Tributary Mapping Configuration section. Corrected incorrect reference made in Section 8.4.5 Line Clock Source.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Scope

The AAL1gator-32/8/4 Programmer's Guide is intended to describe the configurable features and operation of an AAL1gator-32/8/4 from a programmer's perspective. This document may not cover all applications of the AAL1gator-32/8/4. Please contact a PMC-Sierra Applications Engineer for specific uses not covered in this document.

This document is a supplement to the AAL1gator-32 [1], AAL1gator-8 [2], and AAL1gator-4 [3] Longform Datasheets. Both the longform datasheet and the programmer's guide should be studied together to interface the AAL1gator-32/8/4 to an embedded processor. In case of a discrepancy between the programmer's guide and the datasheet, the datasheet will take precedence.

This document is a supplement to the AAL1gator-32/8/4 Software Driver User's Manual for engineers who need detailed information on register accesses and programming procedures.

1.2 Target Audience

This document has been prepared for engineers that design-in the AAL1gator-32/8/4 and require a quick reference for programming the AAL1gator-32/8/4. It is assumed that the reader is familiar with ATM, AAL1, circuit emulation, line interface, and UTOPIA interface technologies.

1.3 Numbering Conventions

The following numbering conventions are used throughout this document:

- Binary 1010b, "011", '1'
- Decimal 129, 8
- Hexadecimal 0x80120, 1FH

1.4 Device Naming Conventions

From this point forward, the term AAL1gator shall refer to all three of the AAL1gator-32/8/4 variants, while features and functionality specific to each variant shall use the name of the individual device.

2 REFERENCES

1. PMC-1981419, PMC-Sierra, Inc., "ATM Adaptation Layer 1 Segmentation and Reassembly Processor-32 Datasheet", September 1999, Issue 2.

2. PMC-2000097, PMC-Sierra, Inc., "ATM Adaptation Layer 1 Segmentation and Reassembly Processor-8 Datasheet", January 2000, Issue 1.

3. PMC-2000098, PMC-Sierra, Inc., "ATM Adaptation Layer 1 Segmentation and Reassembly Processor-4 Datasheet", January 2000, Issue 1.

4. ATM Forum, UTOPIA, an ATM-PHY Layer Specification, Level 1, V. 2.01, Foster City, CA USA, March 1994.

5. ATM Forum, UTOPIA, an ATM-PHY Layer Specification, Level 2, V. 1.0, Foster City, CA USA, June 1995.

6. PMC-1980577, PMC-Sierra, Inc., "SATURN Compatible Scaleable Bandwidth Interconnect (SBI) Specification", October 1998, Issue 3.

3 AAL1GATOR PRODUCT FAMILY OVERVIEW

The AAL1 Segmentation And Reassembly (SAR) Processor (AAL1gator-32/8/4) is a highly integrated and flexible monolithic single chip device that provides DS1, E1, DS3, E3, J2 and STS-1/STM-0 line interface access to an ATM Adaptation Layer One (AAL1) Constant Bit Rate (CBR) ATM network. It arbitrates access to an external SRAM for storage of the configuration, the user data, and the statistics. The device provides a microprocessor interface for configuration, management, and statistics gathering. PMC-Sierra also offers a software device control package for the AAL1gator-32/8/4 device.

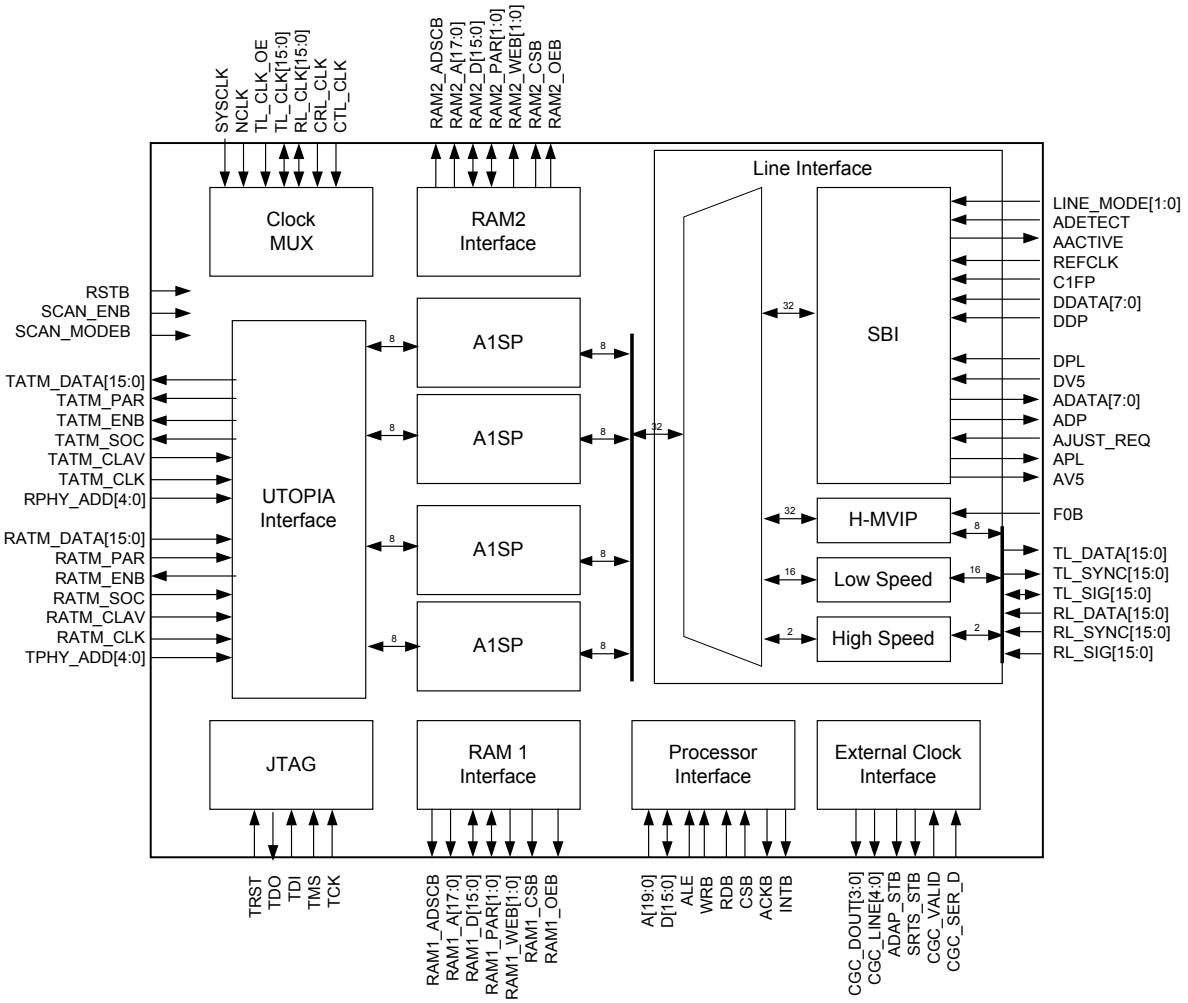
3.1 AAL1gator-32

The AAL1gator-32 contains four AAL1 SAR Processors (A1SP) which work in parallel. The A1SP blocks interface to a common UTOPIA interface on one side and a Line Interface block on the other side which can be configured to support direct clock and data, H-MVIP, or SBI mode. Two of the A1SP blocks share one ram interface and the other two A1SP blocks share the other ram interface. The processor Interface block which also contains the external clock control interface is shared by all blocks.

The AAL1gator-32 is ideal for applications such as multi-service ATM switches, ATM access concentrators, digital cross connects, computer telephony chassis with an ATM infrastructure, wireless local loop back hauls, and ATM Passive Optical Network equipment.

The functional blocks of the AAL1gator-32 are shown in Figure 1.

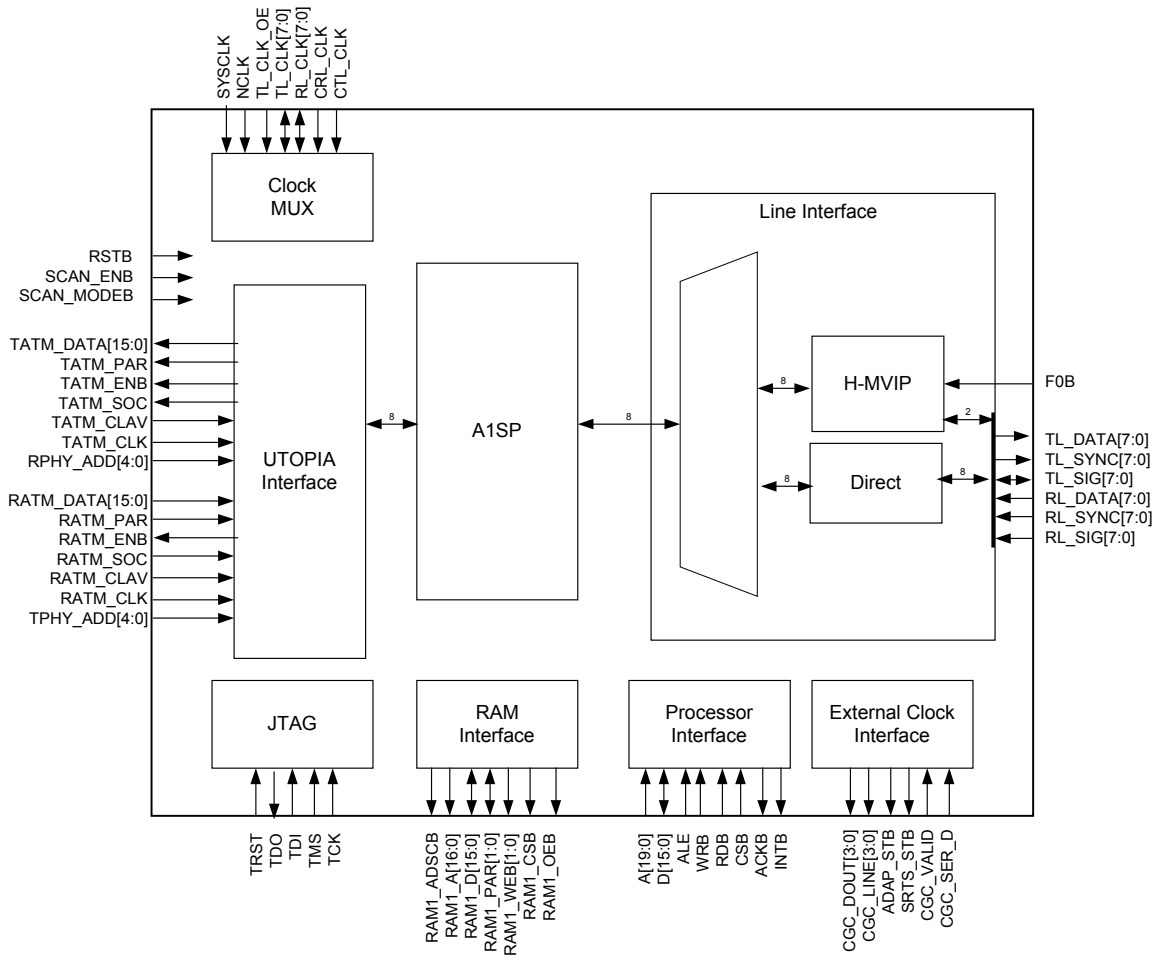
Figure 1 – AAL1gator-32 Block Diagram



3.2 AAL1gator-8

The AAL1gator-8 is a reduced link number version of the AAL1gator-32. The AAL1gator-8 provides low power, DBCES capable, eight link circuit emulation for applications such as Integrated Access Devices (IADs), ATM Multiservice Access Switches, Optical Networking Units and base stations in wireless networks. The AAL1gator-8 provides eight DS1/E1 links or a single DS3/E3/STS-1/STM-0 line interface access to an ATM network. The line interface can be configured to support direct clock and data or H-MVIP mode. The functional blocks of the AAL1gator-8 are shown in Figure 2.

Figure 2 – AAL1gator-8 Block Diagram



3.3 AAL1gator-4

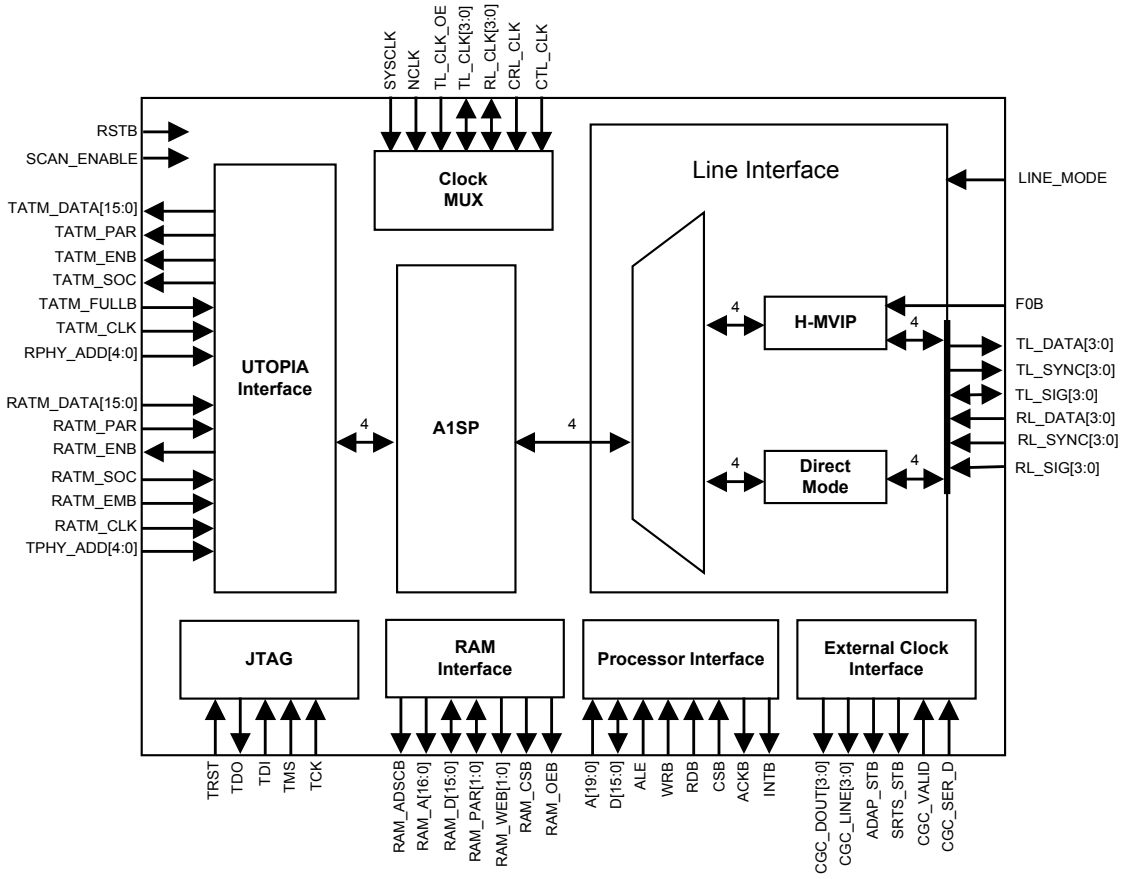
The AAL1gator-4 is a reduced link number version of the AAL1gator-32/8. The AAL1gator-4 provides low power, DBCES capable, four link circuit emulation for applications such as Integrated Access Devices (IADs), ATM Multiservice Access Switches, Optical Networking Units and base stations in wireless networks.

The AAL1gator-4 is a pin-compatible version of the AAL1gator-8 with support for four DS1/E1 links or a single DS3/E3/STS-1/STM-0 link. The line interface can be configured to support direct clock and data or H-MVIP mode. The AAL1gator-

4 is targeted for low link applications such as Optical Networking Units and base stations in wireless networks.

The functional blocks of the AAL1gator-4 are shown in Figure 3.

Figure 3 – AAL1gator-4 Block Diagram



4 REGISTER DESCRIPTION

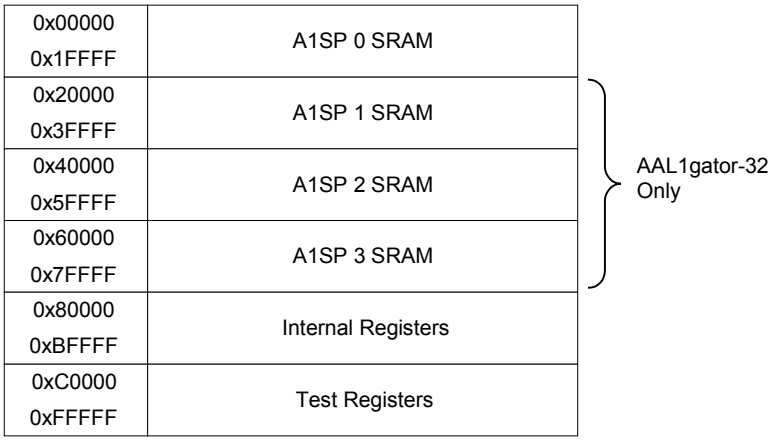
The microprocessor interface is used to configure and monitor the AAL1gator. The address signals (A[19:0]) provide an address bus while the bi-directional data signals (D[15:0]) provide a data bus to allow the AAL1gator device to interface to an external microprocessor. Both read and write transactions are supported.

The microprocessor interface block provides normal and test mode registers which are internal to the AAL1gator, as well as memory mapped registers which are mostly contained in external SRAM. The normal mode registers and memory mapped registers are required for normal operation. Please refer to the datasheets [1,2,3] for information regarding the test mode registers.

Unless otherwise specified, AAL1gator registers are described using the convention **REGISTER_NAME** (20-bit hexadecimal address). Normal mode registers may be specified by its full name or its mnemonic while memory mapped registers have a mnemonic only.

A general memory map for the AAL1gator register set is shown in Figure 4. Figure 4 shows the memory region broken into five blocks. The first four blocks, A1SP0 – A1SP3, are the memory mapped registers which are mostly contained within SRAM. The fifth block, Internal (Normal Mode) Registers, is composed of configuration registers common to the entire chip that are contained internally to the chip. A1SP1 through A1SP3 are only used in the AAL1gator-32. However, the same address space is used in all three devices to maintain software compatibility.

Figure 4 – AAL1gator Memory Map



4.1 Memory Mapped Registers

Memory mapped registers are mostly contained within SRAM and are used for line configuration and configuration of transmit and receive structures for each A1SP.

Figure 5 shows the memory map of an A1SP block within the SRAM. In the AAL1gator-32, the four A1SP blocks are identical and are accessed by taking the 17-bit relative address shown in Figure 5 and appending the 2-bit A1SP identifier to the front to select the particular A1SP block.

Figure 5 – A1SP SRAM Memory Map

0x00000 0x0001F	Control Registers
0x00020 0x07FFF	Transmit Data Structures
0x08000 0x1FFFF	Receive Data Structures

The 2-bit A1SP identifier, with A[19] = 0, is decoded as follows:

- A[18:17] = "00" A1SP0
- A[18:17] = "01" A1SP1
- A[18:17] = "10" A1SP2
- A[18:17] = "11" A1SP3

Notes on Memory Mapped Register Bits:

- All memory locations are readable and writable. Although once processing has begun, writing to some locations is restricted to prevent corruption of structures or data buffers used by the AAL1gator. Any restricted locations are designated below.
- All ports marked as "Reserved" must be initialized to 0 at initial setup. Software modifications to these locations after setup will cause incorrect operation.
- All read/write port bits marked "Not used" must be written with the value 0 to maintain software compatibility with future versions.

- All read-only port bits marked "Not used" are driven with a 0 and should be masked off by the software to maintain compatibility with future versions.

4.2 Normal Mode Registers

Normal mode registers are used to configure and monitor the operation of the AAL1gator. Normal mode registers are selected when A[19] is high and A[18] is low.

Table 1 shows the normal mode register memory map along with the section of this document that describes each block of registers.

Table 1 – Normal Mode Register Memory Map

Address	Register Description	Section
0x8000X	Command Registers	5, 9
0x8010X	RAM Interface Registers	10
0x8012X	UTOPIA Interface Registers	6
0x80200 – 0x80FFF	Line Interface Registers	8
0x81000 – 0x812FF	Interrupt and Status Registers	11
0x82000 – 0x82FFF	Idle Channel Configuration and Status Registers	12

Notes on Normal Mode Register Bits:

- Writing values into unused register bits has no effect. However, to ensure software compatibility with future, feature-enhanced versions of the product, unused register bits must be written with logic zero unless stated otherwise. Reading back unused bits can produce either a logic one or a logic zero; hence, unused register bits should be masked off by software when read.

- All configuration bits that can be written into can also be read back. This allows the processor controlling the AAL1gator to determine the programming state of the block.

- Writable normal mode register bits are cleared to logic zero upon reset unless otherwise noted.

- Writing into read-only normal mode register bit locations does not affect AAL1gator operation unless otherwise noted.

- Certain register bits are reserved. To ensure that the AAL1gator operates as intended, reserved register bits must be written with their default value as indicated by the register bit description.

5 OPERATIONAL PROCEDURES

This section describes the procedure to reset the AAL1gator via software and to initialize the AAL1gator before entering the operating state.

5.1 Software Reset

There are two types of software resets in the AAL1gator: the chip software reset and the A1SP software reset.

5.1.1 Chip Software Reset

The chip software reset is applied by setting the SW_RESET bit in the **Reset and Device ID Register** (0x80000). When set, the entire device is held in reset including all other registers. While set, the external SRAM may not be accessed. Applying a chip software reset also puts the A1SPs into a reset state.

SW_RESET	Function
0	Chip is active.
1	Chip is in reset.

Please see State S1 in section 5.2.2 for the effects of a chip software reset and the steps that need to be taken before a chip software reset can be removed.

5.1.2 A1SP Software Reset

The A1SP software reset is applied to A1SP*n* by setting the An_SW_RESET bit in the **A1SP*n* Command Register** (0x80010, …,13). When set, the corresponding A1SP is held in reset.

An_SW_RESET	Function
0	A1SP*n* is active.
1	A1SP*n* is in reset.

Please see State S2 in section 5.2.3 for the effects of an A1SP software reset and the steps that need to be taken before an A1SP software reset can be removed.

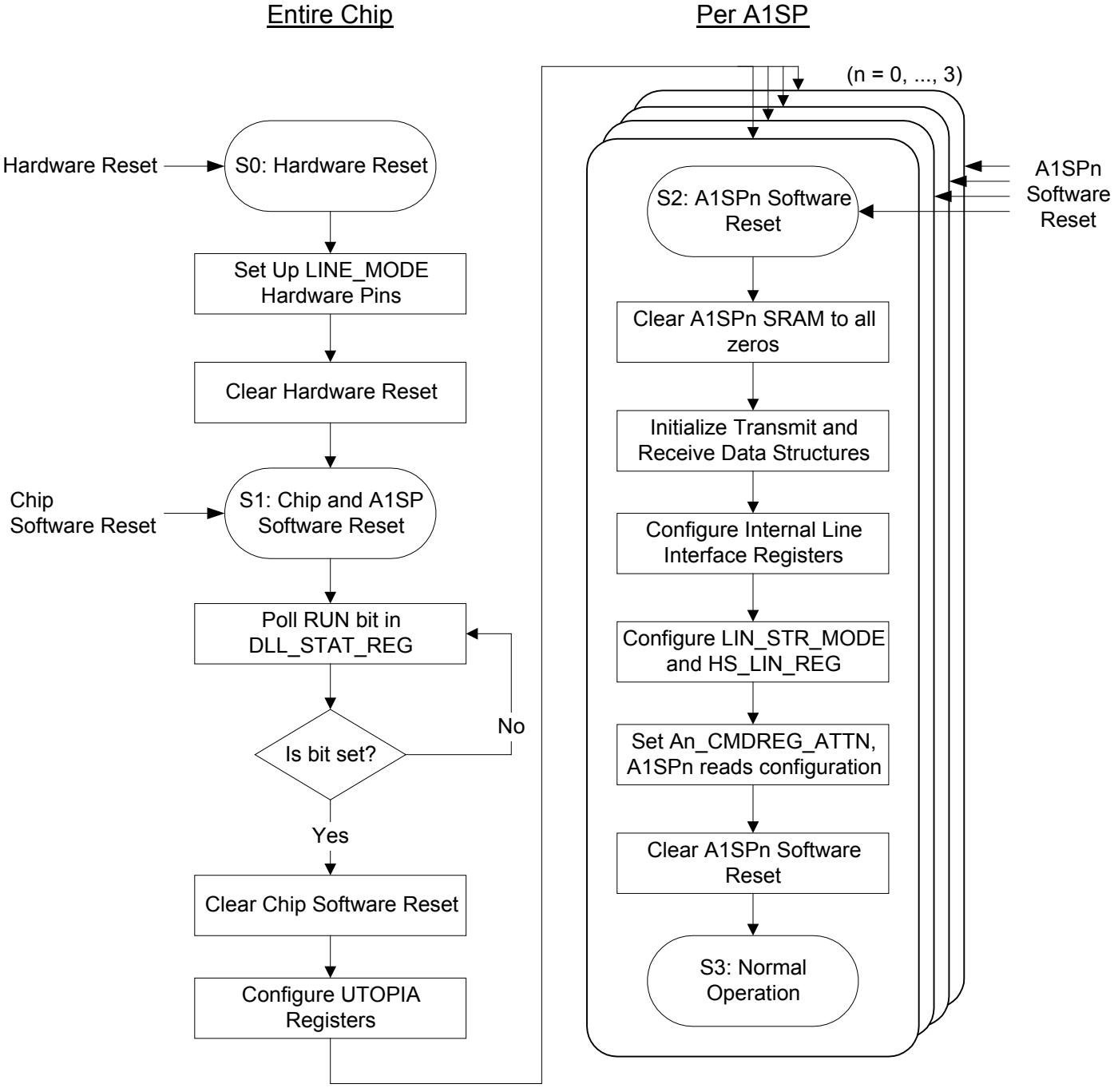
5.2 Configuration Procedure

This section describes the procedure necessary to initialize the AAL1gator after a hardware reset (i.e. - initial power up) and to change the configuration of the AAL1gator at any time. A description of each programming state that exists in the process is provided along with the steps necessary to proceed to the next state.

A flow chart of the sequence of steps is shown in Figure 6. As shown, the three reset states are automatically entered when a hardware reset, chip software reset, or A1SP*n* software reset is applied.

Note: The configuration procedure described is the recommended procedure for programming the AAL1gator. For questions relating to alternative sequences for programming the AAL1gator, please contact a PMC-Sierra Applications Engineer.

Figure 6 – Configuration Steps Flow Chart



5.2.1 State S0: Hardware Reset

The hardware reset state is entered when the RSTB input pin is forced low. This state has the following characteristics:

- All internal registers (0x80000 – 0xBFFFF) are reset to their default states.
- The UTOPIA Interface powers up with all outputs tri-stated.
- The TLCLK_OE input pin controls whether or not the TL_CLK lines are inputs or outputs between the time of hardware reset and the reading of the CLK_SOURCE_TX bits in step 5 of State S2.

The following steps need to be taken to proceed to the next state:

1. The line mode of operation needs to be setup. The LINE_MODE input pins should be tied to a certain level at initial hardware reset and not be changed while out of the reset state. See section 8 for the encoding of the LINE_MODE pins. The line mode cannot be changed by software.
2. Take the AAL1gator out of hardware reset by forcing the RSTB pin high.

5.2.2 State S1: Chip and A1SP Software Reset

The Chip Software Reset state is automatically entered after a hardware reset is removed, or it can be asserted by setting the SW_RESET bit in the **Reset and Device ID Register** (0x80000). All A1SPs are also automatically reset when this state is entered. This state has the following characteristics:

- The entire chip with the exception of the microprocessor interface and the DLL are in reset. The chip is inactive and not processing data.
- External memory can not be accessed.
- Changes to internal registers will not take effect until the Chip Software Reset is removed.

The following steps need to be taken to proceed to the next state:

1. Wait two clock periods of the slowest clock before attempting to write to any other register. An exception to this rule is the DLL register port.
2. Poll the RUN bit in the **DLL Control Status Register** (0x84003) until this bit is set. This ensures that the internal SYSCLK is aligned with the external SYSCLK before proceeding.

3. Configure the UTOPIA Interface registers. See section 6 for a detailed description of configurable features of the UTOPIA Interface.

4. Remove the Chip Software Reset by writing a 0 to the SW_RESET bit in the **Reset and Device ID Register** (0x80000).

5.2.3 State S2: A1SP Software Reset

All A1SPs are still in software reset after a Chip Software Reset is removed. An individual A1SP can also enter the A1SP Software Reset state by setting the An_SW_RESET bit in the corresponding **A1SP*n* Command Register** (0x80010, … 13). This state has the following characteristics:

- External memory can now be accessed.
- The line interface is configured in the mode indicated by the LINE_MODE pins but will only be driving data as if all lines and/or queues are disabled.

The following steps need to be taken for each A1SP in reset before entering the normal operating mode:

1. Clear the section of memory allocated to the reset A1SP to all zeros. If all A1SPs are in reset, then clear the entire SRAM to all zeros. See Figure 4 for the memory map of the AAL1gator. A number of data structures used by the device in reserved areas depends on this initialization. See section 7 for a detailed description of these register accesses.

2. Initialize the transmit and receive data structures by writing to registers with an address offset of 0x00020 – 0x1FFFF in each A1SP SRAM memory map. Some memory locations must only be set up in this state (such as T_SEQNUM_TBL and R_CRC_SYNDROME) while others can also be changed during normal operation. See section 7 for the description of transmit and receive data structures. See Figure 5 for the A1SP SRAM memory map.

3. Configure the internal line interface registers. See section 8 for a detailed description of configurable features of the Line Interface.

4. Initialize the memory mapped registers (LIN_STR_MODE and HS_LIN_REG) which contain the line configuration. See section 8 for a detailed description of these register accesses.

5. Set the An_CMDREG_ATTN bit in the **A1SP*n* Command Register** (0x80010, … 13) so that the configuration data written to the LIN_STR_MODE registers and HS_LIN_REG can be read by the A1SP.

6. Poll the An_CMDREG_ATTN until a '0' is read back to ensure that the configuration read operation is complete.

7. Remove the A1SP Software Reset by writing a '0' to the An_SW_RESET bit in the **A1SP*n* Command Register** (0x80010, … 13).

5.2.4 State S3: Normal Operating Mode

After removing the A1SP Software Reset(s), the device reads the data structures from memory and enters the correct operating mode. The R_CH_TO_QUEUE_TBL will then begin a 640 SYSCLK cycle initialization, which resets each timeslot to playing out conditioned data.

At this point the queues can be initialized as needed. Queues are added by writing to the **An_ADDQ_REG** (0x80020, …, 23) with the number of the queue to be added. There is one add queue FIFO per A1SP. See section 9.2 for details.

Note: Once processing has begun, writing to some locations is restricted to prevent corruption of structures or data buffers used by the AAL1gator.

5.3 Device Identification

Software can identify the AAL1gator by reading the DEV_ID[3:0] and the DEV_TYPE[2:0] bits in the **Reset and Device ID Register** (0x80000).

The DEV_ID bits can be read to provide a binary number indicating the feature version of the AAL1gator device. These bits are incremented only if features are added in a revision of the chip.

The DEV_TYPE bits can be read to distinguish the particular AAL1gator device from the other members of the AAL1gator family of devices as shown in Table 2.

Table 2 – DEV_TYPE Encoding

Device	DEV_TYPE[2:0]
AAL1gator-4	001
AAL1gator-8	010
AAL1gator-32	011

6 CONFIGURING THE UTOPIA INTERFACE

The UTOPIA Interface (UI) manages and responds to all control signals on the UTOPIA bus and passes cells to and from the UTOPIA bus and the four or one A1SP blocks. Both 8-bit and 16-bit UTOPIA interfaces with an optional single parity bit are supported. An 8-bit or 16-bit Any-PHY slave interface is also supported. Each direction can be configured independently and has its own configuration registers.

The following UTOPIA/Any-PHY modes are supported.

- UTOPIA Level One ATM, Master (8-bit only)
- UTOPIA Level One PHY, Slave (8 or 16-bit)
- UTOPIA Level Two PHY, Slave (8 or 16-bit)
- Any-PHY (8 or 16-bit) Slave

UTOPIA Level 1

UTOPIA Level 1 defines the interface between the Physical Layer (PHY) and upper layers such as the ATM Layer and various management entities. The definition allows a common PHY interface in ATM subsystems across a wide range of speeds and media types up to OC-3c rates (155 Mbps). UTOPIA Level 1 has the restriction that only one PHY device can be supported [4].

The AAL1gator devices can be configured as an 8-bit UTOPIA Level 1 Master or an 8-bit or 16-bit UTOPIA Level 1 Slave.

UTOPIA Level 2

UTOPIA Level 2 enhances UTOPIA Level 1 by defining the physical operation of the interface to support up to 31 PHY devices with an aggregate data rate of up to 622 Mbps [5].

