PM73122

AAL1GATOR-32

ATM ADAPTATION LAYER 1 SEGMENTATION AND REASSEMBLY PROCESSOR-32

DATASHEET

PROPRIETARY AND CONFIDENTIAL

RELEASED

ISSUE 7: JUNE 2001

REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	Dec 1998	Document created.
2	Sept 1999	Significant design details added.
3	Nov 1999	Further design details and pinout added.
4	Jan 2000	Updated to reflect functional details based on latest design. Clarified and added further descriptive text. Finalized pinout.
5	May 2000	Added description of the floating CAS nibble capability. (SHIFT_CAS). Added more functional detail.

Issue No.	Issue Date	Details of Change
6	May 2001	Added section: Changes from Rev B to Rev C. Added “BUSMASTER” bit in SBI_BUS_CFG_REG, along with description. Clarified NODROP_IN_START and DROPPED_CELL counter functionality is that NODROP_IN_START has no effect for ROBUST SN Processing and UDF-HS mode. In LIN_STR_MODE added cross reference to HS Operations section for Dual DS3 mode. Corrected “Out of Band Signaling Idle Detection” section with respect to RX and TX CAS. Fixed Robust SN processing Figure. Added note in SN PROCESSING register to clarify that only ROBUST_SN_EN or DISABLE_SN can be set, not both. Only SBI mapping ram has 2 pages not SBI control RAM. Sometimes An_SW_RESET needs to be used with high speed queue. Updated Operations section and An_SW_RESET description. Added times when OFFSET needs to be set to FRAMES_PER_CELL in OFFSET bit description. Clarified MVIP-90 configuration in the Operations section Changed DC_INT from link to tributary. Flipped HIZDATA and HIZIO. In DC Characteristics, made operating current for I/O typical, added .5 ns margin to C1FP hold and SBI Tz. Also applied C1FP timing to C1FP_ADD also. Default value corrected for MIN_DEPTH for DS3 Register. Clarified and/or corrected “INSBI/EXSBI Programming Steps” section, “SBI Operation” section, tributary mapping sequences in “Programming Sequence for SBI” section and lack of depth check support in SBI Synchronous Mode. Added statements describing that UDF_HS Loopback Mode requires that the High Speed Queue be reset, that SPE Activation Must Occur After Tributaries are Enabled, that SPE Activation must occur after tributaries are enabled. Updated applications section with new PMC devices: PM8316 TEMUX-84, PM7341 S/UNI-IMA-84.

Issue No.	Issue Date	Details of Change
7	June 2001	Changed DC_INT_EN to SYNC_INT_EN in operations section. In DC Characteristics, corrected IDDOP(2.7) for HS mode. In Operations section, clarified function of SBI Parity Error Detection and recommended setting the BUSMASTER bit in the PHY SBI device. Added note at beginning of AC Characteristics recommending that transition times on clock inputs is less than 15 ns. Corrected Ram Interface Timing In Memory Mapped Register Section added CSD_BYTES_LEFT register definition in T_QUEUE_TABLE. Added hidden bits in QUE_CREDITS word and added R_DBCES_BM_IN_NEXT to R_TOT_LEFT memory register and changed R_DBCES_BM_IN_NEXT bit to R_DBCES_BM_INACT in R_STATE_0 memory register. Clarified C1FP signal definition in SBI Signal Definition section. Changed “Out of Band” idle detection mode to “Processor Controlled” idle detection mode in Idle Detection section of Functional Description. Updated PCR section in Functional Description Added SRTS patent legal note to footer of last page Corrected T1/E1 Link Rate Table (reversed polarity of C1FP) Corrected cross reference in ADD QUEUE FIFO section Removed equations from partial cell PCR section and replaced with summary table. Added reference by RL_CLK that clock can not be gapped and must have jitter less than .3 UI if using SRTS. Added minor clarifications, including: R_LINE_STATE location not used in UDF-HS mode, explanation of PCR for UDF-ML, removed references to E3 over SBI, added recommendation to tie unused TL_CLK pins high in UDF-HS modes, renamed RPHY_ADD_RSX pin to RPHY_ADD[4]/RSX to match Tx side, removed ‘sampled on rising edge’ from ADETECT pin description, added that PAGE bit is a don’t care when accessing SBI Control RAM’s, clarified max RSTB timing and max 400 SYSCLK rd/wr timing, clarified T1/E1 granularity in SBI mode(at DA1SP level), E1_w_T1_sig mode not supported over SBI, clarified that SN state machines freeze during underruns.

CONTENTS

LIST OF REGISTERS

LIST OF FIGURES

LIST OF TABLES

1 CHANGES FROM REV. A TO REV. B

There are four main changes when transitioning from Rev. A to Rev. B:

1. The DEV_ID has changed from 0000 to 0001.

2. The JTAG version number has changed from 0 to 1.

3. The SHIFT_CAS feature which allows the signaling to be aligned with the first nibble of data has been added. This feature is enabled by setting SHIFT_CAS in the LIN_STR_MODE register.

4. The capability to have a separate C1FP for the ADD and DROP side of the SBI bus has been added. In Rev. A there is only one C1FP pin, but Rev. B has the option to support two separate C1FP pins. The original C1FP pin becomes the C1FP for the drop side of the SBI bus and the new pin, C1FP_ADD, is the C1FP for the add side of the SBI bus. Rev. B defaults to using one C1FP pin which makes it backwards compatible with Rev. A, but two C1FP capability can be enabled by setting TWO_C1FP_EN in the SBI_BUS_CFG_REG register.

2 CHANGES FROM REV B TO REV C

There are three main changes when transitioning from Rev. B to Rev. C:

1. The DEV_ID has changed from 0001 to 0010.
2. The JTAG version number has changed from 1 to 2.
3. A BUSMASTER bit has been added (bit 15 of SBI_BUS_CFG_REG). When set, the AAL1gator32 will drive all SBI bytes that are not driven by other devices. The will prevent parity errors from being detected due to floating bytes.

3 FEATURES

The AAL1 Segmentation And Reassembly (SAR) Processor (AAL1gator-32) is a monolithic single chip device that provides DS1, E1, E3, or DS3 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 device.

- Compliant with the ATM Forum's Circuit Emulation Services (CES) specification (AF-VTOA-0078), and the ITU-T I.363.1

- Supports Dynamic Bandwidth Circuit Emulation Services (DBCES). Compliant with the ATM Forum's DBCES specification (AF-VTOA-0085). Supports idle channel detection via processor intervention, CAS signaling, or data pattern detection. Provides idle channel indication on a per channel basis.

- Supports non-DBCES idle channel detection by activating a queue when any of its constituent time slots are active, and deactivating a queue when all of its constituent time slots are inactive.

- Provides AAL1 segmentation and reassembly of 16 individual E1 or T1 lines, 8 H-MVIP lines at 8 MHz, or 2 E3 or DS3 or STS-1 unstructured lines.

- Using the optional Scalable Bandwidth Interconnect (SBI) Interface, provides AAL1 segmentation and reassembly of up to 32 T1, E1, or 2 DS3 links. In SBI mode can map any SBI tributary to any of the 32 AAL1 links. Supports floating and locked tributaries as well as unframed, framed without CAS and framed with CAS tributaries. CAS is only supported on Synchronous tributaries.

- Provides a standard UTOPIA level 2 Interface which optionally supports parity and runs up to 52 MHz. Only Cell Level Handshaking is supported. In MPHY mode, can act like a single port or 4 port device. The following modes are supported:

    - 8/16-bit Level 2, Multi-Phy Mode (MPHY)

    - 8/16-bit Level 1, SPHY

    - 8-bit Level 1, ATM Master

- Provides an optional 8/16-bit Any-PHY slave interface.
- Supports up to 1024 Virtual Channels (VC).
- Supports n x 64 (consecutive channels) and m x 64 (non-consecutive channels) structured data format.
- Provides transparent transmission of Common Channel Signaling (CCS) and Channel Associated Signaling (CAS). Provides for termination of CAS signaling.
- Allows the CAS nibble to be coincident with either the first or second nibble of the data.
- Provides per-VC data and signaling conditioning in the transmit cell direction and per DS0 data and signaling conditioning in the transmit line direction. Data and signaling conditioning can be individually enabled. Includes DS3 AIS conditioning support in both directions. Transmit line conditioning options include programmable byte pattern, pseudo-random pattern or old data. Conditioning automatically occurs on underruns.
- In Cell Transmit direction, provides per-VC configuration of time slots allocated, CAS signaling support, partial cell size, data and signaling conditioning, ATM Cell header definition. Generates AAL1 sequence numbers, pointers and SRTS values in accordance with ITU-T I.363.1. Multicast connections are supported.
- In Cell Transmit direction provides counters for:
    - Conditioned cells transmitted for each queue
    - Cells which were suppressed for each queue
    - Total number of cells transmitted for each queue
- In Cell Receive direction, provides per-VC configuration of time slots allocated, CAS signaling support, partial cell size, sequence number processing options, cell delay variation tolerance buffer depth, maximum buffer depth. Processes AAL1 headers in accordance with ITU-T I.363.1.
- In Cell Receive direction, supports the Fast Sequence Number processing algorithm on all types of connections and Robust Sequence Number processing on Unstructured Data Format (UDF) connections. Cells are inserted/dropped to maintain bit integrity on lost or misinserted cells. Bit integrity is maintained through any single errored cell or up to six lost cells. Bit

integrity can also optionally be maintained even if an underrun occurs. Pointer bytes, signaling bytes, and bitmask bytes are taken into account. Cell insertion options include a programmable single byte pattern, pseudo-random data, or old data .

- In Cell Receive direction provides counters for the following events which include all counters required by the ATM Forum's CES-IS 2.0 MIB:
    - Incorrect sequence numbers per queue
    - Incorrect sequence number protection fields per queue
    - Total number of received cells per queue
    - Total number of dropped cells per queue
    - Total number of underruns per queue
    - Total number of lost cells per queue
    - Total number of overruns per queue
    - Total number of reframes per queue
    - Total number of pointer parity errors per queue
    - Total number of misinserted cells per queue
    - Total number of OAM or non-data cells received
    - Total number of OAM or non-data cells dropped.
- For each receive queue the following sticky bits are maintained:
    - Cell received
    - Structured pointer rule error detected
    - DBCES bitmask parity error
    - Cell dropped due to blank allocation table
    - Cells dropped due to pointer search
    - Cell dropped due to forced underrun

- Cell dropped due to sequence number processing algorithm
- Valid pointer was received
- Pointer parity error detected
- SRTS resume from an underrun condition
- SRTS underrun occurred
- Resume occurred from an underrun condition
- Pointer reframe occurred
- Overrun condition detected
- Cell received while in an underrun

- Supports AAL0 mode, selectable on a per VC basis.
- Provides system side loopback support. When enabled and the incoming VCI matches the programmable loopback VCI, the cell received on the Receive UTOPIA interface is looped back to the Transmit UTOPIA interface. Alternatively the UTOPIA interface can be put into remote loopback mode where all incoming cells are looped back out. Provides line side loopback, enabled on a per queue basis, which can loop a single channel or any group of channels which can be mapped to a single queue.
- Provides a patented frame based calendar queue service algorithm with anti-clumping add-queue mechanism that produces minimal Cell Delay Variation (CDV). In UDF mode uses non-frame based scheduling to optimize CDV. In addition, four internal cell generation engines work in parallel to further insure low CDV.
- Queues are added by making entries into an add-queue FIFO to minimize queue activation overhead. An offset can be configured when queue is added to distribute cell build times to minimize CDV due to clumping.
- Provides single maskable, open-collector interrupt with master interrupt register to facilitate interrupt processing. The master interrupt register indicates the following conditions each of which can be masked:
  - Error/status condition with one of four AAL1 blocks
  - Ram parity error

- UTOPIA parity error
- Transmit UTOPIA FIFO is full
- Transmit UTOPIA transfer error
- UTOPIA loopback FIFO is full
- UTOPIA runt cell is detected
- SBI error detected

- For each AAL1 block the following conditions can cause an interrupt, each of which can be masked. Separate 64 entry FIFOs per AAL1 block are used to track receive and transmit status.
  - A receive queue sticky bit was just set (individual mask per sticky bit)
  - Receive queue entered underrun state
  - Receive queue exited underrun state
  - DBCES bitmask changed.
  - Receive Status FIFO overflow
  - Transmit Frame Advance FIFO full
  - Reception of OAM cells
  - Change in idle state of a channel enabled for idle channel detection
  - Transmit Channel Idle State change FIFO overflow
  - Line frame resync event
  - Transmit ATM Layer Processor (TALP) FIFO full
- For SBI logic the following conditions can cause an interrupt, each of which can be masked:
  - SBI Add bus FIFO overflow or underrun
  - SBI Add bus C1FP resync

- SBI Add bus depth check error
- SBI Drop bus FIFO overflow or underrun
- SBI Drop bus parity error
- SBI Drop bus depth check error
- SBI Drop bus C1FP resync
- SBI Alarm detected

- Provides a 16-bit microprocessor interface to internal registers, and two external 256K x 16(18) (10 ns) Pipelined Single-Cycle Deselect Synchronous SRAMs, or Synchronous ZBT SRAMs.

- Provides a transmit buffer which can be used for Operations, Administration and Maintenance (OAM) cells as well as any other user-generated cells such as AAL5 cells for ATM signaling. A corresponding receive buffer exists for the reception of OAM cells or non-AAL1 data cells.

- Includes an internal E1/T1 clock synthesizer for each line which can generate a nominal E1/T1 clock or be controlled via Synchronous Residual Time Stamp (SRTS) clock recovery method in Unstructured Data Format (UDF) mode or a programmable weighted moving average adaptive clocking algorithm. DS3 and E3 SRTS or adaptive clocking is supported using an external clock synthesizer and the clock control port.

- The clock synthesizers can also be controlled externally to provide customization of SRTS or adaptive algorithms. SRTS can also be disabled via a hardware input. Adaptive and SRTS information is output to a port for external processing for both low speed and high speed mode, if needed. Buffer depth is provided in units of bytes. The synthesizer can be set to 256 discrete frequencies between either +/-100 ppm for E1 or +/-200 ppm for T1.

- Low-power 2.5 Volt CMOS technology with 3.3 Volt, 5 Volt tolerant I/O.

- 352-pin super ball grid array (SBGA) package.

4 APPLICATIONS

- Multi-service ATM Switch
- ATM Access Concentrator
- Digital Cross Connect
- Computer Telephony Chassis with ATM infrastructure
- Wireless Local Loop Back Haul
- ATM Passive Optical Network Equipment

5 REFERENCES

Applicable Recommendations and Standards.

1. ANSI T1 Recommendation T1.403, Network-to-Customer Installation – DS1 Metallic Interface, NY, NY, 1995.

2. ANSI T1 Recommendation T1.630, Broadband ISDN-ATM Adaptation Layer for Constant Bit Rate Services, Functionality and Specification, NY, NY, 1993.

3. ATM Forum, ATM User Network Interface (UNI) Specification, V 3.1, Foster City, CA USA, September 1994.

4. ATM Forum, Circuit Emulation Service – Interoperability Specification (CES-IS), V. 2.0, Foster City, CA USA, August 1996.

5. ATM Forum, Specifications of (DBCES) Dynamic Bandwidth Utilization – in 64Kbps Time Slot Trunking Over ATM – Using CES, Foster City, CA USA, (AF-VTOA-0085) July 1997.

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

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

8. ITU-T Recommendation G.703, Physical/Electrical Characteristics of Hierarchical Digital Interfaces, April 1991.

9. ITU-T Recommendation I.363.1, B-ISDN ATM Adaptation Layer (AAL) Specification, July 1995.

10. ITU-T Recommendation G.823, The Control of Jitter and Wander within Digital Networks Which Are Based on the 2048 kbit/s Hierarchy, March 1993.

11. ITU-T Recommendation G.824 The Control of Jitter and Wander within Digital Networks Which Are Based on the 1544 kbit/s Hierarchy, March 1993.

12. PMC-971268, “High density T1/E1 framer with integrated VT/TU mapper AND M13 multiplexer” (TEMUX), 2000, Issue 5.

13. GO-MVIP, “MVIP-90 Standard” Release 1.1, October 1994.

14. GO-MVIP, “H-MVIP Standard” Release 1.1a, January 1997.

6 APPLICATION EXAMPLES

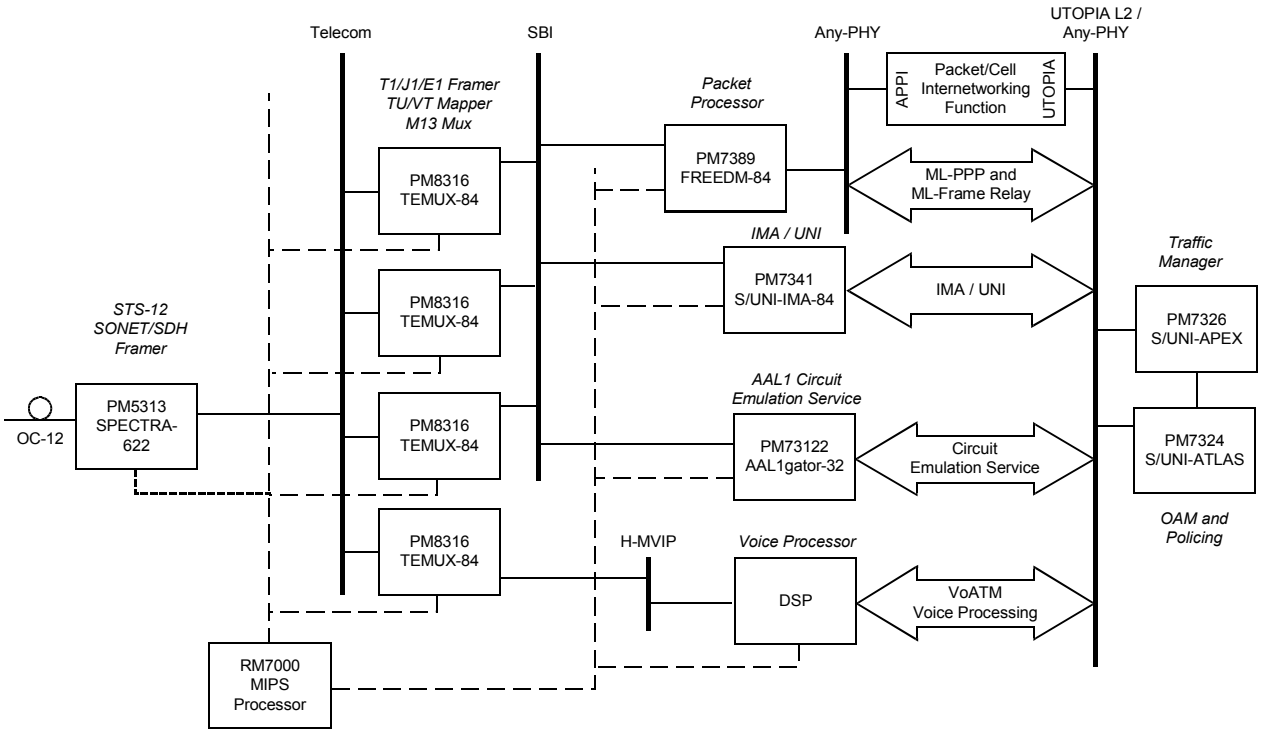
An essential function for ATM networks is to emulate existing Time Division Multiplexing (TDM) circuits. Since most voice and data services are currently provided by TDM circuits, seamless interworking between TDM and ATM has become a system requirement. The ATM Forum has standardized an internetworking function that satisfies this requirement in the Circuit Emulation Service (CES) Specification. The AAL1gator-32 is a direct implementation of that service specification in silicon, including the complex Nx64 channelized service and support of CAS.

6.1 ATM Multi-service Switch

An ATM Multi-service Switch, located at the edge of the wide area network, interfaces to Frame Relay, ATM as well as TDM services and consolidates these different services to ATM cells for transport over a single high-bandwidth ATM core network.

With the AAL1gator-32 and its support for the SBI™ bus, high density Any Service Any Port linecards for ATM Switches can be designed with PMC-Sierra's SPECTRA™, TEMUX™, AAL1gator, S/UNI -IMA, FREEDM™, S/UNI -APEX™ and S/UNI -ATLAS™ products. The design shown in Figure 1 supports a broad spectrum of existing and emerging services including Frame Relay (FR), multi-link Frame Relay, multi-link PPP, Internet Protocol (IP), Dedicated Private Line and Integrated Voice and Data.