With the AAL1gator-32 in UTOPIA Level 2 mode, the device generally responds on the UTOPIA bus as a single port device. However it is possible to configure the sink direction (i.e. the direction from UTOPIA to AAL1gator) as a 4-port device where each A1SP is a different port.

Any-PHY

The Any-PHY interface is a 16 bit, 52 MHz bus that can support up to 832 Mbps of raw bandwidth. By using very few overhead cycles to transmit and receive packets the Any-PHY interface is suitable for designs that need to scale up to OC-12 (622 Mbps) bandwidths. Like POS-PHY and UTOPIA bus interfaces, the Any-PHY interface is a master/slave bus. An Any-PHY bus master can interface with multiple devices.

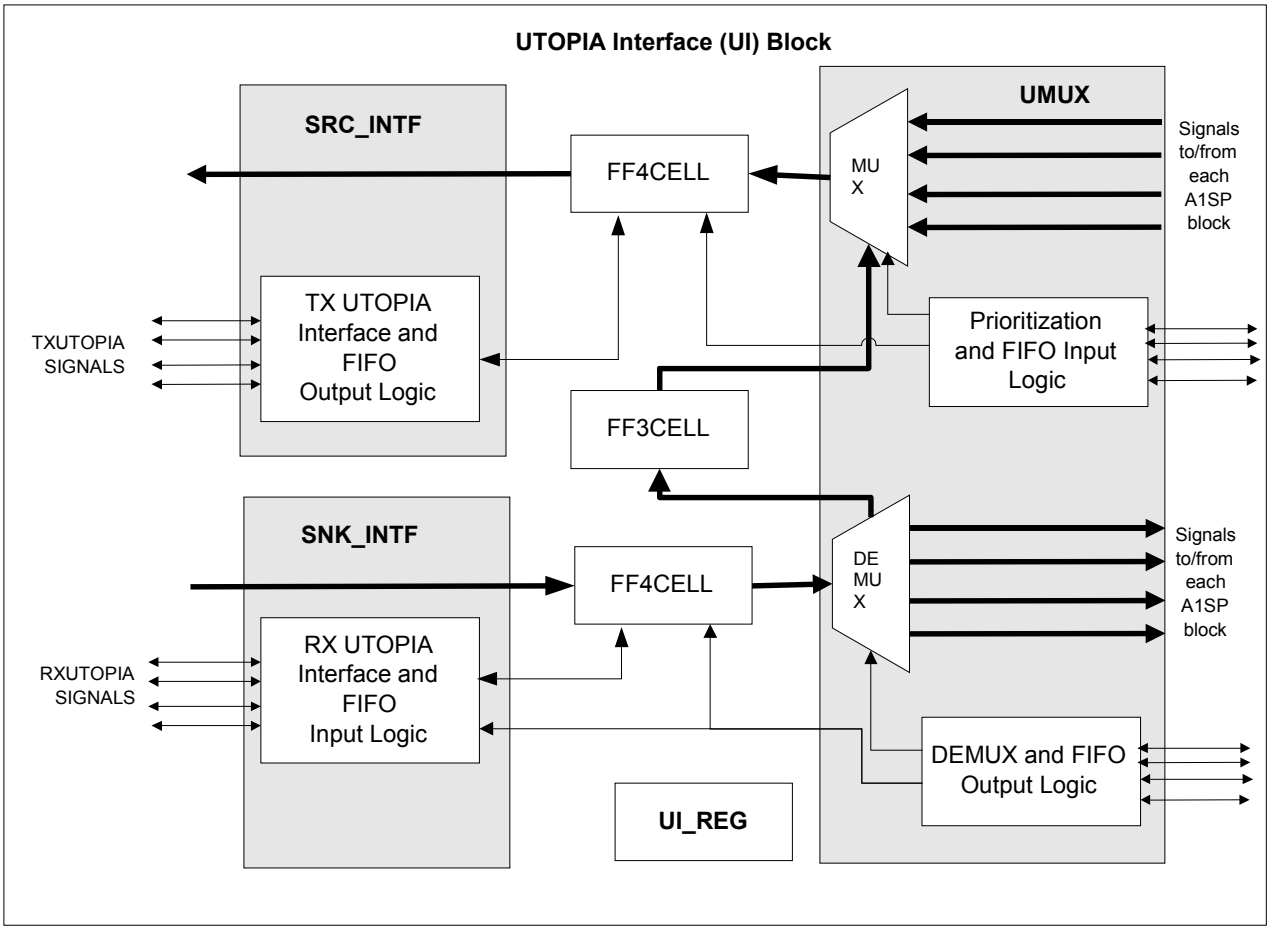
The Any-PHY interface extends beyond the 31 PHY device limit of UTOPIA Level 2 through the use of in-band addressing. One extra word indicating the port address, is prepended to the front of each cell that is transmitted and received.

The AAL1gator devices all support the Any-PHY interface.

Block Diagram

The UI block consists of 6 functions: UI Data Source Interface (SRC_INTF), UI Data Sink Interface(SNK_INTF), 4-cell FIFO (FF4CELL), 3-cell FIFO (FF3CELL), UMUX, and UI_REG. See Figure 7 for the block diagram of the AAL1_UI block.

Figure 7 – UTOPIA Interface Block Diagram



There is very little setup required to configure the UTOPIA Interface. For typical operation, the UI_COMN_CFG_REG, UI_SRC_CFG_REG, and UI_SNK_CFG_REG need to be written to select the mode of operation and the UI_SRC_ADD_CFG and UI_SNK_ADD_CFG need to be programmed for a pre-defined address of the device. Once the registers are written with the proper configuration information, the UI_EN bit in UI_COMN_CFG_REG should be set to enable normal operation.

Aside from the normal configurations, the block can also be placed in loopback where cells received on the UI interface are transmitted back out onto the UI interface.

The following registers control the configuration of the UTOPIA interface:

Address	Register Description	Register Mnemonic
0x80120	UTOPIA Common Configuration Register	UI_COMN_CFG
0x80121	UTOPIA Source Configuration Register	UI_SRC_CFG
0x80122	UTOPIA Sink Configuration Register	UI_SNK_CFG
0x80123	UTOPIA Source Address Configuration Register	UI_SRC_ADD_CFG
0x80124	UTOPIA Sink Address Configuration Register	UI_SNK_ADD_CFG
0x80125	UTOPIA to UTOPIA Loopback VCI Register	UI_U2U_LOOP_VCI

6.1 Common UTOPIA Interface Configuration

General configuration and enabling of the UTOPIA Interface is controlled by the **UI Common Configuration Register** (0x80120). Please note that every time the UTOPIA Interface needs to be reprogrammed, it is recommended to be in the Chip Software Reset state as described in section 5.2.2.

The default configuration is as follows:

Bit	Register	Value
UI_EN	**UI Common Configuration Register** (0x80120)	0
U2U_LOOP	**UI Common Configuration Register** (0x80120)	0
VCI_U2U_LOOP	**UI Common Configuration Register** (0x80120)	0
SHIFT_VCI	**UI Common Configuration Register** (0x80120)	0
VPI_MODE_EN	**UI Common Configuration Register** (0x80120)	0

The UTOPIA Interface is disabled in the default state. In addition, the remote loopback and VCI based loopback modes are disabled, and neither VCI shifting nor VP mode is used for cell header interpretation.

Enabling the UTOPIA Interface

The UI_EN bit enables both the source side and sink side UTOPIA Interface. This bit resets to the disabled state so that the chip resets with all UTOPIA outputs tristated. Once the modes have been configured and the interface

enabled, then the outputs will drive to their correct values. The other UI registers are not affected by this bit.

UI_EN	Function
0	The UI is disabled. All of the UI logic is held in reset and all of the FIFOs are cleared. The AAL1gator will not respond in the UTOPIA interface.
1	The UI is enabled in both directions.

UTOPIA to UTOPIA Loopback Modes

The AAL1gator supports two forms of UTOPIA to UTOPIA loopback; global loopback, where all cells are looped, and VC based loopback, where only a specific VC is used to loopback cells. The 3-cell FIFO is used for loopback.

In global loopback mode, all cells received by the UTOPIA block are sent back out onto the UTOPIA bus (regardless of single or multi-addressing mode). Global loopback is enabled by setting the U2U_LOOP bit:

U2U_LOOP	Function
0	The UI is in normal mode.
1	The UI is in global loopback mode

In VCI based loopback mode, any cell received with a VCI that matches the loopback VCI is sent back out onto the UTOPIA bus. The loopback VCI is programmable by writing the U2U_LOOP_VCI register (please see section 6.4). VCI based loopback is enabled by setting the VCI_U2U_LOOP bit:

VCI_U2U_LOOP	Function
0	The UI is in normal mode.
1	The UI is in VCI based loopback mode

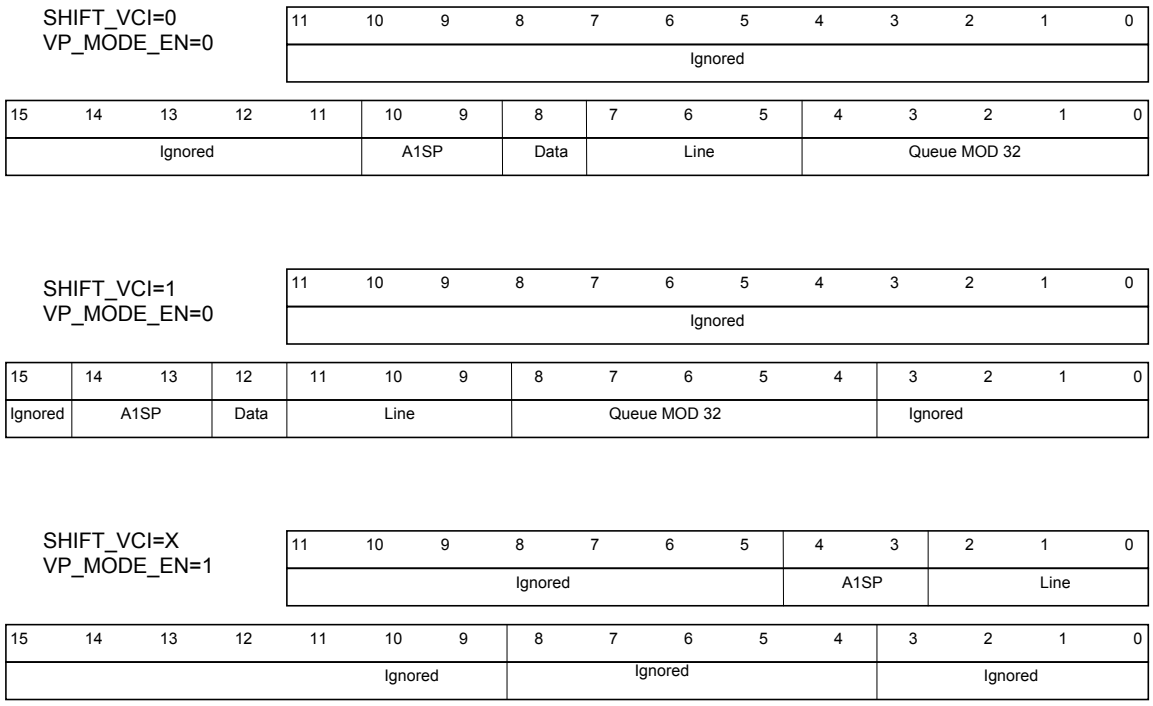
Cell Header Interpretation

The UMUX blocks serves as the bridge between the four or one A1SP blocks and the SNK_INTF and SRC_INTF blocks.

To determine which A1SP to forward a received cell, the UMUX looks at the VPI and VCI bits of each cell (unless in UTOPIA Level 2 multi-port mode or Any-PHY mode, in which case the bottom two address bits are used).

The SHIFT_VCI and VP_MODE_EN bits determine the interpretation of the VCI and VPI bits of the cell header, as shown in Figure 8.

Figure 8 – Cell Header Interpretation



The three possible interpretations are described below:

1. When SHIFT_VCI is low and VP_MODE_EN is low:
   - VCI[10:9] is used to select the A1SP
   - When VCI[8] is set, the cell is a data cell; when VCI[8] is low, the cell is an OAM cell
   - VCI[7:0] is used as the queue number if VCI[8] = 1
2. When SHIFT_VCI is set and VP_MODE_EN is low:
   - VCI[14:13] is used to select the A1SP

- When VCI[12] is set, the cell is a data cell; when VCI[12] is low, the cell is an OAM cell
- VCI[11:4] is used as the queue number if VCI[12] = 1

3. When VP_MODE_EN is set:
   - VPI[4:3] is used to select the A1SP
   - VPI[2:0] selects the line within the A1SP
   - Queue 0 will be assumed and no VCI bits need to be used to indicate queue number
   - If VCI <= 31, then interpret the cell as an OAM cell and place it in the OAM buffer

Note: VP_MODE_EN can only be set if all lines are in UDF mode.

6.2 UTOPIA Source Interface Configuration

The UTOPIA Source Interface (SRC_INTF) block conveys the cells received from the UMUX block to the UTOPIA interface.

Configuration of the UTOPIA source side interface is controlled by the **UI Source Configuration Register** (0x80121). Please note that every time the UTOPIA Interface needs to be reprogrammed, it is recommended to be in the Chip Software Reset state as described in section 5.2.2.

The default configuration is as follows:

Bit	Register	Value
UTOP_MODE[1:0]	**UI Source Configuration Register** (0x80121)	00
ANY-PHY_EN	**UI Source Configuration Register** (0x80121)	0
EVEN_PAR	**UI Source Configuration Register** (0x80121)	0
16_BIT_MODE	**UI Source Configuration Register** (0x80121)	0
CS_MODE_EN	**UI Source Configuration Register** (0x80121)	0
CFG_ADDR[15:0]	**Slave Source Address Configuration Register** (0x80123)	0x0000

In the default state, the source side interface is in UTOPIA Level 1 Master mode with odd parity generation.

Source Side Operating Mode

Depending on the value of UTOP_MODE[1:0] field, the UTOPIA interface will either act as a UTOPIA master (controls the write enable signal) or as a UTOPIA PHY device (controls the cell available signal). As a PHY device the SRC_INTF can either be a UTOPIA Level 1 device, where it is the only device on the UTOPIA bus, or a UTOPIA Level 2 device where other devices can coexist on the UTOPIA bus. As a master device the SRC_INTF can only function as a UTOPIA Level 1 device.

If ANY-PHY_EN is set then the SRC_INTF operates as a single port Any-PHY slave device. In Any-PHY mode in-band addressing is used to allow more than the 32 possible addresses available in UTOPIA mode. One extra word is prepended to the front of each cell that is transmitted. The prepended word indicates the port address sending the cell. The SRC_INTF uses CFG_ADDR[15:0] in the **UI_SRC_ADD_CFG** register (0x80123) for the address prepend.

The operating mode for the source side interface is configured using the UTOP_MODE[1:0], ANY-PHY_EN, and 16_BIT_MODE bits, as summarized in the table below. A "√" denotes that the field is selectable for that mode. An "n/a" denotes that the field is ignored for that mode.

Mode	UTOP_MODE[1:0]	ANY-PHY_EN	16_BIT_MODE
UTOPIA Level 1 Master	00	0	0
UTOPIA Level 1 Slave	01	0	√
UTOPIA Level 2 Single Address Slave	10	0	√
Any-PHY Slave	n/a	1	√

Notes:

- In the source side interface, UTOP_MODE[1:0] = "11" is Reserved and should not be used.
- If 16_BIT_MODE is set, then all 16 bits of the UTOPIA data bus or Any-PHY interface are used.

- If 16_BIT_MODE is low, then only the lower 8 bits are used.

Parity Generation

The EVEN_PAR bit determines the calculated parity across data bytes/words sent out of the source interface.

EVEN_PAR	Function
0	Odd parity
1	Even parity.

Chip Select Enable for Any-PHY Mode

The CS_MODE_EN bit is used to determine the use of the RPHY_ADDR(3)/RCSB input pin. This bit should only be set in Any-PHY mode.

CS_MODE_EN	Function
0	The RPHY_ADDR(3)/RCSB input pin is used as an address bit (RPHY_ADDR(3)) for the source side interface.
1	The RPHY_ADDR(3)/RCSB input pin is used as a chip select (RCSB) for the source side interface.

Slave Source Address

The CFG_ADDR[15:0] bits of the **Slave Source Address Configuration Register** (0x80123) contain the configured slave address used for UTOPIA Level 2 and Any-PHY operation in the source direction. Depending on the mode of the UTOPIA/Any-PHY interface different bits of this field are used. See the following table for details.

	Polling		Selection	
MODE	**PHY_ADDR Pins**	**CFG_ADDR**	**PHY_ADDR Pins**	**CFG_ADDR**
UTOPIA-2 Single-Addr	[4:0]=device	[4:0]=device	[4:0]=device	[4:0]=device
Any-PHY with CSB	[2:0]=device	[2:0]=device	[2:0]=device CFG_ADDR is prepended	[15:0]=device
Any-PHY without CSB	[3:0]=device	[3:0]=device	[3:0]=device CFG_ADDR is prepended	[15:0]=device

Notes:

- In Any-PHY mode in the SRC direction, the AAL1gator will prepend the cell with CFG_ADDR[15:0]. In 8-bit mode, the cell will be prepended with CFG_ADDR[7:0].
- In Any-PHY mode, if CS_MODE_EN = '0', then CFG_ADDR[4:3] = "00".
- In Any-PHY mode, if CS_MODE_EN = '1', then CFG_ADDR[3]="0".

6.3 UTOPIA Sink Interface Configuration

The UTOPIA Sink Interface (SNK_INTF) block receives cells from the UTOPIA interface and sends them to the UMUX interface.

Configuration of the UTOPIA sink side interface is controlled by the **UI Sink Configuration Register** (0x80122). Please note that every time the UTOPIA Interface needs to be reprogrammed, it is recommended to be in the Chip Software Reset state as described in section 5.2.2.

The default configuration is as follows:

Bit	Register	Value
UTOP_MODE[1:0]	**UI Sink Configuration Register** (0x80122)	00
ANY-PHY_EN	**UI Sink Configuration Register** (0x80122)	0
EVEN_PAR	**UI Sink Configuration Register** (0x80122)	0

Bit	Register	Value
16_BIT_MODE	**UI Sink Configuration Register** (0x80122)	0
CS_MODE_EN	**UI Sink Configuration Register** (0x80122)	0
CFG_ADDR[15:0]	**Slave Sink Address Configuration Register** (0x80124)	0x0000

In the default state, the sink side interface is in UTOPIA Level 1 Master mode with odd parity generation.

Sink Side Operating Mode

Depending on the value of the UTOP_MODE[1:0] field, the UTOPIA interface acts either as a UTOPIA master (controls the read enable signal) or as a UTOPIA PHY device (controls the cell available signal). As a PHY device the SNK_INTF can either be a UTOPIA Level 1 device, where it is the only device on the UTOPIA bus, or a UTOPIA Level 2 device where other devices can coexist on the UTOPIA bus. As a master device the SNK_INTF can only function as a UTOPIA Level 1 device.

If ANY-PHY_EN is set then the SNK_INTF operates as a multi port Any-PHY slave device. In Any-PHY mode in-band addressing is used to allow more than the 32 possible addresses available in UTOPIA mode. One extra word is prepended to the front of each cell that is transmitted. The prepended word indicates the port address to receive the cell. The SNK_INTF uses CFG_ADDR[15:2] in the **UI_SNK_ADD_CFG** register (0x80124) to match with the address prepend. If 16_BIT_MODE is low then CFG_ADDR(7:2) is used.

The operating mode for the sink side interface is configured using the UTOP_MODE[1:0], ANY-PHY_EN, and 16_BIT_MODE bits, as summarized in the table below. A "√" denotes that the field is selectable for that mode. An "n/a" denotes that the field is ignored for that mode.

Mode	UTOP_MODE[1:0]	ANY-PHY_EN	16_BIT_MODE
UTOPIA Level 1 Master	00	0	0
UTOPIA Level 1 Slave	01	0	√
UTOPIA Level 2 Single Address Slave	10	0	√
UTOPIA Level 2 Multi-Address Slave	11	0	√

Mode	UTOP_MODE[1:0]	ANY-PHY_EN	16_BIT_MODE
Any-PHY Slave	n/a	1	√

Notes:

- UTOPIA Level 2 Multi-Address Slave mode is only valid in the AAL1gator-32; for the AAL1gator-8 and AAL1gator-4, UTOP_MODE[1:0] = "11" is Reserved and should not be used.
- If 16_BIT_MODE is set, then all 16 bits of the UTOPIA data bus or Any-PHY interface are used.
- If 16_BIT_MODE is low, then only the lower 8 bits are used.

Parity Generation

The EVEN_PAR bit determines the calculated parity across data bytes/words sent out of the source interface.

EVEN_PAR	Function
0	Odd parity
1	Even parity.

Chip Select Enable for Any-PHY Mode

The CS_MODE_EN bit is used to determine the use of the TPHY_ADDR(3)/TCSB input pin. This bit should only be set in Any-PHY mode.

CS_MODE_EN	Function
0	The TPHY_ADDR(3)/TCSB input pin is used as an address bit (TPHY_ADDR(3)) for the sink side interface.
1	The TPHY_ADDR(3)/TCSB input pin is used as a chip select (TCSB) for the sink side interface.

Slave Sink Address

The CFG_ADDR[15:0] bits of the **Slave Sink Address Configuration Register** (0x80124) contain the configured slave address used for UTOPIA Level 2 and Any-PHY operation in the sink direction. Depending on the mode of the

UTOPIA/Any-PHY interface different bits of this field are used. See the following table for details.

	Polling		Selection	
MODE	**PHY_ADDR Pins**	**CFG_ADDR**	**PHY_ADDR Pins**	**CFG_ADDR**
UTOPIA-2 Single-Addr	[4:0]=device	[4:0]=device	[4:0]=device	[4:0]=device
UTOPIA-2 Multi-Addr	[4:2]=device [1:0]=A1SP	[4:2]=device	[4:2]=device [1:0]=A1SP	[4:2]=device
Any-PHY with CSB	[2]=device [1:0]=A1SP	[2]=device	[2]=device [1:0]=A1SP CFG_ADDR is prepended	[15:2]=device
Any-PHY without CSB	[3:2]=device [1:0]=A1SP	[3:2]=device	[3:2]=device [1:0]=A1SP CFG_ADDR is prepended	[15:2]=device

Notes:

- In Any-PHY mode, the upper 14 bits of the prepended address are compared with CFG_ADDR[15:2]. The bottom two bits are not compared with this field and are just used to select the target A1SP. If in 8-bit mode CFG_ADDR[7:2] is used instead.
- In Any-PHY mode, if CS_MODE_EN='0', then CFG_ADDR[4:3] = "00".
- In Any-PHY mode, if CS_MODE_EN='1', then CFG_ADDR[3]="0".

6.4 VCI Based UTOPIA to UTOPIA Loopback

The U2U_LOOP_VCI[15:0] bits in the **U2U_LOOP_VCI** (0x80125) register specify the VCI to be used for VCI based UI to UI loopback. If VCI_U2U_LOOP is set in the **UI_COMN_CFG_REG** (0x80120), then any cell received from the UI bus, with a VCI which matches this programmed VCI, will be sent back out to the UI bus. The value of this register should be changed only when VCI_U2U_LOOP is disabled. U2U_LOOP_VCI[15:0] defaults to 0x0000.

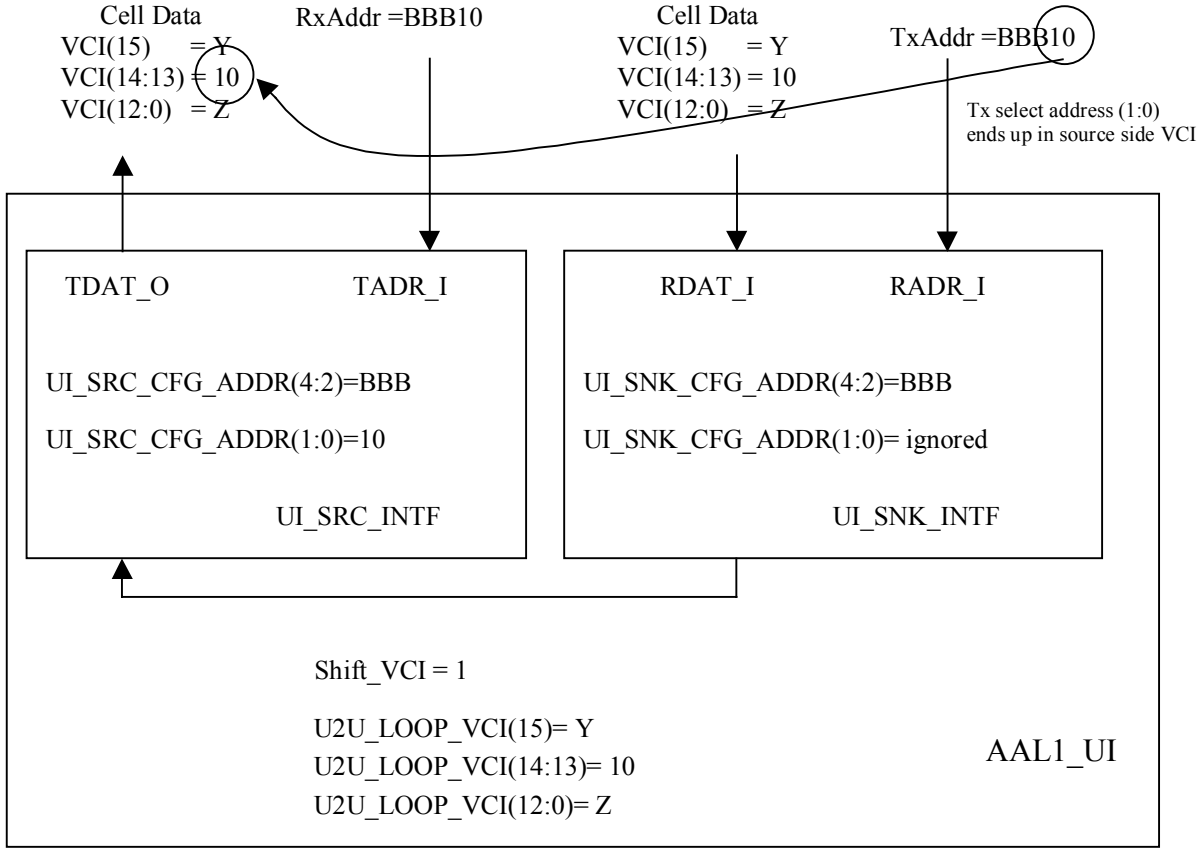
6.4.1 VCI Loopback Setup Example in Multi-Address mode

Internal routing circuitry in UTOPIA Level 2 multi-address mode requires some special consideration of the VCI values and select addresses. An example of a proper configuration for VCI based loopback while in UTOPIA Level 2 multi-address mode is shown in Figure 9. Note that cells sent to the sink interface address TxAddr=bbb10 (where bbb is the configured base address) which have VCI[15]=y and VCI[12:0]=z will be looped back because the resulting internal VCI will match that in the **U2U_LOOP_VCI** (0x80125) register.

To maintain correct functionality, the cell's VCI[14:13] should match that of the lower bits of the select address TxAddr, i.e. VCI[14:13]=10, and thus will appear unchanged when exiting the source side. If Shift_VCI= 0 then all requirements on VCI[14:13] are placed on VCI[10:9] instead.

All loopback cells will appear at the source side interface, regardless of the setting of source configuration address (UI_SRC_CFG_ADR), but to maintain symmetry the source configuration address should be set to RxAddr=bbb10 in this example so that looped back cells appear on the same source address (Rx slave) port as the sink address (Tx slave). An alternate setting would be to set incoming VCI[14:13] = TxAddr(1:0) = RxAddr(1:0) = "00", thus using the sink side base address as the loopback address.

Figure 9 – UTOPIA Level 2 Multi-Address Mode with VCI Based Loopback



7 DATA STRUCTURES

Please note that only the memory locations for A1SP 0 are used for the AAL1gator-8, and that only the memory locations for lines 0 through 3 of A1SP 0 are used for the AAL1gator-4.

7.1 Transmit Data Structures

Figure 10 shows the format of the Transmit Data Structures block.

Figure 10 – Transmit Data Structures Memory Map

00020 0002F	T_SEQNUM_TBL
00030 0003F	T_ADD_QUEUE
00040 003FF	Unused
00400 0047F	T_COND_SIG
00480 004FF	T_COND_DATA
00500 006FF	Unused
00700 007FF	Reserved (Frame Advance FIFO)
00800 00FFF	Reserved (Transmit Calendar)
01000 013FF	Reserved (Transmit Signaling Buffer)
01400 0143F	T_OAM_QUEUE
01440 01FFF	Unused
02000 03FFF	T_QUEUE_TBL
04000 07FFF	Reserved (Transmit Data Buffer)

Note the addresses listed below are the offsets within each A1SP address space as described in section 4. Note "A" in the Addr column in Table 3 means A1SP digit: (000x=A1SP0, 011x=A1SP3)

Table 3 – Transmit Structures Summary

Name	R/W	Org	Size	Addr	Description
P_FILL_CHAR	R/W	1 word	2 bytes	0004H	The empty bytes in a partially filled cell are filled with P_FILL_CHAR.
Reserved(AQ)	R/W	16 words	32 bytes	0030H – 003FH	(Reserved (AQ))
T_SEQNUM_TBL	R/W	16 words	32 bytes	0020H – 002FH	The Transmit Sequence Number Table is initialized according to a table.
T_COND_SIG	R/W	32 bytes x 8 lines	256 bytes	A 0400H - A 047FH	This table stores the signaling to be used when the TX_COND bit in the T_QUEUE_TBL is set.
T_COND_DATA	R/W	32 bytes x 8 lines	256 bytes	0480H-04FFH	This table stores the data to be used when the TX_COND bit in the T_QUEUE_TBL is set.
Reserved	R/W	256 words	512 bytes	0700H-07FFH	Reserved (Frame Advance FIFO).
Reserved	R/W	8 x 128 x 2 words	4 kBytes	0800H-0FFFH	Reserved (Transmit Calendar).
Reserved	R/W	8 x 256 bytes	2 kBytes	1000H-13FFH	Reserved (Transmit Signaling Buffer).
T_OAM_QUEUE	R/W	2 x 32 words	128 bytes	1400H-143FH	The Transmit OAM Queue contains the OAM cells to be transmitted.

Name	R/W	Org	Size	Addr	Description
T_QUEUE_TBL	R/W	256 x 32 words	16 kBytes	2000H-3FFFH	The Transmit Queue Table contains all pointers and variables that are queue-dependent.
Reserved	R/W	8 x 2 K words	32 kBytes	4000H-7FFFH	Reserved (Transmit Data Buffer).

Notes:

All ports marked as "Reserved" must be initialized to 0 at initial setup. Software modifications to these locations after setup will cause incorrect operation.

All read/write port bits marked "Not used" must be written with the value 0 to maintain software compatibility with future versions.

All read-only port bits marked "Not used" are driven with a 0 and should be masked off by the software to maintain compatibility with future versions.

7.1.1 P_FILL_CHAR

Organization: One word

Base address within A1SP: 4H

Type: Read/Write

Function: Contains the fill character for partially filled cells.

Format: Refer to the following table.

Field (Bits)	Description
Not used (15:8)	Write with a 0 to maintain future software compatibility.
P_FILL_CHAR (7:0)	Character used in partially filled cells. Initialize to the desired value.

7.1.2 T_SEQNUM_TBL

Organization: 16 words

Base address within A1SP: 20H

Index: 1H

Type: Read/Write

Function: Stores all possible first bytes in the payload: CSI, SN, and SNP. This table must be loaded into the SRAM on every power cycling.

Initialization: Initialize to the values in the following table

Offset	Data Value
0H	0000H
1H	0017H
2H	002DH
3H	003AH
4H	004EH
5H	0059H
6H	0063H
7H	0074H
8H	008BH
9H	009CH
AH	00A6H
BH	00B1H
CH	00C5H
DH	00D2H
EH	00E8H
FH	00FFH

7.1.3 T_COND_SIG

Organization: 32 bytes x 8 lines

Base address within A1SP: 400H

Index: 10H

Type: Read/Write

Function: Stores the transmit conditioned signaling.

Initialization: Initialize to the conditioned signaling value for the channel. This value typically depends on the type of channel unit that is connected. For example, a Foreign Exchange Office (FXO) needs a different conditioning value than a Foreign Exchange Subscriber (FXS).

Format: One nibble per byte, two bytes per word, 16 words per line. Refer to the following table.

Offset	Name	Description
00000H	T_COND_SIG_0	Transmit conditioned signaling for line 0.
00010H	T_COND_SIG_1	Transmit conditioned signaling for line 1.
00020H	T_COND_SIG_2	Transmit conditioned signaling for line 2.
00030H	T_COND_SIG_3	Transmit conditioned signaling for line 3.
00040H	T_COND_SIG_4	Transmit conditioned signaling for line 4.
00050H	T_COND_SIG_5	Transmit conditioned signaling for line 5.
00060H	T_COND_SIG_6	Transmit conditioned signaling for line 6.
00070H	T_COND_SIG_7	Transmit conditioned signaling for line 7.

T_COND_SIG_n Word Format

Field (Bits)	Description
Not used (15:12)	Write with a 0 to maintain future software compatibility.
T_COND_SIG_A_H (11)	Transmit conditioned A bit for: Offset = ((channel -1) / 2) + line x 16.

Field (Bits)	Description
T_COND_SIG_B_H (10)	Transmit conditioned B bit for: Offset = ((channel -1) / 2) + line x 16.
T_COND_SIG_C_H (9)	Transmit conditioned C bit or A bit if T1 SF for: Offset = ((channel -1) / 2) + line x 16.
T_COND_SIG_D_H (8)	Transmit conditioned D bit or B bit if T1 SF for: Offset = ((channel -1) / 2) + line x 16.
Not used (7:4)	Write with a 0 to maintain future software compatibility.
T_COND_SIG_A_L (3)	Transmit conditioned A bit for: Offset = (channel / 2) + line x 16.
T_COND_SIG_B_L (2)	Transmit conditioned B bit for: Offset = (channel / 2) + line x 16.
T_COND_SIG_C_L (1)	Transmit conditioned C bit or A bit if T1 SF for: Offset = (channel / 2) + line x 16.
T_COND_SIG_D_L (0)	Transmit conditioned D bit or B bit if T1 SF for: Offset = (channel / 2) + line x 16.

7.1.4 T_COND_DATA

Organization: 32 bytes x 8 lines

Base address within A1SP: 480H

Index: 10H

Type: Read/Write

Function: Stores the transmit conditioned data.

Initialization: Initialize to the conditioned data appropriate for the channel, which typically depends on the type of channel connected to the device. For example, data usually needs an FFH value and voice needs a small Pulse Coded Modulation (PCM) value.

Format: Two bytes per word, 16 words per line. Refer to the following table.

Offset	Name	Description
00000H	T_COND_DATA_0	Transmit conditioned data for line 0.
00010H	T_COND_DATA_1	Transmit conditioned data for line 1.
00020H	T_COND_DATA_2	Transmit conditioned data for line 2.
00030H	T_COND_DATA_3	Transmit conditioned data for line 3.
00040H	T_COND_DATA_4	Transmit conditioned data for line 4.
00050H	T_COND_DATA_5	Transmit conditioned data for line 5.
00060H	T_COND_DATA_6	Transmit conditioned data for line 6.
00070H	T_COND_DATA_7	Transmit conditioned data for line 7.

T_COND_DATA_n Word Format

Field (Bits)	Description
T_COND_DATA_H (15:8)	Transmit conditioned data offset = ((channel / 2) + 1) + line x 16.
T_COND_DATA_L (7:0)	Transmit conditioned data offset = (channel / 2) + line x 16.

7.1.5 RESERVED (Transmit Signaling Buffer)

This structure is reserved and need not be initialized to 0. Software modifications to this structure after setup will cause incorrect operation.

Organization: Eight multiframes x 32 DS0s x 8 lines. Each of the eight lines are allocated a separate signaling buffer. Each DS0 generates one new nibble of signaling per multiframe. The data is stored in the buffer in the order it is received from the framer device. Different framers provide the signaling information in different formats, as the following illustration shows, for one multiframe worth of signaling data.

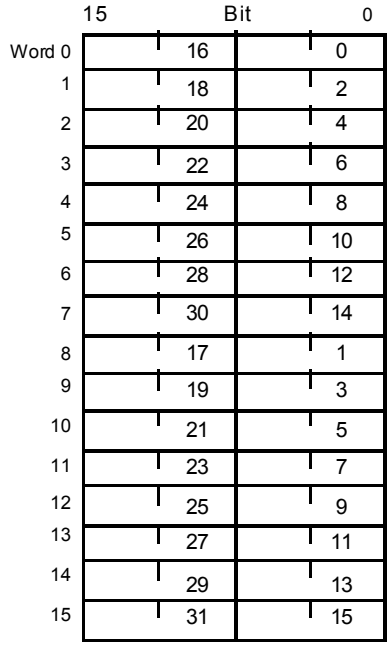
Base address: 01000H

Index: 80H

Type: Read/Write

Function: Stores the outgoing signaling data.

Figure 11 – SDF-MF Format of the T_SIGNALING BUFFER



The upper nibble of each byte is 0.

7.1.6 T_OAM_QUEUE

Organization: 2 cells x 32 words

Base address within A1SP: 01400H

Index: 20H

Type: Read/Write

Function: Stores two transmit OAM cells.

Initialization: An optimization is to initialize to the body of an OAM cell so only the header must be modified before sending.

Format: Refer to the following table.

Offset	Name	Description
01400H	T_OAM_CELL_1	Transmit OAM cell 1.
01420H	T_OAM_CELL_2	Transmit OAM cell 2.

T_OAM_CELL_n Format

Offset	Bits 15:8	Bits 7:0
Word 0	Header 1	Header 2
Word 1	Header 3	Header 4
Word 2	Header 5 (HEC) (Pre-calculated by software)	Bits 7:1 Not used. Set to 0. Bit 0 0 Disables CRC-10 insertion. 1 Enables CRC-10 insertion.
Word 3	Payload 1	Payload 2
.	.	.
.	.	.
.	.	.
Word 26	Payload 47	Payload 48
	If CRC-10 is enabled in Word 2, set data to 0 in Word 26. Word 26 will be replaced by the computed CRC-10 result as the cell is transmitted.	

Note: Programming the HEC (word 2 of T_OAM_CELL_n) is optional if this is already done in the PHY device that is interconnected to the AAL1gator.

7.1.7 T_QUEUE_TBL

Organization: 256 x 32 words

Base address within A1SP: 2000H

Index: 20H

Type: Read/Write

Function: Configures the VCs.

Format: Each queue will be allocated 32 consecutive words.

Offset	Name	Description
0H	Reserved	(Data pointer.) Initialize to FFFFH each time this queue is initialized.
1H	Not used	Initialize to '0' each time this queue is initialized to maintain future software compatibility.
2H	T_COND_CELL_CNT	A 16-bit rollover count of conditioned cells transmitted.
3H	T_SUPPRESS_CNT	A 16-bit rollover count of cells not sent because of a line resynchronization. Or, if in UDF-HS mode, a 16-bit rollover count of cells not sent because TX_ACTIVE is not set. This counter also counts when cells are not sent because SUPPRESS_TRANSMISSION is set.
4H	Not used	Initialize to '0' each time this queue is initialized to maintain future software compatibility.
5H	Reserved	(Sequence number.) Initialize to '0' each time this queue is initialized.
6H	QUEUE_CONFIG	The configuration of the current queue. Initialize to the proper value.
7H	T_CELL_CNT	A 16-bit count of the cells transmitted.
8H	TX_HEAD(1:2)	Header byte 1 in bits 15:8, header byte 2 in bits 7:0.
9H	TX_HEAD(3:4)	Header byte 3 in bits 15:8, header byte 4 in bits 7:0.
AH	TX_HEAD(5)	Header byte 5 (pre-calculated HEC) in bits 15:8.
BH	QUE_CREDITS	A 10-bit quantity representing the number of byte credits accumulated for the queue.
CH	CSD_CONFIG	Stores the average number of bytes in each cell, and carries the number of DS0s for this queue.
DH	Not used	Initialize to '0' each time this queue is initialized to maintain future software compatibility.
EH	T_CHAN_ALLOC(15:0)	A bit table with a bit set per DS0 allocated to this queue for DS0s 15:0 on the line defined by queue / 32.

Offset	Name	Description
FH	T_CHAN_ALLOC(31:16)	A bit table with a bit set per DS0 allocated to this queue for DS0s 31:16 on the line defined by queue / 32.
10H	T_CHAN_LEFT(15:0)	Initialize to the same value as T_CHAN_ALLOC(15:0).
11H	T_CHAN_LEFT(31:16)	Initialize to the same value as T_CHAN_ALLOC(31:16).
12H	TRANSMIT_CONFIG	Controls transmission of data.
13H	Reserved	(T_CUR_ACT_CHAN(15:0)) Initialize to 0 each time this queue is initialized.
14H	Reserved	(T_CUR_ACT_CHAN(31:16)) Initialize to 0 each time this queue is initialized.
15H	Reserved	(T_NEW_ACT_CHAN(15:0)) Initialize to 0 each time this queue is initialized.
16H	Reserved	(T_NEW_ACT_CHAN(15:0)) Initialize to 0 each time this queue is initialized.
17H	Reserved	(CSD_BYTES_LEFT) Only used in DBCES mode. When operating in DBCES mode this register must be initialized to the structure size minus the portion of a structure that fits in the first cell. The formula to calculate this value is: struct_size - ((46-bitmask_size) MOD struct_size) The number of data bytes in the first cell is 47 minus the structure pointer and the bitmask size. The MOD operation determines the number of bytes from the structure that make it into the first cell. This number is then subtracted from the structure size to determine how many bytes are left in the structure after the first cell.
18H-1FH	Not used	Initialize to 0 each time this queue is initialized.

T_COND_CELL_CNT Word Format

Initialize to "0000" and at all other times the word is read only. The word maintained by TALP.

Field (Bits)	Description
T_COND_CELL_CNT (15:0)	A 16-bit rollover count of conditioned cells transmitted. This counter increments when cells with conditioned data is sent. If only signaling is conditioned this counter will not increment.

T_SUPPRESS_CNT Word Format

Initialize to "0000" and at all other times the word is read only. The word is maintained by TALP.

Field (Bits)	Description
T_SUPPRESS_CNT (15:0)	A 16-bit rollover count of cells not sent because of a line resynchronization. Or, if in UDF-HS mode, a 16-bit rollover count of cells not sent because TX_ACTIVE is not set. This counter also counts when cells are not sent because SUPPRESS_TRANSMISSION is set.

QUEUE_CONFIG Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
TX_COND (15)	Sends data and signaling from the transmit conditioned data area according to the conditioning mode selected in the TRANSMIT_CONFIG register. Initialize to the proper value.

Field (Bits)	Description
TX_ACTIVE (14)	Enables this queue. To enable connections: • 1)Assert this bit. • 2)Add this queue to the ADDQ_FIFO Register. To disable connections, clear the TX_ACTIVE bit. This queue is then removed from the calendar queue the next time a cell would have been sent. Once this bit is cleared, the associated queue must not be returned to the add-queue FIFO until FRAMES_PER_CELL frames have passed by. If quick reconfiguration is required and the size of the queue is not going to change (number of allocated channels), then use SUPPRESS_XMT bit to pause queue and reconfigure instead of clearing TX_ACTIVE bit. When reactivating a previously active queue, be sure to reinitialize all the registers in the queue table for that queue.

Field (Bits)	Description
FRAMES_PER_CELL (13:8)	A 6-bit integer specifying the maximum number of frames required to have enough data to construct a cell (round up of BYTE_PER_CELL/number of DS0s assigned) plus 1. For example, for a T1 line in SDF-FR mode with five DS0s, initialize this field to 11. In T1 SDF-MF or SDF-FR modes, the FRAMES_PER_CELL is encoded as the number of 24-frame multiframes required in bit 13 and the number of frames mod 24 in bits 12:8. In all other modes, including unstructured T1 mode, encode this value as the maximum number of 256 bit increments required to create a cell. For unstructured mode with full cells, set this value to 3. • For channels with a single DS0, encode the value 48 as one multiframe and 24 frames. • When calculating the FRAMES_PER_CELL value, do not subtract the bytes used by signaling nibbles from the value. For example, for an SDF-MF, single DS0, full cell connection, use the value 47 + 1 = 48 and not 46 + 1 = 47. • For SDF-MF connections using partial cells, set FRAMES_PER_CELL to (round up of BYTE_PER_CELL/number of DS0s assigned) plus 2. This prevents scheduling more than one cell per frame.
T_CHAN_NO_SIG (7)	Set to 1 to send cells with no signaling when in SDF-MF mode. This is the same as using this queue in SDF-FR mode, which means the structure forms on frame boundaries instead of multiframe boundaries.
T_CHAN_UNSTRUCT (6)	Set to 1 only when sending cells with a single DS0 without a pointer in the SDF-FR mode. To conform to the CES standard V 2.0 when using a single DS0 in SDF-FR mode, no pointer should be used.

Field (Bits)	Description
BYTES_PER_CELL (5:0)	A 6-bit integer specifying how many bytes per cell are required if no structure pointers are used. For UDF_HS mode, this value must be 47. This number must be set so the cell generation rate per queue is slower than once per frame. For unstructured lines, this means between 33 and 47. For structured applications, the BYTES_PER_CELL number must exceed the number of DS0 channels allocated to the queue. For example, a two channel queue may have the number set from 3 to 47. For SDF-MF connections with more than 16 channels allocated, the BYTES_PER_CELL number must exceed the number of DS0 channels allocated to the queue by two. For example, a 17 channel SDF-MF queue may have the number set from 19 to 47. For AAL0 connections this field should be set to 48. This is due to the fact that there is no sequence number byte in AAL0 cells.

T_CELL_CNT Word Format

Initialize to "0000" and at all other times the word is read only. The word is maintained by TALP.

Field (Bits)	Description
T_CELL_CNT (15:0)	A 16-bit count of the data cells transmitted. Rolls to 0 from FFFFH. Initialize to 0. After initialization, do not write to this word.

TX_HEAD(1:2) Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
TX_HEAD(1) (15:8)	First header byte in bits 15:8. Initialize to the proper value.
TX_HEAD(2) (7:0)	Second header byte in bits 7:0. Initialize to the proper value.

TX_HEAD(3:4) Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
TX_HEAD(3) (15:8)	Third header byte in bits 15:8. Initialize to the proper value.
TX_HEAD(4) (7:0)	Fourth header byte in bits 7:0. Initialize to the proper value.

TX_HEAD(5) Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
TX_HEAD(5) (15:8)	Fifth header byte that contains the precalculated HEC word. Initialize to the proper value.
Not used (7:0)	Write with a 0 to maintain compatibility with future software versions.

Note: Programming the HEC is optional if this is already done in the PHY device that is interconnected to the AAL1gator.

QUE_CREDITS Word Format

After initialization this word is read only. The word is maintained by CSD.

Field (Bits)	Description
FRAME_REMAINDER (15:14)	A 2-bit quantity representing the remainder of the division operation the CSD performs when converting the frame differential (expressed in frames) to the frame differential (expressed in eighths of multiframes). This quantity is maintained by the CSD. Initialize to 00b.
Not used (13:10)	Write with a 0 to maintain compatibility with future software versions.
QUEUE_CREDITS (9:0)	A 10-bit quantity representing the number of byte credits accumulated for the queue. It is measured in eighths (three LSBs are fractional bits). Initialize to 47 x 8 (178H) for UDF modes full cells, 46.875 x 8 (177H) for SDF mode full cells, or to the partially filled cell length x 8. For SDF-MF queues start with (177H) and add 8 times the number of signaling bytes which would occur in the first cell. (For 1 DS0 (E1) this is (187H), for 1 DS0(T1) or 2 DS0s (E1) this is (17FH). For all other configurations the initial value should be (177H).

CSD_CONFIG Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
NUM_CHAN (15:10)	A 6-bit integer specifying the number of DS0 channels being carried by this queue. If a queue serves seven DS0s, initialize this field to 7. This field has to be set to 32 in UDF-ML mode. It is not used in UDF-HS mode.

Field (Bits)	Description
AVG_SUB_VALU (9:0)	A 10-bit integer representing the average number of data bytes per cell measured in eighths. The three LSBs represent bits after the fixed decimal point. Initialize to 46.875 (0101110.111) for full cells when in SDF-FR or SDF-MF mode. Initialize to 47 (0101111.000) for full cells when in UDF-ML mode. For partial cells, this value is the same as the partially filled value x 8. This field is not used in UDF-HS mode.

T_CHANNEL_ALLOC(15:0) Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
T_CHANNEL_ALLOC (15:0)	A bit table with a bit set per DS0 allocated to this queue for DS0s 15 to 0 on the line defined by queue / 32. Initialize to the proper value for SDF-MF and SDF-FR modes and to FFFFH for UDF-ML and UDF-HS modes.

T_CHANNEL_ALLOC(31:16) Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
T_CHANNEL_ALLOC (31:16)	A bit table with a bit set per DS0 allocated to this queue for DS0s 31 to 16 on the line defined by queue / 32. Initialize to the proper value for SDF-MF and SDF-FR modes and to FFFFH for UDF-ML and UDF-HS modes.

T_CHANNEL_LEFT(15:0) Word Format

After initialization this word is read only. The word is maintained by TALP.

Field (Bits)	Description
T_CHANNEL_LEFT (15:0)	Initialize to the same value as T_CHAN_ALLOC(15:0).

T_CHANNEL_LEFT(31:16) Word Format

After initialization this word is read only. The word is maintained by TALP.

Field (Bits)	Description
T_CHANNEL_LEFT (31:16)	Initialize to the same value as T_CHAN_ALLOC(31:16).

TRANSMIT_CONFIG Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
SUPPRESS_XMT (15)	Set to 1 to suppress the generation of cells for this queue. Cells are scheduled but not transmitted.
LOOPBACK_ENABLE (14)	Set to 1 to loopback cell to receive side. Set VPI/VCI to corresponding receive queue number.
AAL0_MODE_ENABLE (13)	Set to 1 to build AAL0 cells instead of AAL1.

Field (Bits)	Description
COND_MODE (12:11)	Selects the conditioning mode with the following encoding: 00 Both signaling and data are conditioned 01 Only signaling is conditioned 10 Only data is conditioned 11 reserved The chosen mode takes effect when the TX_COND bit is set in the QUEUE_CONFIG memory register. If data is conditioned the T_COND_CELL_CNT counter will increment. If only signaling is conditioned the T_CELL_CNT will increment as normal.
DBCES_ENABLE (10)	Set to 1 to enable DBCES functionality.
IDLE_DET_ENABLE (9)	Set to 1 to enable idle detection in non-DBCES mode. When in this mode a queue which has all idle channels will have its transmission of cells suppressed. Any suppressed cell will cause the T_SUPPRESS_CNT to be incremented.
Not used (8:0)	Write with a '0' to maintain future software compatibility.

7.1.8 RESERVED (Transmit Data Buffer)

This structure is reserved and must be initialized to 0 at initial setup. Software modifications to this location after setup will cause incorrect operation.

Organization: 4 kBytes x 8 lines - Each line is allocated a separate 128 frame buffer memory. For E1 applications, this is large enough to store eight multiframes (32 DS0s x 16 frames x 8 multiframes = 4096 bytes). In T1 mode, 96 frames or four multiframes are stored (24 ¥ 24 ¥ 4 = 2880bytes). T1 storage uses 32 bytes per frame and 32 frames per multiframe to simplify address generation. Every data byte is stored in the multiframe line buffers in the order in which it arrives.

If E1_with_T1_SIG is set, data is arranged as if in T1 mode.

Base address within A1SP: 4000H

Index(line): 800H

Type: Read/Write

Function: Stores the outgoing data.

Format: Two data bytes per word, 16 words per frame.

T_DATA_BUFFER Word Format

Field (Bits)	Description
T_DATA_H (15:8)	Transmit data for: Channel = (offset mod 16) x 2 + 1. E1 offset = line x 2048 + multiframe x 256 + frame x 16 + (channel - 1) / 2. T1 offset = line x 2048 + multiframe x 512 + frame x 16 + (channel - 1) / 2.
T_DATA_L (7:0)	Transmit data for: Channel = (offset mod 16) x 2. E1 offset = line x 2048 + multiframe x 256 + frame x 16 + channel / 2. T1 offset = line x 2048 + multiframe x 512 + frame x 16 + channel / 2.

7.2 Receive Data Structures

Figure 12 shows the format of the Receive Data Structures block in more detail.

Figure 12 – Receive Data Structures

08000 08001	R_OAM_QUEUE_TBL
08002	R_OAM_CELL_CNT
08003	R_DROPPED_OAM_CELL_CNT
08004 0801F	Unused
08020 0802F	Reserved (SRTS Queue Pointers)
08030 08037	Unused
08038 0803F	R_SRTS_CONFIG
08040 0807F	Unused
08080 080FF	R_CRC_SYNDROME
08100 081FF	Unused
08200 0827F	R_CH_TO_QUEUE_TBL
08280 083FF	Unused
08400 0847F	R_COND_SIG
08480 084FF	R_COND_DATA
08500 087FF	Unused
08800 08FFF	Reserved (Receive SRTS Queue)
09000 09FFF	Reserved (Receive Signaling Buffer)
0A000 0BFFF	R_QUEUE_TBL
0C000 0DFFF	Unused
0E000 0FFFF	R_OAM_QUEUE
10000 1FFFF	Reserved (Receive Data Buffer)

Note the addresses listed below are the offsets within each A1SP address space as described in section 4.

Name	Org	Size	Addr	Description
R_OAM_QUEUE_TBL	2 words	4 bytes	8000H–8001H	Receive OAM head and tail pointers.
R_OAM_CELL_CNT	1 word	2 bytes	8002H	Count of received OAM cells.
R_DROP_OAM_CELL	1 word	2 bytes	8003H	Count of dropped OAM cells.
Reserved	16 words	32 bytes	8020H-802FH	Reserved (SRTS Queue Pointers).
R_SRTS_CONFIG	2 bytes x 8 lines	16 bytes	8038H-803FH	Receive SRTS configuration.
R_CRC_SYNDROME	128 words	256 bytes	8080H-80FFH	Mask of bits. Initialized from a table.
R_CH_TO_QUE_TBL	128 words	256 bytes	8200H-827FH	Receive channel to queue table.
R_COND_SIG	16 x 8 bytes	256 bytes	8400H-847FH	Receive signaling conditioning values.
R_COND_DATA	32 x 8 bytes	256 bytes	8480H-84FFH	Receive data conditioning values.
Reserved	8 x 256 words	4 kBytes	8800H-8FFFH	Reserved (Receive SRTS Queue).
Reserved	8 x 32 x 16 words	8 kBytes	9000H-9FFFH	Reserved (Receive Signaling Buffer).
R_QUEUE_TBL	256 x 32 words	16 kBytes	A000H-BFFFH	Receive queue table.
R_OAM_QUEUE	256 x 64 bytes	16 kBytes	E000H-FFFFH	Receive OAM queue.
Reserved	8 x 512 x 32 bytes	128 kBytes	10000H-1FFFFH	Reserved (Receive Data Buffer).

This section describes the structures used by the receive side of the AAL1gator.

Notes:

- All ports marked as "Reserved" must be initialized to 0 at initial setup. Software modifications to these locations after setup will cause incorrect operation.
- All read/write port bits marked "Not used" must be written with the value 0 to maintain software compatibility with future versions.
- All read-only port bits marked "Not used" are driven with a 0 and should be masked off by the software to maintain compatibility with future versions.

7.2.1 R_OAM_QUEUE_TBL

Organization: 2 words

Base address within A1SP: 8000H

Index: 1H

Type: Read/Write

Function: OAM cells received from the ATM side are stored in a FIFO queue in the memory. Head and tail pointers are used to keep track of the read and write locations of the OAM cell buffers. There are 256 cell buffers in the OAM receive queue. Of these 256 cell buffers, 255 are usable. The 256th buffer is used to detect a full queue as follows:

When the queue is empty, OAM_HEAD = OAM_TAIL = N. When a cell is received, the cell is written into the buffer at index (OAM_TAIL + 1) mod 256, and OAM_TAIL is replaced with (OAM_TAIL + 1) mod 256. When the processor receives an interrupt, it reads the cell at the buffer index (OAM_HEAD + 1) mod 256. After completing the read, it sets OAM_HEAD to (OAM_HEAD + 1) mod 256. This process is continued until OAM_HEAD = OAM_TAIL, at which time the OAM receive queue is empty. The receive OAM interrupt can be cleared by asserting the CLR_RX_OAM_LATCH bit in the CMD_REG.

If an OAM cell arrived between the time the OAM_TAIL was last read and CLR_RX_OAM_LATCH was asserted, this OAM cell's arrival can be detected within the interrupt service routine by re-reading OAM_TAIL after CLR_RX_OAM_LATCH was asserted.

OAM Queue Format

Offset	Name	Description
0	OAM_HEAD	Head pointer
1	OAM_TAIL	Tail pointer

OAM_HEAD Word Format

Field(Bits)	Description
OAM_HEAD (7:0)	The microprocessor should increment to the next cell location when it reads a cell. Initialize to 0.

OAM_TAIL Word Format

Field(Bits)	Description
OAM_TAIL (7:0)	Incremented by the RALP after it writes a cell to the OAM cell queue. Initialize to 0.

7.2.2 R_OAM_CELL_CNT

Organization: 1 word

Base address within A1SP: 8002H

Index: 1H

Type: Read/Write

Function: 16-bit rollover counter that counts the number of OAM cells received. The software must initialize this counter to 0 during reset.

R_OAM_CELL_CNT Word Format

Field(Bits)	Description
R_OAM_CELL_CNT (15:0)	16-bit rollover counter that counts the number of OAM cells received. The software must initialize this counter to 0 during reset. After initialization, do not write to this word.

7.2.3 R_DROP_OAM_CELL

Organization: 1 word

Base address within A1SP: 8003H

Index: 1H

Type: Read/Write

Function: 16-bit rollover counter that counts the number of dropped OAM cells. The software should initialize this counter to 0 during reset.

R_DROP_OAM_CELL Word Format

Field(Bits)	Description
R_DROP_OAM_CELL (15:0)	16-bit rollover counter that counts the number of OAM cells dropped. OAM cells are dropped when more than 255 are present in the receive queue. The software must initialize this counter to 0 during reset. After initialization, do not write to this word.

7.2.4 R_SRTS_CONFIG

Organization: 2 bytes x 8 lines

Base address within A1SP: 8038H

Index: 1H

Type: Read/Write

Function: This table stores the CDVT for the SRTS channel, expressed in the number of queued SRTS nibbles.

Initialization: Initialize to the number of SRTS nibbles equivalent to the CDVT for the data by rounding up. Each frame of CDVT for unstructured applications represent 256 bits. Each SRTS nibble represents 3008 bits, which is the number of data bits in eight cells. Therefore, the number of SRTS nibbles that corresponds to the CDVT can be determined by dividing the CDVT number in frames by 3008 / 256, or 11.75, and rounding up to the next higher integer.

Format: One byte per line. Refer to the following table.

R_SRTS_CONFIG Format

Offset	Name	Description
0_h	R_SRTS_CDVT_0	Receive SRTS CDVT for line 0
1_h	R_SRTS_CDVT_1	Receive SRTS CDVT for line 1
2_h	R_SRTS_CDVT_2	Receive SRTS CDVT for line 2
3_h	R_SRTS_CDVT_3	Receive SRTS CDVT for line 3
4_h	R_SRTS_CDVT_4	Receive SRTS CDVT for line 4
5_h	R_SRTS_CDVT_5	Receive SRTS CDVT for line 5
6_h	R_SRTS_CDVT_6	Receive SRTS CDVT for line 6
7_h	R_SRTS_CDVT_7	Receive SRTS CDVT for line 7

R_SRTS_CDVT_n Word Format

Field(Bits)	Description
Not used (15:5)	Write with 0 to maintain compatibility with future software versions.
R_SRTS_CDVT (4:0)	Receive SRTS CDVT

7.2.5 R_CRC_SYNDROME

Organization: 128 words

Base address within A1SP: 8080H

Index: 1H

Type: Read/Write

Function: This table identifies which bit of the SN/SNP byte has been corrupted, if any. Load after each power cycle. Used internally to perform CRC correction.

R_CRC_SYNDROME Word Format

Field(Bits)	Description
Not used (15:5)	Write with 0 to maintain compatibility with future software versions.
RX_CRC_SYNDROME (4:0)	Mask of bits to change.

Figure 13 – R_CRC_SYNDROME Mask Bit Table Legend

LEGEND

00 No errors
01 Correct bit 0
02 Correct bit 1
04 Correct bit 2
08 Correct bit 3
10 SNP error (no need to correct SN field)

Table 4 – R_CRC_SYNDROME Mask Bit Table

Sequence Number	Offset	Data (Hex)		Sequence Number	Offset	Data (Hex)
0	00	00		4	40	08
0	01	10		4	41	10
0	02	10		4	42	04
0	03	01		4	43	02
0	04	10		4	44	10
0	05	08		4	45	00
0	06	02		4	46	01
0	07	04		4	47	10
0	08	01		4	48	02
0	09	10		4	49	04
0	0A	10		4	4A	10
0	0B	00		4	4B	08
0	0C	04		4	4C	10

Sequence Number	Offset	Data (Hex)		Sequence Number	Offset	Data (Hex)
0	0D	02		4	4D	01
0	0E	08		4	4E	00
0	0F	10		4	4F	10
1	10	02		5	50	01
1	11	04		5	51	10
1	12	10		5	52	10
1	13	08		5	53	00
1	14	10		5	54	04
1	15	01		5	55	02
1	16	00		5	56	08
1	17	10		5	57	10
1	18	08		5	58	00
1	19	10		5	59	10
1	1A	04		5	5A	10
1	1B	02		5	5B	01
1	1C	10		5	5C	10
1	1D	00		5	5D	08
1	1E	01		5	5E	02
1	1F	10		5	5F	04
2	20	04		6	60	10
2	21	02		6	61	01
2	22	08		6	62	00
2	23	10		6	63	10
2	24	01		6	64	02
2	25	10		6	65	04
2	26	10		6	66	10
2	27	00		6	67	08
2	28	10		6	68	10

Sequence Number	Offset	Data (Hex)		Sequence Number	Offset	Data (Hex)
2	29	08		6	69	00
2	2A	02		6	6A	01
2	2B	04		6	6B	10
2	2C	00		6	6C	08
2	2D	10		6	6D	10
2	2E	10		6	6E	04
2	2F	01		6	6F	02
3	30	10		7	70	10
3	31	00		7	71	08
3	32	01		7	72	02
3	33	10		7	73	04
3	34	08		7	74	00
3	35	10		7	75	10
3	36	04		7	76	10
3	37	02		7	77	01
3	38	10		7	78	04
3	39	01		7	79	02
3	3A	0		7	7A	08
3	3B	10		7	7B	10
3	3C	02		7	7C	01
3	3D	04		7	7D	10
3	3E	10		7	7E	10
3	3F	08		7	7F	00

7.2.6 R_CH_TO_QUEUE_TBL

Organization: 128 words (8 lines x 32 DS0s)

Base address within A1SP: 8200H

Index: 1H

Type: Read/Write

Hardware Reset Value: 8080H

Function: This table associates the DS0 with the queue. It allows the transmit line interface to determine the status of the receive queue supplying bytes for the DS0s being processed. This table is located inside the chip and all time slots are initialized to play out conditioned data. The AAL1gator processes two bytes at a time so the values in the following table are in pairs. For unstructured, low speed lines, set all of the queue values to the receive queue number mod 32. In UDF-HS mode, this table is not used. When this queue is in underrun, the AAL1gator reads data for the line from the first word of the R_COND_DATA_0 table.

Format: Refer to the following table.

R_CH_TO_QUEUE_TBL Format

Offset	Name	Description
N	R_CH_TO_QUEUE	Queue numbers and condition bits associated with this pair of channels where: Line = N / 16. Low channel = (N mod 16) x 2. High channel = (N mod 16) x 2 + 1.

R_CH_TO_QUEUE Word Format

Field(Bits)	Description
RX_COND_H (15:14)	Determines the type of data to be played out: Options "00", "01", and "11" are executed only when the queue is in an underrun or resume state. **00**b When the queue is in underrun, freeze signaling and read the data for this channel from the R_COND_DATA table. **01**b When the queue is in underrun, freeze signaling and play out pseudorandom data, which is inserted data from R_COND_DATA, with the MSB controlled by the pseudorandom number algorithm $x^{18} + x^7 + 1$ (not valid for UDF-HS). **10**b Read signaling for this channel from the R_COND_SIG table and the data for this channel from the R_COND_DATA table. **11**b When the queue is in underrun freeze signaling and play out the contents of the buffer.
RX_SIG_COND_H (13)	Overrides the normal signaling with Conditioned signaling **0**b Read signaling as indicated by RX_COND_H **1**b Always read signaling for this channel from the R_COND_SIG table
QUEUE_H (12:8)	Five LSBs of the queue index associated with this DS0. The three MSBs are implicitly those of the line number. Offset = (channel - 1) / 2 + line x 16. For unstructured lines, set to the receive queue number mod 32.

Field(Bits)	Description
RX_COND_L (7:6)	Determines the type of data to be played out: Options "00", "01", and "11" are executed only when the queue is in an underrun or resume state. **00**b When the queue is in underrun, freeze signaling and read the data for this channel from the R_COND_DATA table. **01**b When the queue is in underrun, freeze signaling and play out pseudorandom data, which is inserted data from R_COND_DATA, with the MSB controlled by the pseudorandom number algorithm $x^{18} + x^7 + 1$ (not valid for UDF-HS). **10**b Read signaling for this channel from the R_COND_SIG table and the data for this channel from the R_COND_DATA table. **11**b When the queue is in underrun, freeze signaling and play out the contents of the buffer.
RX_SIG_COND_L (5)	Overrides the normal signaling with Conditioned signaling **0**b Read signaling as indicated by RX_COND_L **1**b Always read signaling for this channel from the R_COND_SIG table
QUEUE_L (4:0)	Five LSBs of the queue index associated with this DS0. The three MSBs are implicitly those of the line number. Offset = channel / 2 + line x 16.