Figure 1 Multi-service Switch Application



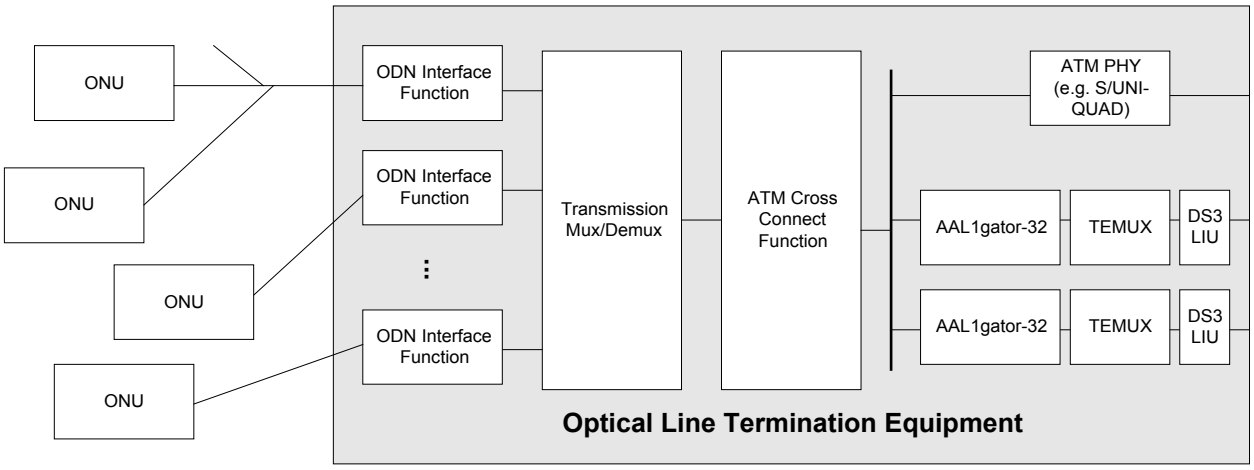
With the dramatic reduction in board space and power, the optimized AAL1gator-32 / TEMUX-84 solution enables a new generation of OC-3 and OC-12 Circuit Emulation Service and Any Service Any Port linecards.

6.2 Passive Optical Network (PON) System

The general architecture of a Passive Optical Network (PON) access network consists of two key elements: the Optical Line Termination (OLT) and the Optical Network Unit (ONU). The OLT is connected to the ONU through a point-to-multipoint Passive Optical Network that consists of fiber, splitters and other passive components. Typically, up to 32 ONUs are connected to a single OLT, depending on the splitting factor. OLTs are typically located in local exchanges and ONUs on street locations, in buildings or even in homes.

Figure 2 shows the use of the AAL1gator-32 in an OLT application supporting CES functions. Note that the PM73123 AAL1gator-8 or PM73124 AAL1gator-4 can be used in ONUs to provide the reciprocal CES functions to the OLT.

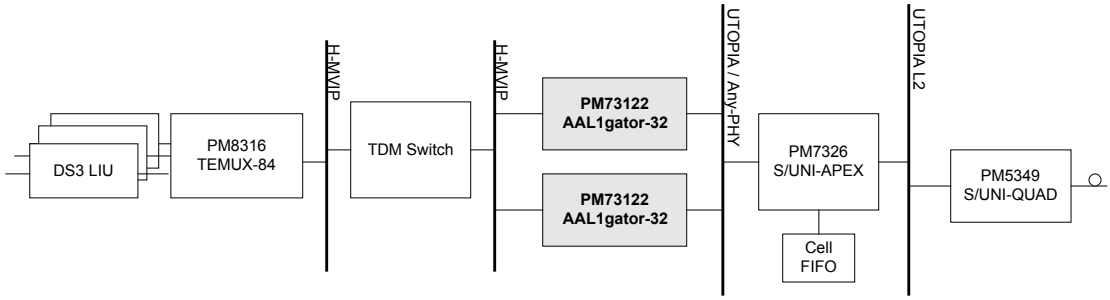
Figure 2 Using the AAL1gator-32 in an ATM Passive Optical Network.



6.3 Digital Access Cross-connect System (DACS) with an ATM Interface

Digital Access Cross-connect systems (DACS) with an ATM uplink to a core ATM switch can use one or more AAL1gator-32s to emulate a TDM service over ATM. DACs with CES capabilities allow the service providers to consolidate legacy private line services onto a high speed ATM backbone network and reduce the number of network elements and physical connections that need to be managed. Figure 3 shows the AAL1gator-32 in a DACS application.

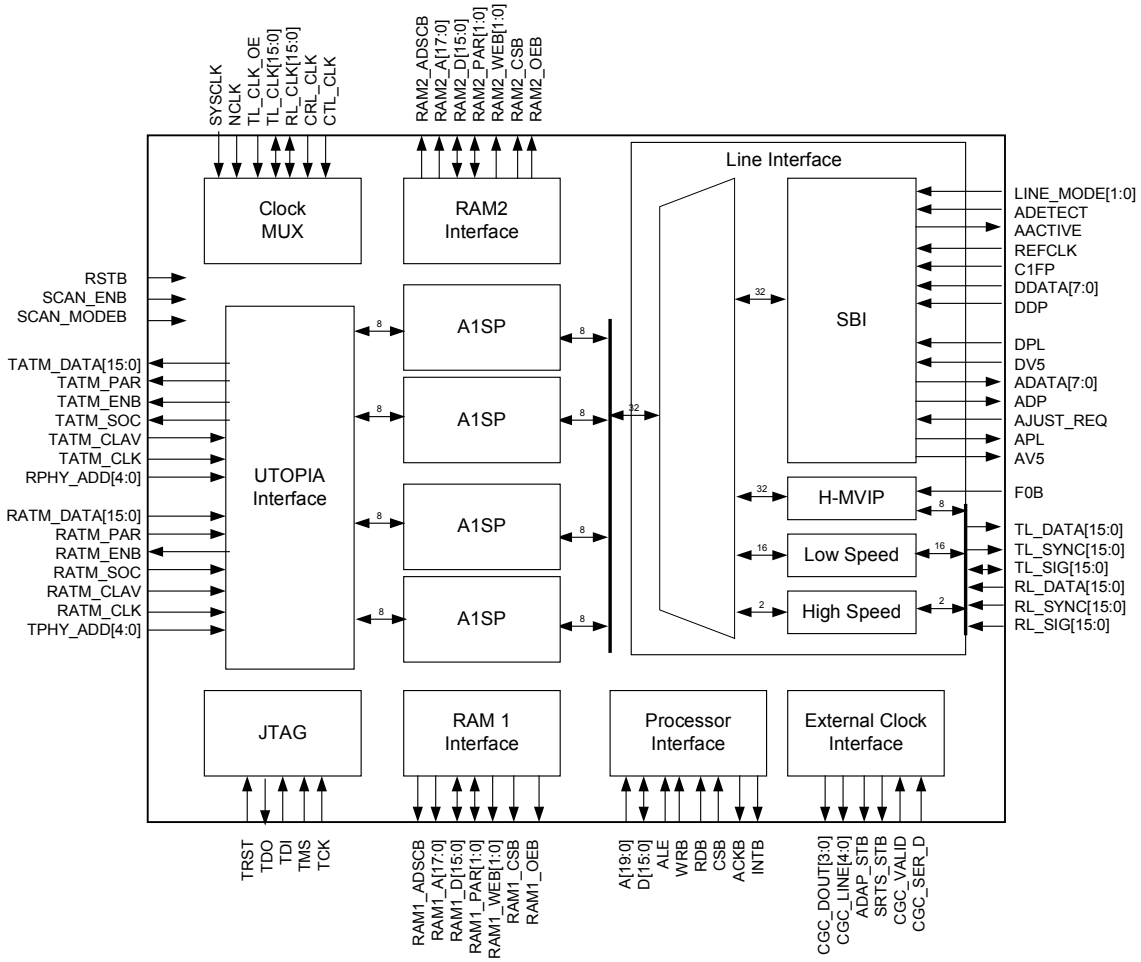
Figure 3 Using the AAL1gator-32 in a DACS application.



7 BLOCK DIAGRAM

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 several different line protocols. 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.

Figure 4 AAL1gator-32 Internal Block Diagram



8 DESCRIPTION

The AAL1 Segmentation And Reassembly (SAR) Processor (AAL1gator-32) is a monolithic single chip device that provides DS1, E1, E3, or DS3 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 device.

9 PIN DIAGRAM

The AAL1gator-32 is manufactured in a 352 pin enhanced ball grid array (SBGA) package.

	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	vss5	vss4	TMS	TDO	RAM1_AD	PCH	RAM1_AD	RAM1_AD	RAM1_AD	RAM1_AD	RL_CLK	TL_CLK	vss3	vss2	RAM1_D	RAM1_D	PCH	TL_DATA	RAM1_OE	RAM1_D	RAM1_D	TL_SYNC	TL_SYNC	RL_SIG [7]	vss1	vss0	A
B	vss9	vdd10	vss8	SYS_CLK	RAM1_AD	RAM1_AD	RAM1_AD	RAM1_AD	RAM1_AD	RAM1_PA	RAM1_W	RL_CLK	RAM1_D	RAM1_D	RAM1_D	RL_SYNC	TL_SYNC	RAM1_D	RAM1_D	CRL_CLK	RL_DATA	LINE_MO	TL_DATA	vss7	vdd9	vss6	B
C	TATM_DA	vss11	vdd12	TDI	RAM1_AD	RAM1_AD	RAM1_AD	RAM1_CS	RAM1_AD	RAM1_AD	RAM1_PA	TL_CLK	RAM1_D	RAM1_D	SCAN_EN	RL_DATA	RAM1_D	RAM1_D	CTL_CLK	RL_SYNC	TL_DATA	TL_SIG [7]	RL_SYNC	vdd11	vss10	TL_CLK	C
D	TATM_DA	TATM_DA	TATM_DA	vdd17	TCLK	RAM1_AD	RAM1_AD	RAM1_AD	vdd16	RAM1_AD	RAM1_W	RAM1_D	vdd15	RAM1_D	RL_SIG [8]	TL_SIG [8]	RAM1_D	vdd14	RL_SIG [9]	TL_SIG [9]	TL_CLK	RL_CLK	vdd13	RL_DATA	TL_SIG [6]	RL_CLK	D
E	TATM_PA	TATM_DA	RPHY_AD	TATM_DA																			TL_SYNC	TL_DATA	RL_SYNC	TL_CLK	E
F	TATM_CL	RPHY_AD	TATM_DA	TATM_DA																			RL_SIG [6]	RL_DATA	TL_SIG [5]	RL_CLK	F
G	TATM_SO	RPHY_AD	RPHY_AD	PCH																			TL_SYNC	TL_DATA	RL_SYNC	TL_SYNC	G
H	TL_CLK	RL_CLK	TATM_EN	RPHY_AD																			RL_SIG [5]	RL_DATA	TL_CLK	RL_SIG [4]	H
J	TL_DATA	TL_CLK	TL_CLK	vdd18																			vdd19	TL_SIG [4]	RL_CLK	RL_DATA	J
K	TATM_DA	TATM_DA	TL_SYNC	RL_CLK																			TL_DATA	RL_SYNC	TL_CLK	TL_DATA	K
L	PCH	TATM_DA	TATM_DA	TL_SIG																			TL_SYNC	TL_SIG [3]	TL_SIG	RL_CLK	L
M	TATM_DA	TATM_DA	TATM_DA	TATM_CL																			TL_DATA	RL_SIG	RL_SYNC	RL_DATA	M
N	Vss13	TL_CLK	RL_SYNC	RL_CLK																			vdd20	TL_SYNC	TL_SIG	vss12	N
P	vss15	RL_CLK		vdd21																			PCH	RL_SIG [3]	TL_DATA	vss14	P
R	RATM_DA	TL_CLK	RL_CLK	RL_CLK																			TL_SIG [2]	TL_SYNC	RL_DATA	RL_SYNC	R
T	TL_CLK	RATM_DA	RATM_DA	RATM_DA																			RL_DATA	RL_CLK	TL_DATA	TL_CLK	T
U	RATM_DA	RATM_EN	RATM_DA	RATM_CL																			RL_SIG [1]	TL_CLK	RL_SYNC	RL_SIG [2]	U
V	RATM_DA	RATM_DA	TPHY_AD	vdd23																			vdd22	RL_CLK	TL_SIG [1]	TL_SYNC	V
W	RATM_SO	TPHY_AD	TPHY_AD	TPHY_AD																			PCH	TL_SYNC	RL_SYNC	TL_DATA	W
Y	RATM_CL	RATM_PA	RATM_DA	RATM_DA																			RL_SYNC	RL_DATA	TL_CLK	RL_DATA	Y
AA	PCH	RATM_DA	TPHY_AD	RATM_DA																			TL_SIG [0]	RL_DATA	RL_SIG	TL_SYNC	AA
AB	RATM_DA	RATM_DA	RATM_DA	A [17]																			RL_SYNC	TL_DATA	RL_SIG	RL_SYNC	AB
AC	RATM_DA	A [19]	SCAN_MO	vdd4	D [15]	A [15]	D [11]	A [11]	vdd3	D [5]	A [4]	TL_SYNC	TL_SIG	vdd2	RDB	TL_SYNC	RL_SIG	vdd1	CGC_DOU	CGC_SER	CGC_LINE	TRSTB	vdd0	RL_DATA	RL_SIG [0]	LINE_MO	AC
AD	A [18]	vss17	vdd8	A [16]	D [12]	A [12]	D [8]	A [9]	D [6]	A [6]	RL_DATA	RL_DATA	D [1]	A [2]	ALE	ACKB	TL_SIG	TL_DATA	CGC_DOU	NCLK	CGC_LINE	CGC_LINE	RSTB	vdd5	vss16	RL_CLK	AD
AE	vss21	vdd8	vss19	D [13]	A [13]	D [9]	A [10]	D [7]	A [7]	D [3]	RL_SIG	RL_SYNC	D [0]	A [3]	A [0]	CSB	INTB	RL_DATA	TL_SIG	CGC_DOU	TL_CLK_O	CGC_LINE	ADAP_ST	vss20	vdd7	vss18	AE
AF	vss27	vss26	D [14]	A [14]	D [10]	D [2]	A [8]	D [4]	A [5]	PCH	TL_DATA	RL_SIG	vss25	vss24	A [1]	WRB	CGC_DOU	TL_DATA	RL_SYNC	TL_SYNC	PCH	CGC_VALI	CGC_LINE	SRTS_ST	vss23	vss22	AF
	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Bottom View of
352 SBGA Package
(35 mm x 35 mm)

10 PIN DESCRIPTION

10.1 UTOPIA Interface Signals (52)

Pin Name	Type	Pin No.	Function
Note signals have different meanings depending on whether the UTOPIA bus is in ATM master mode, PHY mode or Any-PHY mode. The mode is controlled by the UTOP_MODE and ANY-PHY_EN fields in the UI_SRC_CFG and UI_SNK_CFG registers. All outputs are tri-state when the chip is in reset or when UI_EN is disabled in the UI_COMN_CFG register. All outputs have a maximum output current (IMAX) = 8 mA.			
TATM_CLK/RPHY_CLK	Input	F26	**ATM**: Transmit UTOPIA ATM Layer Clock is the synchronization clock input for the TATM interface. **PHY**: Receive UTOPIA/Any-PHY PHY Layer Clock is the synchronization clock input for the RPHY interface Maximum frequency is 52 MHz.
TATM_SOC/RPHY_SOC /RSOP	Output	G26	**ATM**: Transmit UTOPIA ATM Layer Start-Of-Cell is an active high signal asserted by the AAL1gator-32 when TATM_D contains the first valid byte of the cell. **PHY**: Receive Any-PHY/UTOPIA PHY Layer Start-Of-Cell is an active high signal asserted by the AAL1gator-32 when RPHY_D[15:0] contains the first valid word of the cell. AAL1gator-32 drives this signal only when the ATM layer has selected it for a cell transfer. **Any-PHY**: This pin is the Receive Start of Packet (RSOP) signal which functions just like RPHY_SOC.

Pin Name	Type	Pin No.	Function
TATM_D[15]/RPHY_D[15] TATM_D[14]/RPHY_D[14] TATM_D[13]/RPHY_D[13] TATM_D[12]/RPHY_D[12] TATM_D[11]/RPHY_D[11] TATM_D[10]/RPHY_D[10] TATM_D[9]/RPHY_D[9] TATM_D[8]/RPHY_D[8] TATM_D[7]/RPHY_D[7] TATM_D[6]/RPHY_D[6] TATM_D[5]/RPHY_D[5] TATM_D[4]/RPHY_D[4] TATM_D[3]/RPHY_D[3] TATM_D[2]/RPHY_D[2] TATM_D[1]/RPHY_D[1] TATM_D[0]/RPHY_D[0]	Output	D24 E23 C26 D25 F23 D26 E25 F24 K25 L24 K26 L25 P24 M24 M25 M26	**ATM**: Transmit UTOPIA ATM Layer Data Bits 7 to 0 form the byte-wide data driven to the PHY layer. Bit 0 is the Least Significant Bit (LSB). Bit 7 is the Most Significant Bit (MSB) and is the first bit received for the cell from the serial line. Note that only the lower 8 bit of the bus are used in ATM master mode. **PHY**: Receive UTOPIA/Any-PHY PHY Layer Data Bits 15 to 0 form the word-wide data driven to the ATM layer. This bus only driven when the ATM layer has selected the UI_SRC_INTF for a cell transfer. The upper byte is only used if 16_BIT_MODE is set in the UI_SRC_CFG register. Otherwise the upper byte is driven to 0's. Bit 0 is the LSB. Bit 7 is the MSB of the first byte and is the first bit received for the cell from the serial line.
TATM_PAR/ RPHY_PAR	Output	E26	**ATM**: Transmit UTOPIA ATM Layer Parity is a byte parity bit covering TATM_D(7:0). **PHY**: Receive UTOPIA/Any-PHY PHY Layer Parity is either a byte parity covering RPHY_D(7:0) or word parity covering RPHY_D(15:0) depending on the value of 16_BIT_MODE.

Pin Name	Type	Pin No.	Function
TATM_ENB/RPHY_ENB /RENB	Bidi	H24	**ATM**: Transmit UTOPIA ATM Layer Enable is an active low signal asserted by the AAL1gator-32 during cycles when TATM_D contains valid data. It is not asserted until the AAL1gator-32 is ready to send a full cell. **PHY**: Receive UTOPIA/Any-PHY PHY Layer Enable is an active low signal asserted by the ATM layer to indicate RPHY_D and RPHY_SOC will be sampled at the end of the next cycle. If UTOP_MODE in UI_SRC_CFG is set to UTOPIA Level 2 Mode then the AAL1gator-32 will drive data only if RPHY_ADD matches CFG_ADDR in the UI_SRC_ADD_CFG register the cycle before RPHY_ENB goes low. **Any-PHY:** This pin is the RENB input signal, which functions the same as RPHY_ENB. The only difference is that data is driven two cycles after selection instead of just one cycle.

Pin Name	Type	Pin No.	Function
TATM_CLAV/RPHY_CLAV /RPA	Bidi	M23	**ATM**: Transmit UTOPIA ATM Layer Cell Available is an active high signal from the PHY layer device to indicate that there is sufficient room to accept a cell. **PHY**: Receive UTOPIA/Any-PHY PHY Layer Cell Available is an active high signal asserted by the AAL1gator-32 to indicate it is ready to deliver a complete cell. In Utopia Level 2 mode, this signal is driven only when MPHY_ADD matches CFG_ADDR in the UI_SRC_ADD_CFG register in the previous cycle. A pulldown resistor is recommended. **Any-PHY**: This pin is the Receive Packet Available (RPA) signal which functions the same as RPHY_CLAV except for it is activated two cycles after a matching address instead of one.
RPHY_ADD[4]/RSX RPHY_ADD[3]/RCSB RPHY_ADD[2] RPHY_ADD[1] RPHY_ADD[0]	I/O Input Input Input Input	E24 G24 F25 H23 G25	**ATM**: These signals are not used in ATM mode. **PHY**: Receive UTOPIA PHY Layer Address (Bits 4 to 0) which selects the UTOPIA receiver. These inputs are used as an output enable for RPHY_CLAV and to validate the activation of RPHY_ENB. There are internal pull-up resistors. These pins are compared with CFG_ADDR[4:0] in the UI_SRC_CFG_ADDR register. **ANY-PHY:** Receive Start Transfer(RSX) is an active high output which indicates the start of an Any-PHY packet which identifies the location of the prepended address. ANY-PHY_EN in

Pin Name	Type	Pin No.	Function
			UI_SRC_CFG register needs to be set for this function. Receive Chip Select Bar (RCSB) is an active low input which is used to select the AAL1gator-32 when polling in Any-PHY mode. This input is used to decode any Any-PHY address bits greater than RPHY_ADD[2]. This input goes low one cycle after Any-PHY address is valid. ANY-PHY_EN and CS_MODE_EN in UI_SRC_CFG register needs to be set for this function. Otherwise this bit functions as RPHY_ADD[3]. RPHY_ADD[2:0] is the bottom three bits of the Any-PHY address and is used to select the device when polling. These pins are compared with CFG_ADDR[2:0] in the UI_SRC_CFG_ADDR register. Note these pins must be tied to ground when not used.
RATM_CLK/ TPHY_CLK	Input	Y26	**ATM**: Receive UTOPIA ATM Layer Clock is the synchronization clock input for synchronizing the RATM interface. **PHY**: Transmit UTOPIA/Any-PHY PHY Layer Clock is the synchronization clock input for synchronizing the TPHY interface. Maximum frequency is 52 MHz.