7.2.7 R_COND_SIG

Organization: 16 words x 8

Base address within A1SP: 8400H

Index: 10H

Type: Read/Write

Function: This table stores the signaling to be used when RX_SIG_COND_H or RX_SIG_COND_L equals '1' in the R_CH_TO_QUEUE_TBL.

Initialization: Initialize to the conditioned signaling value for the channel. This value typically depends on the type of channel unit that is connected. For example, an FXO channel unit needs a different conditioning value than an FXS channel unit.

Format: One nibble per byte, two bytes per word, 16 words per line. Refer to the following table.

R_COND_SIG Format

Offset	Name	Description
00000H	R_COND_SIG_0	Receive conditioned signaling for line 0.
00010H	R_COND_SIG_1	Receive conditioned signaling for line 1.
00020H	R_COND_SIG_2	Receive conditioned signaling for line 2.
00030H	R_COND_SIG_3	Receive conditioned signaling for line 3.
00040H	R_COND_SIG_4	Receive conditioned signaling for line 4.
00050H	R_COND_SIG_5	Receive conditioned signaling for line 5.
00060H	R_COND_SIG_6	Receive conditioned signaling for line 6.
00070H	R_COND_SIG_7	Receive conditioned signaling for line 7.

R_COND_SIG_n Word Format

Field (Bits)	Description
Not used (15:12)	Write with a '0' to maintain future software compatibility.
R_COND_A_H (11)	Receive conditioned A signaling bit for: Offset = (channel - 1) / 2 + line x 16.
R_COND_B_H (10)	Receive conditioned B signaling bit for: Offset = (channel - 1) / 2 + line x 16.
R_COND_C_H (9)	Receive conditioned C signaling bit or A bit if T1 SF for: Offset = (channel - 1) / 2 + line x 16.
R_COND_D_H (8)	Receive conditioned D signaling bit or B bit if T1 SF for: Offset = (channel - 1) / 2 + line x 16.

Field (Bits)	Description
Not used (7:4)	Write with a 0 to maintain future software compatibility.
R_COND_A_L (3)	Receive conditioned A signaling bit for: Offset = (channel / 2) + line x 16.
R_COND_B_L (2)	Receive conditioned B signaling bit for: Offset = (channel / 2) + line x 16.
R_COND_C_L (1)	Receive conditioned C signaling bit or A bit if T1 SF for: Offset = (channel / 2) + line x 16.
R_COND_D_L (0)	Receive conditioned D signaling bit or B bit if T1 SF for: Offset = (channel / 2) + line x 16.

7.2.8 R_COND_DATA

Organization: 16 words x 8

Base address within A1SP: 8480H

Index: 10H

Type: Read/Write

Function: This table stores the data to be used when RX_COND in the R_CH_TO_QUEUE_TBL equals 00b, 01b, or 10b.

Initialization: Initialize to the conditioned data appropriate for the channel. This typically depends on the type of channel connected to the device. For example, data usually needs an FFH value and voice needs a small PCM value.

Format: Two bytes per word, 16 words per line. Refer to the following table.

R_COND_DATA Format

Offset	Name	Description
00000H	R_COND_DATA_0	Receive conditioned data for line 0.
00010H	R_COND_DATA_1	Receive conditioned data for line 1.

Offset	Name	Description
00020H	R_COND_DATA_2	Receive conditioned data for line 2.
00030H	R_COND_DATA_3	Receive conditioned data for line 3.
00040H	R_COND_DATA_4	Receive conditioned data for line 4.
00050H	R_COND_DATA_5	Receive conditioned data for line 5.
00060H	R_COND_DATA_6	Receive conditioned data for line 6.
00070H	R_COND_DATA_7	Receive conditioned data for line 7.

R_COND_DATA_n Word Format

Field (Bits)	Description
R_COND_DATA_H (15:8)	Receive conditioned data for: Offset = (channel - 1) / 2 + line x 16.
R_COND_DATA_L (7:0)	Receive conditioned data for: Offset = channel / 2 + line x 16.

7.2.9 RESERVED (Receive SRTS Queue)

This structure is reserved. Software modifications to this structure after setup will cause incorrect operation.

Organization: 64 words x 8 lines. Each line is allocated a separate 64-entry queue to store the SRTS receive nibbles.

Base address within A1SP: 8800H

Index: 100H

Type: Read/Write

Function: The receive signaling queue stores the SRTS bits received from the UTOPIA interface.

Initialization: It is not necessary to initialize this structure.

Format: One SRTS nibble per word.

R_SRTS_QUEUE_n Word Format

Field (Bits)	Description
Not used (15:8)	Write with a 0 to maintain future software compatibility.
R_SRTS_VAL (7:4)	Indicates if each SRTS bit contains valid data. When an error occurs which causes a bit to be lost the corresponding bit will be written with a '0'. Each time a new entry is written, the remaining bits which haven't been received yet will also be written with a '0'. So normally this field will be written with a "1000" for the first bit then "1100" for the second bit, then "1110" for the third bit and finally "1111" for the last bit.
R_SRTS (3:0)	Receive SRTS data for line = offset / 64.

7.2.10 RESERVED (Receive Signaling Buffer)

This structure is reserved. Software modifications to this structure after setup will cause incorrect operation.

Organization: 32 x 32 DS0s x 8 lines. Each line is allocated a separate 32 x 32 byte memory. For E1, this allows storage of signaling information for 32 multiframes, unless E1_WITH_T1_SIG is set. T1 applications use only the first 24 bytes of every 32 to store signaling data. In addition, since the transmit data buffer is only 16 multiframes in size, this structure also needs to store only 16 multiframes. Successive multiframes are stored in every other 32-byte buffer. When signaling is frozen due to an underrun, the value in multiframe 0 is used.

Base address within A1SP: 9000H

Index (line): 200H

Type: Read/Write

Function: The receive signaling queue stores the signaling that is received from the UTOPIA interface.

Initialization: The signaling buffer should be initialized to "0". Also, if R_CHAN_NO_SIG is set for some queues and a specific signaling value is desired to be driven for these queues, then the DS0s in those queues must be initialized to the desired value for all multiframes.

Format: Two signaling nibbles per word.

R_SIG_BUFFER_n Word Format

Field (Bits)	Description
Not used (15:13)	Write with a 0 to maintain future software compatibility.
R_SIG_INVALID (12)	Indicates that the stored signaling is invalid. If signaling is not valid due to lost cells, signaling will freeze.
R_SIG_H (11:8)	Receive signaling data for: Channel = (offset mod 16) x 2 + 1. Multiframe = (offset mod 512) / 16. Line = offset / 512. Offset = line x 512 + multiframe x 16 + (channel - 1) / 2.
Not used (7:5)	Write with a 0 to maintain future software compatibility.
R_SIG_INVALID (4)	Indicates that the stored signaling is invalid. If signaling is not valid due to lost cells, signaling will freeze.
R_SIG_L (3:0)	Receive signaling data for: Channel = (offset mod 16) x 2. Multiframe = (offset mod 512) / 16. Line = offset / 512. Offset = line x 512 + multiframe x 16 + channel / 2.

7.2.11 R_QUEUE_TBL

Organization: 256 x 32 words

Base address within A1SP: A000H

Index: 20H

Type: Read/Write

Function: Receive Queue Table contains all the structures and pointers specific to a queue. The RALP and RFTC blocks both use the R_QUEUE_TBL. Some of the words are read by both the blocks but written by only one of the blocks.

Format: Each queue is allocated 32 consecutive words. Each word is 16-bits wide. The organization of the words is as follows.

Table 5 – R_QUEUE_TBL Format

Offset	Name	Description
0H	R_STATE_0	Cell receiver state 0.
1H	R_MP_CONFIG	Bytes per cell and CDVT constant.
2H	R_STATE_1	Cell receiver state 1.
3H	R_LINE_STATE	Line state.
4H	R_MAX_BUF	Receive maximum buffer size.
5H	R_SEQUENCE_ERR	16-bit rollover count of SN errors.
6H	R_INCORRECT_SNP	16-bit rollover count of cells with incorrect SNP.
7H	R_CELL_CNT	16-bit rollover count of played out cells.
8H	R_ERROR_STKY	Receive sticky bits.
9H	R_TOT_SIZE	Total bytes in structure.
AH	R_DATA_LAST	Number of signaling bytes in structure.
BH	R_TOT_LEFT	Number of bytes remaining in the structure. Initialize to 0 each time this queue is initialized.
CH	Not used	Initialize to 0 each time this queue is initialized.
DH	R_SN_CONFIG	Configures sequence number processing algorithm.
EH	R_CHAN_ALLOC (15:0)	A bit table with a bit set per DS0 allocated to this queue for DS0s 15 to 0 on the line defined by queue / 32.
FH	R_CHAN_ALLOC (31:16)	A bit table with a bit set per DS0 allocated to this queue for DS0s 31 to 16 on the line defined by queue / 32.
10H	Reserved (CHNLEFTL)	Initialize to 0 each time this queue is initialized.
11H	Reserved (CHNLEFTH)	Initialize to 0 each time this queue is initialized.
12H	R_DROPPED_CELLS	16-bit rollover count of cells that were received but dropped. Initialize to 0.
13H	R_UNDERRUNS	16-bit rollover count of the occurrences of underrun on this queue. Initialize to 0.

Offset	Name	Description
14H	R_LOST_CELLS	16-bit rollover count of the number of lost cells for this queue. Initialize to 0.
15H	R_OVERRUNS	16-bit rollover count of the occurrences of overrun on this queue. Initialize to 0.
16H	R_PTR_REFRAMES	16-bit rollover count of the occurrences of pointer reframes. Initialize to 0.
17H	R_PTR_PAR_ERR	16-bit rollover count of the occurrences of pointer parity errors. Initialize to 0.
18H	R_MISINSERTED	16-bit rollover count of the occurrences of misinserted cells. Initialize to 0.
19H	R_ROBUST_SN	Write pointer for robust SN processing
1AH	Reserved (CHNACTL)	Initialize to 0 each time this queue is initialized.
1BH	Reserved (CHNACTH)	Initialize to 0 each time this queue is initialized.
1CH	R_RD_PTR_LAST	Read pointer for bit integrity through underrun
1DH-1FH	Not used	Initialize to 0 each time this queue is initialized.
All of these locations must be initialized whenever the queue is initialized.		

R_STATE_0 Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_DBCES_BM_IN_NXT (15)	Indicates that a Bit Mask will be present in the next structure. Used when a ptr is found that locates the structure in the next cell.
R_STRUCT_FOUND (14)	Indicates that the receiver structure was found. Initialize to 0.
Reservd(OLDUNDRN_N) (13)	Initialize to 0 to maintain future software compatibility.
Reservd(UNDRN_2AGO) (12)	Initialize to 0 to maintain future software compatibility.
Reserved(ACTSN) (11:9)	Initialize to 0 to maintain future software compatibility.
SN_STATE (8:6)	Specifies the state of the SN state machine. Initialize to 0.
2ND_LAST_SN (5:3)	Specifies the SN that was received two cells ago. Initialize to 0.
LAST_SN (2:0)	Specifies the last SN that was received. Initialize to 0.

R_MP_CONFIG Word Format

This word is maintained by the microprocessor.

Field (Bits)	Description
R_CHK_PARITY (15)	If set, check the parity on the incoming structure pointer.

Field (Bits)	Description
R_BYTES_CELL (14:9)	A 6-bit integer specifying how many bytes per cell are required if no structure pointers are used. For UDF-HS mode, this must be set to 47. In other modes, set this to the partially filled length. If cells are not partially filled, set this to 47.
R_AAL0_MODE (8)	If set, treats this queue as an AAL0 queue and will write all 48 bytes of payload into the allocated time slots.
R_CDVT (7:0)	Receive Cell Delay Variation Tolerance (R_CDVT) is a constant and is programmed by the microprocessor during initialization. It is used by the RFTC after the receipt of the first cell after an underrun. In T1 SDF-MF, E1_WITH_T1_SIG, or SDF-FR mode, the R_CDVT is expressed as the number of multiframes in bits 7:5 and the number of frames in bits 4:0. In E1 and all other T1 modes, R_CDVT is the number of frames. In unstructured applications, the number of frames refers to the number of 256-bit increments. For T1 unstructured modes, this is equivalent to the number of 165.8 ms periods. For Robust SN Processing, this field represents the CDVT desired plus the number of frames stored in the cell that is conditionally stored

R_STATE_1 Word Format

This word is read-only and is maintained by the RALP. This register is located inside the chip and is reset to “0000”.

Field (Bits)	Description
Reserved (FRC_UNDRN) (15)	Initialize to 0 to maintain future software compatibility.
Reserved (SNCRCST) (14)	Initialize to 0 to maintain future software compatibility.
Reserved (PTRMMST) (13)	Initialize to 0 to maintain future software compatibility.

Field (Bits)	Description
Reserved (FNDPTR) (12)	Initialize to 0 to maintain future software compatibility.
Reserved (FNDFRSTPTR) (11)	Initialize to 0 to maintain future software compatibility.
Reserved (DBCES_EN) (10)	Initialize to 0 to maintain future software compatibility.
Not used (9)	Driven with a 0. Mask on reads to maintain future software compatibility.
R_WRITE_PTR (8:0)	Pointer to the frame to which the cell receiver is writing the last accepted cell.

R_LINE_STATE Word Format

This word is read-only after initialization and is maintained by the RALP and RFTC. This register is located inside the chip and is reset to 9000H.

Field (Bits)	Description
R_UNDERRUN (15)	Indicates that this queue is currently in underrun. Initialize to 1.
R_RESUME (14)	Indicates that this queue is currently in resume state. Initialize to 0.
R_SIG_RESUME (13)	Indicates that this queue is currently in signal resume state. Initialize to 0.
R_LONG_UNDERRUN (12)	Indicates that the rd_ptr has wrapped while the queue was in underrun
Reserved (11:9)	Initialize to 0 to maintain future software compatibility.
R_END_UNDERRUN_PTR (8:0)	Location read pointer needs to reach after an underrun to begin playing out new data. Initialize to 0 to maintain future software compatibility.

R_MAX_BUF Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
R_CHAN_UNSTRUCT (15)	Set to 1 only when receiving cells with a single DS0 without a pointer in the SDF-FR mode. This bit is valid only in SDF-FR mode. To conform to the CES standard V 2.0 when using a single DS0 in SDF-FR mode, no pointer should be used.
R_CHAN_NO_SIG (14)	Set to 1 to receive cells without signaling when the line is in SDF-MF mode. This is the same as using this queue in SDF-FR mode, which means that the structure forms on frame boundaries instead of multiframe boundaries. The R_SIG_BUFFER will never be updated for this queue. However, the TL_SIG output will drive the value that was initialized into this timeslot in T_SIG_BUFFER.
R_CHAN_DISABLE (13)	Set to 1 to drop all cells for this queue. Set to 0 for normal operation. Cells dropped because of this bit are recorded in the ALLOC_TBL_BLANK sticky bit.
BITI_UNDERRUN (12)	Set to 1 to maintain bit integrity through underrun. Set to 0 for normal operation.
DBCES_BIT_MASK (11:10)	Size in bytes-1 of the DBCES bit mask field.
DBCES_EN (9)	Set to 1 to enable DBCES. This bit is only valid in SDF_FR or SDF MF mode.

Field (Bits)	Description
R_MAX_BUF (8:0)	Receiver maximum buffer size. The R_MAX_BUF is coded as the number of frames. In all structured modes, this is the number of frames. In all unstructured modes, this is the number of 256-bit increments. If the amount of data in the receive buffer exceeds R_MAX_BUF, no more data will be written, an overflow will be reported, and the queue will be forced into underrun. The maximum value of R_MAX_BUF is 1FEH for most cases. For T1 structured mode or E1 with T1 signaling, the maximum value is 17EH because not all frames are used.

R_SEQUENCE ERROR Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_SEQUENCE_ERR (15:0)	16-bit rollover count of SN errors. This counter counts transitions from the SYNC state to the OUT_OF_SEQUENCE state. This is the atmfCESAal1SeqErrors count from the CES specification. Note that if SN processing is disabled, this counter will count all out-of-sequence cells. Initialize to 0. Once initialized, do not write to this word.

R_INCORRECT_SNP Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_INCORRECT_SNP (15:0)	16-bit rollover count of cells with SNP errors. This is the atmfCESHdrErrors counter from the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_CELL_CNT Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_CELL_CNT (15:0)	16-bit rollover count of received cells. This is the atmfCESReassCells counter from the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_ERROR_STKY Word Format

Receive sticky bits should be used for statistics gathering purposes only as there is no means of clearing them without the possibility of missing an occurrence. Initialize to 0.

Field (Bits)	Description
TRANSFER (15)	This bit is read then written with the same value each time the AAL1gator receives a cell. This feature allows the processor to determine if the AAL1gator was in the middle of a read then write cycle when the processor cleared the other sticky bits. To accomplish this each time the processor wants to clear sticky bits, it should complement this bit. Then, if an additional read of this bit showed it to be the wrong value, then the AAL1gator has had its sticky word update interrupted.
CELL_RECEIVED (14)	A cell was received.
DBCES_BIT_MASK_ERR (13)	There was a parity error in the DBCES Bit Mask.

Field (Bits)	Description
PTR_RULE_ERROR (12)	There was a violation of a pointer generation rule. A sequence begins with a cell with SN=0 and ends with a cell with SN=7. This condition will be set if no pointer was received, more than 1 pointer was received or an 'out of bounds' pointer was received in the sequence. This condition will only be checked in modes where a pointer is expected and no sequence number error or underrun occurred.
ALLOC_TBL_BLANK (11)	A cell was dropped because of a blank allocation table or because R_CHAN_DISABLE (refer to "R_MAX_BUF Word Format" on page144) was asserted.
POINTER_SEARCH (10)	A cell was dropped because a valid pointer has not yet been found.
FORCED_UNDERRUN (9)	A cell was dropped because a forced underrun condition exists. A forced underrun condition can be caused by overruns and pointer mismatches.
SN_CELL_DROP (8)	A cell was dropped in accordance with the SN Algorithm (as specified in ITU-T Recommendation I.363.1). If Fast SN processing is used this bit will always be set for the first cell if NO_DROP_IN_STAT = 0.
POINTER_RECEIVED (7)	A pointer was received.
PTR_PARITY_ERR (6)	A cell was received with a pointer parity error.
SRTS_RESUME (5)	An SRTS resume has occurred. A valid SRTS value was received and stored in the SRTS FIFO.
SRTS_UNDERRUN (4)	A cell was received while the SRTS queue was in underrun.
RESUME (3)	A resume has occurred: a valid cell was received and stored into the buffer. This cell will be played out after 1 CDVT.
PTR_MISMATCH (2)	A cell was dropped because of a pointer mismatch. This event causes a forced underrun condition.

Field (Bits)	Description
OVERRUN (1)	A cell was dropped due to overrun. The receive buffer exceeded the maximum allowed depth. This event causes a forced underrun condition.
UNDERRUN (0)	A cell was received while the queue was in underrun.

R_TOT_SIZE Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
FRAMES_PER_CELL (15:10)	Average number of frames contained within a single cell. This field is not used in UDF-ML or UDF-HS mode.
R_TOT_SIZE (9:0)	Total bytes minus one in the structure (for example, for an E1 MF VC with two DS0s, R_TOT_SIZE is set to 32). This field is not used in UDF-ML or UDF-HS mode. Three formulas for R_TOT_SIZE are: For T1/E1 SDF-FR: R_TOT_SIZE = no. of DS0s - 1 For T1 SDF-MF: R_TOT_SIZE = $(24 \times \text{no. of DS0s}) + \frac{(\text{no. of DS0s} + 1)}{2} - 1$ For E1 SDF-MF: R_TOT_SIZE = $(16 \times \text{no. of DS0s}) + \frac{(\text{no. of DS0s} + 1)}{2} - 1$

R_DATA_LAST Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
Not used (15:13)	Write with a 0 to maintain future software compatibility.
LAST_CHAN (12:8)	Channel number (0 to 31) of the last DS0 with a bit set in the R_CHAN_ALLOC bit table (refer to "R_CHAN_ALLOC(15:0) Word Format" and "R_CHAN_ALLOC(31:16) Word Format" on page153).
Not used (7:6)	Write with a 0 to maintain future software compatibility.
Reserved (5:4)	Write with a 0 to maintain future software compatibility.
R_DATA_LAST (3:0)	Number of signaling bytes minus one in the structure (for example, for an E1 SDF-MF VC with six DS0s, R_DATA_LAST is set to 2). An E1-SDF-MF VC with seven DS0s is set to 3 as one signaling nibble is unused. Not used in UDF-ML or UDF-HS mode. $$R\_DATA\_LAST = \frac{(no.\ of\ DS0s + 1)}{2} - 1$$

R_TOT_LEFT Word Format

This word is read-only and is maintained by RALP

Field (Bits)	Description
Not used (15:13)	Driven with a 0. Mask on reads to maintain future software compatibility.
R_DBCES_BM_LEFT (12:11)	Total unprocessed bytes remaining in bit mask structure.
R_DBCES_BM_ACT (10)	Activity detected in the Bit Mask. Used to indicated whether any channels in the DBCES structure are active or not.
R_TOT_LEFT (9:0)	Total bytes minus one remaining in the structure. Not used in UDF-ML or UDF-HS mode.

R_SN_CONFIG Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
R_CONDQ_DATA (15:8)	Value of conditioned data inserted into lost cells depending on the value of INSERT_DATA.
ROBUST_SN_EN (7)	Set to 1 to enable the "Robust SN algorithm". Set to a "0' for the "Fast SN Algorithm".
INSERT_DATA (6:5)	Controls the format of the data inserted for lost cells: **00**b Insert AIS **01**b Insert data from R_CONDQ_DATA. **10**b Insert old data from receive buffer. **11**b Insert data from R_CONDQ_DATA with the MSB controlled by the pseudorandom number algorithm $x^{18} + x^{7} + 1$ (not valid for UDF-HS).
DISABLE_SN (4)	If set, sequence number processing is disabled. Statistics will still be kept but no cells will be dropped due to SN errors.
NODROP_IN_START (3)	In the "Fast SN Algorithm" for SN processing, the first cell received will always be dropped because a sequence has not been established yet. This bit disables the automatic dropping of cells while in the START state 0 When SN_STATE equals 000b any received cell will be dropped. 1 When SN_STATE equals 000b any received cell with valid SNP will be accepted.
MAX_INSERT (2:0)	The maximum number of cells that will be inserted when cells are lost. If the number of cells lost exceeds MAX_INSERT, then the queue will be forced into underrun. If this value is set to 000b, it is interpreted the same as 111b, which means that up to seven cells will be inserted.

R_CHAN_ALLOC(15:0) Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
R_CHAN_ALLOC (15:0)	A bit table with a bit set per DS0 allocated to this queue for DS0s 15 to 0 on the line defined by queue /32. In UDF-ML and UDF-HS modes, initialize to FFFFH. (DS0 15 is in bit 15).

R_CHAN_ALLOC(31:16) Word Format

This word is maintained by the microprocessor

Field (Bits)	Description
R_CHAN_ALLOC (31:16)	A bit table with a bit set per DS0 allocated to this queue for DS0s 31 to 16 on the line defined by queue /32. In UDF-ML and UDF-HS modes, initialize to FFFFH. (DS0 31 is in bit 15).

R_DROPPED_CELLS Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_DROPPED_CELLS (15:0)	16-bit rollover count of dropped non-OAM cells. Initialize to 0. Once initialized, do not write to this word. Cells may be dropped due to: • Pointer mismatch. • Overrun. • Blank allocation table • SN processing. • Structured cell received while in underrun but structure start has not been found yet.

R_UNDERRUNS Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_UNDERRUNS (15:0)	16-bit rollover count of the occurrences of an underrun on this queue. This is the atmfCESBufUnderflows counter. Initialize to 0. Once initialized, do not write to this word. Underruns are counted by the RALP, which does not know an underrun occurred until a cell is received while in underrun. To ensure the underrun count is correct, the counter is not incremented until the queue exits the underrun state and enters the resume state underrun condition. To determine if the queue is in underrun, check the level of the R_UNDERRUN bit in R_LINE_STATE register. If this bit is set, then increment the underrun count by one to get the current count.

R_LOST_CELLS Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_LOST_CELLS (15:0)	16-bit rollover count of cells that were detected as lost. This is the atmfCESLostCells counter in the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_OVERRUNS Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_OVERRUNS (15:0)	16-bit rollover count of the occurrences of an overrun on this queue. This is the atmfCESBufOverflows counter in the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_POINTER_REFRAMES Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_POINTER_REFRAME (15:0)	16-bit rollover count of the occurrences of pointer reframes on this queue. This is the atmfCESPointerReframes counter in the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_PTR_PAR_ERR Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_PTR_PAR_ERR (15:0)	16-bit rollover count of the occurrences of pointer parity errors on this queue. This is the atmfCESPointerParityErrors counter in the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_MISINSERTED Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
R_MISINSERTED (15:0)	16-bit rollover count of the occurrences of misinserted cells on this queue. This is the atmfCESMisinsertedCells counter in the CES specification. Initialize to 0. Once initialized, do not write to this word.

R_ROBUST_SN Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
Reserved (15)	Used to indicate when the first cell is received on a RSN connection.

Field (Bits)	Description
R_RSN_RESUME (14)	Indication that the stored cell is the first cell after an underrun.
R_RSN_CHAN_PTR (13:9)	Pointer to the channel number in which to start if dropping a previously stored cell.
R_RSN_WRT_PTR (8:0)	Pointer to the frame to which the cell receiver is writing for Robust SN processing.

R_RD_PTR_LAST Word Format

This word is read-only and is maintained by the RALP

Field (Bits)	Description
Not used (15:9)	Driven with a 0. Mask on reads to maintain future software compatibility.
R_RD_PTR_LAST (8:0)	Pointer to the frame that was last read when the last cell was received. This is used to determine whether more than 6 cells have been lost when a SN error occurs to help maintain bit integrity through underrun.

7.2.12 R_OAM_QUEUE

Organization: 256 cells x 64 bytes

Base address within A1SP: E000H

Index: 20H

Type: Read/Write

Function: The receive signaling queue stores the signaling received from the UTOPIA interface.

Initialization: It is not necessary to initialize this structure.

Format: Two data bytes per word

R_OAM_QUEUE Format

Offset	Name	Description
00000H	R_OAM_CELL_0	Receive OAM cell 0
00010H	R_OAM_CELL_1	Receive OAM cell 1
.	.	.
.	.	.
.	.	.
01FFFH	R_OAM_CELL_255	Receive OAM cell 255

R_OAM_CELL_n Format

Offset	Bits (15:8)	Bits (7:0)
Word 0	Header 1	Header 2
Word 1	Header 3	Header 4
Word 2	Header 4 (HEC)	Blank
Word 3	Payload 1	Payload 2
.	.	.
.	.	.
.	.	.
Word 26	Payload 47	Payload 48
Word 27	CRC_10_PASS	

CRC_10_PASS Word Format

Field (Bits)	Description
CRC_10_PASS (15)	The CRC_10_PASS bit is set if the cell passes the CRC-10 check.
Not used (14:0)	Write with a 0 to maintain future software compatibility.

7.2.13 RESERVED (Receive Data Buffer)

This structure is reserved and must be initialized to 0 at initial setup. If RX_COND for some channels is set to "11" (insert old data during underrun),

then those channels may need to be initialized to some other value if "0" data is unacceptable, since all the queues will reset to the underrun state. Software modifications to this location after setup will cause incorrect operation.

Organization: Each line has a separate receive data buffer consisting of 512 frame buffers. Each frame buffer can store 32 bytes. For E1 structured data applications, this allows storage of 512frames or 32 multiframes of data. Structured T1 applications use only the first 24 bytes of each frame buffer for data storage. Also, only the first 24 frame buffers of every 32 are used to store T1 structured data frames. This provides 384 frames of storage, or 16 multiframes. Unstructured applications store 256 bits of data in every frame buffer. For E1 with T1 signaling, use T1 structure but with 32 channels.

Base address within A1SP: 10000H

Index (line): 2000H

Type: Read/Write

Function: The data buffers store receive data information. The data is stored in the buffers in the order that they will be played out to the lines.

Initialization: Initial to 0 at startup. If RX_COND for some channels is set to "11" (insert old data during underrun), then those channels may need to be initialized to some other value if "0" data is unacceptable.

Format: Two data bytes per word.

R_DATA_BUFFER_n Word Format

Field (Bits)	Description
R_DATA_H (15:8)	Receive data for: Channel = (offset mod 16) x 2 + 1. E1 frame = (offset mod 256) / 16. T1 frame = (offset mod 512) / 16. E1 multiframe = (offset mod 8192) / 256. T1 multiframe = (offset mod 8192) / 512. Line = offset / 8192. E1 offset = line x 8192 + multiframe(E1) x 256 + frame(E1) x 16 + (chan-1) / 2. T1 offset = line x 8192 + multiframe(T1) x 512 + frame(T1) x 16 + (chan-1) / 2.
R_DATA_L (7:0)	Receive data for: Channel = (offset mod 16) x 2. E1 frame = (offset mod 256) / 16. T1 frame = (offset mod 512) / 16. E1 multiframe = (offset mod 8192) / 256. T1 multiframe = (offset mod 8192) / 512. Line = offset / 8192. E1 offset = line x 8192 + multiframe(E1) x 256 + frame(E1) x 16 + channel / 2. T1 offset = line x 8192 + multiframe(T1) x 512 + frame(T1) x 16 + channel / 2.

8 CONFIGURING THE LINE INTERFACE

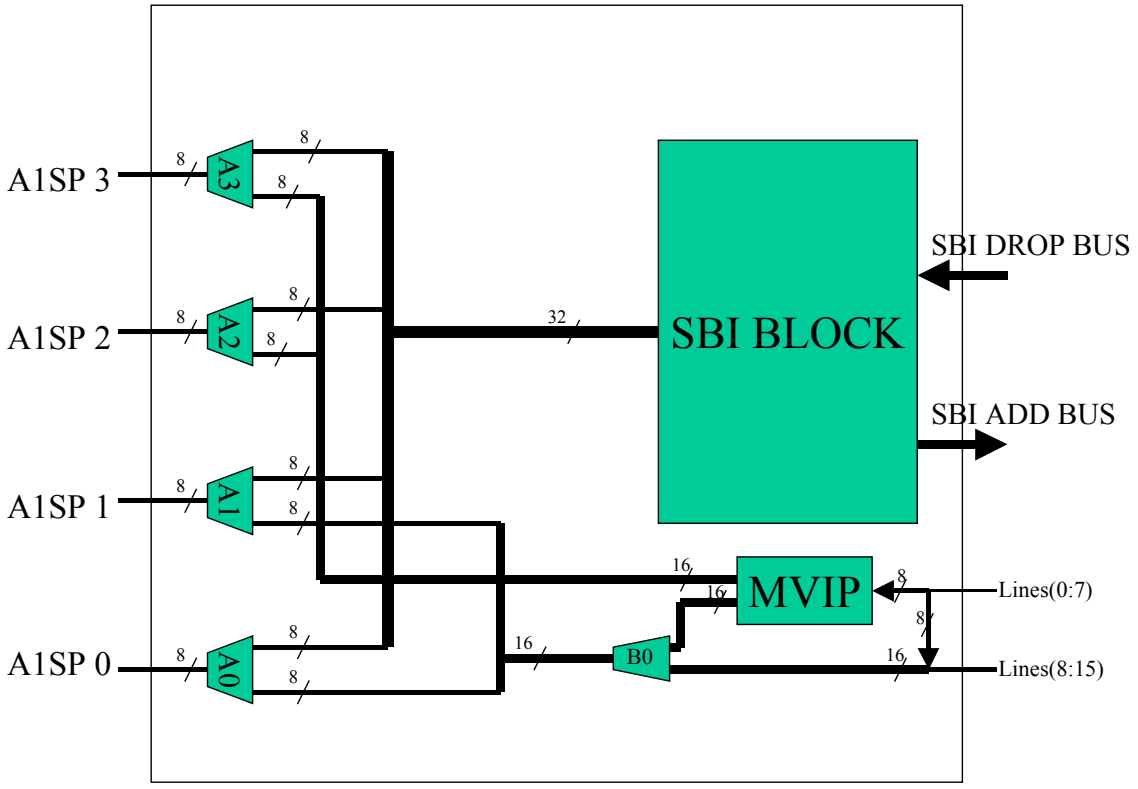
The line interface block is responsible for passing the TDM data between the A1SP blocks and converting it to the appropriate protocol used on the external lines. The mode of the module is determined by the value of the LINE_MODE pins during hardware reset. This mode can be read by software in the LINE_MODE bits of the **LINE_MODE_REG** (0x80210). The following encoding is used:

LINE_MODE [1:0]	Line Interface Mode	AAL1gator-32	AAL1gator-8	AAL1gator-4
"00"	Direct Low Speed	Supports 16 T1/E1 links.	Supports 8 T1/E1 links.	Supports 4 T1/E1 links.
"01"	SBI	Supports 32 T1/E1 links or 2 T3 links.	Not Supported	Not Supported
"10"	H-MVIP	Supports 32 T1/E1 links.	Supports 8 T1/E1 links.	Supports 4 T1/E1 links.
"11"	High Speed	Supports 2 T3/E3/STS-1/STM-0 links.	Supports 1 T3/E3/STS-1/STM-0 link.	Supports 1 T3/E3/STS-1/STM-0 link.

Note: SBI mode is not supported in the AAL1gator-8 and the AAL1gator-4.

Figure 14 shows the block diagram for the Line Interface Block of the AAL1gator-32. The block consists of the SBI Block, an H-MVIP Block and mux/demux logic.

Figure 14 – Line Interface Block Architecture



The A0 to A3 mux/demux logic selects between SBI links and non-SBI links. The B0 mux/demux logic selects between the H-MVIP data or external direct links. The upper 16 lines can only be used in H-MVIP, or SBI mode. In high speed mode, external lines 0, and 2 are used, but they are mapped to internal links 0 and 16.

8.1 Conventions

The following conventions are used in this section:

The 32/8/4 lines, which connect the AAL1_LI to the A1SP blocks, are called **local links**. The lines on the external interface are called **external lines**.

The direction from the local links to the external line interface is call the **transmit** direction. The direction from the external line interface to the local links is called the **receive** direction.

The individual data streams within the SBI interface are known as **tributaries**. Once inside the AAL1_LI, these data streams are known as **links**. In SBI mode links and tributary numbering starts at '1'.

With respect to the SBI interface the **Add** direction is the one where data is placed onto the SBI, the **Drop** direction is the one where data is extracted from the SBI. This block is intended to be used as a Link Layer Device on the SBI.

With respect to the internal links, the **lower** group refers to links 0 through 15 and the **upper** group refers to links 16 through 31. Note that when using the SBI bus the link numbering starts at '1'. Therefore when using SBI the **lower** group refers to links 1 through 16 and the **upper** group refers to links 17 through 32.

8.2 Register Summary

Table 6 is a summary of the normal mode registers that configure the AAL1gator Line Interface. Table 7 summarizes the bit fields within these registers. Note that there is an LS_Ln_CFG_REG for each of the 16/8/4 low speed lines.

Table 6 – Internal Line Configuration Registers

Address	Register Description	Register Mnemonic
0x80200 – 0x8020F	Low Speed Line n Configuration Registers	LS_Ln_CFG_REG
0x80210	Line Mode Register	LINE_MODE_REG

Table 7 – Internal Line Configuration Register Bits

Register Mnemonic	Bit	Type	Function	Default
LS_Ln_CFG_REG	Bit 1	R/W	MVIP_EN	0
	Bit 0	R/W	MF_SYNC_MODE	0
LINE_MODE_REG	Bit 1:0	RO	LINE_MODE	00

Table 8 is a summary of the memory mapped configuration structures that configure the AAL1gator Line Interface. There is a LIN_STR_MODE register for each of the (up to) eight lines (n = 0, …, 7) within each A1SP. Writes to these registers will not take effect until the An_CMDREG_ATTN bit is set in the An_CMD_REG for that A1SP. Note that the addresses listed below are the offsets within each A1SP address space.

Table 8 – A1SP and Line Configuration Structures Summary

Address	Name	Org	Size	Description
0001_h	HS_LIN_REG	1 word	2 bytes	The High Speed Line Register provides overall mode information.
0010_h - 0017_h	LIN_STR_MODE_n	1 word	2 bytes	The Line Structure Mode register identifies which data structure type will be supported for each line. This is selectable on a line basis.

The bit fields of the LS_Ln_CFG_REG and LIN_STR_MODE registers and the modes in which these bits are applicable are shown in Table 9. A "√" denotes that the field is selectable for that mode. An "n/a" denotes that the field is ignored for that mode. External memory should be cleared to all zeros at initialization. Please see each mode's section for a description of the bit fields applicable to that mode.

Table 9 – Applicability of Line Configuration Bits

Bit	Field	Direct Low Speed		SBI	H-MVIP	High Speed
		SDF	UDF			
LS_Ln_CFG_REG:						
1	MVIP_EN	√	0	0	0	0
0	MF_SYNC_MODE	√	n/a	n/a	0	n/a
LIN_STR_MODE:						
15	LOW_CDV	n/a	√	√	n/a	n/a
14	REF_VAL_ENABLE	√	n/a	√	√	n/a
13	T1_MODE	√	n/a	√	0	√
12	E1_WITH_T1_SIG	√	n/a	n/a	√	n/a
11	HI_RES_SYNTH	√	√	√	n/a	n/a
10	Reserved	0	0	0	0	0
9	MF_ALIGN_EN	√	n/a	√	√	n/a
8	Not Used	0	0	0	0	0

Bit	Field	Direct Low Speed		SBI	H-MVIP	High Speed
		SDF	UDF			
7	GEN_SYNC	√	n/a	0	0	n/a
6:4	CLK_SOURCE_TX	√	√	√	n/a	√
3	CLK_SOURCE_RX	√	√	√	n/a	√
2	SRTS_EN	0	√	√	0	√
1:0	FR_STRUCT	√	10	√	√	n/a

8.3 Direct Low Speed Mode

Direct Low Speed mode is used for interconnecting to standard T1 or E1 framers and to devices which support the MVIP-90 protocol. For the AAL1gator, this mode is mainly a pass through mode between the external 16/8/4 lines and the lower 16/8/4 local links. In the AAL1gator-32, the lower 16 local links connect to A1SP 0 and A1SP 1. The second RAM interface of the AAL1gator-32 cannot be used and is not needed when using Direct Low Speed Mode.

DS1 / E1 Links

When configured in Direct Low Speed Mode, the AAL1gator-32, AAL1gator-8 and AAL1gator-4 support sixteen, eight and four DS1/E1 links respectively where each link comprises a clock, data, frame pulse, and signaling pin for each direction. In addition, low speed clear channel data streams can be passed in this mode. A common clock pin is also available, which can be shared across all receive lines or all transmit lines and is selectable on a per line basis

Some framers also share a clock and signaling pin, where the clock pin becomes a signaling pin when signaling is required or remains as a clock pin when individual clocks per line are required. When this pin carries signaling information a common clock is used, which is shared across all lines. This option can be configured on a per line basis.

MVIP-90

2 Mbps MVIP mode is also supported where the line is handled in accordance with the MVIP-90 specification. MVIP mode can be individually selected per line for all lines. Tri-stating of individual time slots is not supported. There is a common 4 MHz clock and a common framing reference signal.

Unstructured J2

Using the clock and data pins in Direct Low Speed Mode, the AAL1gator devices support unstructured J2 transport over ATM. The AAL1gator-32 supports up to six unstructured J2 lines while the AAL1gator-8 and AAL1gator-4 support three unstructured J2 lines.

8.3.1 Line Format and Frame Structure

Selection between T1 or E1 clock/data lines is based on the value of T1_MODE in the LIN_STR_MODE memory register for each line. Selection between standard clock/data or MVIP-90 line format is based on the value of MVIP_EN in the LS_Ln_CFG_REG register for each line.

The frame structure in Direct Low Speed Mode can be Structured Data Format – Frame (SDF-FR), Structured Data Format – Multiframe (SDF-MF) or Unstructured Data Format – Multi-Line (UDF-ML) and is determined by the value of FR_STRUCT[1:0] in the LIN_STR_MODE memory register for each line as follows:

FR_STRUCT[1:0]	Frame Structure	Description
00		Not Used.
01	SDF-FR	A structured connection where CAS signaling is not being transported (basic service).
10	UDF-ML	An unstructured clear channel bit stream for line speeds < 15 Mbps (supports 8 lines per A1SP if all are under 2.5 Mbps).
11	SDF-MF	A structured connection where CAS signaling is being transported.

Notes:

- MVIP-90 lines can not be configured for UDF-ML.
- If a mixture of CAS and non-CAS connections are being made on the same line, then put the line in SDF-MF mode and set R_CHAN_NO_SIG and T_CHAN_NO_SIG in the queue tables for the connections not carrying CAS.

Table 10 summarizes configuration of line format and frame structure.

Table 10 – Configuration of Line Format and Frame Structure

Mode		T1_MODE	MVIP_EN	FR_STRUCT[1:0]
T1	SDF-FR	1	0	01
	SDF-MF	1	0	11
	UDF-ML	n/a	0	10
E1	SDF-FR	0	0	01
	SDF-MF	0	0	11
	UDF-ML	n/a	0	10
J2	UDF-ML	n/a	0	10
MVIP-90	SDF-FR	0	1	01
	SDF-MF	0	1	11

8.3.2 Line Clock Source

Several clocking options exist in this mode and are controlled by the value of the CLK_SOURCE bits in the LIN_STR_MODE register for each line.

8.3.2.1 Receive Line Clock Source

In the receive direction, the line clock source has two options based on the value of the CLK_SOURCE_RX bit in LIN_STR_MODE:

CLK_SOURCE_RX	Function
0	The line receives its clock from the external clock (RL_CLK[n] pin) associated with that line.
1	The line receives its clock from the common external clock (CRL_CLK pin).

8.3.2.2 Transmit Line Clock Source

In the transmit direction, eight possible options exist and are controlled by the value of the CLK_SOURCE_TX bits in the LIN_STR_MODE memory register for each line. When read by the A1SP, this value will override the setting defined by the TLCLK_OE input pin. If switching from an external to an internal clock or visa versa, make sure there are not two clocks driving simultaneously. The options are listed below:

CLK_SOURCE_TX[2:0]	Function
000	Clock is an input on pin TL_CLK[n].
001	Clock is an input on pin RL_CLK[n] (loop timing mode).
010	Clock is internally synthesized as a nominal T1 or E1 clock based on SYS_CLK and the value of T1_MODE. The clock is output on the TL_CLK[n] pin.
011	Clock is internally synthesized based on the received SRTS values. The clock is output on TL_CLK[n] pin.
100	Clock is internally synthesized using the adaptive algorithm which uses receive buffer depth to control TL_CLK[n]. The clock is output on TL_CLK[n] pin.
101	Clock is internally synthesized based on values received on the external clock control interface. This mode is used for external implementations of SRTS or Adaptive clocking. The clock is output on TL_CLK[n] pin.
110	The line uses the common external clock (CTL_CLK pin).
111	The line uses the common external clock (CTL_CLK pin) and signaling data (TL_SIG[n]) is driven onto the TL_CLK[n] pin.

8.3.2.2.1 AAL1 Clock Generation Control

The Clock Generation Control (CGC) block is responsible for generating the synthesized transmit line clocks. Options “010” through “101” for CLK_SOURCE_TX involve the CCG block.

A given line clock is synthesized internally using a 38.88 MHz system clock (SYS_CLK). Only T1 or E1 clocks can be generated internally. Any other frequency clock must be generated externally and passed into the AAL1gator as an input. The frequency of the synthesized clock can be controlled via an external input, the internal SRTS algorithm, or the internal adaptive algorithm. Each line clock can be controlled independently. To assist an external source in determining what frequency to use, SRTS and adaptive information is output using the external interface.

8.3.2.2.1.1 Synchronous Residual Time Stamp

The SRTS block within the CGC block receives SRTS values and uses the values to determine the frequency to be synthesized. When enabled, the local SRTS values, which are calculated within this SRTS block, are subtracted from the SRTS values received in the cells received by RALP, which represent the remote SRTS values. This SRTS difference is sent by the SRTS block to the Frequency Synthesizer to indicate what frequencies should be synthesized for each line. The SRTS difference is also played out externally.

SRTS functionality is enabled by setting the SRTS_EN bit in the LIN_STR_MODE memory register. SRTS is supported for unstructured data formats on a per-line basis. Leave this bit clear for structured data formats.

SRTS_EN	Function
0	SRTS disabled. The CSI bits of the odd transmit AAL1 cells are set to 0 and the received SRTS bits are ignored.
1	SRTS enabled. The insertion of the transmit SRTS bits is enabled for this line and the received SRTS bits are accumulated.

8.3.2.2.1.2 Adaptive Clocking Algorithm

The Adaptive block determines the appropriate line clock frequencies based on the buffer depth received from the A1SP. Every time a cell is received on a particular line, the Adaptive block is given the current depth of the receive buffer. If the buffer depth is increasing, then the local line clock is running slower than the remote line clock. If the buffer depth is decreasing, then the local line clock is running faster than the remote line clock. Therefore, the Adaptive block will adjust the local line clock according to the buffer depth by passing the appropriate value to the Frequency Synthesizer.

Adaptive clocking is only supported for unstructured connections inside the AAL1gator. If adaptive clocking is desired for structured connections, it will need to be processed externally.

8.3.2.2.1.3 Frequency Synthesizer

The Frequency Synthesizer block synthesizes any one of 256 possible frequencies centered around either the T1 or E1 nominal frequency based upon an 8-bit select value. The synthesized frequency is derived from the 38.88 MHz system clock.

The Frequency Synthesizer block receives an 8-bit two's complement number to select a frequency setting. Although possible (with 8 bits) to input a value of -128 to 127 this input is limited internally to –83 to 88 for T1 and -128 to 111 for E1. This is done in order to not exceed the frequency range specification of +/- 200ppm for T1 and +/- 100ppm for E1. Based on this value, the Frequency Synthesizer synthesizes a clock for each line. The line frequency is synthesized by dividing down the 38.88 MHz system clock. The method for dividing down the system clock is dependent on whether the line is in T1 or E1 mode.

The synthesizers can be set to operate in normal or high resolution mode based on the value of HI_RES_SYNTH in the LIN_STR_MODE memory register.

HI_RES_SYNTH	Function
0	Normal mode: only the 4 high order bits of the frequency setting value are monitored to generate 1 of 16 frequencies. SRTS requires normal mode.
1	High resolution mode: all 8 bits are monitored to generate 1 of 256 frequencies. Adaptive clocking requires high resolution mode.

Table 11 indicates the resolution modes that must be used in the four CLK_SOURCE_TX options that use the CGC block. A "√" indicates that either normal or high resolution mode can be selected for that clock source option.

Table 11 – Frequency Synthesis Resolution Modes

CLK_SOURCE_TX[2:0]	Synthesized Clock Type	HI_RES_SYNTH
010	Nominal T1 or E1	√
011	SRTS	0
100	Adaptive	1
101	Externally controlled	√

8.3.3 Synchronization

8.3.3.1 Receive Direction

The Line Interface Block accepts deframed data from the external lines. The data, signaling and synchronization signals are received from the external interface. The external lines can support data rates up to 15 Mbps per line. The falling edge of RL_CLK[n] is used to clock in the data and is used as the active

edge for all receive logic in this mode. For structured data, the Line Interface Block uses the external synchronization input signals (RL_SYNC[n]) as either frame pulses or multi-frame pulses. Whether the Line Interface Block interprets RL_SYNC[n] as a frame pulse or a multi-frame pulse is determined by the value of MF_SYNC_MODE in the LS_Ln_CFG_REG register for that line.

MF_SYNC_MODE	Function
0	Sync signals are frame sync signals.
1	Sync signals are multi-frame sync signals.

Notes:

- If the line is configured for UDF-ML mode (unstructured), the data will be passed as a clear channel bit stream and RL_SYNC[n] will be ignored. In this case, leave MF_SYNC_MODE clear as default.

- In the receive direction, synchronization is always controlled by the external line interface.

In T1/E1 mode, the first time RL_SYNC[n] is sampled high after being low indicates the first bit of a frame or multi-frame. In MVIP-90 mode, the first time RL_SYNC[n] is sampled low after being high indicates the first bit of a frame. For T1 structured data, a frame is completed every 193 bits. For E1 or MVIP-90 structured data, a frame is completed every 32 bytes.

8.3.3.2 Transmit Direction

In the line transmit direction, for structured data, the Line Interface Block takes the TL_SYNC[n] signal and depending on the value of MF_SYNC_MODE, interprets the signal as either a frame pulse or multi-frame pulse. The MF_SYNC_MODE bit from section 8.3.3.1 configures both the receive and transmit directions and its encoding is repeated below:

MF_SYNC_MODE	Function
0	Sync signals are frame sync signals.
1	Sync signals are multi-frame sync signals.

Note: If the line is configured for UDF-ML mode (unstructured), the data will be played out as a clear channel bit stream and TL_SYNC[n] will be ignored. In this case, leave MF_SYNC_MODE clear as default.

In the transmit direction, synchronization can be controlled from either the local link side or the external line side based on the value of GEN_SYNC in the LIN_STR_MODE memory register.

GEN_SYNC	Function
0	TL_SYNC[n] is received from the corresponding external framing device and is an input for this line.
1	TL_SYNC[n] is generated internally by the AAL1gator and is an output for this line.

Note: If no synchronization is required, then leave GEN_SYNC clear as default.

In T1/E1 mode, the first time TL_SYNC[n] is sampled high after being low indicates the first bit of a frame or multi-frame. In MVIP-90 mode, the first time TL_SYNC[n] is sampled low after being high indicates the first bit of a frame. For T1 structured data, a frame is completed every 193 bits. For E1 or MVIP-90 structured data, a frame is completed every 32 bytes.

8.3.4 CAS Signaling

This section is applicable to structured data only. Unstructured data is sent and received without regard to the byte alignment of data within a frame and is placed in the frame buffer in the order in which it arrives.

In the receive direction, signaling is accumulated on RL_SIG[n] from the framer over an entire multi-frame, so signaling only has to be sampled once per multi-frame. The AAL1gator reads signaling during the last frame of every multi-frame. The AAL1gator reads the signaling nibble for each channel when it reads the last nibble of each channel's data.

In the transmit direction, signaling data is driven on TL_SIG[n] for all frames of any multi-frame and will change only on multi-frame boundaries. The signaling nibble is valid for each channel when the last nibble of each channel's data is being driven.

In T1 mode, a multi-frame can either be 12 or 24 frames of 24 timeslots depending on if the line is in Super Frame (SF) or Extended Super Frame (ESF) mode. The AAL1gator accommodates the T1 Super Frame (SF) mode by treating it like the Extended Super Frame (ESF) format and updating signaling

data only on the last frame of odd SF multiframes. Figure 15 shows an example of a T1 frame in the receive direction.

Figure 15 – Capture of T1 Signaling Bits

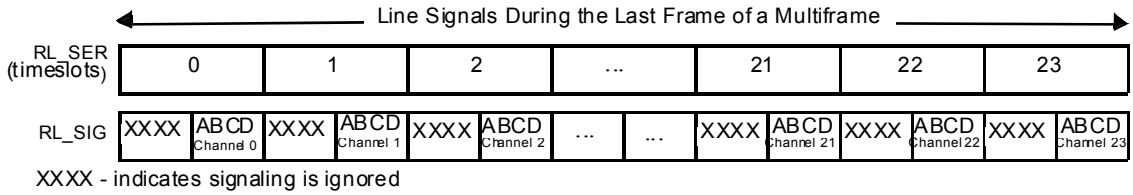
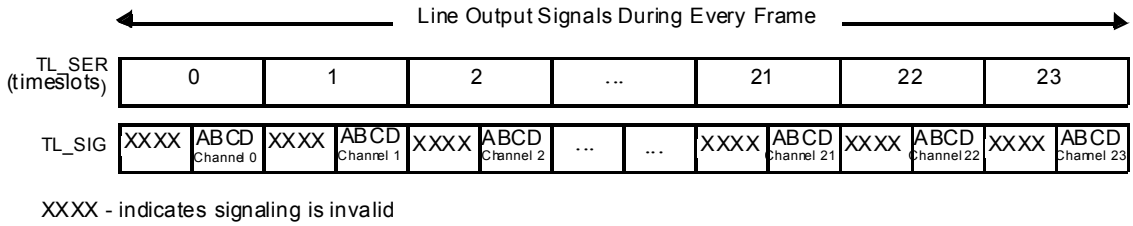


Figure 16 shows an example of a T1 frame in the transmit direction.

Figure 16 – Output of T1 Signaling Bits



Normally an E1 multiframe consists of 16 frames of 32 timeslots, where signaling changes on multiframe boundaries. A special case of E1 mode exists that permits the use of T1 signaling with E1 framing based on the value of E1_WITH_T1_SIG in the LIN_STR_MODE memory register. When E1_WITH_T1_SIG is set and the line is in E1 SDF-MF mode, the AAL1gator will use a multiframe consisting of 24 frames of 32 timeslots.

E1_WITH_T1_SIG	Function
0	Use E1 signaling. Signaling is updated every 16 frames.
1	Use T1 signaling. Signaling is updated every 24 frames instead of every 16 frames.

Figure 17 shows an example of an E1 frame in the receive direction.

Figure 17 – Capture of E1 Signaling Bits

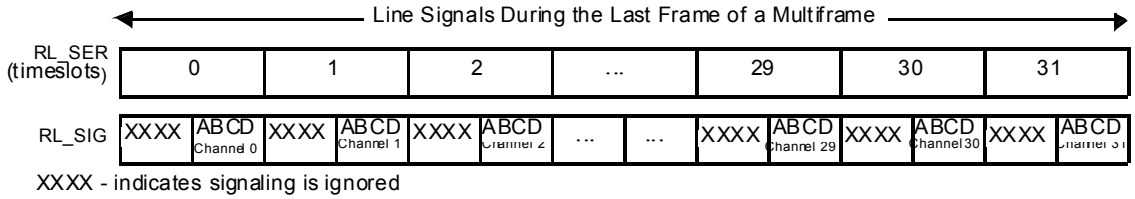
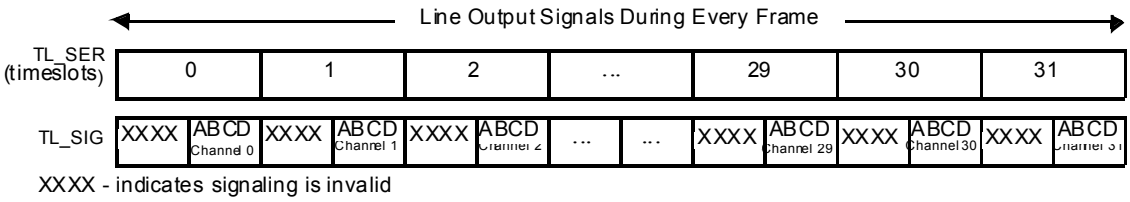


Figure 18 shows an example of an E1 frame in the transmit direction.

Figure 18 – Output of E1 Signaling Bits



Note: The AAL1gator treats all 32 timeslots identically. Although E1 data streams contain 30 timeslots of channel data, 1 timeslot of framing (timeslot 0) and one time slot that can either be signaling or data (time slot 16), data and signaling for all 32 timeslots are stored in memory and can be sent and received in cells.

8.3.5 Other Per-Line Options

Multiframe Alignment

By default, in SDF-MF mode, only the CDVT value is taken into account when determining when to play out data. This provides predictable delay but causes a difference in MF alignment on both sides of the ATM network.

To align the MF structure in the cell with the external MF pulse, MF_ALIGN_EN should be set. If MF_ALIGN_EN is set, then RALP will not only queue up one CDVT worth of time of data but will also delay the write pointer to the next MF boundary. This method will ensure MF alignment across the ATM network but will add variable delay that could be between 0 and 3 ms.

Usually, MF_ALIGN_EN is not set because predictable delay is more important than MF alignment and a variable delay is undesirable. Note that

MF_ALIGN_EN is only used for lines in SDF-MF mode. MF_ALIGN_EN is encoded as follows:

MF_ALIGN_EN	Function
0	Only the CDVT value is taken into account when determining when to play out data.
1	The CDVT and the multiframe boundaries are taken into account; the data is aligned with the next MF boundary after CDVT.

Reference Value Generation

This bit enables the generation of the reference value for this line used when adding queues. This bit is used for structured modes only.

REF_VAL_ENABLE	Function
0	Reference value generation is disabled. The first cell of a queue is scheduled relative to frame 0.
1	Reference value generation is enabled. The reference value increments to emulate a single DS0 queue every time the current frame value is equal to the current reference value.

Low CDV for Unstructured Mode

For UDF-ML lines, the LOW_CDV bit of the LIN_STR_MODE memory register can be set to cause cells to be scheduled every 47 bytes instead of using frame based scheduling. This eliminates the CDV caused by the scheduler. This mode can only be used in UDF-ML mode when BYTES_PER_CELL is 47. This mode cannot be used when the queue is configured for partial cells or for AAL0 mode.

LOW_CDV	Function
0	Unstructured line is not in low CDV mode. AAL1gator uses frame based scheduling.
1	Unstructured line is in low CDV mode. AAL1gator uses byte based scheduling where cells are scheduled every 47 bytes.

8.4 SBI Mode

The Scaleable Bandwidth Interconnect (SBI) is a synchronous, time-division multiplexed bus designed to transfer, in a pin-efficient manner, data belonging to a number of independently timed links of varying bandwidth.

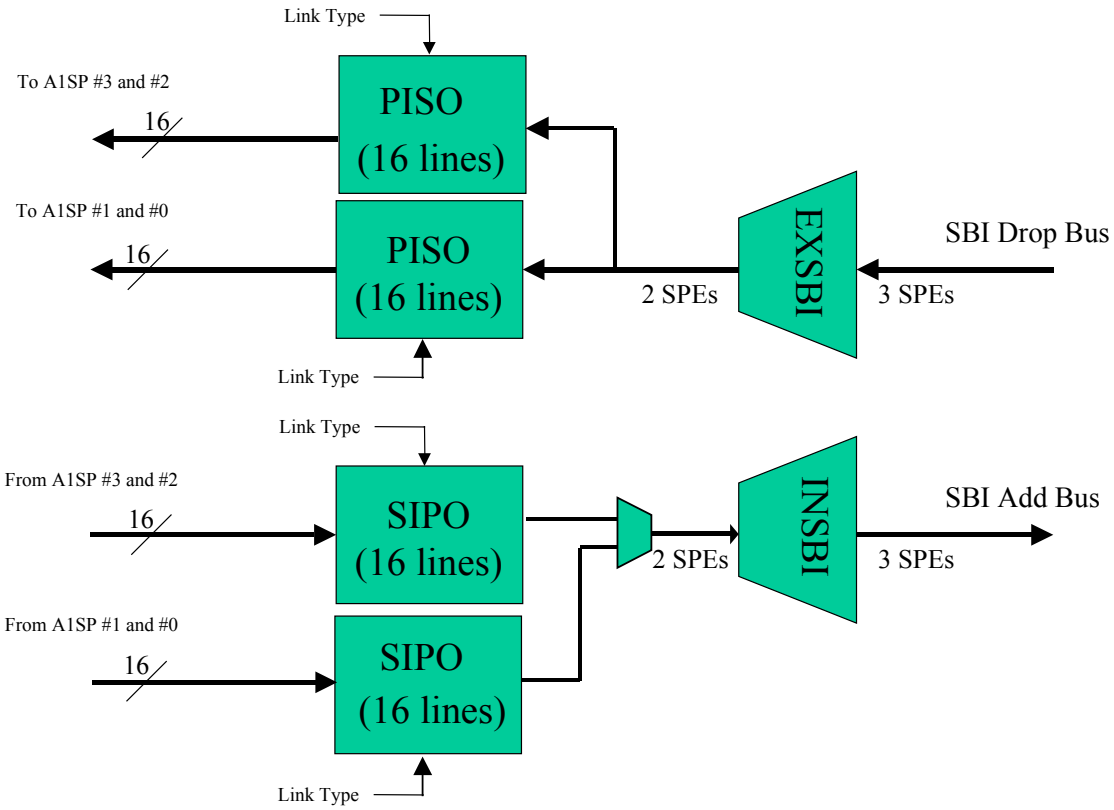
The SBI multiplexing structure is modeled on the SONET/SDH standards. The SONET/SDH virtual tributary structure is used to carry T1 and E1 links. Unchannelised DS3 payloads follow a byte synchronous structure modeled on the SONET/SDH format. The multiplexed links are separated into three Synchronous Payload Envelopes (SPEs). Each envelope may be configured independently to carry up to 28 T1s, 21 E1s or a DS3.

Full details of the operation of the SBI interface are provided in the SBI Compatibility Specification [6].

The Scaleable Bandwidth Interconnect (SBI) Line Interface mode is used in the AAL1gator-32 to interface to the PM8315 TEMUX chip or any other PHY layer chip that has an SBI Interface on it. The SBI Block within the Line Interface Block interfaces to the SBI bus on one side and inputs/outputs up to 32 T1 links, 32 E1 links or two DS3 links to the A1SP blocks. It can also support a mix of links made up of a combination of 16 T1 or E1 links or 1 DS3 link. In SBI mode the upper 16 links make up one configurable unit and the lower 16 links make up another unit.

The SBI interface block is made up of 4 main components: Extract SBI Block (EXSBI), Insert SBI BLOCK (INSBI), Parallel In to Serial Out Converter (PISO), and Serial In to Parallel Out Converter (SIPO). There are two PISO and SIPO blocks, each of which processes 16 links. See Figure 19 for the SBI Block architecture.

Figure 19 – SBI Block Architecture



The EXSBI is responsible for extracting links from the SBI Drop bus and switching them to either SPE#1 which is mapped to the lower 16 links or SPE#2 which is mapped to the upper 16 links. The links are output to an internal parallel bus.

The INSBI is responsible for taking links from the internal parallel bus and inserting the remapped links to the SBI Add bus.

The PISO block is responsible for taking the internal parallel bus from EXSBI and serializing the links for the local link interface. It is also responsible for generating the receive serial clocks for the local links going to each A1SP block. There is one PISO block for each group of 16 links.

The SIPO block is responsible for taking the serial links from the local link interface and putting them onto the internal parallel bus to INSBI. It is also responsible for handling the generation of the transmit serial clock for the local links coming from the A1SP blocks, if the local link is receiving its transmit timing from the SBI. There is one SIPO block for each group of 16 links.

8.4.1 Programming the SBI Interface

To have a clean start up, the following programming sequences are recommended when setting up the SBI interface:

General Rules

- After clearing Chip Software Reset, the internal rams go through an internal initialization process that takes at least 500 µs. The BUSY bit in the **Extract/Insert Tributary RAM Indirect Access Control Registers** (0x80404/0x80504) should be polled until low before attempting any INSBI/EXSBI read and write accesses.

- General SBI control registers outside of EXSBI/INSBI can be configured at any time.

- At initialization, the SBI tributary receiver should be enabled before the corresponding SBI tributary transmitter (i.e. – configure the EXSBI before configuring the INSBI)

- If DC_EN in the **Extract/Insert Control Register** (0x80400/0x80500) is disabled and an overrun/underrun condition is reported for a link, the link should be reset by writing to the tributary control register for the tributary corresponding to that link. If DC_EN is set, then the tributary will automatically be reset.

- To configure individual link to be in synchronous mode, the SPEn_SYNCH bit in the **SBI_BUS_CFG_REG** (0x80300) must be clear and the corresponding SYNC_LINK[n] bit in **SBI_SYNC_LINK_REGL/H** (0x80304/0x80305) must be set. Furthermore, the SYNCH_TRIB bit in the **Insert Tributary Control Indirect Access Data Register** (0x80506) must also be set.

- It is important to note that in synchronous mode, a tributary must derive its clocking from the SBI bus. In other words, that tributary must be configured to be clock slave to the SBI bus by clearing the CLK_MSTR bit in both **Extract and Insert Control RAM Indirect Access Data** (0x80406/0x80506) registers

- The TS_EN bit in **SBI Bus Configuration Register** (0x80300) must be set to globally enable tributary to link mapping. If this bit is set then mapping may occur on a tributary by tributary basis. If this bit is a zero, which is the default value, then the first 16 links are mapped to the first 16 links of SPE1 and the upper 16 links are mapped to the first 16 links of SPE2.

- There is also a TS_EN bit in the **Extract/Insert Control Register** (0x80400/0x80500), which operates in a similar fashion. To map an individual tributary to a link, both TS_EN bit on the Extract and on the Insert side must be set in addition to the global TS_EN bit in the SBI Bus Configuration Register.

Tributary Mapping Sequence

1. Configure all tributaries in the SBI Extract Tributary Mapping RAM by following the steps described in section 8.4.3.2.1. Configure all tributaries in the SBI Insert Tributary Mapping RAM by following the steps described in section 8.4.4.2.1.

2. Set the global TS_EN bit in the **SBI Bus Configuration Register** (0x83000).

3. Set the global TS_EN bit in the **EXSBI/INSBI Control Registers** (0x84000/0x85000).

4. Configure all of the SBI Extract Tributary Control RAM by following the steps described in section 8.4.3.2.2. Configure all of the SBI Insert Tributary Control RAM by following the steps described in section 8.4.4.2.2.