Pin Name	Type	Pin No.	Function
RATM_SOC/ TPHY_SOC /TSOP	Input	W26	This signal has two definitions depending on whether the UTOPIA is in ATM mode or PHY mode. **ATM**: Receive UTOPIA ATM Layer Start-Of-Cell is an active high signal asserted by the PHY layer when RATM_D contains the first valid byte of a cell. **PHY**: Transmit UTOPIA/Any-PHY PHY Layer Start-Of-Cell is an active high signal asserted by the ATM layer when TPHY_D contains the first valid byte of a cell. **Any-PHY**: This pin is the Transmit Start of Packet (TSOP) signal which functions just like TPHY_SOC. This signal is optional in this mode. If unused, tie low.
RATM_D[15]/TPHY_D[15] RATM_D[14]/TPHY_D[14] RATM_D[13]/TPHY_D[13] RATM_D[12]/TPHY_D[12] RATM_D[11]/TPHY_D[11] RATM_D[10]/TPHY_D[10] RATM_D[9]/TPHY_D[9] RATM_D[8]/TPHY_D[8] RATM_D[7]/TPHY_D[7] RATM_D[6]/TPHY_D[6] RATM_D[5]/TPHY_D[5] RATM_D[4]/TPHY_D[4] RATM_D[3]/TPHY_D[3] RATM_D[2]/TPHY_D[2] RATM_D[1]/TPHY_D[1] RATM_D[0]/TPHY_D[0]	Input	AB24 AA23 AC26 AB25 Y23 AB26 AA25 Y24 V25 U24 V26 T23 T24 U26 T25 R26	**ATM**: Receive UTOPIA ATM Layer Data Bits 7 to 0 form the byte-wide data from the PHY layer device. Bit 0 is the LSB. Bit 7 is the MSB. This is the first bit of the cell, which will be transmitted on the serial line. The upper byte is not used in ATM mode. **PHY**: Transmit UTOPIA/Any-PHY PHY Layer Data Bits 15 to 0 form the word-wide data from the ATM layer device. Bit 0 is the LSB. Bit 7 is the MSB of the first byte. This is the first bit of the cell, which will be transmitted on the serial line. The upper byte is only used if 16_BIT_MODE is set in the UI_SNK_CFG register.

Pin Name	Type	Pin No.	Function
RATM_PAR/ TPHY_PAR	Input	Y25	**ATM**: Receive UTOPIA ATM Layer Parity is a byte odd parity bit covering RATM_D(7:0) or word odd parity covering RATM_D(15:0) depending on the value of 16_BIT_MODE. **PHY**: Transmit UTOPIA/Any-PHY PHY Layer Parity is either a byte odd parity covering TPHY_D(7:0) or word odd parity covering TPHY_D(15:0) depending on the value of 16_BIT_MODE.
RATM_ENB/TPHY_ENB	Bidi	U25	**ATM**: Receive UTOPIA ATM Layer Enable is an active low signal asserted by the AAL1gator-32 to indicate RATM_D and RATM_SOC will be sampled at the end of the next cycle. It will not be asserted until the AAL1gator-32 is ready to receive a full cell. **PHY**: Transmit UTOPIA/Any-PHY PHY Layer Enable is an active low signal asserted by the ATM layer device during cycles when TPHY_D[15:0] contain valid data. The AAL1gator-32 will accept data only if TPHY_ADD matches CFG_ADDR in the UI_SNK_CFG register the cycle before TPHY_ENB goes low **Any-PHY**: This pin is the TENB input signal, which functions the same as TPHY_ENB.

Pin Name	Type	Pin No.	Function
RATM_CLAV/TPHY_CLAV	Bidi	U23	**ATM**: Receive UTOPIA ATM Layer Cell Available is an active high signal asserted by the PHY layer to indicate that there is a cell available to send. **PHY**: Receive UTOPIA/Any-PHY PHY Layer Cell Available is an active high signal asserted by the AAL1gator-32 to indicate there is a cell-space available. The AAL1gator-32 drives this signal only when TPHY_ADD matches CFG_ADDR in the UI_SNK_CFG register in the previous cycle. A pulldown resistor is recommended. **Any-PHY**: This pin is the Transmit Packet Available (TPA) signal which functions the same as TPHY_CLAV except for it is activated two cycles after a matching address instead of one.
TPHY_ADD[4]/TSX TPHY_ADD[3]/TCSB TPHY_ADD[2] TPHY_ADD[1] TPHY_ADD[0]	Input	AA24 W23 W24 W25 V24	**ATM**: These signals are not used in ATM mode. **PHY**: Transmit UTOPIA PHY Layer Address (Bits 4 to 0) which selects the UTOPIA transmitter. These inputs are used as an output enable for TPHY_CLAV and to validate the activation of TPHY_ENB. There are internal pull-up resistors. These pins are compared with CFG_ADDR[4:0] in the UI_SNK_CFG_ADDR register. **ANY-PHY:** Transmit Start Transfer(TSX) is an active high input which indicates the start of an Any-PHY packet which identifies the location of the prepended address. ANY-PHY_EN in UI_SNK_CFG register needs to be set for this

Pin Name	Type	Pin No.	Function
			function. Transmit Chip Select Bar (TCSB) is an active low input which is used to select the AAL1gator-32 when polling in Any-PHY mode. This input is used to decode any Any-PHY address bits greater than TPHY_ADD[2]. This input goes low one cycle after Any-PHY address is valid. ANY-PHY_EN and CS_MODE_EN in UI_SNK_CFG register needs to be set for this function. Otherwise this bit functions as TPHY_ADD[3]. TPHY_ADD[2:0] is the bottom three bits of the Any-PHY address and is used to select the device when polling. These pins are compared with CFG_ADDR[2:0] in the UI_SNK_CFG_ADDR register. Note these pins must be tied to ground when not used.

10.2 Microprocessor Interface Signals (43)

Pin Name	Type	Pin No.	Function
D[15] D[14] D[13] D[12] D[11] D[10] D[9] D[8] D[7] D[6] D[5] D[4] D[3] D[2] D[1] D[0]	I/O	AC22 AF24 AE23 AD22 AC20 AF22 AE21 AD20 AE19 AD18 AC17 AF19 AE17 AF21 AD14 AE14	The bi-directional data signals (D[15:0]) provide a data bus to allow the AAL1gator-32 device to interface to an external micro-processor. Both read and write transactions are supported. The microprocessor interface is used to configure and monitor the AAL1gator-32 device. Maximum output current (IMAX) = 6 mA
A[19] A[18] A[17] A[16] A[15] A[14] A[13] A[12] A[11] A[10] A[9] A[8] A[7] A[6] A[5] A[4] A[3] A[2] A[1] A[0]	Input	AC25 AD26 AB23 AD23 AC21 AF23 AE22 AD21 AC19 AE20 AD19 AF20 AE18 AD17 AF18 AC16 AE13 AD13 AF12 AE12	The address signals (A[19:0]) provide an address bus to allow the AAL1gator-32 device to interface to an external micro-processor.

Pin Name	Type	Pin No.	Function
ALE	Input	AD12	The address latch enable signal (ALE) latches the A[19:0] signals during the address phase of a bus transaction. When ALE is set high, the address latches are transparent. When ALE is set low, the address latches hold the address provided on A[19:0]. ALE has an internal pull-up resistor.
WRB	Input	AF11	The write strobe signal (WRB) qualifies write accesses to the AAL1gator-32 device. When CSB is set low, the D[15:0] bus contents are clocked into the addressed register on the rising edge of WRB. Note that if CSB, WRB and RDB are all low, all chip outputs are tristated. Therefore WRB and RDB should never be active at the same time during functional operation.
RDB	Input	AC12	The read strobe signal (RDB) qualifies read accesses to the AAL1gator-32 device. When CSB is set low, the AAL1gator-32 device drives the D[15:0] bus with the contents of the addressed register on the falling edge of RDB. Note that if CSB, WRB and RDB are all low, all chip outputs are tristated. Therefore WRB and RDB should never be active at the same time during functional operation.
CSB	Input	AE11	The chip select signal (CSB) qualifies read/write accesses to the AAL1gator-32 device. The CSB signal must be set low during read and write accesses. When CSB is set high, the microprocessor interface signals are ignored by the AAL1gator-32 device. If CSB is not required (register accesses controlled only by WRB and RDB) then CSB should be connected to an inverted version of the RSTB signal. Note that if CSB, WRB and RDB are all low, all chip outputs are tristated.

Pin Name	Type	Pin No.	Function
ACKB	Open-Drain Output	AD11	The ACKB is an active low signal which indicates when processor read data is valid or when a processor write operation has completed. When inactive this signal is tristated. ACKB is an open drain output and should be pulled high externally with a fast resistor. Maximum output current (IMAX) = 6 mA
INTB	Open-Drain Output	AE10	The interrupt signal (INTB) is an active low signal indicating that an enabled bit in the MSTR_INTR_REG register was set. When INTB is set low, the interrupt is active and enabled. When INTB is tristate, there is no interrupt pending or it is disabled. INTB is an open drain output and should be pulled high externally with a fast resistor. Maximum output current (IMAX) = 6 mA

10.3 Ram 1 Interface Signals(41)

Pin Name	Type	Pin No.	Function
RAM1_D[15] RAM1_D[14] RAM1_D[13] RAM1_D[12] RAM1_D[11] RAM1_D[10] RAM1_D[9] RAM1_D[8] RAM1_D[7] RAM1_D[6] RAM1_D[5] RAM1_D[4] RAM1_D[3] RAM1_D[2] RAM1_D[1] RAM1_D[0]	I/O	A7 B8 C9 D10 B9 C10 A6 A11 B12 A12 D13 C13 B13 B14 C14 D15	The bi-directional data signals (RAM1_D[15:0]) provide a data bus to allow the AAL1gator-32 device to access an external 256Kx16(18) RAM. RAM1 is used for A1SP blocks 0 and 1. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
RAM1_A[17] RAM1_A[16] RAM1_A[15] RAM1_A[14] RAM1_A[13] RAM1_A[12] RAM1_A[11] RAM1_A[10] RAM1_A[9] RAM1_A[8] RAM1_A[7] RAM1_A[6] RAM1_A[5] RAM1_A[4] RAM1_A[3] RAM1_A[2] RAM1_A[1] RAM1_A[0]	Output	A18 C17 B18 A19 D17 C18 B19 A20 B20 A17 D19 C20 A22 D20 C21 B22 D21 C22	The address signals (RAM1_A[17:0]) provide an address bus to allow the AAL1gator-32 device to address an external 256Kx16(18) RAM. Maximum output current (IMAX) = 6 mA
RAM1_OEB	Output	A8	RAM1 Output Enable is an active low signal that enables the SRAM to drive data. Maximum output current (IMAX) = 6 mA.
RAM1_WEB[1]	Output	B16	RAM1 Write Enable One is an active low signal for the high-byte write. Maximum output current (IMAX) = 6 mA.
RAM1_WEB[0]	Output	D16	RAM1 Write Enable Zero is an active low signal for the low-byte write. Maximum output current (IMAX) = 6 mA.
RAM1_CSB	Output	C19	RAM1 Chip Select is an active low chip-select signal for external memory. Maximum output current (IMAX) = 6 mA.

Pin Name	Type	Pin No.	Function
RAM1_ADSCB /RAM1_R/WB	Output	B21	This signal has different meanings depending upon the type of SSRAM that the AAL1gator-32 is programmed to interface to. **Pipelined Single-Cycle Deselect SSRAM**: RAM1 Address Status Control is an active low output for external memory and is used to cause a new external address to be loaded into the RAM. **Pipelined ZBT SSRAM**: RAM1 R/W indicates the direction of the transfer. Maximum output current (IMAX) = 6 mA.
RAM1_PAR[1] RAM1_PAR[0]	I/O	C16 B17	RAM1 Parity is a two bit bi-directional signal that indicates odd parity for the upper and lower byte of RAM1_D[15:0]. Maximum output current (IMAX) = 6 mA

Note: For different modes of the line interface the I/O is redefined. For Direct Low Speed mode there are 16 pairs of bi-directional lines, which can support links up to 2.5 Mbps. For H-MVIP mode there are eight pairs of 8 Mbps bidirectional lines, which are compatible with the H-MVIP specification. For High Speed mode there are two lines, which can support unchannelized data streams up to 45 Mbps. And lastly, there is the SBI mode, which supports one SBI interface. For H-MVIP mode, high speed (HS) mode and SBI mode the upper 8 Direct Low Speed lines become the 2nd ram interface. For SBI mode the bottom 8 Direct Low Speed lines become the SBI interface.

Table 1 defines which signal tables need to be used for each possible mode. Select the mode of the line interface that will be used and refer to the tables listed. Table 2 on page 48 shows how pins are shared between the different modes.

Table 1 Line Interface Signal Table Selection

Line Mode	Line Interface Table	RAM2 Interface Table
Direct Low Speed	Direct Low Speed	No
H-MVIP	H-MVIP	Yes (if using upper 4 lines)
SBI	SBI	Yes

Line Mode	Line Interface Table	RAM2 Interface Table
High Speed	High Speed	Yes (if using 2 lines)

10.4 Line Interface Signals(Direct Low Speed)(132)

Pin Name	Type	Pin No.	Function
LINE_MODE[1] LINE_MODE[0]	Input	B5 AC1	Determines the mode of operation for the line interface: 00)Direct Low Speed Mode 01)SBI Mode 10) H-MVIP Mode 11) High Speed Mode Note: In Direct Low Speed Mode, one UDF-HS (51 Mbps) line can be supported. In High Speed Mode, two UDF-HS (51 Mbps) lines can be supported. In SBI Mode, two UDF-HS (DS3) lines can be supported.
TL_SYNC[15] TL_SYNC[14] TL_SYNC[13] TL_SYNC[12] TL_SYNC[11] TL_SYNC[10] TL_SYNC[9] TL_SYNC[8] TL_SYNC[7] TL_SYNC[6] TL_SYNC[5] TL_SYNC[4] TL_SYNC[3] TL_SYNC[2] TL_SYNC[1] TL_SYNC[0]	I/O	K24 AC15 AC11 AF7 AA1 N3 A5 B10 A4 E4 G4 G1 L4 R3 V1 W3	Transmit Line Synchronization 15 to 0 are the transmit frame synchronization indicators used in SDF-MF and SDF-FR modes. Depending on the value of MF_SYNC_MODE in the LI_CFG_REG register for the line, these signals either indicate a frame boundary or a multi-frame boundary. Depending on the value of GEN_SYNC in the LIN_STR_MODE register for that line, the sync signal is either received from the corresponding framer device 0 to 15 or it is generated internally The Default mode of this signal is to be a frame sync input. When the MVIP_EN bit is set in LS_Ln_CFG_REG then TL_SYNC[0] is the F0B pin; the common frame sync. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
TL_DATA[15] TL_DATA[14] TL_DATA[13] TL_DATA[12] TL_DATA[11] TL_DATA[10] TL_DATA[9] TL_DATA[8] TL_DATA[7] TL_DATA[6] TL_DATA[5] TL_DATA[4] TL_DATA[3] TL_DATA[2] TL_DATA[1] TL_DATA[0]	Output	J26 AF16 AF9 AD9 P2 M4 C6 A9 B4 E3 G3 K4 K1 T2 W1 AB3	Transmit Line Serial Data 15 to 0 carry the received data to the corresponding framer devices. Maximum output current (IMAX) = 6 mA
TL_SIG[15] TL_SIG[14] TL_SIG[13] TL_SIG[12] TL_SIG[11] TL_SIG[10] TL_SIG[9] TL_SIG[8] TL_SIG[7] TL_SIG[6] TL_SIG[5] TL_SIG[4] TL_SIG[3] TL_SIG[2] TL_SIG[1] TL_SIG[0]	Output	L23 AC14 AD10 AE8 N2 L2 D7 D11 C5 D2 F2 J3 L3 R4 V2 AA4	Transmit Line Signal 15 to 0 are the CAS signaling outputs to the corresponding framer devices in SDF-MF mode. This is the default function for this pin. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
TL_CLK[15]/TSM[15] TL_CLK[14]/TSM[14] TL_CLK[13]/TSM[13] TL_CLK[12]/TSM[12] TL_CLK[11]/TSM[11] TL_CLK[10]/TSM[10] TL_CLK[9]/TSM[9] TL_CLK[8]/TSM[8] TL_CLK[7]/TSM[7] TL_CLK[6]/TSM[6] TL_CLK[5]/TSM[5] TL_CLK[4]/TSM[4] TL_CLK[3]/TSM[3] TL_CLK[2]/TSM[2] TL_CLK[1]/TSM[1] TL_CLK[0]/TSM[0]	I/O	J25 N25 R25 T26 H26 J24 A15 C15 D6 C1 E1 H2 K2 T1 U3 Y2	Transmit Line Channel Clock 15 to 0 are the clock lines for the sixteen lines. They clock the data from the AAL1gator-32 to the corresponding framer devices. Depending on the value of the TLCLK_OE pin and the CLK_SOURCE_TX field in the LIN_STR_MODE memory register, these pins are either outputs or inputs. If TLCLK_OUTPUT_EN is high, these pins are outputs and the clock is sourced internally at power up. This can later be changed by the CLK_SOURCE_TX field. Note that if CLK_SOURCE_TX /= “000” then this pin is an output, even if it is not driving a clock. A clock will only be driven if in E1 or T1 mode and either the internal clock synthesizer is being used or the clock is being looped. CLK_SOURCE_TX = “001”, “010, “011”, “100”, or “101”) Note that if UDF_HS=1 in the HS_LIN_REG, TL_CLK[7:1] should be tied high. Transmit Signaling Mirror is a copy of the TL_SIG output. In Direct Low Speed mode, if CLK_SOURCE_TX=”111” then signaling is output on this pin. This option is used with devices that share the same pin for clock and signaling. In this mode CTL_CLK is used as the line clock. Maximum output current (IMAX) = 6 mA.
CTL_CLK	Input	C8	Common Transmit Line Clock is a transmit line clock which can be shared across all lines. Whether this clock is used or not for a given line is dependent on the value of CLK_SOURCE_TX in the LINE_STR_MODE memory register for that line.

Pin Name	Type	Pin No.	Function
RL_SYNC[15] RL_SYNC[14] RL_SYNC[13] RL_SYNC[12] RL_SYNC[11] RL_SYNC[10] RL_SYNC[9] RL_SYNC[8] RL_SYNC[7] RL_SYNC[6] RL_SYNC[5] RL_SYNC[4] RL_SYNC[3] RL_SYNC[2] RL_SYNC[1] RL_SYNC[0]	Input	N24 AE15 AF8 Y4 AB1 M2 C7 B11 C4 E2 G2 K3 R1 U2 W2 AB4	Receive Line Synchronization 15 to 0 are the receive frame synchronization indicators used in SDF-MF and SDF-FR modes. Depending on the value of MF_SYNC_MODE in the LI_CFG_REG register for the line, these signals either indicate a frame boundary or a multi-frame boundary. Tie to ground if unused.
RL_DATA[15] RL_DATA[14] RL_DATA[13] RL_DATA[12] RL_DATA[11] RL_DATA[10] RL_DATA[9] RL_DATA[8] RL_DATA[7] RL_DATA[6] RL_DATA[5] RL_DATA[4] RL_DATA[3] RL_DATA[2] RL_DATA[1] RL_DATA[0]	Input	AD16 AD15 AE9 AA3 Y3 M1 B6 C11 D3 F3 H3 J1 R2 T4 Y1 AC3	Receive Line Serial Data 15 to 0 carries the receive data from the corresponding framer devices.

Pin Name	Type	Pin No.	Function
RL_SIG[15] RL_SIG[14] RL_SIG[13] RL_SIG[12] RL_SIG[11] RL_SIG[10] RL_SIG[9] RL_SIG[8] RL_SIG[7] RL_SIG[6] RL_SIG[5] RL_SIG[4] RL_SIG[3] RL_SIG[2] RL_SIG[1] RL_SIG[0]	Input	AE16 AF15 AC10 AB2 AA2 M3 D8 D12 A3 F4 H4 H1 P3 U1 U4 AC2	Receive Line Signaling 15 to 0 carries the CAS data from the corresponding framer devices.
RL_CLK[15] RL_CLK[14] RL_CLK[13] RL_CLK[12] RL_CLK[11] RL_CLK[10] RL_CLK[9] RL_CLK[8] RL_CLK[7] RL_CLK[6] RL_CLK[5] RL_CLK[4] RL_CLK[3] RL_CLK[2] RL_CLK[1] RL_CLK[0]	Input	N23 P25 R24 R23 K23 H25 B15 A16 D5 D1 F1 J2 L1 T3 V3 AD1	Receive Line Clock 15 to 0 is the clock received from the corresponding framer device used to clock in RL_DATA, RL_SIG, and RL_SYNC.