5. Enable the SPEs last by setting the SPEn_ENBL bits in the **SBI Bus Configuration Register** (0x83000).

Note: It is important to note that if tributary mapping needs to be done on any tributary, then all tributary mapping must be done to all tributaries in the SBI (i.e., 84 tributaries if all 3 SPEs are T1).

Synchronous Configuration

Even though there is no strict sequence to be followed when configuring a link to be synchronous or asynchronous mode, there are a few important notes to be aware of:

- If any of the 32 links is configured for synchronous mode, then the DC_EN and DC_INT_EN bits in the **Extract/Insert Control Registers** (0x84000/0x85000) must be disabled by clearing the bits. This must be done because Depth Check logic does not support synchronous mode.

- If any link/tributary within an E1 SPE is in synchronous mode, the C1FP pulses **must not** be 500 µs apart (2Khz rate) but 6 ms instead, which is once every 48 SBI frames.

8.4.2 General SBI Configuration

General SBI configuration is performed by programming bits within the following internal registers:

Table 12 – General SBI Register Memory Map

Offset	Register Description	Register Mnemonic
These registers are selected when the address = 0x803XX.		
0x00	SBI Bus Configuration Register	SBI_BUS_CFG_REG
0x01	SBI Link Configuration Register	SBI_LNK_CFG_REG
0x02	SBI Link Disable Low	SBI_LINKL_DISABLE
0x03	SBI Link Disable High	SBI_LINKH_DISABLE
0x04	SBI Sync Link Low	SBI_SYNC_LINKL
0x05	SBI Sync Link High	SBI_SYNC_LINKH
0x06	Reserved	
0x07	Reserved	
0x08	SBI Extract Bus Alarm Interrupt Register Low	EXT_ALRM_INTL
0x09	SBI Extract Bus Alarm Interrupt Register High	EXT_ALRM_INTH
0x0A	SBI Extract Bus Alarm Status Register Low	EXT_ALRM_STATL
0x0B	SBI Extract Bus Alarm Status Register High	EXT_ALRM_STATH
0x0C	SBI Insert Bus Alarm Insert Register Low	INS_ALRM_REGL
0x0D	SBI Insert Bus Alarm Insert Register High	INS_ALRM_REGH

The default configuration is as follows:

Bit	Register	Value
SPE1_TYP[1:0]	**SBI_BUS_CFG_REG** (0x80300)	00
SPE2_TYP[1:0]	**SBI_BUS_CFG_REG** (0x80300)	00
SPE3_TYP[1:0]	**SBI_BUS_CFG_REG** (0x80300)	00
SPE1_ENBL	**SBI_BUS_CFG_REG** (0x80300)	0
SPE2_ENBL	**SBI_BUS_CFG_REG** (0x80300)	0
SPE3_ENBL	**SBI_BUS_CFG_REG** (0x80300)	0

Bit	Register	Value
SPE1_SYNCH	**SBI_BUS_CFG_REG** (0x80300)	0
SPE2_SYNCH	**SBI_BUS_CFG_REG** (0x80300)	0
SPE3_SYNCH	**SBI_BUS_CFG_REG** (0x80300)	0
CLK_MSTR	**SBI_BUS_CFG_REG** (0x80300)	0
TS_EN	**SBI_BUS_CFG_REG** (0x80300)	0
LINK_TYPL[1:0]	**SBI_LNK_CFG_REG** (0x80301)	00
FST_CK_FRQL [1:0]	**SBI_LNK_CFG_REG** (0x80301)	00
CKOUT_KILLL	**SBI_LNK_CFG_REG** (0x80301)	0
LINK_TYPH[1:0]	**SBI_LNK_CFG_REG** (0x80301)	00
FST_CK_FRQH [1:0]	**SBI_LNK_CFG_REG** (0x80301)	00
CKOUT_KILLH	**SBI_LNK_CFG_REG** (0x80301)	0
LINK_DIS[15:0]	**SBI_LINK_DIS_REGL** (0x80302)	0x00
LINK_DIS[31:16]	**SBI_LINK_DIS_REGH** (0x80303)	0x00
SYNC_LINK[15:0]	**SBI_SYNC_LINK_REGL** (0x80304)	0x00
SYNC_LINK [31:16]	**SBI_SYNC_LINK_REGH** (0x80305)	0x00
SBI_ALARM_INT [15:0]	**EXT_ALRM_INT_REGL** (0x80308)	0x00
SBI_ALARM_INT [31:16]	**EXT_ALRM_INT_REGH** (0x80309)	0x00
SBI_ALRM_STAT [15:0]	**EXT_ALRM_STAT_REGL** (0x8030A)	0x00
SBI_ALRM_STAT [31:16]	**EXT_ALRM_STAT_REGH** (0x8030B)	0x00
SBI_ALRM_INS [15:0]	**INS_ALRM_REGL** (0x8030C)	0x00
SBI_ALRM_INS [31:16]	**INS_ALRM_REGH** (0x8030D)	0x00

The default indicates that all three Synchronous Payload Envelopes are configured for T1 links and the FASTCLK input operates at a frequency of 51.84 MHz. The three SPEs are disabled in the default state.

8.4.2.1 SBI Bus Configuration and Tributary Override Control

The SBI bus contains 3 Synchronous Payload Envelopes (SPE) which can be configured to be E1, T1, or DS3. **SBI_BUS_CFG_REG** (0x80300) defines the payload type for each SPE, enables each SPE and provides some override control for the individual tributaries.

SPE*n* Payload Type

The SBI bus contains 3 Synchronous Payload Envelopes (SPE) which can be configured to be E1, T1, or DS3. The SPE*n*_TYP fields identify the payload type of each SPE. The encoding for SPE*n*_TYPE is:

Payload Type	SPE*n*_TYP Value
T1	b"00"
E1	b"01"
DS3	b"10"
Reserved	b"11"

SPE*n* Enable

The SPE*n*_ENBL field is used to enable or disable an entire SPE on the SBI. All SPEs default to being disabled.

SPEn_ENBL	Function
0	SPE*n* is disabled.
1	SPE*n* is enabled.

Synchronous Mode for All Tributaries within SPE*n*

The SPE*n*_SYNCH field is used to specify that all tributaries within the SPE should operate in synchronous mode. This can be used as an override mechanism for configuring all tributaries within the particular SPE to be in synchronous mode irrespective of the TRIB_CTL settings.

SPE*n*_ENBL	Function
0	The value of the TRIB_CTL settings will be used when determining if an individual tributary within SPE*n* is in synchronous mode.
1	All tributaries within SPE*n* will be forced to be in synchronous mode.

Clock Master Mode for All Tributaries within SPE*n*

The CLK_MSTR field is used as an override mechanism for configuring all tributaries to be clock masters irrespective of the TRIB_CTL settings. This TRIB_CTL settings default so that all tributaries are clock slaves.

CLK_MSTR	Function
0	The value of the TRIB_CTL settings will be used when determining if an individual tributary within SPE*n* is a clock master.
1	All tributaries within SPE*n* will be forced to be clock masters.

One to One Tributary to Link Mapping

The TimeSwitch Enable (TS_EN) field is used as an override mechanism for configuring Tributary to Link mapping to be one to one irrespective of the TRIB_MAP settings.

TS_EN	Function
0	The first 16 links are mapped to the first 16 links of SPE1 and the upper 16 links are mapped to the first 16 links of SPE2.
1	The value of the TS_EN bit in the TRIB_CTL registers will be used to determine tributary to link mapping.

8.4.2.2 SBI Link Configuration

The AAL1gator-32 supports up to 32 links which, in SBI mode, can be mapped to any tributary on the SBI bus. The lower 16 links and the upper 16 links can be configured as two separate groups to handle 16 E1 or T1 links. Link 0 and link

16 can alternatively be configured for DS3 mode. A mix of configurations is possible, but all links within the same group have to be configured the same.

SBI_LNK_CFG_REG (0x80301) configures the two groups of links in SBI mode. The High group contains the upper 16 links and is configured independently from the low group which contains the bottom 16 links.

Per-Group Link Type

The LINK_TYPL field defines the link type for the lower group of links. The LINK_TYPH field defines the link type for the higher group of links.

If LINK_TYP*x* is set to E1 or T1 mode, then the T1_MODE bit in the LIN_STR_MODE memory register associated with the links in this group needs to be set to the same value. For example, if LINK_TYPH is set to T1 mode then T1_MODE bit must be set in the LIN_STR_MODE register for links 16-31 (lines 0-7 in both A1SP 2 and A1SP 3).

If the LINK_TYP*x* is set to DS3 mode, then the UDF_HS mode bit should be set in the HS_LIN_REG memory register in either A1SP0 or A1SP2.

The encoding of these fields is as follows:

Payload Type	LINK_TYP*x*[1:0]	T1_MODE	UDF_HS
T1	b"00"	1	0
E1	b"01"	0	0
DS3	b"10"	√	1
Reserved	b"11"		

Note: The T1_MODE bit is selectable for DS3 payload types as described in section 8.6.1.2. All configurable features covered in section 8.6 for high speed lines is applicable for DS3 links conveyed via SBI.

DS3 Fast Clock Generation

It is strongly recommended that the DS3 clock be passed externally to the AAL1gator-32. However, DS3 links may use a separate clock input, FASTCLK, for clock generation. The use of FASTCLK will adversely affect jitter performance.

Three different FASTCLK frequencies are supported in each case. The FST_CK_FRQL field selects the frequency being used for link number 0. The

FST_CK_FRQH field selects the frequency being used for link number 16. The encoding of FST_CK_FRQ*x* is given below:

FST_CK_FRQ*x* [1:0]	FASTCLK Frequency
"00"	51.84 MHz
"01"	44.928 MHz
"10"	Reserved
"11"	66 MHz

Note: Illegal values on FST_CK_FRQx[1:0] will result in the default frequency of 51.84MHz being used.

Clock Kill

The CKOUT_KILLL field is used to disable the clock on all the lower 16 links. The CKOUT_KILLH field is used to disable the clock on all the upper 16 links.

CKOUT_KILLx	Function
0	Clocks are not disabled.
1	Clocks are disabled.

8.4.2.3 SBI Link Disable

The Link Disable *n* (LINK_DIS*n*) bit allows individual serial SBI local links to be disabled. LINK_DIS[15:0] is located in **SBI_LINK_DIS_REGL** (0x80302) while LINK_DIS[31:16] is located in **SBI_LINK_DIS_REGH** (0x80303). All serial links default to being enabled.

LINK_DIS*n*	Function
0	Link *n* is enabled.
1	Link *n* is disabled.

8.4.2.4 Synchronous/Asynchronous Mode for Local Links

The SYNC Link *n* (SYNC_LINK*n*) bit must be set if the SBI local link is mapped to an SBI tributary which is operating in synchronous mode. An SBI tributary is in

synchronous mode when either the corresponding SPE*n*_SYNCH is set in the SBI_BUS_CFG register or TRIB_SYNCH is set in the corresponding Insert Tributary Control Register. SYNC_LINK[15:0] is located in **SBI_SYNC_LINK_REGL** (0x80304) while SYNC_LINK[31:16] is located in **SBI_SYNC_LINK_REGH** (0x80305).

SYNC_LINKn	Function
0	Link *n* is operating in asynchronous mode. The serial clock output to link *n* will be sourced from the PISO block.
1	Link *n* is operating in synchronous mode. The serial clock output to link *n* will be sourced from the Insert SBI Block.

In synchronous mode, transmit frame/multi-frame pulses are sourced by the INSBI block.

In asynchronous mode, transmit frame/multi-frame pulses are sourced from the SBI tributaries. Therefore, GEN_SYNC in the LIN_STR_MODE memory register must be set to '0'.

8.4.2.5 SBI Alarm Configuration

SBI Extract Alarm State Transition Indication

When set, SBI_ALRM_INT*n* indicates that the SBI alarm state has changed on the SBI tributary mapped to this link. This bit is cleared upon reading. Read SBI_ALRM_STATL to see the current state. When a bit is set in either EXT_ALRM_INTL or EXT_ALRM_INTH and SBI_ALARM is enabled, the SBI_ALARM bit will be set in the MSTR_INTR register, which will activate INTB.

SBI_ALRM_INT[15:0] is located in **EXT_ALRM_INT_REGL** (0x80308) and SBI_ALRM_INT[31:16] is located in **EXT_ALRM_INT_REGH** (0x80309).

SBI Extract Alarm Status

SBI_ALRM_STATn indicates the current state of the Extract SBI ALARM signal on the SBI tributary mapped to this link. The SBI_ALRM_INT*n* bit will be set in the SBI_ALRM_INTL or SBI_ALRM_INTH register whenever this bit changes.

SBI_ALRM_STAT[15:0] is located in **EXT_ALRM_STAT_REGL** (0x8030A) and SBI_ALRM_STAT[31:16] is located in **EXT_ALRM_STAT_REGH** (0x8030B).

SBI_ALRM_STAT*n*	Function
0	The ALARM signal for the SBI tributary mapped to link *n* is not active.
1	The ALARM signal for the SBI tributary mapped to link *n* is active.

SBI Insert Alarm Indicator Activation

When set, the SBI_ALRM_INS*n* bit will activate the Insert SBI Alarm indicator for the tributary mapped to this link. SBI_ALRM_INS[15:0] is located in **INS_ALRM_REGL** (0x8030C) and SBI_ALRM_INS[31:16] is located in **INS_ALRM_REGH** (0x8030D).

8.4.3 Extract SBI Block Configuration

The EXSBI demaps up to 32 T1 links, 32 E1 links, two DS3 links from the SBI shared bus. The T1/E1 links can be unframed or framed and channelized, with or without CAS support. The DS3 link can also be unframed or framed. The links, which the EXSBI processes can originate from any SPE but all links within the SPE must be of the same type. Note that link/tributary numbering starts from '1' in SBI mode.

EXSBI configuration is performed by programming bits within the following internal registers:

Table 13 – EXSBI Block Register Memory Map

Offset	Register Description	Register Mnemonic
These registers are selected when the address = 0x804XX.		
0x00	Extract Control Register	EXT_CTL
0x01	Extract FIFO Under Run Interrupt Status Register	EXT_FI_URI
0x02	Extract FIFO Over Run Interrupt Status Register	EXT_FI_ORI
0x03	Extract Tributary RAM Indirect Access Address Register	EXT_TRIAD
0x04	Extract Tributary RAM Indirect Access Control Register	EXT_TRIAC

0x05	Extract Tributary Mapping RAM Indirect Access Data Register	EXT_TRIB_MAP
0x06	Extract Tributary Control RAM Indirect Access Data Register	EXT_TRIB_CTL
0x07	SBI Parity Error Interrupt Status Register	SBI_PERR
0x08	MIN_DEPTH for T1 and E1 Register	EXT_MD_T1E1
0x09	MIN_DEPTH for DS3 Register	EXT_MD_DS3
0x0A	T1 Threshold Register	EXT_T1_THR
0x0B	E1 Threshold Register	EXT_E1_THR
0x0C	DS3 Threshold Register	EXT_DS3_THR
0x0D	Reserved	--
0x0E	Depth Check Reset Interrupt Status Register	EXT_DCR_INT
0x0F	Extract Master Interrupt Register	EXT_MSTR_INT

8.4.3.1 SBI Extract Control

The SBI Extract block is controlled by programming bits within the **EXT_CTL** (0x80400) register. The default configuration is as follows:

Bit	Register	Value
SBI_PAR_CTL	**Extract Control Register** (0x80400)	1
SBI_PERR_EN	**Extract Control Register** (0x80400)	0
TS_EN	**Extract Control Register** (0x80400)	0
FIFO_UDR_EN	**Extract Control Register** (0x80400)	0
FIFO_OVR_EN	**Extract Control Register** (0x80400)	0
SYNC_INT_EN	**Extract Control Register** (0x80400)	0
DC_EN	**Extract Control Register** (0x80400)	1
APAGE	**Extract Control Register** (0x80400)	0

The default indicates that odd parity mode is used for checking the SBI parity signal, all EXSBI interrupts are disabled, tributary to link mapping is disabled, depth check logic is enabled, and page 0 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator-32 links.

SBI Parity Mode

The SBI_PAR_CTL bit is used to configure the Parity mode for checking of the SBI parity signal, DDP, as follows:

SBI_PAR_CTL	Function
0	Even parity checking.
1	Odd parity checking.

Extract SBI Interrupt Enables

There are four interrupt enable bits which are used to enable the generation of an interrupt or set of interrupts when the corresponding interrupt condition is detected. When the interrupt enable bit is '0', the corresponding interrupt is disabled. When the interrupt enable bit is '1', the corresponding interrupt is enabled.

Interrupt Enable Bit	Interrupt Condition
SBI_PERR_EN	SBI Parity Error
FIFO_UDR_EN	FIFO underrun
FIFO_OVR_EN	FIFO overrun
SYNC_INT_EN	Depth Check sync error, C1FP sync error, or SBIIP sync error

Note: Regardless of whether SBI Parity Error interrupts are enabled or disabled, the SBI Parity checker logic will update the **SBI Parity Error Interrupt Status Register** (0x80407).

One to One Tributary to Link Mapping

The TS_EN bit is used to enable the SBI tributary to SBIIP link mapping capability when the TS_EN bit in the SBI_BUS_CFG_REG also enables mapping.

TS_EN	Function
0	Mapping will be fixed to a one to one mapping and will not be programmable.

TS_EN	Function
1	SBI tributary to local link mapping is enabled and is specified by the contents of the Extract Tributary Mapping Register RAM when TS_EN in SBI_BUS_CFG_REG is also set.

Depth Check Logic

This bit enables the Depth Check Logic and is encoded as follows:

DC_EN	Function
0	Depth checker logic is disabled.
1	The depth checker logic will periodically monitor the Data/Framing FIFO Depth and compare it against the write and read pointers. If there is a discrepancy the tributary is synchronously reset by the depth checker.

Note: When in synchronous mode, the DC_EN bit must be clear.

Active Page Selection

The tributary mapping and control configuration RAMs active page select bit (APAGE) controls the selection of one of two pages in the tributary mapping and control configuration RAMs to be the active page. Changes of the active page as a result of write accesses to APAGE will be synchronized to SBI multi-frame boundaries at C1FP.

APAGE	Function
0	The configuration in page 0 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator links.
1	The configuration in page 1 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator links.

8.4.3.2 SBI Extract Tributary Configuration

SBI Extract tributary configuration information is read from and written to the SBI Extract tributary mapping and control RAM.

Please note that the BUSY bit in the **Extract Tributary RAM Indirect Access Control Register** (0x80404) might not be cleared for up to 4.32 µs after either a mapping page switch or a Control RAM or Mapping RAM access.

8.4.3.2.1 SBI Extract Tributary Mapping Configuration

The default tributary to link mapping is straight through i.e. 1:1. Therefore, Link Group LOW, LINK 1 inside the AAL1gator-32 will be mapped by default to SPE1, LINK 1 on the SBI side and so on up to SPE1, LINK 16. Link Group HIGH, LINK 1 inside the AAL1gator will be mapped by default to SPE2, LINK 1 on the SBI side and so on up to SPE2, LINK 16. Any tributary can be mapped to any link, unless the tributary is configured for DS3. In that case the tributary can only be mapped to link 1 or link 17. Note link numbering starts at '1'.

SBI Extract mapping information is read from and written to the SBI Extract tributary mapping RAM. The tributary to link mapping for an SBI tributary in the receive direction is configured using the following procedure:

1. Poll the BUSY bit of the **Extract Tributary RAM Indirect Access Control Register** (0x80404) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. Specify the tributary to which the mapping register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below
SPE[1:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below
MAP_REG	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	1

   The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the mapping register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be set to specify that the Extract Tributary Mapping Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

3. Specify the tributary to link mapping to be written to the tributary mapping RAM by writing the following register:

Bit	Register	Value
LINK[4:0]	**Extract Tributary Mapping RAM Indirect Access Data Register** (0x80405)	See below
LINK_GRP_LOW	**Extract Tributary Mapping RAM Indirect Access Data Register** (0x80405)	See below
LINK_GRP_HIGH	**Extract Tributary Mapping RAM Indirect Access Data Register** (0x80405)	See below

The LINK[4:0] and LINK_GRP_LOW/HIGH fields are used to specify which link of which link group within the AAL1gator-32 is mapped to the SBI tributary specified in step 2. LINK_GRP_LOW/HIGH specifies the link group while LINK[4:0] specifies the link number. Legal values for LINK[4:0] are b"00001" through b"10000".

4. Trigger an indirect write operation on the tributary mapping RAM by writing the following register:

Bit	Register	Value
PAGE	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	See below
RWB	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	0

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary mapping configuration RAM. Note that the BUSY and HST_ADDR_ERR bits in the **Extract Tributary RAM Indirect Access Control Register** (0x80404) are read only and are not affected by write operations.

Notes:

- Following a configuration change, which generates a Configuration Reset, a tributary may not become active for up to 1 ms following the change.
- Mapping of more than one tributary to a link or more than one link to a tributary is not allowed.
- The mapping RAM only needs to be configured if tributary mapping is enabled (TS_EN='1'). If tributary mapping is disabled, the default 1:1 mapping is automatically assumed and no mapping RAM configuration is necessary.

- Configuration changes should be done while the tributary is disabled.
- Whether tributary mapping is performed or using default 1:1 mapping, the control RAM needs to be configured for any active tributaries. All tributaries are initialized to being disabled.

8.4.3.2.2 SBI Extract Tributary Control Configuration

SBI Extract tributary control information is read from and written to the SBI Extract tributary control RAM.

The default configuration of each tributary is as follows:

Bit	Register	Value
TRIB_ENBL	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	0
Reserved	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	0
TRIB_TYP[1:0]	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	00
CLK_MSTR	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	0
CLK_MODE[1:0]	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	00

In the default state, the tributary is disabled, the tributary type is Structured with CAS, the tributary is a clock slave, and EXT_CKCTL[1:0] is used for clocking.

The tributary control information for an SBI tributary in the receive direction is configured using the following procedure:

1. Poll the BUSY bit of the **Extract Tributary RAM Indirect Access Control Register** (0x80404) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.
2. Specify the tributary to which the control register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below

Bit	Register	Value
SPE[1:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below
MAP_REG	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	0

The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the control register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be cleared to specify that the Extract Tributary Control Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

3. Specify the configuration data to be written to the tributary control RAM by writing the following register:

Bit	Register	Value
TRIB_ENBL	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	See below
Reserved	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	0
TRIB_TYP[1:0]	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	See below
CLK_MSTR	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	See below
CLK_MODE[1:0]	**Extract Tributary Control RAM Indirect Access Data Register** (0x80406)	See below

The TRIB_ENBL bit is used to enable the tributary. Writing to the tributary control RAM with the TRIB_ENBL bit set enables the EXSBI to take data from an SBI tributary and transmit that data to the local link mapped to that tributary.

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in the table below. TRIB_TYP should correspond with the FR_STRUCT setting in the LIN_STR_MODE memory register for the link associated with this tributary in the AAL1gator-32 line setup.

CAS Enabled	Framed	Transparent VT (Floating)	TRIB_TYP	FR_STRUCT	Description
True	True	False	00	11	Structured with CAS
False	True	False	01	01	Structured without CAS
False	False	False	10	10	Unstructured
False	False	True	11		Transparent VT (not supported)

Notes:

- CAS can only be enabled for a Structured (framed) tributary.
- "Framed" means framing information available – may be channelized or unchannelized.
- "Unframed" means no framing information available.
- If a mixture of CAS and non-CAS connections are being made on the same line, then put the line in Structured with CAS mode and set R_CHAN_NO_SIG and T_CHAN_NO_SIG in the queue tables for the connections not carrying CAS.

The CLK_MSTR bit is used to specify whether the tributary is a clock master or a clock slave. Setting CLK_MSTR will configure the tributary to be a clock master and clearing CLK_MSTR will configure the tributary to be clock slave. The CLK_MSTR configuration bits are ORed with the CLK_MSTR bit in the SBI_BUS_CFG_REG to allow the chip level to force master mode for all tributaries.

When in clock slave mode, the CLK_MODE[1:0] field determines what type of clocking is used for the tributary. CLK_MOD[1:0] is used directly as the CLK_MODE bits of the EXT_LINKRATE output of the Extract TSB. Phase mode is recommended. CLK_MODE is encoded as follows:

CLK_MODE[1:0]	Function
00	Use EXT_CKCTL[1:0]

CLK_MODE[1:0]	Function
01	Use only ClkRate field of EXT_LINKRATE
10	Use only Phase field of EXT_LINKRATE
11	Reserved

4. Trigger an indirect write operation on the tributary mapping RAM by writing the following register:

Bit	Register	Value
PAGE	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	See below
RWB	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	0

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary control configuration RAM. Note that the BUSY and HST_ADDR_ERR bits in the **Extract Tributary RAM Indirect Access Control Register** (0x80404) are read only and are not affected by write operations.

Notes:

- Following a configuration change, which generates a Configuration Reset, a tributary may not become active for up to 1 ms following the change.
- Any write to a Tributary Control register for a tributary will generate a configuration reset on that tributary, irrespective of whether the data written to the tributary control register is unchanged from the previous value.

8.4.3.2.3 SBI Extract Tributary Configuration Read Procedure

The SBI tributary mapping and control configuration in the receive direction can be read using the following procedure:

1. Poll the BUSY bit of the **Extract Tributary RAM Indirect Access Control Register** (0x80404) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. Specify the tributary to which the mapping register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below
SPE[1:0]	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below
MAP_REG	**Extract Tributary RAM Indirect Access Address Register** (0x80403)	See below

The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the mapping register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be cleared to specify that the Extract Tributary Control Registers are addressed and needs to be set to specify that the Extract Tributary Mapping Registers are addressed.

3. Trigger an indirect read operation on the tributary mapping or control RAM by writing the following register:

Bit	Register	Value
PAGE	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	See below
RWB	**Extract Tributary RAM Indirect Access Control Register** (0x80404)	1

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary mapping or control configuration RAMs. Note that the BUSY and HST_ADDR_ERR bits in the **Extract Tributary RAM Indirect Access Control Register** (0x80404) are read only and are not affected by write operations.

4. Poll the BUSY bit of the **Extract Tributary RAM Indirect Access Control Register** (0x80404) until it is zero. When BUSY is polled to be zero, data from an indirect read operation is available in either the **Extract Tributary Mapping RAM Indirect Access Data Register** (0x80405) or the **Extract Tributary Control RAM Indirect Access Data Register** (0x80406) depending on the value of MAP_REG from step 2.

5. Read the HST_ADDR_ERR bit of the **Extract Tributary RAM Indirect Access Control Register** (0x80404). A value of HST_ADDR_ERR = '1' indicates that an illegal host access was attempted. An illegal host access occurs when an attempt is made to access an out of range tributary. This bit is cleared upon a read access.

8.4.3.3 Depth and Threshold Values

The following fields are used to modify the minimum depth (MIN_DEP), minimum threshold (MIN_THR) and maximum threshold (MAX_THR) for T1, E1, and DS3 tributaries. These values should remain at their default settings.

Bits	Register	Default
MIN_DEP_T1[3:0]	**MIN_DEPTH for T1 and E1 Register** (0x80408)	0x7
MIN_DEP_E1[3:0]	**MIN_DEPTH for T1 and E1 Register** (0x80408)	0x7
MIN_DEP_DS3[3:0]	**MIN_DEPTH for DS3 Register** (0x80409)	0x4
Reserved[3:0]	**MIN_DEPTH for DS3 Register** (0x80409)	0x6
MAX_THR_T1[3:0]	**T1 Threshold Register** (0x8040A)	0xD
MIN_THR_T1[3:0]	**T1 Threshold Register** (0x8040A)	0x2
MAX_THR_E1[3:0]	**E1 Threshold Register** (0x8040B)	0xD
MIN_THR_E1[3:0]	**E1 Threshold Register** (0x8040B)	0x2
MAX_THR_DS3[3:0]	**DS3 Threshold Register** (0x8040C)	0xD
MIN_THR_DS3[3:0]	**DS3 Threshold Register** (0x8040C)	0x2

8.4.3.4 Extract SBI Interrupts

The following master interrupt status bits should be read to detect interrupt conditions in the Extract SBI block:

Bit	Register
EXT_C1FP_INT	**Extract Master Interrupt Status Register** (0x8040F)
EXT_SYNC_INT	**Extract Master Interrupt Status Register** (0x8040F)
EXT_FIFO_OVR_INT	**Extract Master Interrupt Status Register** (0x8040F)

Bit	Register
EXT_FIFO_UDR_INT	**Extract Master Interrupt Status Register** (0x8040F)
EXT_SBI_PERR_INT	**Extract Master Interrupt Status Register** (0x8040F)
EXT_DC_INT	**Extract Master Interrupt Status Register** (0x8040F)

C1FP Realignment

EXT_C1FP_INT is set when a C1FP realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the **EXT_CTL** register (0x80400). This bit is cleared upon a read access.

SBIIP_SYNC Realignment

EXT_SYNC_INT is set when a SBIIP_SYNC realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the **EXT_CTL** register (0x80400). This bit is cleared upon a read access.

FIFO Overrun

EXT_FIFO_OVR_INT is set when a FIFO Overrun Interrupt is pending. This bit will not be set if FIFO_OVR_EN is low in the **EXT_CTL** register (0x80400).

The FIFO_OVRI bit of the **Extract FIFO Overrun Interrupt Status Register** (0x80402) is also set when a FIFO overrun is detected and is cleared upon a read access. Only one error can be reported at a time. However errors are latched internally so that if multiple errors occur, any pending errors will be reported when the first one is cleared.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Extract FIFO Overrun Interrupt Status Register** (0x80402) are used to specify which link of which Link Group was associated with the FIFO buffer in which the overrun was detected.

FIFO Underrun

EXT_FIFO_UDR_INT is set when a FIFO Underrun Interrupt is pending. This bit will not be set if FIFO_UDR_EN is low in the **EXT_CTL** register (0x80400).

The FIFO_UDRI bit of the **Extract FIFO Underrun Interrupt Status Register** (0x80401) is also set when a FIFO underrun is detected and is cleared upon a read access. Only one error can be reported at a time. However errors

are latched internally so that if multiple errors occur, any pending errors will be reported when the first one is cleared.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Extract FIFO Underrun Interrupt Status Register** (0x80401) are used to specify which link of which Link Group was associated with the FIFO buffer in which the underrun was detected.

SBI Parity Error

EXT_SBI_PERR_INT is set when a SBI Parity Error Interrupt is pending. This bit will not be set if SBI_PERR_EN is low in the **EXT_CTL** register (0x80400).

The PERRI bit of the **SBI Parity Error Interrupt Status Register** (0x80407) is also set when an SBI parity error has been detected and is cleared upon a read access.

The TRIB[4:0] and SPE[1:0] fields of the **SBI Parity Error Interrupt Status Register** (0x80407) are used to specify the SBI tributary for which a parity error was detected. These fields are only valid only when PERRI is set.

Depth Check Error

EXT_DC_INT is set when an Extract Depth Check Interrupt is pending. his bit will not be set if SYNC_INT_EN is low in the **EXT_CTL** register (0x80400).

The DCR_INTI bit of the **Extract Depth Check Interrupt Status Register** (0x8040E) is also set when a Depth Check error is detected and is cleared upon a read access. This error is detected when the internal FIFO pointers do match the expected internal FIFO depth.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Extract Depth Check Interrupt Status Register** (0x8040E) are used to specify which link of which Link Group was associated with Depth Check error. These fields are only valid when DCR_INTI is set.

8.4.4 Insert SBI Block Configuration

The INSBI maps up to 32 T1 or E1 links, or two DS3 links to the SBI shared bus. The T1/E1 links can be unframed or framed and channelized, with or without CAS support. The DS3 link can also be unframed or framed. The links which the INSBI processes can be mapped into any tributary in any SPE but all tributaries within an SPE must be all of the same type. Note that link/tributary numbering starts from '1' in SBI mode.

INSBI configuration is performed by programming bits within the following internal registers:

Table 14 – INSBI Block Register Memory Map

Offset	Register Description	Register Mnemonic
These registers are selected when the address = 0x805XX.		
0x00	Insert Control Register	INS_CTL
0x01	Insert FIFO Underrun Interrupt Status Register	INS_FI_URI
0x02	Insert FIFO Overrun Interrupt Status Register	INS_FI_ORI
0x03	Insert Tributary Register Indirect Access Address Register	INS_TRIAD
0x04	Insert Tributary Register Indirect Access Control Register	INS_TRIAC
0x05	Insert Tributary Mapping Register Indirect Access Data Register	INS_TRIB_MAP
0x06	Insert Tributary Control Register Indirect Access Data Register	INS_TRIB_CTL
0x07	MIN_DEPTH for T1 and E1 Register	INS_MD_T1E1
0x08	MIN_DEPTH for DS3 Register	INS_MD_DS3
0x09	MIN_THR and MAX_THR for T1 Register	INS_THR_T1
0x0A	MIN_THR and MAX_THR for E1 Register	INS_THR_E1
0x0B	MIN_THR and MAX_THR for DS3 Register	INS_THR_DS3
0x0C	Reserved	
0x0D	Reserved	
0x0E	Reserved	
0x0F	Reserved	
0x10	Reserved	
0x11	Depth Check Reset Interrupt Status Register	INS_DCR_INT
0x12	Insert Master Interrupt Register	INS_MSTR_INT

8.4.4.1 SBI Insert Control

The SBI Insert block is controlled by programming bits within the **INS_CTL** (0x80500) register. The default configuration is as follows:

Bit	Register	Value
SBI_PAR_CTL	**Insert Control Register** (0x80500)	1
TS_EN	**Insert Control Register** (0x80500)	0
FIFO_UDR_EN	**Insert Control Register** (0x80500)	0
FIFO_OVR_EN	**Insert Control Register** (0x80500)	0
SYNC_INT_EN	**Insert Control Register** (0x80500)	0
DC_EN	**Insert Control Register** (0x80500)	1
APAGE	**Insert Control Register** (0x80500)	0

The default indicates that odd parity mode is used for checking the SBI parity signal, all INSBI interrupts are disabled, tributary to link mapping is disabled, depth check logic is enabled, and page 0 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator-32 links.

SBI Parity Mode

The SBI_PAR_CTL bit is used to configure the Parity mode for generation of the SBI data parity signal, ADP, as follows:

SBI_PAR_CTL	Function
0	Even parity generation.
1	Odd parity generation.

One to One Tributary to Link Mapping

The TS_EN bit is used to enable the SBI tributary to AAL1gator link mapping capability when the TS_EN bit in the SBI_BUS_CFG_REG also enables mapping.

TS_EN	Function
0	Mapping will be fixed to a one to one mapping and will not be programmable.

TS_EN	Function
1	SBI tributary to AAL1gator link mapping is enabled and is specified by the contents of the Insert Tributary Mapping Register RAM when TS_EN in SBI_BUS_CFG_REG is also set.

Insert SBI Interrupt Enables

There are three interrupt enable bits which are used to enable the generation of an interrupt or set of interrupts when the corresponding interrupt condition is detected. When the interrupt enable bit is '0', the corresponding interrupt is disabled. When the interrupt enable bit is '1', the corresponding interrupt is enabled.

Interrupt Enable Bit	Interrupt Condition
FIFO_UDR_EN	FIFO underrun
FIFO_OVR_EN	FIFO overrun
SYNC_INT_EN	Depth Check sync error, C1FP sync error, or SBIIP sync error

Depth Check Logic

This bit enables the Depth Check Logic and is encoded as follows:

DC_EN	Function
0	Depth checker logic is disabled.
1	The depth checker logic will periodically monitor the Data/Framing FIFO Depth and compare it against the write and read pointers. If there is a discrepancy the tributary is synchronously reset by the depth checker.

Note: When in synchronous mode, the DC_EN bit must be clear.

Active Page Selection

The tributary mapping and control configuration RAMs active page select bit (APAGE) controls the selection of one of two pages in the tributary mapping and control configuration RAMs to be the active page. Changes of the active page

as a result of write accesses to APAGE will be synchronized to SBI multi-frame boundaries at C1FP.

APAGE	Function
0	The configuration in page 0 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator links.
1	The configuration in page 1 of the tributary mapping and control configuration RAMs is used to associate incoming tributaries to AAL1gator links.

8.4.4.2 SBI Insert Tributary Configuration

SBI Insert tributary configuration information is read from and written to the SBI Insert tributary mapping and control RAM.

Please note that the BUSY bit in the **Extract Tributary RAM Indirect Access Control Register** (0x80404) might not be cleared for up to 4.32 µs after either a mapping page switch or a Control RAM or Mapping RAM access.

8.4.4.2.1 SBI Insert Tributary Mapping Configuration

The default tributary to link mapping is straight through i.e. 1:1. Therefore, Link Group LOW, LINK 1 inside the AAL1gator-32 will be mapped by default to SPE1, LINK 1 on the SBI side and so on up to SPE1, LINK 16. Link Group HIGH, LINK 1 inside the AAL1gator will be mapped by default to SPE2, LINK 1 on the SBI side and so on up to SPE2, LINK 16. Any tributary can be mapped to any link, unless the tributary is configured for DS3. In that case the tributary can only be mapped to link 1 or link 17. Please note that link numbering starts at '1'.

SBI Insert mapping information is read from and written to the SBI Insert tributary mapping RAM. The tributary to link mapping for an SBI tributary in the transmit direction is configured using the following procedure:

1. Poll the BUSY bit of the **Insert Tributary RAM Indirect Access Control Register** (0x80504) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. Specify the tributary to which the mapping register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below
SPE[1:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below
MAP_REG	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	1

The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the mapping register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be set to specify that the Insert Tributary Mapping Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

3. Specify the tributary to link mapping to be written to the tributary mapping RAM by writing the following register:

Bit	Register	Value
LINK[4:0]	**Insert Tributary Mapping RAM Indirect Access Data Register** (0x80505)	See below
LINK_GRP_LOW	**Insert Tributary Mapping RAM Indirect Access Data Register** (0x80505)	See below
LINK_GRP_HIGH	**Insert Tributary Mapping RAM Indirect Access Data Register** (0x80505)	See below

The LINK[4:0] and LINK_GRP_LOW/HIGH fields are used to specify which link of which link group within the AAL1gator is mapped to the SBI tributary specified in step 2. LINK_GRP_LOW/HIGH specifies the link group while LINK[4:0] specifies the link number. Legal values for LINK[4:0] are b"00001" through b"10000".

4. Trigger an indirect write operation on the tributary mapping RAM by writing the following register:

Bit	Register	Value
PAGE	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	See below

Bit	Register	Value
RWB	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	0

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary mapping configuration RAM. Note that the BUSY and HST_ADDR_ERR bits in the **Insert Tributary RAM Indirect Access Control Register** (0x80504) are read only and are not affected by write operations.

Notes:

- Following a configuration change, which generates a Configuration Reset, a tributary may not become active for up to 1 ms following the change.
- Mapping of more than one tributary to a link or more than one link to a tributary is not allowed.
- Whether tributary mapping is performed or default 1:1 mapping is used, the above steps must be done and must be repeated for every tributary access.

8.4.4.2.2 SBI Insert Tributary Control Configuration

SBI Insert tributary control information is read from and written to the SBI Insert tributary control RAM.

The default configuration of each tributary is as follows:

Bit	Register	Value
TRIB_ENBL	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	0
TRIB_TYP[1:0]	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	00
CLK_MSTR	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	0
SYNCH_TRIB	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	0

In the default state, the tributary is disabled, the tributary type is Structured with CAS, the tributary is a clock slave, and the tributary is free to float (i.e. – is in asynchronous mode).

The tributary control information for an SBI tributary in the receive direction is configured using the following procedure:

1. Poll the BUSY bit of the **Insert Tributary RAM Indirect Access Control Register** (0x80504) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. Specify the tributary to which the control register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below
SPE[1:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below
MAP_REG	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	0

The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the control register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be cleared to specify that the Insert Tributary Control Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

3. Specify the configuration data to be written to the tributary control RAM by writing the following register:

Bit	Register	Value
TRIB_ENBL	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	See below
TRIB_TYP[1:0]	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	See below
CLK_MSTR	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	See below
SYNCH_TRIB	**Insert Tributary Control RAM Indirect Access Data Register** (0x80506)	See below

The TRIB_ENBL bit is used to enable the tributary. Writing to the tributary control RAM with the TRIB_ENBL bit set enables the INSBI to take tributary

data from an AAL1gator link and transmit that data to the SBI tributary mapped to that link.

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in the table below. TRIB_TYP should correspond with the FR_STRUCT setting in the LIN_STR_MODE memory register for the link associated with this tributary in the AAL1gator-32 line setup.

CAS Enabled	Framed	Transparent VT (Floating)	TRIB_TYP	FR_STRUCT	Description
True	True	False	00	11	Structured with CAS
False	True	False	01	01	Structured without CAS
False	False	False	10	10	Unstructured
False	False	True	11		Transparent VT (not supported)

Notes:

- CAS can only be enabled for a Structured (framed) tributary.
- "Framed" means framing information available – may be channelized or unchannelized.
- "Unframed" means no framing information available.
- If a mixture of CAS and non-CAS connections are being made on the same line, then put the line in Structured with CAS mode and set R_CHAN_NO_SIG and T_CHAN_NO_SIG in the queue tables for the connections not carrying CAS.

The CLK_MSTR bit is used to specify whether the tributary is a clock master or a clock slave. Setting CLK_MSTR will configure the tributary to be a clock master and clearing CLK_MSTR will configure the tributary to be clock slave. The CLK_MSTR configuration bits are ORed with the CLK_MSTR bit in the SBI_BUS_CFG_REG to allow the chip level to force master mode for all tributaries.

The SYNCH_TRIB bit is used to indicate whether the tributary is locked to the SBI SPE (i.e. – is in synchronous mode). If this bit is set than the tributary is

locked. If this bit is not set, then the tributary is free to float (i.e. – is in asynchronous mode).

4. Trigger an indirect write operation on the tributary mapping RAM by writing the following register:

Bit	Register	Value
PAGE	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	See below
RWB	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	0

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary control configuration RAM. Note that the BUSY and HST_ADDR_ERR bits in the **Insert Tributary RAM Indirect Access Control Register** (0x80504) are read only and are not affected by write operations.

Notes:

- Following a configuration change, which generates a Configuration Reset, a tributary may not become active for up to 1 ms following the change.
- Any write to a Tributary Control register for a tributary will generate a configuration reset on that tributary, irrespective of whether the data written to the tributary control register is unchanged from the previous value.

8.4.4.2.3 SBI Insert Tributary Configuration Read Procedure

The SBI tributary mapping and control configuration in the receive direction can be read using the following procedure:

1. Poll the BUSY bit of the **Insert Tributary RAM Indirect Access Control Register** (0x80504) until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.
2. Specify the tributary to which the mapping register operation will apply to by writing the following register:

Bit	Register	Value
TRIB[4:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below

Bit	Register	Value
SPE[1:0]	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below
MAP_REG	**Insert Tributary RAM Indirect Access Address Register** (0x80503)	See below

The TRIB[4:0] and SPE[1:0] fields are used to specify which SBI tributary the mapping register write operation will apply to. Legal values for TRIB[4:0] are b"00001" through b"11100". Legal values for SPE[1:0] are b"01" through b"11". The MAP_REG bit needs to be cleared to specify that the Insert Tributary Control Registers are addressed and needs to be set to specify that the Insert Tributary Mapping Registers are addressed.

3. Trigger an indirect read operation on the tributary mapping or control RAM by writing the following register:

Bit	Register	Value
PAGE	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	See below
RWB	**Insert Tributary RAM Indirect Access Control Register** (0x80504)	1

The PAGE bit selects between the two pages (page 0 or page 1) in the tributary mapping or control configuration RAMs. Note that the BUSY and HST_ADDR_ERR bits in the **Insert Tributary RAM Indirect Access Control Register** (0x80504) are read only and are not affected by write operations.

4. Poll the BUSY bit of the **Insert Tributary RAM Indirect Access Control Register** (0x80504) until it is zero. When BUSY is polled to be zero, data from an indirect read operation is available in either the **Insert Tributary Mapping RAM Indirect Access Data Register** (0x80505) or the **Insert Tributary Control RAM Indirect Access Data Register** (0x80506) depending on the value of MAP_REG from step 2.

5. Read the HST_ADDR_ERR bit of the **Insert Tributary RAM Indirect Access Control Register** (0x80504). A value of HST_ADDR_ERR = '1' indicates that an illegal host access was attempted. An illegal host access occurs when an attempt is made to access an out of range tributary. This bit is cleared upon a read access.

8.4.4.3 Depth and Threshold Values

The following fields are used to modify the minimum depth (MIN_DEP), minimum threshold (MIN_THR) and maximum threshold (MAX_THR) for T1, E1, and DS3 tributaries. In most cases, these values should remain at their default settings. However, for T1/E1 applications, where the AAL1gator-32 is a clock master, the MAX_THR value should be changed to 0xA.

Bits	Register	Default
MIN_DEP_T1[3:0]	**MIN_DEPTH for T1 and E1 Register** (0x80507)	0x7
MIN_DEP_E1[3:0]	**MIN_DEPTH for T1 and E1 Register** (0x80507)	0x7
MIN_DEP_DS3[3:0]	**MIN_DEPTH for DS3 Register** (0x80508)	0xC
Reserved[3:0]	**MIN_DEPTH for DS3 Register** (0x80508)	0x0
MAX_THR_T1[3:0]	**T1 Threshold Register** (0x80509)	0xE
MIN_THR_T1[3:0]	**T1 Threshold Register** (0x80509)	0x6
MAX_THR_E1[3:0]	**E1 Threshold Register** (0x8050A)	0xE
MIN_THR_E1[3:0]	**E1 Threshold Register** (0x8050A)	0x2
MAX_THR_DS3[3:0]	**DS3 Threshold Register** (0x8050B)	0x6
MIN_THR_DS3[3:0]	**DS3 Threshold Register** (0x8050B)	0x8

8.4.4.4 Insert SBI Interrupts

The following master interrupt status bits should be read to detect interrupt conditions in the Insert SBI block:

Bit	Register
EXT_C1FP_INT	**Insert Master Interrupt Status Register** (0x80512)
EXT_SYNC_INT	**Insert Master Interrupt Status Register** (0x80512)
EXT_FIFO_OVR_INT	**Insert Master Interrupt Status Register** (0x80512)
EXT_FIFO_UDR_INT	**Insert Master Interrupt Status Register** (0x80512)
EXT_DC_INT	**Insert Master Interrupt Status Register** (0x80512)

C1FP Realignment

INS_C1FP_INT is set when a C1FP realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the **INS_CTL** register (0x80500). This bit is cleared upon a read access.

SBIIP_SYNC Realignment

INS_SYNC_INT is set when a SBIIP_SYNC realignment has been detected. This is an internal bus error. This bit will not be set if SYNC_INT_EN is low in the **INS_CTL** register (0x80500). This bit is cleared upon a read access.

FIFO Overrun

INS_FIFO_OVR_INT is set when a FIFO Overrun Interrupt is pending. This bit will not be set if FIFO_OVR_EN is low in the **INS_CTL** register (0x80500).

The FIFO_OVRI bit of the **Insert FIFO Overrun Interrupt Status Register** (0x80502) is also set when a FIFO overrun is detected and is cleared upon a read access. Only one error can be reported at a time. However errors are latched internally so that if multiple errors occur, any pending errors will be reported when the first one is cleared.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Insert FIFO Overrun Interrupt Status Register** (0x80502) are used to specify which link of which Link Group was associated with the FIFO buffer in which the overrun was detected.

FIFO Underrun

INS_FIFO_UDR_INT is set when a FIFO Underrun Interrupt is pending. This bit will not be set if FIFO_UDR_EN is low in the **INS_CTL** register (0x80500).

The FIFO_UDRI bit of the **Insert FIFO Underrun Interrupt Status Register** (0x80501) is also set when a FIFO underrun is detected and is cleared upon a read access. Only one error can be reported at a time. However errors are latched internally so that if multiple errors occur, any pending errors will be reported when the first one is cleared.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Insert FIFO Underrun Interrupt Status Register** (0x80501) are used to specify which link of which Link Group was associated with the FIFO buffer in which the underrun was detected.

Depth Check Error

INS_DC_INT is set when an Insert Depth Check Interrupt is pending. This bit will not be set if SYNC_INT_EN is low in the **INS_CTL** register (0x80500).

The DCR_INTI bit of the **Insert Depth Check Interrupt Status Register** (0x80511) is also set when a Depth Check error is detected and is cleared upon a read access. This error is detected when the internal FIFO pointers do match the expected internal FIFO depth.

The LINK[4:0] and LINK_GRP_LOW/HIGH fields of the **Insert Depth Check Interrupt Status Register** (0x80511) are used to specify which link of which Link Group was associated with Depth Check error. These fields are only valid when DCR_INTI is set.

8.4.5 Line Clock Source

Several clocking options exist in this mode and are controlled by the value of the CLK_SOURCE bits in the LIN_STR_MODE register for each line. The CLK_SOURCE bits must be in a compatible setting with respect to the CLK_MSTR setting discussed in sections 8.4.4.2.2 and 8.4.3.2.2.

8.4.5.1 Receive Line Clock Source

In the receive direction, the line clock source has two options based on the value of the CLK_SOURCE_RX bit in LIN_STR_MODE. This bit should only be set when supporting DS3 tributaries.

CLK_SOURCE_RX	Function
0	The line receives its clock from the external clock (RL_CLK[n] pin) associated with that line.
1	The line receives its clock from the SBI.

8.4.5.2 Transmit Line Clock Source

In the transmit direction, seven possible options exist and are controlled by the value of the CLK_SOURCE_TX bits in the LIN_STR_MODE memory register for each line. This value will override the setting defined by the TLCLK_OE input pin. If switching from an external to an internal clock or visa versa, make sure there are not two clocks driving simultaneously.

The seven options are listed in the following table. All options but the last require that the AAL1gator is the SBI clock master for that line in the Add direction.

CLK_SOURCE_TX[2:0]	Function
000	Clock is an input on pin TL_CLK[n].
001	Clock is an input on pin RL_CLK[n] (loop timing mode).
010	Clock is internally synthesized as a nominal T1 or E1 clock based on SYS_CLK and the value of T1_MODE. The clock is output on the TL_CLK[n] pin.
011	Clock is internally synthesized based on the received SRTS values. The clock is output on TL_CLK[n] pin.
100	Clock is internally synthesized using the adaptive algorithm which uses receive buffer depth to control TL_CLK[n]. The clock is output on TL_CLK[n] pin.
101	Clock is internally synthesized based on values received on the external clock control interface. This mode is used for external implementations of SRTS or Adaptive clocking. The clock is output on TL_CLK[n] pin.
110	Not Valid in SBI mode
111	SBI is clock master, use SBI clock.

Please see 8.3.2.2.1 for a description of the CGC block which implements internally synthesized clocks. All configurable features in that section are applicable for lines conveyed via SBI with CLK_SOURCE_TX options “010” through “101”.

8.4.6 Other Per-Line Options

Low CDV for Unstructured Mode

For unstructured lines conveyed via SBI, the LOW_CDV bit of the LIN_STR_MODE memory register can be set to cause cells to be scheduled every 47 bytes instead of using frame based scheduling. This eliminates the CDV caused by the scheduler. This mode can only be used in unstuctured mode when BYTES_PER_CELL is 47. This mode cannot be used when the queue is configured for partial cells or for AAL0 mode.

LOW_CDV	Function
0	Unstructured line is not in low CDV mode. AAL1gator uses frame based scheduling.
1	Unstructured line is in low CDV mode. AAL1gator uses byte based scheduling where cells are scheduled every 47 bytes.

8.5 H-MVIP Mode

The High Speed Multi-Vendor Integration Protocol (H-MVIP) bus provides synchronous, time-division multiplexed transport of N x 64 kbps constant bit rate (CBR) data streams. The H-MVIP bus consists of a group of related clock signals and serial data streams which operate at 8 Mbps.

In H-MVIP mode, each incoming external 8 Mbps H-MVIP data stream and breaks it into 4 separate local 2Mbps data streams. The bytes are taken off the bus in round robin fashion and sent to separate 2Mbps links. In the outgoing direction the H-MVIP block takes each group of four 2Mbps local links and combines them into one external 8 Mbps H-MVIP data stream.

In H-MVIP mode, there is a common 16 MHz clock, a 4 MHz clock, and a common framing reference signal. Separate data pins and signaling pins are provided in each direction for each line. Individual time slots cannot be disabled.

DS1 / E1 Links

The AAL1gator-32, AAL1gator-8 and AAL1gator-4 support eight, two and one H-MVIP line, respectively, where each H-MVIP line supports four DS1/E1 links each. Each link can be individually configured for either Basic Service (no CAS) or Service with CAS.

Structured J2 Links

By mapping a J2 frame into an H-MVIP frame, the AAL1gator can circuit emulate individual N x 64 kbps timeslots or groups of timeslots. Note that because the H-MVIP data stream is internally divided into 4 separate 2 Mbps streams, groups of timeslots are required to be within a 2 Mbps stream boundary of thirty-two timeslots. The AAL1gator-32, AAL1gator-8 and AAL1gator-4 support eight, two and one Structured J2 Basic Service links respectively.

8.5.1 Line Format and Frame Structure

For H-MVIP mode, the T1_MODE bit in the LIN_STR_MODE register and the MVIP_EN bit in the LS_Ln_CFG_REG should both be low.

The frame structure for H-MVIP lines can be Structured Data Format – Frame (SDF-FR) or Structured Data Format – Multiframe (SDF-MF) and is determined by the value of FR_STRUCT[1:0] in the LIN_STR_MODE memory register for each line as follows:

FR_STRUCT[1:0]	Frame Structure	Description
00		Not Used.
01	SDF-FR	A structured connection where CAS signaling is not being transported (basic service).
10		Not valid in H-MVIP mode.
11	SDF-MF	A structured connection where CAS signaling is being transported.

Note: If a mixture of CAS and non-CAS connections are being made on the same line, then put the line in SDF-MF mode and set R_CHAN_NO_SIG and T_CHAN_NO_SIG in the queue tables for the connections not carrying CAS.

8.5.2 Line Clock Source

In H-MVIP mode there is a common 16 Mhz clock (HMVIP16CLK) whose every other rising edge is used to sample data on all external lines in the receive direction and whose every other falling edge is used to source data on all external lines in the transmit direction. There is also a common 4 Mhz clock (HMVIP4CLK) whose falling edge is used to sample the frame pulse. Internally a 2 Mhz clock is generated for each link which connects to the A1SP block(s).

Because H-MVIP lines all run off the same clock, there is only one clocking option available in this mode. Therefore the CLK_SOURCE values are not used. The HMVIP16CLK and HMVIP4CLK will always be used and a clock will be expected on these pins. SRTS_EN should stay low in this mode.

8.5.3 Synchronization

A common frame pulse, F0B, is used to indicate the start of a frame for all external lines. Individual local link frame pulses are derived from F0B. This signal is always an input.

In H-MVIP mode, synchronization is always controlled from the external interface and the sync signal is always considered to be a frame synchronization signal. Therefore MF_SYNC_MODE and GEN_SYNC should be inactive (stay low) for all lines when this mode is in use.

8.5.4 CAS Signaling

Signaling is passed through as received and sent with the corresponding data byte. Signaling format is the same as in Direct Low Speed mode, where the last nibble of each timeslot carries the CAS signaling bits. The E1_WITH_T1_SIG bit in LIN_STR_MODE can be configured in H-MVIP mode. See section 8.3.4 for a detailed discussion of signaling. E1_WITH_T1_SIG is encoded as follows:

E1_WITH_T1_SIG	Function
0	Use E1 signaling. Signaling is updated every 16 frames.
1	Use T1 signaling. Signaling is updated every 24 frames instead of every 16 frames.

8.6 High Speed Mode

The High Speed Line Interface mode is used to interface to high speed clear channel, unstructured data streams. The Line Interface Block just passes the clock and data between the external lines and the corresponding local link. Only clock and data are used. The data is passed as a clear channel bit stream and data rates are supported up to 45 Mbps.

Timing recovery is supported externally. The AAL1gator's External Clock Interface outputs various signals including SRTS and Adaptive Status values.

DS3 / E3 / STS-1 / STM-0 Links

In High Speed Mode, the AAL1gator-32 supports two DS3 , E3 or STS-1/STM-0 links where external lines 0, and 2 are connected to local links 0 and 16 respectively.

In High Speed Mode, the AAL1gator-8 and AAL1gator-4 support one DS3 , E3 or STS-1/STM-0 link where external line 0 is connected to local link 0.

8.6.1 High Speed Line Configuration

The configuration of the high speed lines is stored in the HS_LIN_REG memory register for the corresponding A1SP. All settings are only applicable to line 0 of the A1SP. For AAL1gator-32, A1SP 0 and A1SP 2 are used for the high speed lines. For AAL1gator-8 and AAL1gator-4, A1SP 0 is used for the high speed line.

HS_LIN_REG has the following bit format. Please note that writes to this register will not take into effect until the An_CMDREG_ATTN bit is set in the An_CMD_REG.

Table 15 – HS_LIN_REG Format

Bit	Field	Description
15:14	Reserved	Initialize to 0.
13:6	Not used	Write with a 0 to maintain future software compatibility.
5	HS_GEN_DS3_AIS	See section 6.6.1.1, Transmit Conditioning.
4	HS_TX_COND	See section 6.6.1.1, Transmit Conditioning.
3	HS_RX_COND	Fetch receive conditioning data from the R_COND_DATA buffer for line 0, channels 0 and 1. High-speed mode (line 0 only).
2	UDF_HS	See discussion below.
1:0	Unused	Write with a 0 to maintain future software compatibility.

Line 0 of the A1SP is configured for Unstructured Data Format – High Speed (UDF-HS) mode by setting the UDF_HS bit in HS_LIN_REG. For AAL1gator-32, A1SP0 and A1SP2 need to be configured for UDF-HS mode. For AAL1gator-8 and AAL1gator-4, A1SP 0 needs to be configured for UDF-HS mode.

UDF_HS	Function
0	Disables the UDF-HS (T3/E3/STS-1/STM-0) mode.
1	Enables the UDF-HS (T3/E3/STS-1/STM-0) mode. If this mode is selected, the T_QUEUE_TBL and R_QUEUE_TBL entry index 0 are used.

8.6.1.1 Transmit Cell Conditioning

Under certain alarm conditions such as Loss of Signal (LOS), an Alarm Indication Signal (AIS) needs to be transmitted downstream in the cell transmit direction. This means that cells need to be generated which carry an AIS pattern (conditioned data).

For E3 links, this can be done by setting the HS_TX_COND bit in the HS_LIN_REG register. When this bit is set, cells with an all ones pattern will be generated.

However, a DS3 AIS signal is a framed "1010" pattern. This signal can be generated by setting the HS_GEN_DS3_AIS bit in the HS_LIN_REG register.

8.6.1.2 Receive Cell Conditioning

In High Speed Mode, the HS_RX_COND bit in HS_LIN_REG is set to force the playing out of conditioned data. Also, when no data is present in the VC receive buffer, the AAL1gator declares an underrun condition and automatically plays out conditioned data.

In High Speed Mode, the T1_MODE bit in LIN_STR_MODE is used to determine the type of conditioned data being played out in the above two situations:

T1_MODE	Function
0	The conditioned data from the R_COND_DATA buffer for line 0, channels 0 and 1 is played out.
1	The framed DS3 AIS pattern is played out.

Note: There is no old data or pseudorandom data options available for UDF-HS mode.

8.6.2 Line Clock Source

8.6.2.1 Receive Line Clock Source

In the receive direction, the line clock source must be the RL_CLK[n] input pin associated with the high speed line. Therefore, CLK_SOURCE_RX must be set to '0'.

8.6.2.2 Transmit Line Clock Source

In the transmit direction, two possible options exist and are controlled by the value of the CLK_SOURCE_TX bits in the LIN_STR_MODE memory register for each line. The options are listed below:

CLK_SOURCE_TX[2:0]	Function
000	Clock is an input on pin TL_CLK[n].
001	Clock is an input on pin RL_CLK[n] (loop timing mode).

8.6.2.2.1 Synchronous Residual Time Stamp

SRTS functionality is enabled by setting the SRTS_EN bit in the LIN_STR_MODE memory register. SRTS is supported for unstructured data formats on a per-line basis.

SRTS_EN	Function
0	SRTS disabled. The CSI bits of the odd transmit AAL1 cells are set to '0' and the received SRTS bits are ignored.
1	SRTS enabled. The insertion of the transmit SRTS bits is enabled for this line and the received SRTS bits are accumulated.

8.6.3 Other Per-Line Options

By default, a high speed queue is always configured for low CDV and the LOW_CDV bit of the LIN_STR_MODE is ignored.

9 CONFIGURING THE A1SP BLOCKS

The following internal registers are used to configure the A1SP blocks. Please note that only the registers for A1SP 0 are used in the AAL1gator-8 and AAL1gator-4.

9.1 Sending OAM Cells

The A*n*_SEND_OAM bits in the **A1SP*n* Command Register** (0x80010, … 13) are used to send OAM cells in the TX OAM buffer as follows:

Bit	Function	Description
2	A*n*_SEND_OAM_1	A write of 1 causes the cell in the TX OAM buffer 1 for the corresponding A1SP to be sent. Reads as a 0 when the cell has been sent.
1	A*n*_SEND_OAM_0	A write of 1 causes the cell in the TX OAM buffer 0 for the corresponding A1SP to be sent. Reads as a 0 when the cell has been sent.

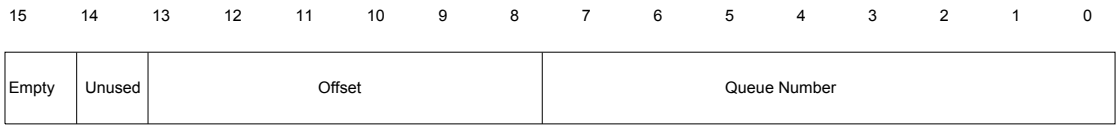
9.2 Adding Queues

In order to add a queue the processor has to write the ADDQ_FIFO. The ADDQ_FIFO consists of 64 16-bit entries and is accessed using a single address. The format of the ADDQ_FIFO word is shown in Figure 20. The first byte specifies the number of the queue to be added. The next six bits represent an offset that is used to spread the scheduling of cells across multiple frames. This helps to avoid the problem of clumping, which refers to contention with other cells scheduled during the same frame.

The upper bit indicates whether or not the ADDQ_FIFO is empty. This bit can be polled after adding queues to find out when they all have been added. The Empty bit indicates Empty status when it is set. The amount of time it takes to empty the FIFO is dependent on how full it is, whether TALP is processing cells and whether there is back pressure on the UTOPIA bus. ADDQ_FIFO entries can only be processed when TALP is idle. If the TALP_FIFO fills and prevents TALP from processing cells, this will prevent ADDQ_FIFO entries from being processed.

Note that the Offset and Queue Number fields are write only and cannot be read. The Empty field is read only and cannot be written.

Figure 20 – ADDQ_FIFO Word Structure



Queues are added by writing to the **A*n*_ADDQ_FIFO** (0x80020, …, 23) register with the number of the queue to be added. The fields of the A*n*_ADDQ_FIFO register is summarized below:

Bit	Function	Description
15	EMPTY	This field indicates when the Add Queue FIFO is empty. It can be polled to determine when the A1SP module has finished processing the ADDQ_FIFO entries that were in the FIFO.
13:8	OFFSET	This field indicates the offset from the cell scheduling reference value. This offset can be used to spread the scheduling of cells across multiple frames in order to control clumping Note that this field is invalid on reads.
7:0	QUEUE_NUM	The number of the queue being added. Note that this field is invalid on reads.

9.3 A1SP Clock Configuration

The **A1SP*n* Clock Configuration Register** (0x80030, … 33) is used to configure the internal adaptive clocking algorithm and the dividing down of the network-derived clock, NCLK, used for SRTS as follows:

Bit	Function	Description
9	NCLK_DIV_EN	When set, the NCLK is divided down as specified by the NCLK_DIV field. Otherwise the NCLK is passed directly from the NCLK pin to this A1SP.
8:5	NCLK_DIV	When NCLK_DIV_EN is set, this field indicates how much to divide down the NCLK for this A1SP. The clock is divided by ((NCLK_DIV + 1) *2). This field is "0000" by default so the NCLK is divided by 2. If this field is "1111" then NCLK is divided by 32.

Bit	Function	Description
4:0	ADAP_FILT_ SIZE	When a line is configured to use the internal adaptive clocking algorithm, this field defines the size of the filtering window. The filter size is a power of 2 where ADAP_FILT_SIZE represents the exponent (i.e. $2^{ADAP\_FILT\_SIZE}$). The adaptive algorithm determines the clock frequency by averaging the byte difference over $2^{ADAP\_FILT\_SIZE}$ number of samples. The maximum value is "10000" or 16. The default value is 2^{00000B} = 1.

10 CONFIGURING THE RAM INTERFACE

The AAL1gator has a separate SRAM and processor interface. The RAM Interface is the central arbiter for all memory accesses. It provides a priority mechanism that incorporates fairness to satisfy all real-time requirements of the various blocks. All blocks requesting a data transfer with the common memory supply the address, control signals, and the data, if the requested data transfer is a write, to the RAMI. When the RAMI actually grants the transfer, it provides a grant signal to the requesting block, indicating that the transfer has been performed. The memory is arbitrated on a cycle-by-cycle basis. No device is granted the bus for an indefinite time.

For the AAL1gator-32, either one or two external 256K x 16 or 256K x 18 (10 ns) SRAMs are needed, depending on the mode of operation. Two rams are required if the AAL1gator-32 is configured to use the SBI interface, the upper four 8 Mbps MVIP interfaces, or the upper two High speed interfaces.

For the AAL1gator-8 and the AAL1gator-4, one external 128K x 16 or 128K x 18 (10 ns) SRAM is needed.