Pin Name	Type	Pin No.	Function
CRL_CLK	Input	B7	Common Receive Line Clock is a receive line clock which can be shared across all lines. Whether this clock is used or not for a given line is dependent on the value of CLK_SOURCE_RX in the LIN_STR_MODE memory register for that line. When the MVIP_EN bit is set in LS_Ln_CFG_REG then this is the C4B input; the common 4.096 MHz clock.

10.5 Line Interface Signals(H-MVIP)(37)

Pin Name	Type	Pin No.	Function
LINE_MODE[1] LINE_MODE[0]	Input	B5 AC1	Determines the mode of operation for the line interface: 00) Direct Low Speed Mode 01) SBI Mode 10) H-MVIP Mode 11) High Speed Mode
F0B	Input	W3	Frame Sync 0 is the active low frame synchronization input signal used to indicate the start of a frame.
TL_DATA[7] TL_DATA[6] TL_DATA[5] TL_DATA[4] TL_DATA[3] TL_DATA[2] TL_DATA[1] TL_DATA[0]	Output	B4 E3 G3 K4 K1 T2 W1 AB3	Transmit Line Serial Data 7 to 0 carry the received data to the corresponding framer devices. or an H_MVIP backplane. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
TL_SIG[7] TL_SIG[6] TL_SIG[5] TL_SIG[4] TL_SIG[3] TL_SIG[2] TL_SIG[1] TL_SIG[0]	Output	C5 D2 F2 J3 L3 R4 V2 AA4	Transmit Line Signal 7 to 0 are the CAS signaling outputs to the corresponding framer devices in SDF-MF mode. H-MVIP does not support signaling directly, but these signals can be used to transport signaling if needed. Maximum output current (IMAX) = 6 mA
C16B	Input	C8	Clock 16 MHz is the clock used to transfer data across the H-MVIP bus. The clock runs twice as fast as the data rate. This common clock is used in both the receive and transmit direction.
RL_DATA[7] RL_DATA[6] RL_DATA[5] RL_DATA[4] RL_DATA[3] RL_DATA[2] RL_DATA[1] RL_DATA[0]	Input	D3 F3 H3 J1 R2 T4 Y1 AC3	Receive Line Serial Data 7 to 0 carries the receive data from the corresponding framer devices or H-MVIP backplane.
RL_SIG[7] RL_SIG[6] RL_SIG[5] RL_SIG[4] RL_SIG[3] RL_SIG[2] RL_SIG[1] RL_SIG[0]	Input	A3 F4 H4 H1 P3 U1 U4 AC2	Receive Line Signaling 15 to 0 carries the CAS data from the corresponding framer devices. H-MVIP does not support signaling directly, but these signals can be used to transport signaling if needed.
C4B	Input	B7	Clock 4 MHz is the clock used for generating and sampling F0B. This common clock is used in both the receive and transmit direction.

10.6 SBI Interface Signals (only used in SBI mode)(64)

Pin Name	Type	Pin No.	Function
LINE_MODE[1] LINE_MODE[0]	Input	B5 AC1	Determines the mode of operation for the line interface: 00) Direct Low Speed Mode 01) SBI Mode 10) H-MVIP Mode 11) High Speed Mode
REFCLK	Input	C8	**Reference Clock (REFCLK).** This signal is an externally generated 19.44MHz +/-50ppm clock with a nominal 50% duty cycle. Since the ADD and DROP busses are locked together this clock is common to both the add and drop sides of the SBI BUS.

Pin Name	Type	Pin No.	Function
TL_CLK[31] TL_CLK[30] TL_CLK[29] TL_CLK[28] TL_CLK[27] TL_CLK[26] TL_CLK[25] TL_CLK[24] TL_CLK[23] TL_CLK[22] TL_CLK[21] TL_CLK[20] TL_CLK[19] TL_CLK[18] TL_CLK[17] TL_CLK[16] TL_CLK[15] TL_CLK[14] TL_CLK[13] TL_CLK[12] TL_CLK[11] TL_CLK[10] TL_CLK[9] TL_CLK[8] TL_CLK[7] TL_CLK[6] TL_CLK[5] TL_CLK[4] TL_CLK[3] TL_CLK[2] TL_CLK[1] TL_CLK[0]	Input	N23 P25 R24 R23 K23 H25 B15 A16 D5 D1 F1 J2 R1 U2 W2 AB4 J25 N25 R25 T26 H26 J24 A15 C15 D6 C1 E1 H2 K2 T1 U3 Y2	Transmit Serial Clock input. When SRTS is used in DS3 mode or a clock with better jitter characteristics is desired the TL_CLK[16] and TL_CLK[0] pins should be used to connect the externally generated transmit line clock. When generating the TL_CLK with external logic the TL_CLK for all 32 lines can be accessed. These pins should only be used when CLK_SOURCE_TX = “000”. Note that if CLK_SOURCE_TX is not equal to “000”, TL_CLK[15:0] is an output and must not be driven externally. Note that if UDF_HS=1 in the A1SP0 HS_LIN_REG, TL_CLK[7:1] should be tied high, and if UDF_HS=1 in the A1SP2 HS_LIN_REG, TL_CLK[23:17] should be tied high.
RL_CLK[2] RL_CLK[0]	Input	T3 AD1	When SRTS is used in DS3 mode or a clock with better jitter characteristics is desired the RL_CLK pins should be used to connect the externally recovered receive line clock.

Pin Name	Type	Pin No.	Function
C1FP	Input	AB3	**Active High C1 Frame Pulse (C1FP).** This signal is externally generated to indicate the first C1 octet on the SBI BUS. If TWO_C1FP_EN is low then the ADD and DROP busses are locked together and this signal is common to both the ADD and DROP sides of the SBI BUS. If TWO_C1FP_EN is high, then this signal is the DROP side C1FP. This frame pulse indicator is a single REFCLK signal long and is updated on the rising edge of REFCLK. This signal is sampled on the rising edge of REFCLK. This signal also indicates multiframe alignment which occurs every 4 frames, therefore this signal is pulsed once every fourth C1 octet to produce a 2KHz multiframe signal. The frame pulse does not need to be repeated every 2KHz. The AAL1gator-32 will synchronize to this signal and is also be able to flywheel in its absence. When any tributary on the SBI bus is in synchronous mode the C1FP signal is used to indicate T1 and E1 multiframe alignment and must be pulsed on 48 SBI frame boundaries.
C1FP_ADD	Input	B4	**C1 Frame Pulse for Add bus.** This pin can optionally be used as the C1FP pulse for the Add bus if the Add bus and Drop bus need to be offset from each other. To use this pin the TWO_C1FP_EN bit must be set in the SBI_BUS_CFG_REG.
DDATA[7] DDATA[6] DDATA[5] DDATA[4] DDATA[3] DDATA[2] DDATA[1] DDATA[0]	Input	D2 E3 F2 G3 J3 K4 L3 K1	**Drop Bus Data (DDATA[7:0]).** The Drop data bus is a time division multiplexed bus which transports tributaries by assigning them to fixed octets within the SBI BUS structure. Multiple PHY devices can drive this bus at uniquely assigned tributary columns within the SBI BUS structure. DDATA[7:0] is sampled on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
DDP	Input	R4	**Drop Bus Data Parity (DDP).** This signal carries the even or odd parity for the drop bus signals. The parity calculation encompasses DDATA[7:0], DPL and DV5 signals. The selection of even or odd parity is made via SBI_PAR_CTL bit of Extract Control Register. Multiple PHY devices can drive this signal at uniquely assigned tributary columns within the SBI BUS structure. This parity signal is intended to detect multiple sources in the column assignment. DDP is sampled on the rising edge of REFCLK.
DPL	Input	V2	**Active High Drop Bus Payload (DPL).** This active high signal indicates valid data within the SBI BUS structure. This signal is asserted during all octets making up a tributary. This signal goes high during the V3 octet within a tributary to accommodate negative timing adjustments between the tributary rate and the fixed SBI BUS structure. This signal goes low during the octet after the V3 octet within a tributary to accommodate positive timing adjustments between the tributary rate and the fixed SBI BUS structure. Multiple PHY devices can drive this signal at uniquely assigned tributary columns within the SBI BUS structure. DPL is sampled on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
DV5	Input	T2	**Active High Drop Bus Payload Indicator (DV5).** This active high signal locates the position of the floating payloads for each tributary within the SBI BUS structure. Timing differences between the port timing and the SBI BUS timing are indicated by adjustments of this payload pointer relative to the fixed SBI BUS structure. Multiple PHY devices can drive this signal at uniquely assigned tributary columns within the SBI BUS structure. All movements indicated by this signal must be accompanied by appropriate adjustments in the DPL signal. DV5 is sampled on the rising edge of REFCLK.
ADATA[7] ADATA[6] ADATA[5] ADATA[4] ADATA[3] ADATA[2] ADATA[1] ADATA[0]	Output	G2 H3 H1 K3 J1 P3 R2 U1	**Add Data (ADATA[7:0]).** The Add data bus is a time division multiplexed bus which transports tributaries by assigning them to fixed octets within the SBI BUS structure. The AAL1gator-32 drives ADATA[7:0] only at uniquely assigned tributary columns within the SBI BUS structure. ADATA[7:0] is asserted on the rising edge of REFCLK. Maximum output current (IMAX) = 8 mA
ADP	Output	T4	**Add Bus Data Parity (ADP).** This signal carries the even or odd parity for the add bus signals. The parity calculation encompasses ADATA[7:0], APL and AV5 signals. The selection of even or odd parity is made via SBI_PAR_CTL bit of Insert Control Register The AAL1gator drives ADP only at uniquely assigned tributary columns within the SBI BUS structure. ADP is asserted on the rising edge of REFCLK. Maximum output current (IMAX) = 8 mA

Pin Name	Type	Pin No.	Function
APL	Output	U4	**Active High Add Bus Payload (APL).** This active high signal indicates valid data within the SBI BUS structure. This active high signal is asserted during all octets making up a tributary. This signal goes high during the V3 or H3 octet within a tributary to accommodate negative timing adjustments between the tributary rate and the fixed SBI BUS structure. This signal goes low during the octet after the V3 or H3 octet within a tributary to accommodate positive timing adjustments between the tributary rate and the fixed SBI BUS structure. The AAL1gator-32 drives APL only at uniquely assigned tributary columns within the SBI BUS structure. APL is asserted on the rising edge of REFCLK. Maximum output current (IMAX) = 8 mA
AV5	Output	Y1	**Active High Add Bus Payload Indicator (AV5).** This active high signal locates the position of the floating payload for each tributary within the add bus structure. The AAL1gator-32 drives AV5 only at uniquely assigned tributary columns within the SBI BUS structure. All movements indicated by this signal are accompanied by appropriate adjustments in the APL signal. AV5 is asserted on the rising edge of REFCLK. Maximum output current (IMAX) = 8 mA

Pin Name	Type	Pin No.	Function
AJUST_REQ	Input	W1	**Active High Add Bus Justification Request (AJUST_REQ).** This signal is used to speed up or slow down the AAL1gator-32 which is sending data to the PHY. This signal is only used when the PHY layer device is the timing master for the transmit direction. This active high signal indicates negative timing adjustments when asserted high during the V3 or H3 octet, depending on the tributary type. In response to this the AAL1gator-32 will send an extra byte in the V3 or H3 octet of the next frame. This signal indicates positive timing adjustments when asserted high during the octet following the V3 or H3 octet, depending on the tributary type. The AAL1gator-32 will respond to this by not sending an octet during the V3 or H3 octet of the next frame. All timing adjustments from the AAL1gator-32 in response to the justification request will still set the payload and payload indicators appropriately for timing adjustments. In synchronous T1 and E1 modes this signal is unused and must be held low. AJUST_REQ is sampled on the rising edge of REFCLK.
AACTIVE	Output	AC3	**Add Bus Active Indicator (AACTIVE).** This active high signal is asserted high during all octets when driving data and control signals, ADATA[7:0], ADP, APL and AV5, onto the bus. All other SBI Link Layer devices (e.g. other AAL1gator-32 on the common SBI bus) driving the bus listen to this signal to detect multiple sources driving the bus which can occur due to configuration problems AACTIVE is asserted on the rising edge of REFCLK. Maximum output current (IMAX) = 8 mA

Pin Name	Type	Pin No.	Function
ADETECT	Input	AA4	**Add Bus Active Detector (ADETECT).** This input listens to the OR of all other SBI Link Layer bus masters. The AAL1gator-32 will listen to the OR of all other Link Layer AACTIVE signals. When the AAL1gator-32 is driving AACTIVE high and detects ADETECT is high from another device it backs off driving the bus to minimize or eliminate contention. ADETECT is an asynchronous signal which is used to disable the tristate drivers on the ADD bus. This input must be tied low when not used.

10.7 Line Interface Signals(High Speed)(10)

Pin Name	Type	Pin No.	Function
LINE_MODE[1] LINE_MODE[0]	Input	B5 AC1	Determines the mode of operation for the line interface: 00) Direct Low Speed Mode 01) SBI Mode 10) H-MVIP Mode 11) High Speed Mode
TL_DATA[2] TL_DATA[0]	Output	T2 AB3	Transmit Line Serial Data 2 and 0 carry the received data to the corresponding framer devices. Maximum output current (IMAX) = 6 mA
TL_CLK[2] TL_CLK[0]	I/O	T1 Y2	Transmit Line Channel Clock 2 and 0 are the clock lines for the two high speed lines. They clock the data from the AAL1gator-32 to the corresponding framer devices. The clock is always an input in high speed mode. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
RL_DATA[2] RL_DATA[0]	Input	T4 AC3	Receive Line Serial Data 2 and 0 carries the receive data from the corresponding framer devices.
RL_CLK[2] RL_CLK[0]	Input	T3 AD1	Receive Line Clock 2 and 0 is the clock received from the corresponding framer device used to clock in RL_DATA[2] and RL_DATA[0].

10.8 RAM 2 Interface Signals (only used in H-MVIP, HS, and SBI modes)(41)

Pin Name	Type	Pin No.	Function
RAM2_D[15] RAM2_D[14] RAM2_D[13] RAM2_D[12] RAM2_D[11] RAM2_D[10] RAM2_D[9] RAM2_D[8] RAM2_D[7] RAM2_D[6] RAM2_D[5] RAM2_D[4] RAM2_D[3] RAM2_D[2] RAM2_D[1] RAM2_D[0]	I/O	AC11 AF9 AD10 AD9 AE8 AF7 AA1 P2 N3 M4 L2 C6 D7 A5 A9 D11	The bi-directional data signals (RAM2_D[15:0]) provide a data bus to allow the AAL1gator-32 device to access an external 256Kx16(18) RAM. RAM2 is used for A1SP blocks 2 and 3. Maximum output current (IMAX) = 6 mA

Pin Name	Type	Pin No.	Function
RAM2_A[17] RAM2_A[16] RAM2_A[15] RAM2_A[14] RAM2_A[13] RAM2_A[12] RAM2_A[11] RAM2_A[10] RAM2_A[9] RAM2_A[8] RAM2_A[7] RAM2_A[6] RAM2_A[5] RAM2_A[4] RAM2_A[3] RAM2_A[2] RAM2_A[1] RAM2_A[0]	Output	AD16 AE16 AD15 AE15 AF15 AE9 AF8 AC10 AB2 AA3 Y4 AB1 AA2 Y3 M1 M2 M3 B6	The address signals (RAM2_A[17:0]) provide an address bus to allow the AAL1gator-32 device to address an external 256Kx16(18) RAM. Maximum output current (IMAX) = 6 mA
RAM2_OEB	Output	C7	RAM2 Output Enable is an active low signal that enables the SSRAM to drive data. Maximum output current (IMAX) = 6 mA.
RAM2_WEB[1]	Output	D8	RAM2 Write Enable One is an active low signal for the high-byte write. Maximum output current (IMAX) = 6 mA.
RAM2_WEB[0]	Output	C11	RAM2 Write Enable Zero is an active low signal for the low-byte write. Maximum output current (IMAX) = 6 mA.
RAM2_CSB	Output	B11	RAM2 Chip Select is an active low chip-select signal for external memory. Maximum output current (IMAX) = 6 mA.

Pin Name	Type	Pin No.	Function
RAM2_ADSCB RAM2_R/WB	Output	D12	This signal has different meanings depending upon the type of SSRAM that the AAL1gator-32 is programmed to interface to. **Pipelined Single-Cycle Deselect SSRAM**: RAM2 Address Status Control is an active low output for external memory and is used to cause a new external address to be loaded into the RAM. **Pipelined ZBT SSRAM**: RAM2 R/W indicates the direction of the transfer. Maximum output current (IMAX) = 6 mA.
RAM2_PAR[1] RAM2_PAR[0]	I/O	N2 B10	RAM2 Parity is a two bit bi-directional signal that indicates odd parity for the upper and lower byte of RAM2_D[15:0]. Maximum output current (IMAX) = 6 mA

10.9 Summary of Line Interface Signals

The following table shows all modes at the same time and shows how pins are redefined for the different modes.

Table 2 Line Interface Summary

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
TL_SYNC[15]				K24
TL_SYNC[14]				AC15
TL_SYNC[13]	RAM2_D[15]	RAM2_D[15]	RAM2_D[15]	AC11
TL_SYNC[12]	RAM2_D[10]	RAM2_D[10]	RAM2_D[10]	AF7
TL_SYNC[11]	RAM2_D[9]	RAM2_D[9]	RAM2_D[9]	AA1
TL_SYNC[10]	RAM2_D[7]	RAM2_D[7]	RAM2_D[7]	N3
TL_SYNC[9]	RAM2_D[2]	RAM2_D[2]	RAM2_D[2]	A5
TL_SYNC[8]	RAM2_P[0]	RAM2_P[0]	RAM2_P[0]	B10
TL_SYNC[7]				A4

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
TL_SYNC[6]				E4
TL_SYNC[5]				G4
TL_SYNC[4]				G1
TL_SYNC[3]				L4
TL_SYNC[2]				R3
TL_SYNC[1]				V1
TL_SYNC[0]	F0B			W3
TL_DATA[15]				J26
TL_DATA[14]				AF16
TL_DATA[13]	RAM2_D[14]	RAM2_D[14]	RAM2_D[14]	AF9
TL_DATA[12]	RAM2_D[12]	RAM2_D[12]	RAM2_D[12]	AD9
TL_DATA[11]	RAM2_D[8]	RAM2_D[8]	RAM2_D[8]	P2
TL_DATA[10]	RAM2_D[6]	RAM2_D[6]	RAM2_D[6]	M4
TL_DATA[9]	RAM2_D[4]	RAM2_D[4]	RAM2_D[4]	C6
TL_DATA[8]	RAM2_D[1]	RAM2_D[1]	RAM2_D[1]	A9
TL_DATA[7]	TL_DATA[7]	C1FP_ADD		B4
TL_DATA[6]	TL_DATA[6]	DDATA[6]		E3
TL_DATA[5]	TL_DATA[5]	DDATA[4]		G3
TL_DATA[4]	TL_DATA[4]	DDATA[2]		K4
TL_DATA[3]	TL_DATA[3]	DDATA[0]		K1
TL_DATA[2]	TL_DATA[2]	DV5	TL_DATA[2]	T2
TL_DATA[1]	TL_DATA[1]	AJUST_REQ		W1
TL_DATA[0]	TL_DATA[0]	C1FP	TL_DATA[0]	AB3
TL_SIG[15]				L23
TL_SIG[14]				AC14
TL_SIG[13]	RAM2_D[13]	RAM2_D[13]	RAM2_D[13]	AD10
TL_SIG[12]	RAM2_D[11]	RAM2_D[11]	RAM2_D[11]	AE8
TL_SIG[11]	RAM2_P[1]	RAM2_P[1]	RAM2_P[1]	N2

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
TL_SIG[10]	RAM2_D[5]	RAM2_D[5]	RAM2_D[5]	L2
TL_SIG[9]	RAM2_D[3]	RAM2_D[3]	RAM2_D[3]	D7
TL_SIG[8]	RAM2_D[0]	RAM2_D[0]	RAM2_D[0]	D11
TL_SIG[7]	TL_SIG[7]			C5
TL_SIG[6]	TL_SIG[6]	DDATA[7]		D2
TL_SIG[5]	TL_SIG[5]	DDATA[5]		F2
TL_SIG[4]	TL_SIG[4]	DDATA[3]		J3
TL_SIG[3]	TL_SIG[3]	DDATA[1]		L3
TL_SIG[2]	TL_SIG[2]	DDP		R4
TL_SIG[1]	TL_SIG[1]	DPL		V2
TL_SIG[0]	TL_SIG[0]	ADETECT		AA4
TL_CLK[15]		TL_CLK[15]		J25
TL_CLK[14]		TL_CLK[14]		N25
TL_CLK[13]		TL_CLK[13]		R25
TL_CLK[12]		TL_CLK[12]		T26
TL_CLK[11]		TL_CLK[11]		H26
TL_CLK[10]		TL_CLK[10]		J24
TL_CLK[9]		TL_CLK[9]		A15
TL_CLK[8]		TL_CLK[8]		C15
TL_CLK[7]		TL_CLK[7]		D6
TL_CLK[6]		TL_CLK[6]		C1
TL_CLK[5]		TL_CLK[5]		E1
TL_CLK[4]		TL_CLK[4]		H2
TL_CLK[3]		TL_CLK[3]		K2
TL_CLK[2]		TL_CLK[2]	TL_CLK[2]	T1
TL_CLK[1]		TL_CLK[1]		U3
TL_CLK[0]		TL_CLK[0]	TL_CLK[0]	Y2
CTL_CLK	C16B	REFCLK		C8

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
RL_SYNC[15]				N24
RL_SYNC[14]	RAM2_A[14]	RAM2_A[14]	RAM2_A[14]	AE15
RL_SYNC[13]	RAM2_A[11]	RAM2_A[11]	RAM2_A[11]	AF8
RL_SYNC[12]	RAM2_A[7]	RAM2_A[7]	RAM2_A[7]	Y4
RL_SYNC[11]	RAM2_A[6]	RAM2_A[6]	RAM2_A[6]	AB1
RL_SYNC[10]	RAM2_A[2]	RAM2_A[2]	RAM2_A[2]	M2
RL_SYNC[9]	RAM2_OEB	RAM2_OEB	RAM2_OEB	C7
RL_SYNC[8]	RAM2_CSB	RAM2_CSB	RAM2_CSB	B11
RL_SYNC[7]				C4
RL_SYNC[6]				E2
RL_SYNC[5]		ADATA[7]		G2
RL_SYNC[4]		ADATA[4]		K3
RL_SYNC[3]		TL_CLK[19]		R1
RL_SYNC[2]		TL_CLK[18]		U2
RL_SYNC[1]		TL_CLK[17]		W2
RL_SYNC[0]		TL_CLK[16]		AB4
RL_DATA[15]	RAM2_A[17]	RAM2_A[17]	RAM2_A[17]	AD16
RL_DATA[14]	RAM2_A[15]	RAM2_A[15]	RAM2_A[15]	AD15
RL_DATA[13]	RAM2_A[12]	RAM2_A[12]	RAM2_A[12]	AE9
RL_DATA[12]	RAM2_A[8]	RAM2_A[8]	RAM2_A[8]	AA3
RL_DATA[11]	RAM2_A[4]	RAM2_A[4]	RAM2_A[4]	Y3
RL_DATA[10]	RAM2_A[3]	RAM2_A[3]	RAM2_A[3]	M1
RL_DATA[9]	RAM2_A[0]	RAM2_A[0]	RAM2_A[0]	B6
RL_DATA[8]	RAM2_WEB[0]	RAM2_WEB[0]	RAM2_WEB[0]	C11
RL_DATA[7]	RL_DATA[7]			D3
RL_DATA[6]	RL_DATA[6]			F3
RL_DATA[5]	RL_DATA[5]	ADATA[6]		H3
RL_DATA[4]	RL_DATA[4]	ADATA[3]		J1

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
RL_DATA[3]	RL_DATA[3]	ADATA[1]		R2
RL_DATA[2]	RL_DATA[2]	ADP	RL_DATA[2]	T4
RL_DATA[1]	RL_DATA[1]	AV5		Y1
RL_DATA[0]	RL_DATA[0]	AACTIVE	RL_DATA[0]	AC3
RL_SIG[15]	RAM2_A[16]	RAM2_A[16]	RAM2_A[16]	AE16
RL_SIG[14]	RAM2_A[13]	RAM2_A[13]	RAM2_A[13]	AF15
RL_SIG[13]	RAM2_A[10]	RAM2_A[10]	RAM2_A[10]	AC10
RL_SIG[12]	RAM2_A[9]	RAM2_A[9]	RAM2_A[9]	AB2
RL_SIG[11]	RAM2_A[5]	RAM2_A[5]	RAM2_A[5]	AA2
RL_SIG[10]	RAM2_A[1]	RAM2_A[1]	RAM2_A[1]	M3
RL_SIG[9]	RAM2_WEB[1]	RAM2_WEB[1]	RAM2_WEB[1]	D8
RL_SIG[8]	RAM2_ADSCB	RAM2_ADSCB	RAM2_ADSCB	D12
RL_SIG[7]	RL_SIG[7]			A3
RL_SIG[6]	RL_SIG[6]			F4
RL_SIG[5]	RL_SIG[5]			H4
RL_SIG[4]	RL_SIG[4]	ADATA[5]		H1
RL_SIG[3]	RL_SIG[3]	ADATA[2]		P3
RL_SIG[2]	RL_SIG[2]	ADATA[0]		U1
RL_SIG[1]	RL_SIG[1]	APL		U4
RL_SIG[0]	RL_SIG[0]			AC2
RL_CLK[15]		TL_CLK[31]		N23
RL_CLK[14]		TL_CLK[30]		P25
RL_CLK[13]		TL_CLK[29]		R24
RL_CLK[12]		TL_CLK[28]		R23
RL_CLK[11]		TL_CLK[27]		K23
RL_CLK[10]		TL_CLK[26]		H25
RL_CLK[9]		TL_CLK[25]		B15
RL_CLK[8]		TL_CLK[24]		A16

Direct Low Speed	H-MVIP	SBI	High Speed	Pin
RL_CLK[7]		TL_CLK[23]		D5
RL_CLK[6]		TL_CLK[22]		D1
RL_CLK[5]		TL_CLK[21]		F1
RL_CLK[4]		TL_CLK[20]		J2
RL_CLK[3]				L1
RL_CLK[2]		RL_CLK[2]	RL_CLK[2]	T3
RL_CLK[1]				V3
RL_CLK[0]		RL_CLK[0]	RL_CLK[0]	AD1
CRL_CLK	C4B			B7

10.10 Clock Generation Control Interface(18)

Pin Name	Type	Pin No.	Function
CGC_DOUT[3] CGC _DOUT[2] CGC _DOUT[1] CGC _DOUT[0]	Output	AC8 AF10 AE7 AD8	External Clock Generation Control Data Out Bits 3 to 0 form the SRTS correction code when SRTS_STBH is asserted; otherwise CGC_DOUT[3:0] bits form the channel status and frame difference when ADAP_STBH is asserted.
CGC _LINE[4] CGC _LINE[3] CGC _LINE[2] CGC _LINE[1] CGC _LINE[0]	Output	AD5 AC6 AF4 AE5 AD6	CGC Line Bits 4 to 0 form the line CGC_DOUT corresponds to when SRTS_STBH is asserted; otherwise CGC_LINE[4:0] bits form the adaptive state machine index when ADAP_STBH is asserted.
SRTS_STBH	Output	AF3	SRTS Strobe indicates that an SRTS value is present on CGC_DOUT[3:0]. CGC_LINE[4:0] indicates the line the SRTS code controls.
ADAP_STBH	Output	AE4	Adaptive Strobe indicates that the channel status and byte difference are being played out on the CGC_DOUT[3:0]. The nibbles are identified by the values on CGC_LINE[4:0].

Pin Name	Type	Pin No.	Function
NCLK/ SRTS_DISB	Input	AD7	Network Clock is the ATM network-derived clock used for SRTS. If this signal is tied low, SRTS is disabled. Internally this clock can be divided independently for each A1SP block. This clock should be 2.43 MHz for T1 and E1mode, 38.88 MHz for E3 mode and 77.76 MHz for DS3 mode.
TL_CLK_OE	Input	AE6	Transmit Line Clock Output Enable controls whether or not the TL_CLK lines are inputs or outputs between the time of hardware reset and when the CLK_SOURCE_TX bits are read. If high, all TL_CLK pins are outputs. If low, all TL_CLK pins are inputs. There is an internal pull-up resistor, so all TL_CLK pins are outputs if the pin is not connected. The value of this input is overwritten by the CLK_SOURCE_TX bits in the LIN_STR_MODE memory register.
CGC_SER_D	Input	AC7	External Clock Generation Control Serial Data is an input used to allow external clock control circuitry to pass frequency information into the internal clock synthesizer.
CGC_VALID	Input	AF5	External Clock Generation Control Valid signal is an active high input indicating that the data on CGC_SER_D is valid. This signal must transition from a low to a high at the first valid data on CGC_SER_D and must stay high through the whole clock control word.

10.11 JTAG/TEST Signals(5)

Pin Name	Type	Pin No.	Function
TCLK	Input	D22	The test clock signal provides timing for test operations tat can be carried out using the JTAG test access port.
TMS	Input Internal Pull-up	A24	The test mode select signal controls the test operations that can be carried out using the JTAG test access port. Maintain TMS tied high when not using JTAG logic.

Pin Name	Type	Pin No.	Function
TDI	Input Internal Pull-up	C23	The test data input signal is JTAG serial input data
TDO	Output	A23	The test data output signal is JTAG serial output data.
SCAN_ENB	Input Internal Pull-up	C12	An active low signal which, in SCAN mode, is used to shift data. This signal should be tied high for normal operation.
SCAN_MODEB	Input Internal Pull-up	AC24	When tied low enable SCAN mode. This signal should be tied high for normal operation.
TRSTB	Schmitt Trigger Input Internal Pull-up	AC5	The active low test reset signal is an asynchronous reset for the JTAG circuitry. If JTAG logic will be used, one option is to connect TRSTB to the RSTB input, and keep TMS tied high while RSTB is high; this maintains the JTAG logic in reset during normal operation. If JTAG logic will not be used, use the option described above, or simply ground TRSTB.

10.12 General Signals(3+power/gnd)

Pin Name	Type	Pin No.	Function
RSTB	Schmitt Trigger Input Internal Pull-up	AD4	Reset is an active low asynchronous hardware reset. When RSTB is forced low, all of the AAL1gator's internal registers are reset to their default states.

Pin Name	Type	Pin No.	Function
SYS_CLK	Input	B23	System Clock. The maximum frequency is 45 MHz. This clock is used to clock the majority of the logic inside the chip and also determines the speed of the memory interface and the external clock control interface. This clock is also used for clock synthesis. When clock synthesis is enabled this clock must be 38.88 MHz.
VDD3.3 (PPH, PQH)	Power	AE2 AE25 B2 B25 C3 C24 D4 D9 D14 D18 D23 J4 J23 N4 P23 V4 V23 AC4 AC9 AC13 AC18 AC23 AD3 AD24	**Power (VDD3.3).** The VDD3.3 pins should be connected to a well decoupled +3.3V DC power supply. These pins power the output ports of the device. PQH pins are “quiet” power pads.
VDD2.5 (PCH)	Power	A10 A21 G23 P4 L26 AA26 W4 AF6 AF17	**Power (VDD2.5).** The VDD2.5 pins should be connected to a well decoupled +2.5V DC power supply. These pins power the core of the device.

Pin Name	Type	Pin No.	Function
VSS (PPL, PQL, PCL)	Ground	A1 A2 A13 A14 A25 A26 B1 B3 B24 B26 C2 C25 N1 N26 P1 P26 AD2 AD25 AE1 AE3 AE24 AE26 AF1 AF2 AF13 AF14 AF25 AF26	**Ground (VSS).** The VSS pins should be connected to GND. PPL pins are ground pins for ports. PQL pins are "quiet" ground pins for ports. PCL pins are core ground pins. All grounds should be connected together.

Notes on Pin Description:

- All AAL1gator-32 inputs and bi-directionals present minimum capacitive loading and are 5V tolerant.
- The AAL1gator-32 SBI and UTOPIA/Any-PHY outputs and bi-directional pins have 8 mA drive capability. TDO, the CGC bus outputs and microprocessor bus outputs and bi-directional pins have 4 mA drive capability. Any other outputs and bi-directional pins have 6 mA drive capability.
- All AAL1gator-32 outputs can be tristated under control of the IEEE P1149.1 test access port, even those which do not tristate under normal operation. All outputs and bi-directionals are 5 V tolerant when tristated.

- All clock inputs (except TL_CLK) are Schmitt triggered. Inputs RPHY_ADD_RSX, RL_DATA[15:0], RL_SIG[15:8], RPHY_ADDR[3:0], TPHY_ADDR[4:0], RL_CLK[15:0], RL_SYNC[15:8,3:1], TL_SYNC[15:8], TL_CLK[15:0], RATM_DATA[15:0], RATM_PAR, RATM_CLK, RATM_SOC, TATM_CLK, D[15:0], RAM1_PAR[1:0], WRB, CSB, RDB, NCLK, CRL_CLK, CTL_CLK, SCAN_ENB, SCAN_MODEB, CGC_SER_D, CGC_VALID, RSTB, ALE, TL_CLK_OE, TMS, TCLK, TDI and TRSTB have internal pull-up resistors.

- Power to the VDD3.3 pins should be applied *before* power to the VDD2.5 pins is applied. Similarly, power to the VDD2.5 pins should be removed *before* power to the VDD3.3 pins is removed.

11 FUNCTIONAL DESCRIPTION

The AAL1gator-32 is divided into the following major blocks, all of which are explained in this section:

- UTOPIA Interface Block (UTOPIAI)
- AAL1 SAR Processing Block (A1SP)
- Processor Interface Block (PROCI)
- RAM Interface Block (RAMI)
- Line Interface Block (LINEI)
- JTAG

11.1 UTOPIA Interface Block (UI)

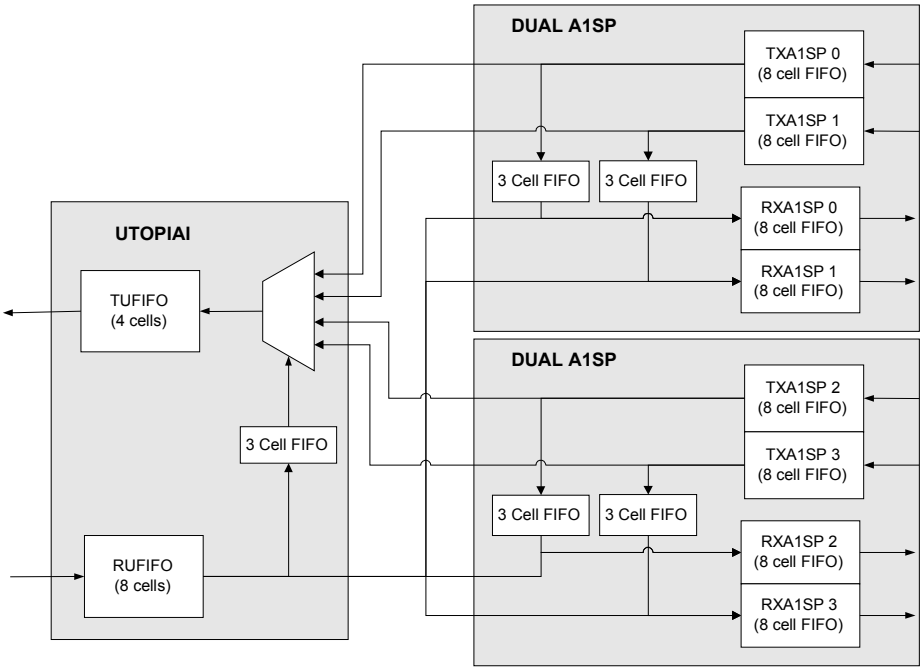
The UI manages and responds to all control signals on the UTOPIA bus and passes cells to and from the UTOPIA bus and the two Dual A1SP blocks. Both 8-bit and 16-bit UTOPIA interfaces with an optional single parity bit are supported. Each direction can be configured independently and has its own address configuration register.

The following UTOPIA modes are supported.

- UTOPIA Level One Master (8-bit only)
- UTOPIA Level One PHY
- UTOPIA Level Two PHY
- Any-PHY PHY

In the sink direction, the UI uses a 8-cell deep FIFO for buffering cells as they wait to be sent to the Dual A1SP blocks. In addition, each Dual A1SP contains two 8-cell deep FIFOs (one per A1SP) with separate interfaces to allow each A1SP to process data at its own pace. In the source direction, the UI uses a 4-cell deep FIFOs for holding cells before they are sent out onto the UTOPIA bus. Also, each Dual A1SP contains two 8-cell deep FIFOs (one per A1SP), again with separate interfaces. The data flow showing the FIFOs is shown in Figure 5.

Figure 5 Data Flow and Buffering in the UI and Dual A1SP Blocks



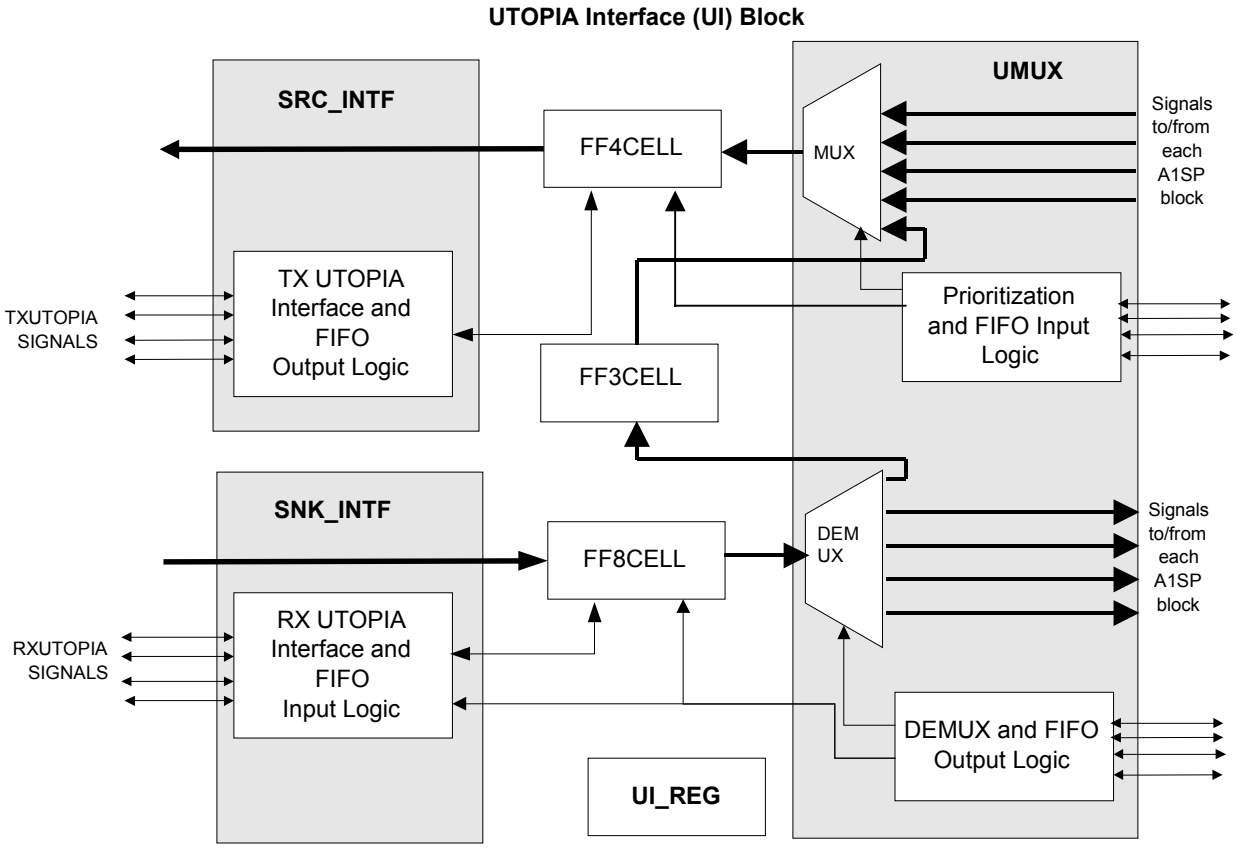
In UTOPIA Level Two mode, the AAL1gator-32 generally responds on the UTOPIA bus as a single port device. However, it is possible to configure the sink direction as a 4-port device where each A1SP is a different port.

For UTOPIA to UTOPIA loopback, there is a 3-cell FIFO in the UI Block. Line-side to Line-side loopback is done in the A1SP Blocks.