The SRAM interface runs at the same frequency as SYS_CLK and can run up to 45 Mhz.

The RAM interface is configured in the **RAM Configuration Register** (0x80100). The default configuration indicates use of pipelined SSRAM protocol and odd parity generation/checking for the RAM interfaces:

Bit	Register	Value
SSRAM_ZBT_MODE	**RAM Configuration Register** (0x80100)	0
RAM_EVEN_PAR	**RAM Configuration Register** (0x80100)	0

SRAM Selection

Either a synchronous SRAM with a single cycle deselect or a pipelined ZBT or ZBT compatible SRAM can be used. For most applications the synchronous SRAM is sufficient, but if additional performance is needed, such as in cross connect applications which need to use partial cells to lower delay, the ZBT ram is recommended. The SSRAM_ZBT_MODE bit selects between SSRAM and ZBT SRAM:

SSRAM_ZBT_MODE	Function
0	The pipelined SSRAM protocol is used on the RAM interfaces allowing glueless connection to pipelined SSRAMs.
1	The ZBT SRAM protocol is used allowing glueless connection to pipelined ZBT SRAMS.

Parity Selection

The RAM_EVEN_PAR bit selects even or odd parity for the RAM I/F as follows:

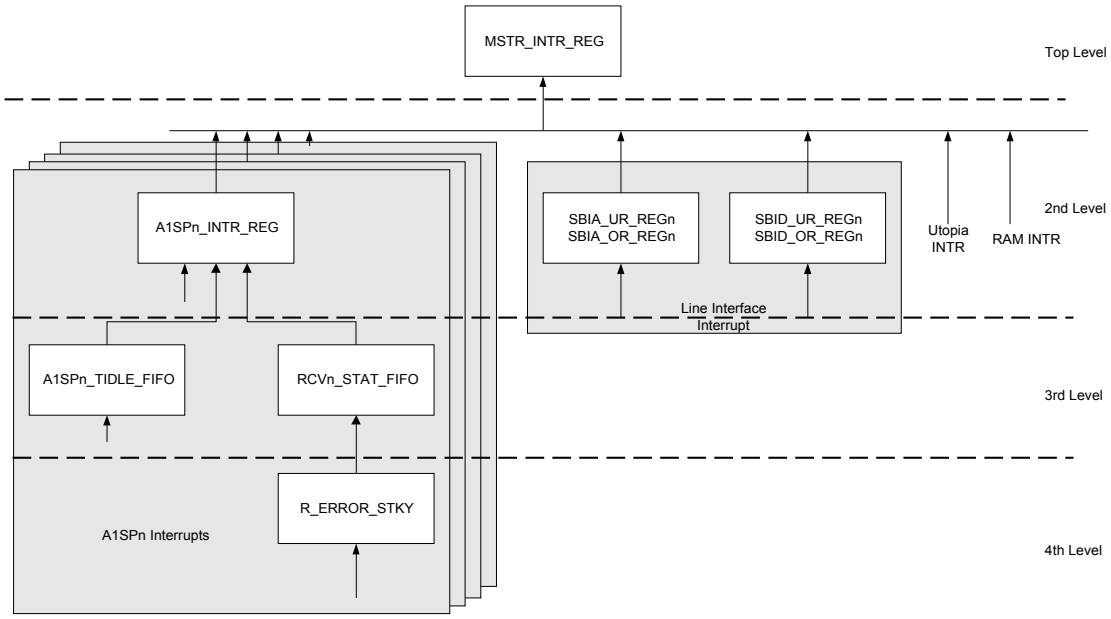
RAM_EVEN_PAR	Function
0	Odd parity is generated and checked.
1	Even parity is generated and checked.

11 INTERRUPTS

The interrupt logic has several layers and can be sourced from any of eight blocks. The eight blocks are the UTOPIA block, the two RAM interface blocks, the four or one A1SP blocks and the Line Interface block. The top layer of the interrupt logic, the **Master Interrupt Register** (0x80000), indicates from which block the interrupt came from. Once the block is determined the processor can access the appropriate block to determine the interrupt cause.

Figure 21 shows the registers in the interrupt tree. The microprocessor traverses the tree based on the value of individuals bits within each register.

Figure 21 – Interrupt Hierarchy



Interrupt and Status Registers Memory Map

These registers indicate the current status of the device and any conditions that might require processor attention:

Address	Register Description	Register Mnemonic
0x81000	Master Interrupt Register	MSTR_INTR_REG

Address	Register Description	Register Mnemonic
0x81010,…,13	A1SP*n* Interrupt Register	A1SP*n*_INTR_REG
0x81020,…,23	A1SP*n* Status Register	A1SP*n*_STAT_REG
0x81030,…,33	A1SP*n* Transmit Idle State FIFO	A1SP*n*_TIDLE_FIFO
0x81040,…,43	A1SP*n* Receive Status FIFO	A1SP*n*_RSTAT_FIFO
0x81100	Master Interrupt Enable Register	MSTR_INTR_EN_REG
0x81110,…,13	A1SP*n* Interrupt Enable Register	A1SP*n*_INTR_EN
0x81140,…,43	A1SP*n* Receive Status FIFO Enable Register	A1SP*n*_RSTAT_EN
0x81150,…,53	A1SP*n* Receive Queue Error Enable	A1SP*n*_RCV_Q_ERR_EN

Note: In the AAL1gator-8 and AAL1gator-4, only the A1SP0 registers are used.

11.1 Master Interrupts

The **Master Interrupt Register** (0x81000) is at the top of the Interrupt Tree. It indicates which lower level interrupt registers have interrupts pending. The UTOPIA Interface error bits and RAM parity error bits are cleared on read, the other bits are current status and will remain set as long as the underlying condition remains active. For bits which are cleared on read, a logic '0' indicates that no further interrupts of this type has been encountered while a logic '1' indicates that another such interrupt has been encountered. For bits which indicate current status, a logic '0' indicates that no interrupt is pending while a logic '1' indicates that the interrupt is still pending.

The enable bits in the **Master Interrupt Enable Register** (0x81100) control the corresponding interrupt bits in the **Master Interrupt Register** (0x81000). When an enable bit in the **Master Interrupt Enable Register** (0x81100) is set to a logic '1', the corresponding error event from the **Master Interrupt Register** (0x81000) will cause the active low interrupt signal to the microprocessor, INTB, to go active.

The following interrupts are at the top of the interrupt tree and have status and enable bits in the **Master Interrupt Register** (0x81000) and **Master Interrupt Enable Register** (0x81100) respectively. A bit type of RO specifies an interrupt bit that indicates current status while a bit type of R2C specifies an interrupt bit that is cleared on read. The interrupt description column describes the condition when the interrupt bit is a logic '1'.

Bit (Position)	Type	Interrupt Description when Bit = '1'
A1SP0_INTR(0)	RO	Interrupt pending from A1SP0. Read A1SP0_INTR_REG to determine cause.
A1SP1_INTR(1)	RO	Interrupt pending from A1SP1. Read A1SP0_INTR_REG to determine cause.
A1SP2_INTR(2)	RO	Interrupt pending from A1SP2. Read A1SP0_INTR_REG to determine cause.
A1SP3_INTR(3)	RO	Interrupt pending from A1SP3. Read A1SP0_INTR_REG to determine cause.
RAM1_PAR_ERR(4)	R2C	Parity encountered in RAM1 interface.
RAM2_PAR_ERR(5)	R2C	Parity encountered in RAM2 interface.
UTOP_PAR_ERR(6)	R2C	Parity encountered in UTOPIA interface.
T_UTOP_FULL(7)	R2C	Transmit UTOPIA FIFO is full. If it is still full after a read, the bit will set again.
T_UTOP_XFR_ERR(8)	R2C	The Transmit UTOPIA Interface was requested to send a cell when it did not have one available.
UTOP_LFIFO_FULL(9)	R2C	UTOPIA loopback FIFO is full. If it is still full after a read, the bit will set again.
R_UTOP_RUNT_CL(10)	R2C	A runt cell (less than 53 bytes) was received.
SBI_ADD_INTR(11)	RO	An interrupt is pending related to the SBI Add Bus, also known as the Insert SBI bus. Read the Insert Master Interrupt Status Register (INS_MSTR_INT) to find the cause of the interrupt.
SBI_DROP_INTR(12)	RO	An interrupt is pending related to the SBI-Drop Bus, also known as the Extract SBI bus. Read the Extract Master Interrupt Status Register (EXT_MSTR_INT) to find the cause of the interrupt.
SBI_ALARM(13)	RO	An SBI Alarm is pending in either the SBI_ALARM_REGH or SBI_ALARM_REGL registers.

Notes:

- The A1SP1_INTR, A1SP2_INTR , A1SP3_INTR , RAM2_PAR_ERR, SBI_ADD_INTR, SBI_DROP_INTR, and SBI_ALARM interrupts are not used in the AAL1gator-8 and AAL1gator-4.

- The RAM2_PAR_ERR is not used if the 2nd RAM interface is not needed for the AAL1gator-32.

11.2 UTOPIA Interrupts

The UTOPIA block sources five interrupts directly to the **Master Interrupt Register** (0x81000). The five interrupts are Transmit UTOPIA FIFO full, Loopback FIFO full, UTOPIA parity error, runt cell error, and UTOPIA transfer error. The first interrupt indicates that the Transmit UTOPIA four cell FIFO has filled. The second interrupt indicates that the UTOPIA loopback FIFO has filled. The third interrupt indicates there was a parity error on the data received on the UTOPIA interface. The fourth interrupt indicates cell less than 53 bytes was detected. The fifth interrupt indicates the receive UTOPIA interface was requested to send a cell when it did not have one available.

11.3 RAM Interface Interrupts

The two RAM interface blocks each source a parity error interrupt directly to the Master Interrupt Register.

11.4 Line Interface Interrupts

The Line Interface sources three interrupts: SBI_DROP_INTR and SBI_ADD_INTR and SBI_ALARM_INTR. The SBI_DROP_INTR indicates there is an interrupt pending related to the SBI-Drop bus, also known as the Extract SBI bus. The Extract Master Interrupt Status Register needs to be read to determine the cause of the interrupt. The SBI_ADD_INTR indicates there is an interrupt pending related to the SBI-Add bus, also known as the Insert SBI bus. The Insert Master Interrupt Status Register needs to be read to determine the cause of the interrupt. The SBI_ALARM_INTR indicates that an alarm was detected on a link from the SBI. SBI_ALARM_REGH and SBI_ALARM_REGL registers will identify which link failed. Please see section 8 for the description of SBI alarm and interrupt registers.

11.5 A1SP Interrupts

The A1SP blocks each source an interrupt to the **Master Interrupt Register** (0x81000). Within each A1SP block, the A1SP*n* interrupt register indicates the source of the interrupt within the A1SP block. Since many indications provided by the A1SP interrupt structure are per channel or per queue, there are 2 FIFOs

provided for per channel indications. There are six possible sources of an interrupt for the A1SP_INTR_REG: A1SP*n* Receive Status FIFO not empty, A1SP*n* Transmit Idle State FIFO not empty, A1SP*n* Receive Status FIFO overflow, Transmit Idle State FIFO overflow, OAM interrupt, and Frame Advance FIFO overflow. The OAM interrupt indicates that the Receive OAM Queue is not empty. In other words, this interrupt can be thought of as an active low Receive OAM Queue Empty signal. The A1SP*n* Receive Status FIFO overflow, A1SP Transmit Idle State FIFO overflow, and Frame Advance FIFO overflow interrupts simply indicate that the FIFOs have overflowed. The A1SP*n* Receive Status FIFO not empty interrupt indicates the A1SP*n* Receive Status FIFO is not empty and the A1SP*n* Transmit Idle State FIFO not empty interrupts do the same. The next two sections discuss how these FIFOs operate.

Since some of the conditions are transitory, the **A1SP*n*_INTR_REG** (0x81010, …,13) captures whether the condition has occurred since the last time the register was read. The bits in this register are set upon entry into the indicated condition and are cleared when this register is read. If any of these conditions still exist the corresponding bit will not be set again until the condition ends and then occurs. The interrupt description column describes the condition when the bit in A1SP*n*_INTR_REG is a logic '1'. The condition does not exist if the bit is a logic '0'.

Bit (Position)	Interrupt Description when Bit = '1'
FR_ADV_FIFO_FULL(0)	The Frame Advance FIFO has entered the full state since the last time the register was read.
OAM_INTR(1)	A1SP*n* has received a new OAM cell.
TIDLE_FIFO_EMPB(2)	The Transmit Idle State FIFO has entered the not empty state.
TIDLE_FIFO_FULL(3)	The Transmit Idle State FIFO has entered the full state.
RSTAT_FIFO_EMPB(4)	The Receive Status FIFO has entered the not empty state.
RSTAT_FIFO_FULL(5)	The Receive Status FIFO has entered the full state.
TALP_FIFO_FULL(6)	The TALP FIFO has entered the full state.

Read the **A1SP*n*_STAT_REG** (0x81020, …, 23) for current status. If any bit is set in this register and the corresponding enable bit is set in **A1SP*n*_INTR_EN_REG** (0x81110, …, 13), the A1SP*n*_INTR bit will be set in **MSTR_INTR_REG** (0x81000). The interrupt description column in the table below describes the condition when the bit in A1SP*n*_STAT_REG is a logic '1'. The condition does not exist if the bit is a logic '0'.

Bit (Position)	Interrupt Description when Bit = '1'
FR_ADV_FIFO_FULL(0)	The Frame Advance FIFO is full.
OAM_INTR(1)	A1SP*n* has received a new OAM cell.
TIDLE_FIFO_EMPB(2)	The Transmit Idle State FIFO is not empty.
TIDLE_FIFO_FULL(3)	The Transmit Idle State FIFO is full.
RSTAT_FIFO_EMPB(4)	The Receive Status FIFO is not empty.
RSTAT_FIFO_FULL(5)	The Receive Status FIFO is full.
TALP_FIFO_FULL(6)	The TALP FIFO is full.

11.5.1 A1SP*n* Receive Status FIFO

The Receive Status FIFO (RCV_STAT_FIFO) consists of 64 entries and is contained internal to the chip. The FIFO is accessed using a single address. When the FIFO transitions from empty to not empty, the RSTAT_FIFO_EMPB bit in both the A1SP*n*_INTR_REG and the A1SP*n*_STAT_REG will go active. When there are no longer any entries in the FIFO, the STAT_FIFO_EMPB bit in the A1SP*n*_STAT_REG will go inactive.

Each entry within the Interrupt FIFO indicates the queue responsible for the interrupt and one of four possible causes: DBCES bitmask change, exiting underrun, entering underrun, and receive queue error. The first cause reports a change in the bit mask for DBCES. The second two causes simply report a change in the underrun status, while the third cause indicates an error has occurred on the receive side. To find out the specific cause of the error, the processor should access the R_ERROR_STKY register for the queue responsible for the interrupt.

The **A1SP*n* Receive Status FIFO** register (0x81040, …, 43) is the read port of a 64 word FIFO that is used to capture receive status events on a first come first serve basis. If the FIFO overflows the RSTAT_FIFO_FULL bit will be set in the A1SP*n*_INTR_REG. The presence of data in this FIFO will set the RSTAT_FIFO_EMPB bit in the A1SP*n*_INTR_REG. The **RSTAT_EN_REG** (0x81140, …, 43) controls whether the corresponding errors or status conditions cause an entry to be written into the RSTAT_FIFO. There is a separate RSTAT_FIFO for each A1SP block. The following bits are used to report error or status conditions:

RSTAT_FIFO_REG Bit	Function
RECEIVE_QUEUE_ERR	An error or status condition occurred on the receive queue identified in QUEUE_NUMBER. Read sticky bit register for this queue to determine actual event.
ENTER_UNDERRUN	The queue identified by QUEUE_NUMBER just entered the underrun state. If the queue is in DBCES mode, this may also indicate that all channels have gone idle.
EXIT_UNDERRUN	The queue identified by QUEUE_NUMBER just exited the underrun state.
BITMASK_CHANGE	This condition is only valid if DBCES mode has been enabled for this queue. If the bit is set it indicates that the bitmask for active channels has changed. Read R_CHAN_ACT in the R_QUEUE_TBL to determine current bit mask.
T_LINE_RESYNC	The transmit line identified by QUEUE_NUMBER(7:5) entered a resync state.
R_LINE_RESYNC	The receive line identified by QUEUE_NUMBER(7:5) entered a resync state.
QUEUE_NUMBER [7:0]	Identifies the queue on which the reported event occurred. Note that the queue number is modulus 256. The top two bits of the queue number will be the number of the A1SP block that is being read.

11.5.2 A1SP*n* Transmit Idle State FIFO

The Transmit Idle State FIFO consists of 64 entries and is contained internal to the chip. The FIFO is accessed using a single address port. When the FIFO transitions from empty to not empty, the TX_IDLE_FIFO_EMPB bit in both the A1SP*n*_INTR_REG and the A1SP*n*_STAT_REG will go active. When there are no longer any entries in the FIFO, the Transmit Idle State FIFO not empty interrupt bit in the A1SP*n*_STAT_REG will go inactive.

Each entry within the Transmit Idle State FIFO indicates the channel responsible for the interrupt and certain status information depending on the selected DBCES mode.

The **A1SP*n*_TIDLE_FIFO** (0x81030, …, 33) register is the read port of a 64 word FIFO that is used to indicate changes in the activity status (active or idle) on a given channel on a first come first serve basis. If the FIFO overflows the

TX_IDLE_FIFO_FULL bit will be set in the A1SP*n*_INTR_REG. When this FIFO goes from an empty to a non-empty condition the TX_IDLE_FIFO_EMPB bit in the A1SP*n*_INTR_REG will be set. The presence of data in this FIFO will set the TX_IDLE_FIFO_EMPB bit in the A1SP*n*_STAT_REG. Read A1SP*n*_STAT_REG to determine when FIFO goes empty again. If idle detection is not enabled on a given channel then the channel will not write to this FIFO.

The function of the CHAN_STATUS[15:0] bits is dependent on which of the two idle detection modes is used: automatic or processor. The idle detection mode is controlled by the value of IDLE_CFG_Ln_C*x* for the respective line and channel number in the Idle Configuration Detection Table (see section 7).

11.5.2.1 Automatic Idle Detection with either CAS or Pattern Matching

In this mode when either CAS or Pattern Matching indicates a change in the active status of a channel, an entry will be written into the FIFO depending on the state of IDLE_CFG_Ln_C*x* for that channel.

Bits	Function
CHAN_STATUS[0]	STATUS: When clear, indicates that the channel contained in the CHAN_NUM field is in the idle state. When set, indicates that the channel contained in the CHAN_NUM field is in the active state.
CHAN_STATUS[15:8]	CHAN_NUM [7:0]: This field indicates the channel that encountered a change in activity status.

11.5.2.2 Processor Idle Detection

In this mode, any changes in the CAS value in either direction will cause an entry to be written to the FIFO if Processor Idle detection is enabled for this line. Note that filtering of CAS changes is only done once by the AAL1gator. If any additional filtering is required, this must be done in the external framers:

Bits	Function
CHAN_STATUS[3:0]	TX_CAS: This field indicates the current value of the transmit CAS ABCD bits.
CHAN_STATUS[7:4]	RX_CAS: This field indicates the current value of the receive CAS ABCD bits.
CHAN_STATUS[15:8]	CHAN_NUM [7:0]: This field indicates the channel that encountered a change in activity status.

11.5.3 Receive Queue Error Enables

The enable bits in the **RCV_Q_ERR_EN** (0x81150, …, 53) register control what is done when R_ERROR_STKY bits are set in the R_QUEUE_TBL. It controls which types of error/status conditions cause the RECEIVE_QUEUE_ERR indication in the RCVn_STAT_FIFO to be set. All queues are configured the same way. Only the first enabled condition which occurs for a given queue will cause an entry to be made in the RCVn_STAT_FIFO until the sticky bit register is cleared. So usually you should only enable bits that will not occur normally. The default configuration is as follows:

Bit	Register	Value
UNDERRUN	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
OVERRUN	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
PTR_MISMATCH	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
RESUME	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
SRTS_UNDERRUN	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
SRTS_RESUME	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
PTR_PARITY_ERR	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
POINTER_RECEIVED	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
SN_CELL_DROP	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
FORCED_UNDERRUN	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
POINTER_SEARCH	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
ALLOC_TBL_BLANK	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
PTR_RULE_ERROR	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
DBCES_BM_ERR	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
CELL_RECEIVED	**RCV_Q_ERR_EN** (0x81150, …, 53)	0
Reserved	**RCV_Q_ERR_EN** (0x81150, …, 53)	0

12 IDLE CHANNEL DETECTION CONFIGURATION AND STATUS

The following registers control how idle channel detection is configured for the AAL1gator and indicate active/idle channel status for all channels on the chip.

Offset	Register Description	Register Mnemonic
0x000 – 0x00F	A1SP *n* Receive Channel Active Table	RX_ACTIVE_TBL*n*
0x010 – 0x01F	A1SP *n* Receive Pending Channel Table	RX_PENDING_TBL*n*
0x100 - 0x1FF	A1SP *n* Change Pointer Table	RX_CHANGE_PTR*n*
0x200 - 0x20F	A1SP *n* Transmit Channel Active Table	TX_ACTIVE_TBL*n*
0x210 – 0x217	A1SP *n* Pattern Matching Line Configuration	PAT_MTCH _CFG*n*
0x220	A1SP *n* Idle Detection Configuration Table	IDLE_CFG_TBL*n*
0x300 – 0x3FF	A1SP *n* CAS/Pattern Matching Configuration Table	CAS_P_CFG_TBL*n*

The base addresses for Idle Channel registers for the four A1SPs are as follows:

A1SP	Base Address
0	0x82000
1	0x82400
2	0x82800
3	0x82C00

Note: In the AAL1gator-8 and AAL1gator-4, only the A1SP 0 registers are used.

12.1 Receive Channel Active Table

The Receive Channel Active Table contains the receive active channel status for A1SP *n* block which contains *m* lines of 32 channels (*m* x 32 channels total)

where *m* = 8 for the AAL1gator-32 and AAL1gator-8 and *m* = 4 for the AAL1gator-4. Therefore, lines 4 to 7 are not used in the AAL1gator-4. The status for each channel is composed of a 1 bit field (RX_CHAN_ACTIVE) and therefore each word contains the status for 16 channels. The structure of the table is shown below.

LINE = OFFSET (MOD 32) / 2

CHANNEL (ADDRESS) = OFFSET (MOD 2)

CHANNEL (BIT LOCATION) = CHANNEL (MOD 16)

Addr. Offset	Line	Channel
0x000	0	RX_CHAN_ACTIVE[15:0]
0x001	0	RX_CHAN_ACTIVE[31:16]
…	…	RX_CHAN_ACTIVE[15:0]
…	…	RX_CHAN_ACTIVE[31:16]
0x00E	7	RX_CHAN_ACTIVE[15:0]
0x00F	7	RX_CHAN_ACTIVE[31:16]

The one bit RX_CHAN_ACTIVE[n] field indicates the active status of the receive channel based on DBCES bit mask. This field is only valid if DBCES is enabled for the queue which is associated with this channel. On read:

RX_CHAN_ACTIVE[n]	Function
0	Receive Channel n on the particular A1SP and line is inactive.
1	Receive Channel n on the particular A1SP and line is active.

12.2 Receive Pending Table

The A1SP n RX Pending Table contains the receive pending channel status for A1SP n block which contains m lines of 32 channels (m x 32 channels total) where m = 8 for the AAL1gator-32 and AAL1gator-8 and m = 4 for the AAL1gator-4. Therefore, lines 4 to 7 are not used in the AAL1gator-4. The status for each channel is composed of a 1 bit field (RX_PENDING) and

therefore each word contains the status for 16 channels. The structure of the table is shown below.

LINE = OFFSET (MOD 32) / 2

CHANNEL (ADDRESS) = OFFSET (MOD 2)

CHANNEL (BIT LOCATION) = CHANNEL (MOD 16)

Addr. Offset	Line	Channel
0x000	0	RX_PENDING [15:0]
0x001	0	RX_PENDING [31:16]
..	….	RX_PENDING [15:0]
..	….	RX_PENDING [31:16]
0x00E	7	RX_PENDING [15:0]
0x00F	7	RX_PENDING [31:16]

The one bit RX_PENDING[n] field indicates the change pending status of the receive channel based on DBCES bit mask. This field is only valid if DBCES is enabled for the queue which is associated with this channel. This bit is set when the RALP detects a change in the bit mask, and is reset when the RFTC updates the active table with the pending change. On read:

RX_PENDING[n]	Function
0	No change in state is pending for this channel.
1	A state change is pending for this channel.

12.3 Receive Change Pointer Table

The RX Change Pointer Table contains the frame pointers, indicating in what frame a channel should change its active state. The new active state is stored along with the frame pointer. This table is generated by the RALP when it gets the bit mask updates in DBCES mode and consumed by the RTFC. When the RFTC gets to the frame specified in the pointer, if the pending bit is set in the pending table, the RFTC updates the active table and plays out the appropriate data depending upon the state of the channel. The structure of the table is shown below. Note that lines 4-7 are not used in the AAL1gator-4.

This table is for chip use only and should not be modified after initialization. Initialize to all "0"s.

LINE = OFFSET (MOD 256) / 32

CHANNEL (ADDRESS) = OFFSET (MOD 32)

Addr. Offset	Line	Channel
0x100	0	CHANGE_PTR_0
0x101	0	CHANGE_PTR_1
..	…	…
..	…	…
0x1FE	7	CHANGE_PTR _30
0x1FF	7	CHANGE_PTR _31

Bits	Function
CHANGE_PTR[8:0]	FRAME_PTR: Indicates the value of frame pointer in which the active status of the channel should change to the value indicated in ACTIVE.
CHANGE_PTR [9]	ACTIVE: Indicates the state which should become current at the frame indicated by FRAME_PTR. When set, the channel should change to an inactive state. When low, the channel should change to an active state.

12.4 Transmit Channel Active Table

The A1SP n TX Channel Active Table contains the transmit active channel status for each A1SP block which each contain *m* lines of 32 channels (4 x *m* x 32 channels total) where *m* = 8 for the AAL1gator-32 and AAL1gator-8 and *m* = 4 for the AAL1gator-4. Therefore, lines 4 to 7 are not used in the AAL1gator-4. The status for each channel is composed of a 1 bit field (TX_CHAN_ACTIVE) and therefore each word contains the status for 16 channels. The structure of the table is shown below. This table should be initialized to all '0's.

A1SP = OFFSET / 16

LINE = OFFSET (MOD 32) / 2

CHANNEL (ADDRESS) = OFFSET (MOD 2)

CHANNEL (BIT LOCATION) = CHANNEL (MOD 16)

Addr. Offset	Line	Channel
0x200	0	TX_CHAN_ACTIVE[15:0]
0x201	0	TX_CHAN_ACTIVE[31:16]
…	…	TX_CHAN_ACTIVE[15:0]
…	…	TX_CHAN_ACTIVE[31:16]
0x20E	7	TX_CHAN_ACTIVE[15:0]
0x20F	7	TX_CHAN_ACTIVE[31:16]

The one bit TX_CHAN_ACTIVE[n] field indicates the active status of the channel. This field is only valid if IDLE_CFG for the associated channel is not equal to "00" (disabled). If IDLE_CFG is equal to "10" (Automatic, CAS) or "11" (Automatic, Pattern) then this field is read only and is updated by the AAL1gator. If IDLE_CFG equals "01" (processor) then the processor activates and deactivates channels by writing this bit. On read/write:

TX_CHAN_ACTIVE[n]	Function
0	Channel is inactive.
1	Channel is active.

Note: Channels within a 16-bit word should not mix automatic detection with processor IDLE_CFG modes because there can be contention in updating these fields.

12.5 Pattern Matching Line Configuration

The A1SP *n* Pattern Matching Line Configuration table contains the configuration for each line that is running in pattern matching idle detection mode. When the received pattern matches the programmed pattern within the bounds set within this table; the channel is considered to be idle. Please note that lines 4 to 7 are not used in the AAL1gator-4.

A1SP = OFFSET / 8

LINE = OFFSET

Addr. Offset	Line	Bit	Channel
0x210	0	15:8	Reserved
0x210	0	7:0	INTVL_LEN
…	….		….
…	….		….
0x217	7	15:8	Reserved
0x217	7	7:0	INTVL_LEN

The INTVL_LEN[7:0] field defines the interval length in increments of 12 ms (T1) or 16 ms (E1). The interval length is calculated by taking the number from this field, adding 1, and multiplying the result by 12/16 ms ((INTVL_LEN + 1) * 12/16 ms).

12.6 Idle Detection Configuration Table

The A1SP *n* Idle Detection Configuration Table contains the idle detection configuration for the A1SP n block which contains m lines where m = 8 for the AAL1gator-32 and AAL1gator-8 and m = 4 for the AAL1gator-4. Therefore, IDLE_CFG_4 through IDLE_CFG_7 are not used in the AAL1gator-4. The configuration for each line is composed of a 2 bit field (IDLE_CFG) and therefore the register contains the configuration for up to 8 lines. The structure of the table is shown below.

Addr. Offset	Line Configuration
0x220	IDLE_CFG_m

The two-bit IDLE_CFG_m field defines the idle detection mode. Selection of the four possible modes is shown below:

IDLE_CFG_m	Idle Detection Mode
00	Idle Detection Disabled.
01	Processor controlled activation/deactivation of channels.
10	Automatic activation/deactivation of channels using CAS matching.
11	Automatic activation/deactivation of channels using pattern matching.

12.7 CAS/Pattern Matching Configuration

The A1SP *n* CAS/Pattern Matching Configuration Table contains the configuration fields for automatic idle channel detection or processor idle detection depending on the value of IDLE_CFG. The A1SP block contains m lines of 32 channels (*m* x 32 channels total per A1SP) where *m* = 8 for the AAL1gator-32 and AAL1gator-8 and *m* = 4 for the AAL1gator-4. Therefore, lines 4 to 7 are not used in the AAL1gator-4. The function of these fields changes depending on whether IDLE_CFG indicates CAS mode, Pattern Matching, or Processor Idle Detection mode for the associated channel. The configuration field for each channel is composed of a 16 bit field (AUTO_CONFIG/PROC_CONFIG) and therefore each word contains the status for 1 channel. The structure of the table is shown below.

LINE = OFFSET (MOD 256) / 32

CHANNEL (ADDRESS) = OFFSET (MOD 32)

Addr. Offset	Line	Channel
0x300	0	AUTO_CONFIG_0/PROC_CONFIG_0
0x301	0	AUTO_CONFIG_1/PROC_CONFIG_1
...	…	…
...	…	…
0x3FE	7	AUTO_CONFIG_30/PROC_CONFIG_30
0x3FF	7	AUTO_CONFIG_31/PROC_CONFIG_31

AUTO_CONFIG_n has two different formats. If IDLE_CFG = "10" (CAS mode) it has one format. If IDLE_CFG = "11" it has a different format. If IDLE_CFG = "00" AUTO_CONFIG is reserved and should not be written. If IDLE_CFG = "01" the field uses the PROC_CONFIG_n format which is different than the AUTO_CONFIG_n format.

12.7.1 CAS Matching Format

The CAS matching format is used when IDLE_CFG_*m* = "10" and is described in the following table:

Bit	Function	Description
15:12	RX_MASK	These bits are used as a mask on the RX_CAS field. When a bit is set in these mask fields the bit will not be factored into the pattern matching function and therefore will be considered a "don't care" bit.
11:8	TX_MASK	These bits are used as a mask on the TX_CAS field. When a bit is set in these mask fields the bit will not be factored into the pattern matching function and therefore will be considered a "don't care" bit.
7:4	RX_CAS	Indicates the value of CAS that when received from the ATM network indicates an idle condition. This value will be masked by RX_MASK.
3:0	TX_CAS	Indicates the value of CAS that when received on the line indicates an idle condition. This value will be masked by TX_MASK.

12.7.2 Pattern Matching Format

The Pattern Matching Format is used when IDLE_CFG_*m* = "11" and is described in the following table:

Bit	Function	Description
15:8	PAT_MASK	When a bit is set in this mask field the bit will not be factored into the pattern matching function and therefore will be considered a "don't care" bit.
7:0	IDLE_PATTERN	When this programmed pattern matches the received byte for the associated channel, the channel is considered to be idle. The conditions which qualify a match are controlled by the PAT_MTCH_CFG register.

12.7.3 Processor Idle Detection Format

The Processor Idle Detection Format is used when IDLE_CFG_*m* = "01" and is described in the following table:

Bit	Function	Description
15:12	RX_MASK	These bits are used as a mask on the RX_CAS field. Only changes in unmasked CAS bits will cause an interrupt to the processor.
11:8	TX_MASK	These bits are used as a mask on the TX_CAS field. Only changes in unmasked CAS bits will cause an interrupt to the processor.
7:0	Reserved	Reserved for internal use. Initialize to '0'.

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Application Information: apps@pmc-sierra.com
(604) 415-4533
Web Site: http://www.pmc-sierra.com

PMC-199-1820 (P2) Issue date: April 2001