The UI_EN bit in the UI_COMN_CFG register 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 UI block consists of 7 functions: UI Data Source Interface (SRC_INTF), UI Data Sink Interface(SNK_INTF), 8-cell FIFO (FF8CELL), 4-cell FIFO (FF4CELL), 3-cell FIFO (FF3CELL), UMUX, and UI_REG. See Figure 6 for the block diagram of the AAL1_UI block.

Figure 6 UI Block Diagram



11.1.1 UTOPIA Source Interface (SRC_INTF)

The SRC_INTF block (shown in Figure 6) conveys the cells received from the UMUX block to the UTOPIA interface. Depending on the value of UTOP_MODE field in the UI_SRC_CFG register, the UTOPIA interface will either act as an UTOPIA master (controls the write enable signal) or as an UTOPIA PHY device (controls the cell available signal). As a PHY device, the SRC_INTF can either be a UTOPIA Level One device, where it is the only device on the UTOPIA bus, or a UTOPIA Level Two device where other devices can coexist on the UTOPIA bus. As a master device, the SRC_INTF can only function as a UTOPIA Level One device.

If 16_BIT_MODE is set in the UI_SRC_CFG register then all 16 bits of the UTOPIA data bus are used. 16_BIT_MODE must be ‘0’ in UTOPIA master mode.

In master mode, the SRC_INTF block sources TATM_D, TATM_PAR, TATM_SOC, and TATM_ENB while receiving TATM_CLAV. The Start-Of-Cell (SOC) indication is generated coincident with the first word (only 8-bit mode is supported) of each cell that is transmitted on TATM_D. TATM_D, TATM_PAR and TATM_SOC are driven at all times. The TATM_ENB signal indicates which clock cycles contain valid data for the UTOPIA bus. The device will not assert the TATM_ENB signal until it has a full cell to send and the target device has activated TATM_CLAV. The TATM_CLAV signal indicates whether the target device is able to accept cells or not. Only cell level handshaking is supported. If the target device is unable to accept any additional cells it must deactivate TATM_CLAV no later than byte 49 of the current cell. No additional cells will be sent until TATM_CLAV is activated.

In PHY mode, the SRC_INTF block sources RPHY_D[15:0], RPHY_PAR, RPHY_SOC, and RPHY_CLAV, while receiving RPHY_ENB. The SOC indication is generated coincident with the first word (8-bit or 16-bit) of each cell that is transmitted on RPHY_D[15:0]. In PHY mode, the RPHY_D[15:0], RPHY_PAR, and RATM_SOC signals are driven only when valid data is being sent; otherwise they are tristated.

In UTOPIA Level 1 PHY mode, RPHY_CLAV is activated whenever a complete cell is available to be sent. It remains active until the last byte has been read of the last available complete cell. A cell is sent one cycle after RPHY_ENB goes low. If RPHY_ENB goes high during the cell transfer, data is not sent each cycle following one where RPHY_ENB is high.

RPHY_ADD[4:0] is an input and is used only in UTOPIA Level Two mode. Any bus cycle following one where RPHY_ADD[4:0] matches CFG_ADDR(4:0) in the UI_SRC_ADD_CFG register, the UI Block will drive RPHY_CLAV. Otherwise RPHY_CLAV is tri-stated. If, in addition, during the previous cycle RPHY_ENB was high and it is low in the current cycle, then the device is selected and the SRC_INTF begins transmitting a cell the next cycle.

Parity is driven on TATM_PAR(RPHY_PAR) whenever TATM_D(RPHY_D[15:0]) is driven. EVEN_PAR will determine whether even parity or odd parity is generated. Since odd parity is required by the ATM Forum, EVEN_PAR is intended to be used for error checking only.

The AAL1gator-32 can tolerate temporary de-assertions of TATM_CLAV/RPHY_ENB), but it is assumed that enough UTOPIA bandwidth is present to accept the cells that the AAL1gator-32 can produce in a timely manner. Once the 4-Cell FIFO fills up in the UI, cells will begin filling up in the 8-cell FIFO in each A1SP block. Anytime the UTOPIA FIFO fills up the T_UTOP_FULL interrupt will go active in the MSTR_INTR_REG if it is enabled. This FIFO can fill during normal operation and is not usually an indication of an

error. However, the A1SP FIFOs should not normally fill. If they do fill it indicates there is some congestion, which is impacting the UTOPIA interface and the TALP_FIFO_FULL bit will go active in A1SPn_INTR_REG. When the TALP FIFO fills, then TALP is no longer able to build cells and data will start building up in the transmit buffer and the frame_advance_fifo will fill. If this continues so that the FR_ADV_FIFO_FULL bit goes active then data has been lost and the transmit queues need to be reset. The T_UTOP_FULL indicator can be used to determine when the UTOPIA Interface clears. It may also be desirable to disable UI_EN so that the stored cells can be flushed.

The SRC_INTF circuit controls when a cell is transmitted from the internal 4 cell FIFO. Since the UTOPIA can transmit cells at higher speeds than the TALP, and since it is expected to see applications in a shared UTOPIA environment, cell transmission from the SRC_INTF commences only when there is a full cell worth of data available to transmit. The cell is then transmitted to the interface at the UTOPIA TATM_CLK rate, in accordance with the TATM_FULLB/RPHY_ENB) input. The maximum supported clock rate is 52 MHz.

11.1.1.1 Any-PHY Mode

If ANY-PHY_EN is set in the UI_SRC_CFG register then the SRC_INTF operates as a single port Any-PHY slave device. In Any-PHY mode the RPHY_ADDR(4) pin becomes the RSX pin and depending on the value of CS_MODE_EN, the RPHY_ADDR(3) pin may become the RCSB signal instead.

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 for the address prepend. If 16_BIT_MODE is low then only the lower 8 bits are used.

During the cycle that the prepend address is active on the bus, RSX pulses high.

Because of the large number of possible ports, in the source direction, device addresses are used for polling and device selection, instead of port addresses. (Each device may control many ports) When a device is selected to send a cell, the PHY device prepends the port address in front of the cell. Since, in this direction the AAL1gator-32 is only a single port, the device address and port address are the same. However, the AAL1gator-32 has only a limited number of address pins. To accommodate systems, which are using a mix of different port density Any-PHY devices, the RCSB signal is available to handle any additional external decoding that is required. In Any-PHY mode, PHY devices respond with RPHY_CLAV 2 cycles after their address is on the bus instead of the one cycle

required in UTOPIA mode. However, the timing of RCSB matches UTOPIA timing so that a full cycle for external decoding is available.

Table 3 shows how the CFG_ADDR field is used in different modes.

Table 3 CFG_ADDR and PHY_ADDR Bit Usage in SRC direction

	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-32 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='1' then CFG_ADDR[4:3] = "00".
- In Any-PHY mode, if CS_MODE_EN='0' then CFG_ADDR[4]="0".

11.1.2 UTOPIA Sink Interface (SNK_INTF)

The SNK_INTF block receives cells from the UTOPIA interface and sends them to the UMUX interface. Depending on the value of the UTOP_MODE field in the UI_SNK_CFG register, the UTOPIA interface acts either as an UTOPIA master (controls the read enable signal) or as an UTOPIA PHY device (controls the cell available signal). As a PHY device the SNK_INTF can either be a UTOPIA Level One device, where it is the only device on the UTOPIA bus, or a UTOPIA Level Two device where other devices can coexist on the UTOPIA bus. As a master device the SNK_INTF can only function as a UTOPIA Level One device.

If 16_BIT_MODE is set in the UI_SNK_CFG register then all 16 bits of the UTOPIA data bus are used. 16_BIT_MODE must be '0' in UTOPIA master mode.

In master mode, the SNK_INTF block receives RATM_D, RATM_PAR, RATM_SOC, and RATM_CLAV while driving RATM_ENB. Once the UI is enabled in this mode, and, if the RATM_CLAV input signal is asserted, the SNK_INTF block waits for an RATM_SOC signal from the PHY layer. Once the RATM_SOC signal arrives, the cell is accepted as soon as possible. The Start-Of-Cell (SOC) indication is received coincident with the first word (only 8-bit mode is supported) of each cell that is received on RATM_D. An 8 cell FIFO allows the interface to accept data at the maximum rate. If the FIFO fills, the RATM_ENB signal will not be asserted again until the device is ready to accept an entire cell. The RATM_ENB signal depends only on the cell space and is independent of the state of the RATM_CLAV signal. The RATM_CLAV signal indicates whether the target device has a cell to send or not. Only cell level handshaking is supported.

In PHY mode, the SNK_INTF block receives TPHY_D[15:0], TPHY_SOC, and TPHY_ENB while driving TPHY_CLAV. The cell available (TPHY_CLAV) signal indicates when the device is ready to receive a complete cell. In UTOPIA Level One mode, TPHY_CLAV is always driven.

In UTOPIA Level Two mode, SNK_INTF normally responds as a single address device. However there may be situations in some systems where groups of cells targetted to a given A1SP may be clumped together. If one of the 8-cell A1SP FIFO fills up so that it backs up into the 8-cell sink UTOPIA FIFO then a head-of-line blocking problem can exist. To alleviate such a situation, the sink direction can be configured as four separate addresses, where the bottom two bits of the address indicate which A1SP is targeted to receive the cell. When polling any of the A1SP addresses, a full indication will be given when the A1SP FIFO associated with that address, reaches a 3/4 full state (room for 2 more cells) or the UI cell sink FIFO already has two cells for that address. This will always allow room for any cells that may still be queued in the sink UTOPIA FIFO and prevent head-of-line blocking. Full indications will be given for a specific port until both full conditions are cleared.

When responding as a single address, TPHY_CLAV is driven the cycle following ones in which TPHY_ADDR(4:0) matches CFG_ADDR(4:0) in UI_SNK_ADD_CFG register. When responding as 4 addresses, TPHY_CLAV is driven the cycle following ones in which TPHY_ADDR(4:2) matches CFG_ADDR(4:2) in UI_SNK_ADD_CFG register. Otherwise TPHY_CLAV is tri-stated. If, in addition to an address match, during the previous cycle TPHY_ENB was high and it is low in the current cycle, then the device is selected and the SRC_INTF begins accepting the cell that is being received.

The SNK_INTF block waits for an SOC. When an SOC signal arrives, a counter is started, and 53 bytes are received. If a new SOC occurs within a cell, the counter reinitializes. This means that the corrupted cell will be dropped and the

second good cell will be received. The SNK_INTF block stores the cell in the receive FIFO. If the receive FIFO becomes full, it stops receiving cells. The bytes are written to the FIFO with RATM_CLK. RATM_CLK is an input to the AAL1gator-32. The maximum supported clock rate is 52 MHz.

Parity is always checked and a parity error will cause an interrupt if the UTOP_PAR_ERR_EN bit is set in the MSTR_INTR_EN_REG. FORCE_EVEN_PARITY will determine whether even parity or odd parity is checked. Since odd parity is required by the ATM Forum, FORCE_EVEN_PARITY is intended to be used for error checking only. If an error is detected the UTOP_PAR_ERR bit in the MSTR_INTR_REG is set, and the corresponding enable bit is set in the MSTR_INTR_EN_REG then INTB will go active. Any cell received with bad parity will still be processed as normal.

11.1.2.1 Any-PHY Mode

If ANY-PHY_EN is set in the UI_SNK_CFG register then the SNK_INTF operates as a multi port Any-PHY slave device. In Any-PHY mode the TPHY_ADDR[4] pin becomes the TSX pin and depending on the value of CS_MODE_EN, the TPHY_ADDR(3) pin may become the TCSB signal instead.

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 to match with the address prepend. If 16_BIT_MODE is low then CFG_ADDR(7:2) is used.

During the cycle that the prepend address is active on the bus, the TSX input pulses high.

In the sink direction, port addresses are used for polling and device selection, instead of device addresses. Since, in this direction the AAL1gator-32 has four ports, the AAL1gator-32 will use the upper 14 bits of the UI_SNK_ADD_CFG register for address compares and use the lower two bits to determine which A1SP is being polled or selected. However the AAL1gator-32 has only a limited number of address pins. To accommodate systems, which are using a mix of different port density Any-PHY devices, the TCSB signal is available to handle any additional external decoding that is required. In Any-PHY mode, PHY devices respond with TPHY_CLAV 2 cycles after their address is on the bus instead of the one cycle required in UTOPIA mode. However the timing of TCSB matches UTOPIA timing so that a full cycle for external decoding is available.

Table 4 shows how the CFG_ADDR field is used in different modes.

Table 4 CFG_ADDR and PHY_ADDR Bit Usage in SNK direction

	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 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 addr is prepended	[15:2]=device

Notes:

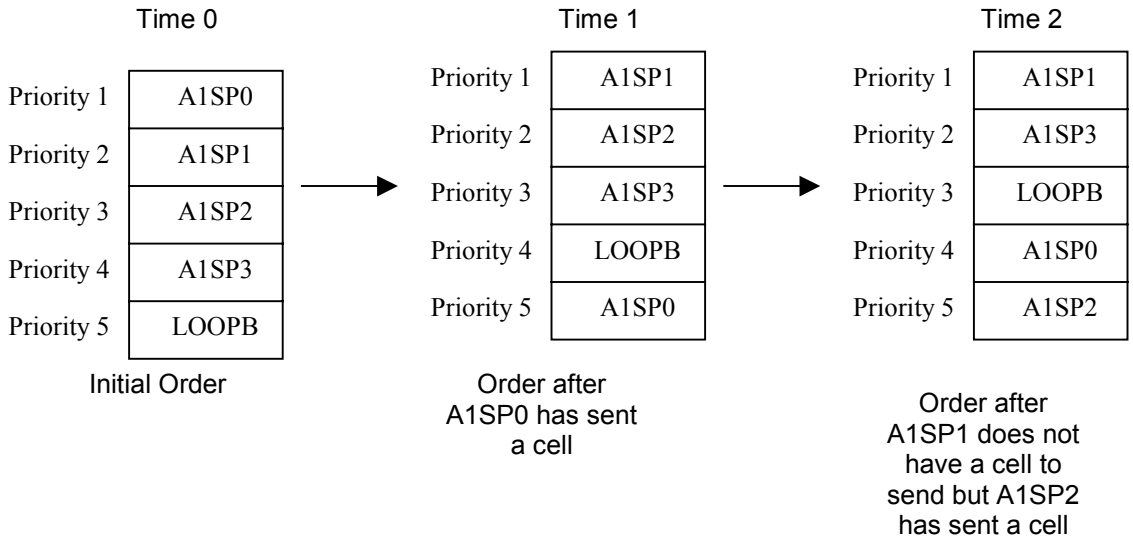
- In Any-PHY mode, if CS_MODE_EN='1' then CFG_ADDR[4:3] = "00". Else if CS_MODE_EN='0' then CFG_ADDR[4]="0".
- In Any-PHY mode the upper 14 bits of the prepended address are compared with CFG_ADDR[15:2]. The bottom two bits are not compared with this field and are just used to select the target A1SP.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

11.1.3 UTOPIA Mux Block (UMUX)

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

In the source direction, the UMUX polls each of the four A1SP blocks and the Loopback FIFO using a least recently serviced algorithm to determine cell availability. In this algorithm, once a particular source is serviced, it is put at the lowest priority of all five sources. When a higher priority source is serviced, the lower priority sources below it move up in the priority list. Thus, excluding the initial start up, the source which has been serviced least recently will have the highest priority. Figure 7 below shows an example of the changing priority list as cells are taken from A1SP0 and A1SP2.

Figure 7 Source Priority Servicing Example



When an A1SP is operated in high-speed mode, its companion A1SP within the dual A1SP is left idle. If the remaining two A1SPs are not in high speed mode, it is advantageous to provide the high-speed A1SP with more opportunities to be serviced with a higher priority than the low-speed A1SPs. To support this, the initial value of priority list can be programmed to be different than the default and allow, for example, two entries of the five slots for the high-speed A1SP and no entry for the idle A1SP. Thus, giving the high-speed A1SP 2/5 of the available bandwidth. The initial order is loaded via the UI_Src_Poll_List register When the Utopia interface is not enabled (UI_EN=0 in the UI_COMN_CFG register), the value in the UI_Src_Poll_List register is loaded into the priority table as its initial value. During operation, the initial entries will be transferred among the priority list but the same number of entries for a particular A1SP will continue to exist indefinitely until a reset event occurs.

If the SRC_INTF FIFO has room for a cell, the UMUX polls the current highest priority A1SP block (or Loopback FIFO) to determine if it has a cell. If so, a cell is read from the selected A1SP block and transferred into the SRC_INTF FIFO. If the highest priority source does not have a cell, the next lowest priority A1SP FIFO is examined and the process continues until all sources in the list are polled. Each A1SP block has an 8-cell FIFO.

In the sink direction, the UMUX waits until the SNK_INTF FIFO has a cell to send. Once the SNK_INTF FIFO has a cell to send, the UMUX polls the A1SP associated with the cell for availability. Once the A1SP has room for the cell, the UMUX reads the cell out of the SNK_INTF FIFO and places it in the A1SP FIFO.

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

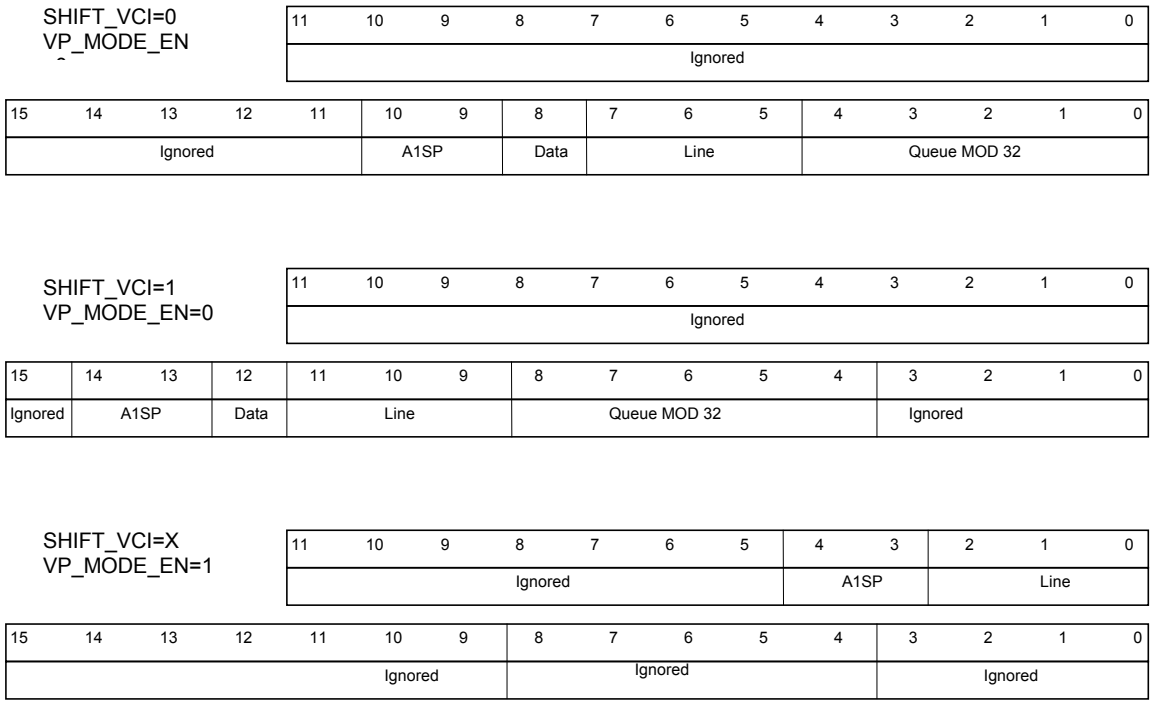
1. If SHIFT_VCI bit in the UI_COMN_CFG register is low and VP_MODE_EN is low then VCI(10:9) are used.

2. If SHIFT_VCI bit in the UI_COMN_CFG register is set and VP_MODE_EN is low then VCI(14:13) are used.

3. If VP_MODE_EN is set then VPI(4:3) are used.

Refer to Figure 8 for details of how VPI and VCI are interpreted by UMUX.

Figure 8 Cell Header Interpretation



The UMUX also 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. In global loopback all cells received by the UTOPIA block are sent back out onto the UTOPIA bus. Global loopback is enabled by setting the U2U_LOOP bit in the UI_COMN_CFG register. In VC based loopback mode, any cell received with a VC that matches the loopback VC is sent back out onto the UTOPIA bus. VC based loopback is enabled by setting the VCI_U2U_LOOP bit in the UI_COMN_CFG register. The loopback

VC is programmable by writing the U2U_LOOP_VCI register. The 3-cell FIFO is used for loopback.

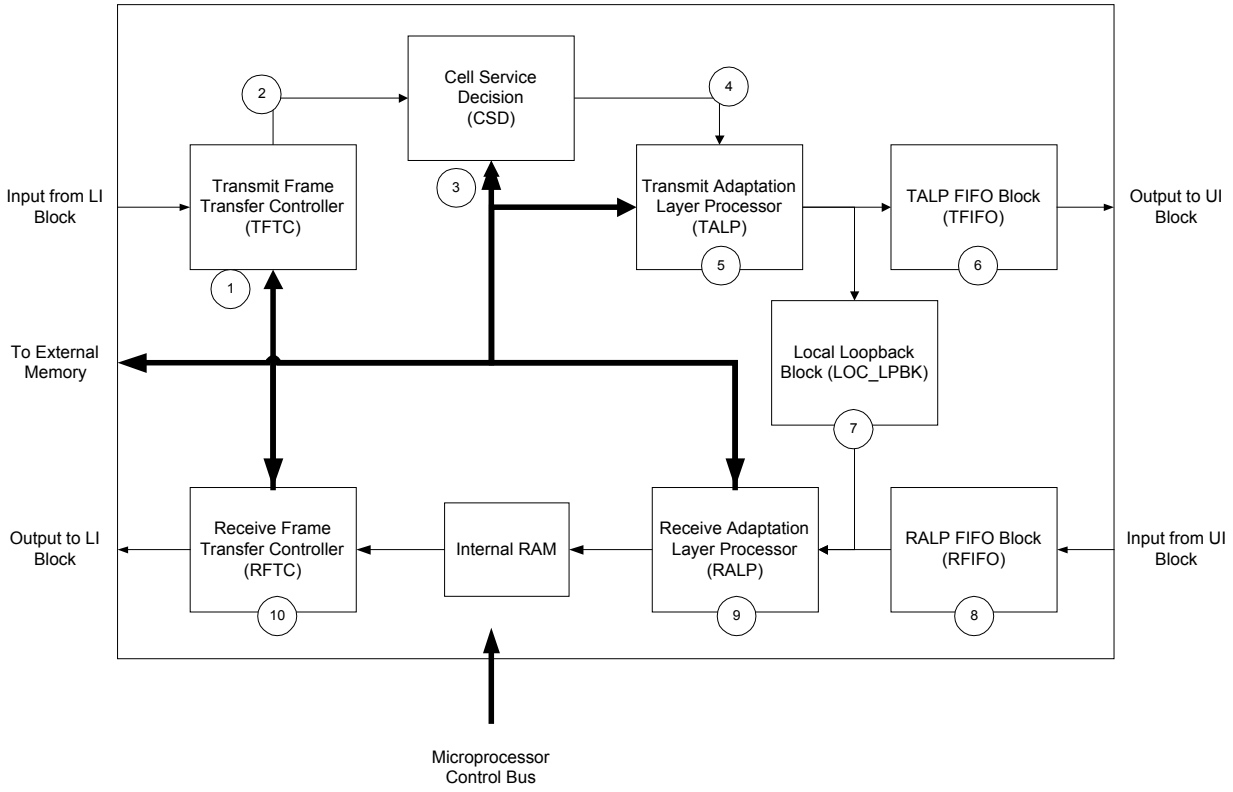
11.2 AAL1 SAR Processing Block (A1SP)

The A1SP block is the main AAL1 SAR processing block. Each block processes 8 E1/T1 lines. This block is replicated four times to maintain throughput and minimize Cell Delay Variation. This block has the following major components.

- Transmit Frame Transfer Controller (TFTC) block
- Cell Service Decision (CSD) block
- Transmit Adaptation Layer Processor (TALP) block
- TALP FIFO (TFIFO) block
- Local Loopback Block (LOC_LPBK) block
- Receive Frame Transfer Controller (RFTC) block
- Receive Adaptation Layer Processor (RALP) block
- RALP FIFO (RFIFO) block

Figure 9 shows a block diagram of the AAL1gator-32 and the sequence of events used to segment and reassemble the CBR data.

Figure 9 A1SP Block Diagram



1. TFTC stores line data into the memory 16 bits at a time.

2. When the TFTC finishes writing a complete frame into the memory, it notifies the CSD of a frame completion by writing the line and frame number into a FIFO. Idle channel detection is processed here if enabled.

3. The CSD checks a frame-based table for queues having sufficient data to generate a cell. For each queue with enough data to generate a cell, the CSD schedules the next cell generation occurrence in the table.

4. The CSD commands the TALP to generate a cell from the available data for each of the ready queues identified in step 3.

5. The TALP generates the cell from the data and signaling buffers and writes the cell into the TALP FIFO.

6. The TFIFO block buffers cells which will be transmitted out to the UTOPIA Interface block.

7. If local loopback is enabled the cell is looped to RALP.

8. The RFIFO block buffers cells received from the UTOPIA Interface block.

9. The RALP performs pointer searches, checks for overrun and underrun conditions, detects SN mismatches, checks for OAM cells, and extracts the line data from the cells, and places the data into the receive buffer.

10. The RFTC plays the receiver buffer data onto the lines.

Four types of data are supported by the A1SP.

1. UDF-ML (Unstructured Data Format- Multi-Line). Unstructured bit stream for line speeds < 15 Mbps. (supports 8 lines per A1SP) if all are under 2.5 Mbps).

2. UDF-HS (Unstructured Data Format- High Speed). Unstructured bit stream for line speeds under 45 Mbps. (Only one line supported per A1SP).

3. SDF-FR (Structured Data Format- Frame). Channelized data without CAS signaling. (Frame based structure).

4. SDF-MF (Structured Data Format- Multi-Frame). Channelized data with CAS signaling. (Multi-Frame based structure).

11.2.1 AAL1 SAR Transmit Side (TxA1SP)

11.2.1.1 Transmit Frame Transfer Controller (TFTC)

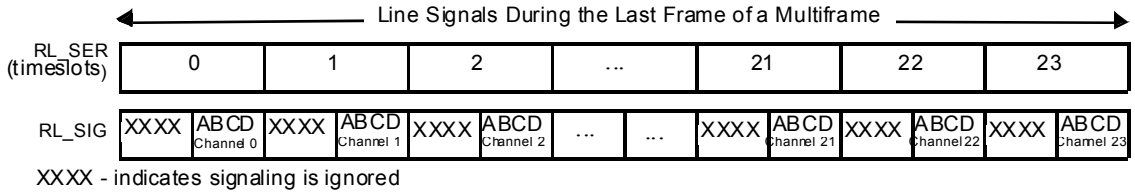
The TFTC accepts deframed data from Line Interface Block. For structured data, the TFTC uses the synchronization supplied by the Line Interface Block to perform a serial-to-parallel conversion on the incoming data and then places this data into a multiframe buffer in the order in which it arrives.

The TFTC monitors the frame sync signals and will realign when an edge is seen on these signals that does not correspond to where it expects it to occur. It is not necessary to provide an edge at the beginning of every frame or multiframe. The AAL1gator-32 reads signaling during the last frame of every multiframe. For T1 mode, the AAL1gator-32 reads signaling on the 24^{th} frame of the multiframe. For E1 mode, the AAL1gator-32 reads signaling on the 16^{th} frame of the multiframe.

A special case of E1 mode exists that permits the use of T1 signaling with E1 framing. Normally an E1 multiframe consists of 16 frames of 32 timeslots, where signaling changes on multiframe boundaries. When E1_WITH_T1_SIG is set in LIN_STR_MODE and the line is in E1 mode, the TFTC will use a multiframe consisting of 24 frames of 32 timeslots. In this mode, the AAL1gator-32 reads signaling on the 24^{th} frame of the multiframe.

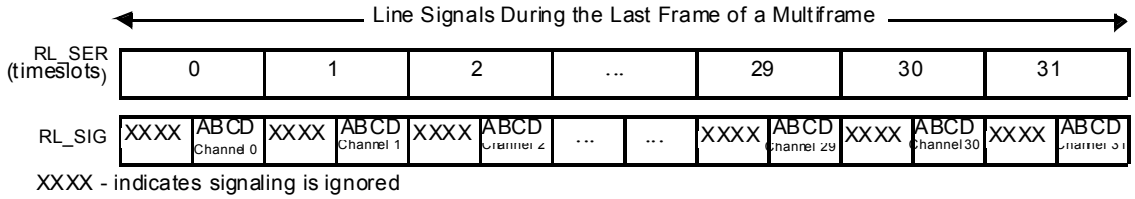
The AAL1gator-32 reads the signaling nibble for each channel when it reads the last nibble of each channel's data unless the SHIFT_CAS bit in the LIN_STR_MODE register is set. If the SHIFT_CAS bit is set then the AAL1gator-32 reads the signaling nibble for each channel when it reads the first nibble of each channel's data. See Figure 10 for an example of a T1 frame. See Figure 11 for an example of an E1 frame.

Figure 10 Capture of T1 Signaling Bits (SHIFT_CAS=0)



XXXX - indicates signaling is ignored

Figure 11 Capture of E1 Signaling Bits (SHIFT_CAS=0)



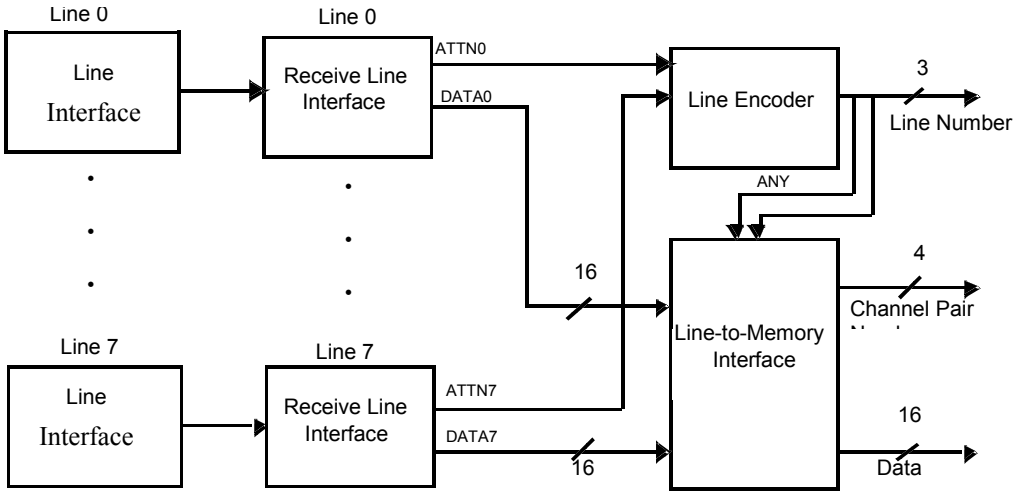
XXXX - indicates signaling is ignored

Note:

- AAL1gator-32 treats all 32 timeslots identically. Although E1 data streams contain 30 timeslots of channel data and 2 timeslots of control (timeslots 0 and 16), data and signaling for all 32 timeslots are stored in memory and can be sent and received in cells.

Unstructured data is 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. Figure 12 shows the basic components of the TFTC.

Figure 12 Transmit Frame Transfer Controller



The receive line interface is primarily a serial-to-parallel converter. Serial data, which is derived from the RL_DATA signal from the LI Block, is supplied to a shift register. The shift register clock is the RL_CLK input from the external framer. When the data has been properly shifted in, it is transferred to a 2-byte holding register by an internally derived channel clock. This clock is derived from the line clock and the framing information.

The channel clock also informs the line-to-memory interface that two data bytes are available from the line. When the two bytes are available, a line attention signal is sent to the line encoder block. However, because the channel clock is an asynchronous input to the line-to-memory interface, it is passed through a synchronizer before it is supplied to the line encoder. Since there are eight potential lines and each of them provides its own channel clock, they are synchronized before being submitted to the line encoder.

The TFTC accommodates the T1 Super Frame (SF) mode by treating it like the Extended Super Frame (ESF) format. The TFTC ignores every other frame pulse and captures signaling data only on the last frame of odd SF multiframes. The formatting of data in the signaling buffers is highly dependent on the operating mode. Refer to section 7.6.6 “RESERVED (Transmit Signaling Buffer)” on page 136 for more information on the transmit signaling buffer.

Figure 13 shows the format of the transmit data buffer for ESF-formatted T1 data for lines that are in the SDF-MF mode.

Figure 13 T1 ESF SDF-MF Format of the T_DATA_BUFFER

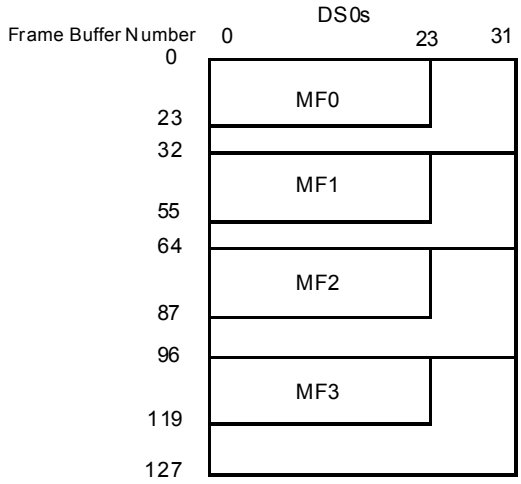


Figure 14 shows the format of the transmit data buffer for SF-formatted T1 data for lines that are in the SDF-MF mode.

Figure 14 T1 SF-SDF-MF Format of the T_DATA_BUFFER

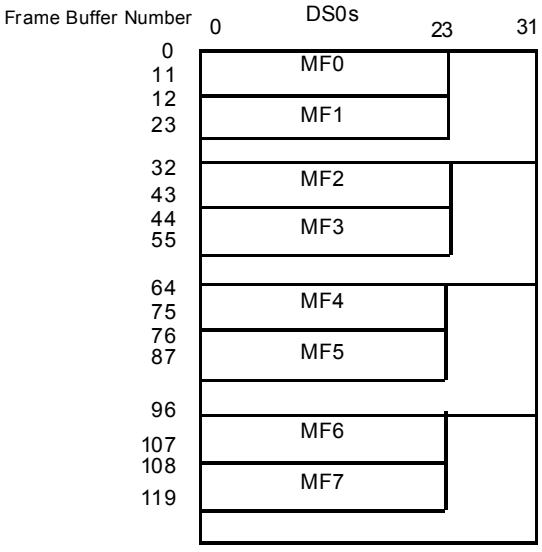


Figure 15 shows the format of the transmit data buffer for T1 data for lines that are in the SDF-FR mode.

Figure 15 T1 SDF-FR Format of the T_DATA_BUFFER

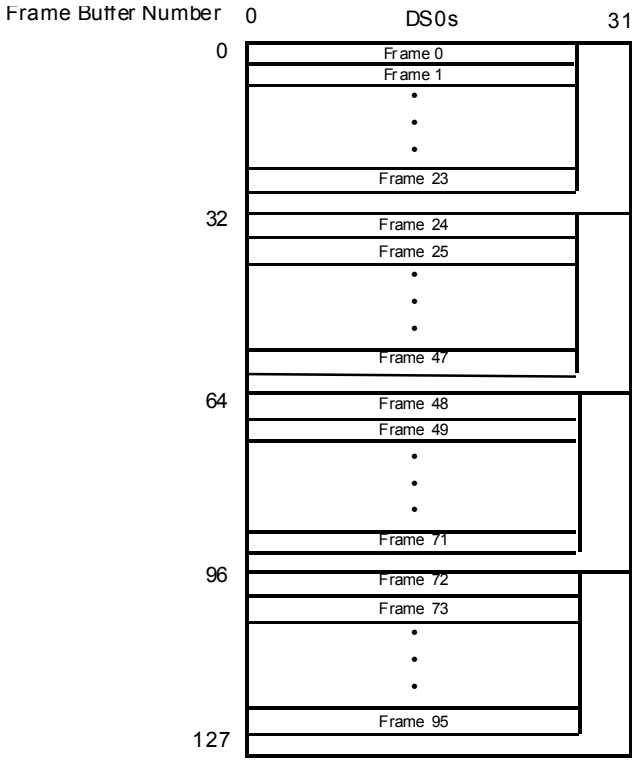


Figure 16 shows the format of the transmit data buffer for E1 data for lines that are in the SDF-MF mode.

Figure 16 E1 SDF-MF Format of the T_DATA_BUFFER

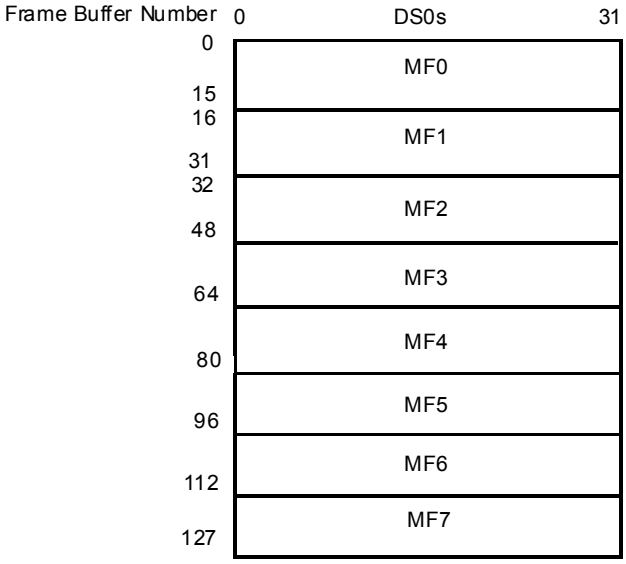


Figure 17 shows the format of the transmit data buffer for E1 data using T1 signaling, for lines that are in SDF-MF mode

Figure 17 E1 SDF-MF with T1 Signaling Format of the T_DATA_BUFFER

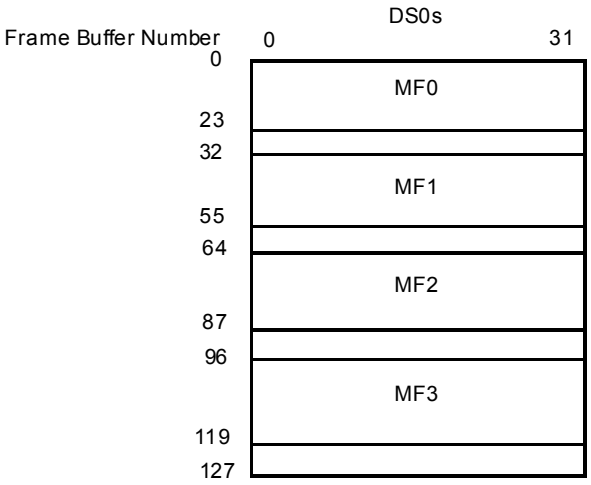


Figure 18 shows the format of the transmit data buffer for E1 data for lines that are in the SDF-FR mode.

Figure 18 E1 SDF-FR Format of the T_DATA_BUFFER

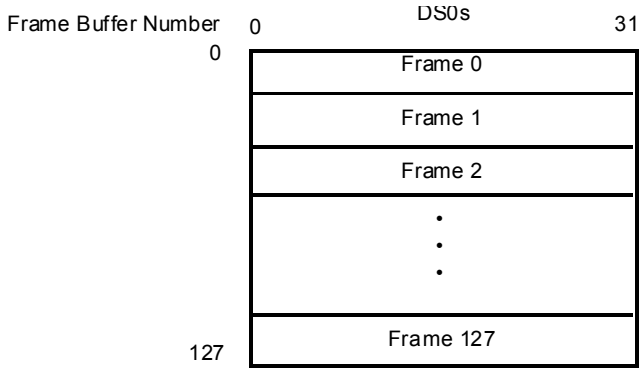


Figure 19 shows the format of the transmit data buffer for lines that are in UDF-ML mode.

Figure 19 Unstructured Format of the T_DATA_BUFFER

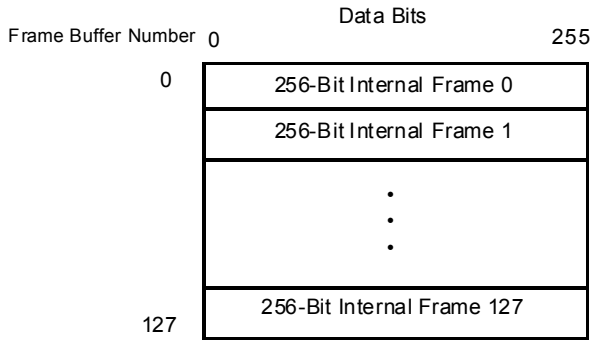


Figure 20, Figure 21, Figure 22 and Figure 23 show the contents of the transmit signaling buffer for the different signaling modes. In all cases the upper nibble of each byte is “0000”.

Figure 20 SDF-MF T1 ESF Format of the T_SIGNALING_BUFFER

Multiframe	Byte Address 0							Byte Address 31
0	Channel 0 ABCD	Channel 1 ABCD	• • •	Channel 22 ABCD	Channel 23 ABCD	Channel 24 Not Used	• • •	Channel 31 Not Used
1								
2								
3								

Figure 21 SDF-MF T1 SF Format of the T_SIGNALING BUFFER

Multiframe/2	Byte Address 0							Byte Address 31
0	Channel 0 ABAB	Channel 1 ABAB	• • •	Channel 22 ABAB	Channel 23 ABAB	Channel 24 Not Used	• • •	Channel 31 Not Used
1								
2								
3								

Figure 22 SDF-MF E1 Format of the T_SIGNALING_BUFFER

Multiframe	Byte Address				
0	Channel 0 ABCD	Channel 1 ABCD	• • •	Channel 30 ABCD	Channel 31 ABCD
7					

Figure 23 SDF-MF E1 with T1 Signaling Format of the T_SIGNALING_BUFFER

Multiframe	Byte Address				
0	Channel 0 ABCD	Channel 1 ABCD	• • •	Channel 30 ABCD	Channel 31 ABCD
3					

11.2.1.1.1 Transmit Conditioning

The T_COND_DATA structure allows conditional data to be defined on a per-DS0 basis and the T_COND_SIG structure allows conditioned signaling to be defined on a per-DS0 basis. The TX_COND bit in the T_QUEUE_TBL allows the cell building logic (described in Section 11.2.1.3 Transmit Adaptation Layer Processor (TALP) on page 93) to be directed to build cells from the conditioned data and signaling. To control whether conditioned data, conditioned signaling, or both is used, set TX_COND_MODE in the TX_CONFIG register to the appropriate value. The TX_COND bit and TX_COND_MODE bits can be set on a per-queue basis.

By having independent control over whether signaling or data is conditioned, it is possible to substitute the signaling which is carried in the CAS bits across the ATM network while still passing the data received off the line. This is useful for applications that may not be receiving the signaling with the data.

For HS_UDF mode the HS_TX_COND bit needs to be set in the HS_LIN_REG register. When this bit is set cells with an all ones pattern will be generated. The CMD_REG_ATTN bit needs to be written to a ‘1’ after the HS_TX_COND bit is set for this function to take affect.

Under certain alarm conditions such as Loss of Signal (LOS), an Alarm Indication Signal (AIS) needs to be transmitted downstream. This means that cells need to be generated which carry an AIS pattern. The AAL1gator-32 does not do any alarm processing and is dependent on the external framer for this functionality. The framer would notify the processor of any alarm conditions and then the processor would switch a particular queue from normal mode to a conditioned mode by setting the TX_COND bit in the T_QUEUE_TBL.

Most AIS signals are an all ones pattern, so cells with this pattern can be generated by setting T_COND_DATA to “FF”x and T_COND_SIG to “F”x. In E3

mode this can be done by setting HS_TX_COND bit to a '1'. 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. The CMD_REG_ATTN bit needs to be written to a '1' after the HS_GEN_DS3_AIS bit or the HS_TX_COND bit is set for this function to take affect.

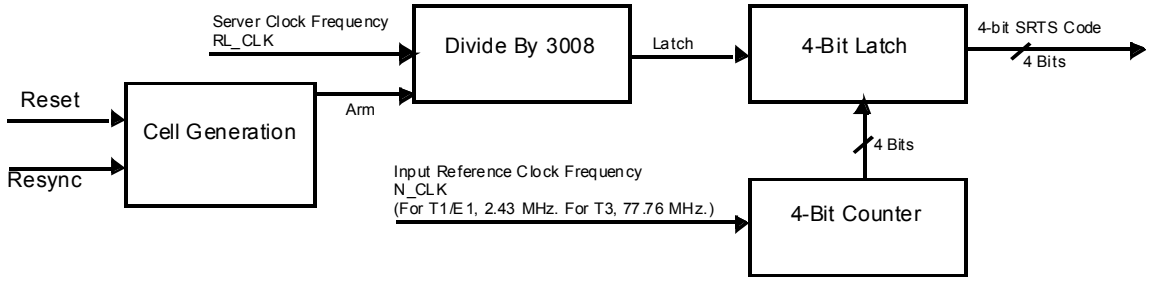
11.2.1.1.2 Transmit Signaling Freezing

Signaling freezing is a required function when transporting CAS. This function holds the signaling unchanged when the incoming line fails. The PMC-Sierra framers provide this function. If a framer is used that does not support signaling freezing, this function must be provided externally.

11.2.1.1.3 SRTS for the Transmit Side

The transmit side supports SRTS only for unstructured data formats on a per-line basis. SRTS support requires an input reference clock, NCLK. The input reference frequency is defined as $155.52/2^n$ MHz, where n is chosen such that the reference clock frequency is greater than the frequency being transmitted, but less than twice the frequency being transmitted (2 x RL_CLK > NCLK > RL_CLK). For T1 or E1 implementation, the input reference clock frequency must be 2.43 MHz. The transmit side can accept a reference clock speed of up to 77.76 MHz, which is required for DS3 applications. Figure 24 on page 81 shows the process implemented for each UDF line enabled for SRTS, regardless of the reference frequency. One bit of resulting 4-bit SRTS code is then inserted into the CSI bit of each of the odd numbered cells for that line. There are four odd cells in each 8 cell sequence, so each one carries a different SRTS bit. If the line does not supply SRTS, then all odd CSI bits are set to 0. The 3008 divider is the number of data bits in eight cells (8 x 8 x 47). The divider is aligned on the first cell generation after a reset or a resynchronization to the cell generation process.

Figure 24 Transmit Side SRTS Function



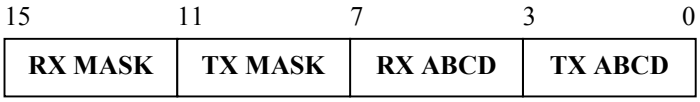
11.2.1.1.4 Idle Detection

Idle detection will be performed on a per queue basis using one of the following three methods: channel associated signaling (CAS), out of band signaling (processor controlled), or pattern matching. The status of each channel is stored in the Active/Idle bit table. The mode for each channel is controlled by the value of IDLE_CFG_n in the Idle Detection Configuration Table. The lower 16 channels or upper 16 channels of a line must not mix CAS or pattern matching mode with processor controlled mode. This is to avoid contention updating the active channel table.

11.2.1.1.4.1 CAS Idle Detection

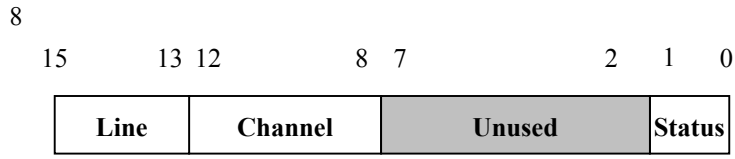
CAS idle detection looks at the ABCD bits in both the receive and transmit direction and compares them to values programmed on a per channel basis by the processor. If two consecutive CAS values match in both the receive and transmit direction the channel is considered to be idle. The format of the register (AUTO_CONFIG_n) in the CAS/Pattern Matching Configuration Table, which the processor programs with the idle ABCD patterns, is pictured below in Figure 25. The register also provides mask fields for the receive and transmit directions which allow any one of the ABCD bits to be ignored when looking for a match.

Figure 25 CAS Idle Detection Configuration Register Structure



During CAS idle detection, a word is written to the Transmit Idle Interrupt FIFO every time the status changes from active->idle or idle->active. TIDLE_FIFO_EMPB is set as long as this FIFO contains any unread entries, which will result in a maskable interrupt. The structure of the word contained in the FIFO is shown in Figure 26. The upper eight bits indicate the channel that encountered the status change and bit 0 indicates the status of the channel (Active =1; Idle = 0). The processor accesses the FIFO by reading the A1SPn_TIDLE_FIFO register which will contain the top element of the FIFO.

Figure 26 CAS Idle Detection Interrupt Word



11.2.1.1.4.2 Processor Controlled Idle Detection

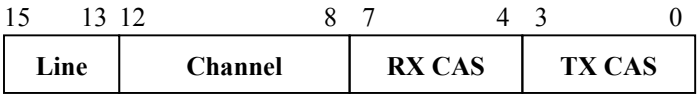
In Processor controlled idle detection mode, it is the responsibility of the processor to add or drop channels. This mode is used in conjunction with common channel signaling (CCS) or if the processor wants to make its own determination of channel activity based on the CAS bits.

During the processor controlled idle detection, a word is written to the Transmit Idle Interrupt FIFO every time the value of the CAS nibble changes and then remains stable for one additional multiframe. TIDLE_FIFO_EMPB is set as long as this FIFO contains any unread entries, which will result in a maskable interrupt. The structure of the word contained in the FIFO is shown in Figure 27. The first eight bits indicate the channel, which encountered a change in the value of CAS. The next four bits indicate the RX CAS value and the final four bits indicate the TX CAS value.

TX refers to the CAS incoming on the TDM interface (SBI Drop bus, H-MVIP RL_SIG or direct low speed mode RL_SIG), and RX refers to CAS incoming in ATM cells at the UTOPIA Cell Sink interface.

Based on these new CAS values the processor can make a determination if the channel should be marked as active or idle. The CAS values are de-bounced internally one time, and any additional debounce must be done external to the chip.

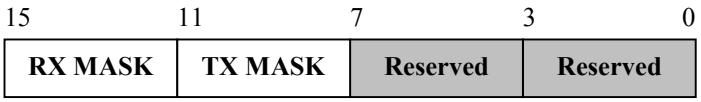
Figure 27 Processor Controlled Idle Detection Interrupt Word



The processor will also be able to mask out portions of the CAS and therefore receive interrupts only when particular bits of the CAS change. Figure 28 shows the structure of AUTO_CONFIG_n field in the CAS/Pattern Matching Configuration Table. The lower byte is reserved and is used in detecting

changes in the CAS values. Therefore, to avoid contention, the upper byte should only be written when the idle detection is disabled.

Figure 28 Processor Controlled Configuration Register Structure



Once the processor determines that the status of a channel should change, the processor should then write the TX Channel Active Table. The processor does this by accessing the table 16 bits at a time. In most situations the processor will want to change a subset of the 16 channels accessed. Therefore, a read-modify-write will have to be performed.

Figure 29 shows the structure of the Active/Idle bit table. The index represents the value that needs to be added to the base address of the table in order to access the status for the channels located at that index.

Note that since the processor writes 16 bits at a time it is recommended that processor intervention and automatic mode is not mixed within a group of 16 channels to avoid contention problems.

Figure 29 TX Channel Active/Idle Bit Table Structure

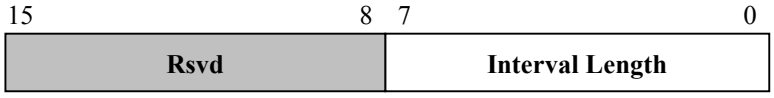
Index	Line	Channel Status															
0	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
2	1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
●	●								●								
●	●								●								
●	●								●								
13	6	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
14	7	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
15	7	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

11.2.1.1.4.3 Pattern Match Idle Detection

Pattern match idle detection compares the received byte with a programmed pattern and a mask. If there is a mismatch of received data with the programmed pattern during a programmable length of time, then the channel is considered active. Otherwise, if the received channel bytes match the unmasked pattern bits over the programmable length of time, the channel is considered in an idle state and cell transmission will be suppressed.

Interval length refers to the amount of time that the patterns must match for it to be considered a match event. This value is programmed in the Pattern Matching Line Configuration register for the associated line. The Interval length is programmed in units of 12 ms for T1 and units of 16 ms for E1. Since this is an 8 bit field, the maximum length of time is 3.1 (+/- 12 ms) seconds for T1 and 4.1 (+/- 16 ms) seconds for E1.

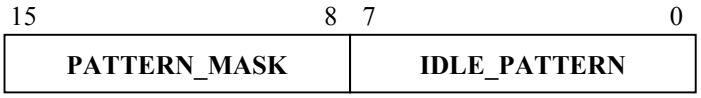
Figure 30 PAT_MTCH_CFG Register Structure



Internal Length = Duration of time data must match before declaring an idle condition

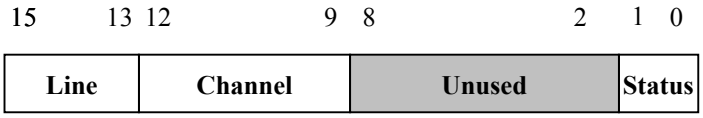
Figure 31 shows the structure of the AUTO_CONFIG_n field in the CAS/Pattern Matching Configuration Table. The lower byte contains the pattern the received byte should be compared against. The upper byte is a mask field that can be used to control which bits are monitored. Since the chip will be updating this field during normal operation, it is best if the processor writes to the lower byte of the register only during reset in order to avoid contention. .

Figure 31 Pattern Match Idle Detection Register Structure



During pattern match idle detection, a word is written to TIDLE_FIFO every time the status changes from active->idle or idle->active. An interrupt is generated as long as this FIFO contains any unread entries. The structure of the word contained in the FIFO is shown in Figure 32. The upper eight bits indicate the channel that encountered the status change and bit 0 indicates the status of the channel (Active =1; Idle = 0). The processor accesses the FIFO by reading the Status Interrupt register, which will contain the top element of the FIFO.

Figure 32 Pattern Match Idle Detection Interrupt Word



11.2.1.2 Cell Service Decision (CSD) Circuit

The CSD circuit determines which cells are to be sent and when. It determines this by implementing Transmit Calendar bit tables and Active/Idle bit tables. When the TALP builds a cell, the CSD circuit performs a complex calculation using credits to determine the frame in which the next cell from that queue should be sent. The CSD circuit schedules a cell only when a cell is built by the TALP. If SUPPRESS_TRANSMISSION bit in the TX_CONFIG word is set, then the cell is scheduled, however, the cell is not transmitted.

In non-DBCES mode, a queue can be placed in idle detection mode by setting the IDLE_DET_ENABLE in the TRANSMIT_CONFIG word within the queue table. When a queue is set for idle detection mode and all the channels on a given queue are inactive, cells are scheduled, but no cells are actually sent. This mode also requires that one of the idle detection methods is enabled for all the channels of the given queue. This can be done by programming the A1SP Idle Detection Configuration Table.

The following steps (as well as Figure 33 on page 88) describe how the CSD circuit schedules cells for the TALP to build.

1) Once the TFTC writes a complete frame into external memory, it writes the line number and frame number of this frame into the FR_ADVANCE_FIFO. The CSD circuit reads the line-frame number pair from the FR_ADVANCE_FIFO and uses it as an index into the Transmit Calendar. The Transmit Calendar is composed of eight-bit tables, one per line. Each bit table consists of 128 entries, one per frame buffer. Each entry consists of 32 bits, one per queue. For each bit set in the indexed entry in the Transmit Calendar, the CSD will schedule the frame in which the next cell can be built for the corresponding queue, and notify the TALP that enough data is available to build a cell for that queue.

2) The CSD circuit processes all queues from the Transmit Calendar entry starting with the lowest queue number and proceeding to the highest. The processing steps are as follows:

a) The CSD circuit obtains the QUE_CREDITS, and subtracts the average number of credits per cell from it. The average number of credits, AVG_SUB_VALU, is the number of credits that will be spent sending the current cell. For structured lines, the average number of credits per cell is 46 7/8. For unstructured lines, the average number of credits per cell is 47.

b) Next, the CSD circuit computes the frame location for the next service by subtracting the remaining credits from 47. It divides the result by the number of channels, NUM_CHAN, dedicated to that queue. The number of channels is calculated based upon the Active/Idle bit table and the channels allocated to the queue. If the chip is in non-DBCES mode, NUM_CHAN is equal to the number of channels allocated to the queue. If the chip is in DBCES mode, NUM_CHAN is equal to the number of allocated channels that are active, which is determined from the Active/Idle table. The result is a frame differential.

c) The CSD then adds this frame differential to the present frame location to determine the frame number of the next frame in which the TALP can build a cell. The CSD circuit then sets a bit in the corresponding entry in the Transmit Calendar and writes to the QUEUE_CREDITS.

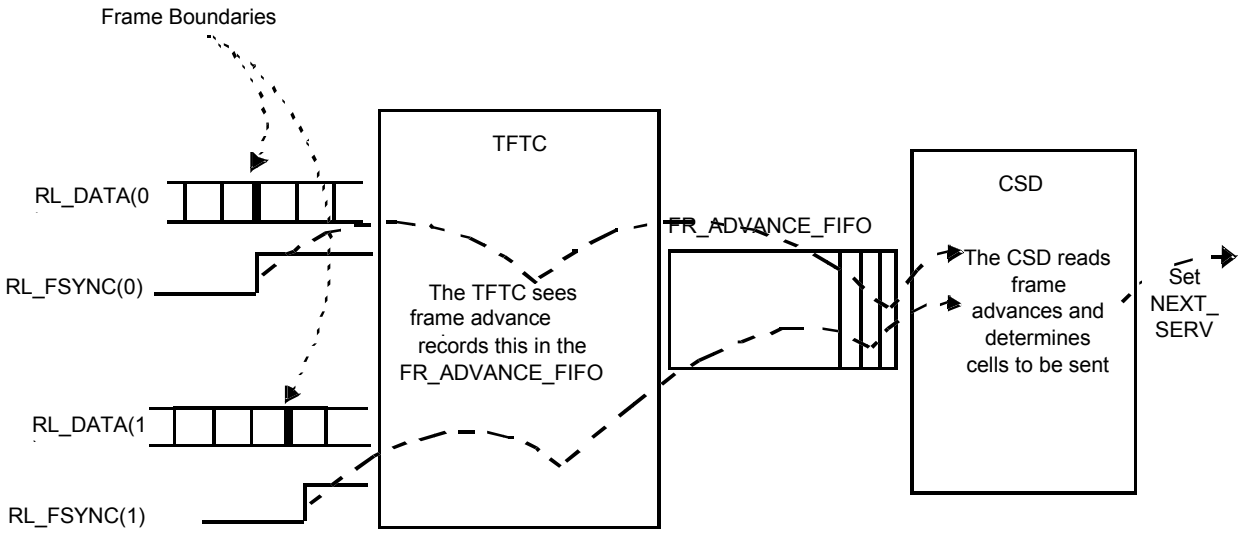
d) The CSD circuit then adds the new credits back to the credit total for the frame increment number. The number of new credits is equal to the frame differential computed earlier, multiplied by the number of channels for that queue. Once a queue is identified as requiring service, its identity is written to the NEXT_SERV location.

e) The CSD circuit obtains the next queue for that frame and repeats steps a. through d. The CSD circuit continues this process until there are no more active queues for that frame.

3) After servicing all the queues for that frame, the CSD circuit advances to the next active line located in the line queue. If there are no active lines, the CSD circuit returns to the idle state to wait for the next line to request service.

Figure 33 shows how the CSD assigns credits to determine in which frames cells should be sent.

Figure 33 Frame Advance FIFO Operation



The following is an example of the calculations the CSD circuit performs. This example assumes a structured line with four channels allocated to one queue in non-DBCES mode.

1) The TFTC writes Line 3 and Frame 4 to the FR_ADVANCE_FIFO.

2) The CSD circuit determines the queue for which a cell is ready by finding a set bit in the Transmit Calendar. In this example, it is queue number 100.

3) The CSD circuit reads the number of credits for queue number 100. The number of credits is always greater than 47 because it is ready for service. In this example, QUE_CREDITS = 59.375.

4) The CSD circuit subtracts AVG_SUB_VALU, the average number of credits spent per cell. (Remember: For structured lines, the average number of

credits per cell is 46-7/8. For unstructured lines, the average number of credits per cell is 47.)

Credits = 59.375 – 46.875

Credits = 12.5

5) The frame differential for the next service is computed from the number of credits needed to exceed 47 and NUM_CHAN, the number of channels allocated per frame.

47 – 12.5 = 34.5

34.5 / 4 = 8.625

Round 8.625 up, so the frame differential is 9.

6) Therefore, the next cell will be sent nine frames ahead of the current cell.

Next frame = present frame number + 9

7) The CSD circuit computes the number of credits for those nine frames and adds the result to the total.

New credits = 9 x 4 = 36

QUE_CREDITS = 36 + 12.5 = 48.5

If the queue is on a line in SDF-MF mode, the CSD makes a signaling adjustment to the QUE_CREDITS before writing this value to memory. (If the queue is not in SDF-MF mode, the signaling adjustment is not made and the QUE_CREDITS calculated in Step 7 is written to memory.)

The calculation determines the number of signaling bytes in the structure, then generates an average number of signaling bytes inserted into cells per frame, and finally multiplies this average number by the frame differential to adjust the QUE_CREDITS.

8) The CSD converts the frame differential from units of frames to units of one-eighth of multiframes.

In performing this calculation, the CSD also uses the FRAME_REMAINDER value from the QUE_CREDITS location in the T_QUEUE_TBL. This example assumes that FRAME_REMAINDER = 1 from the previous calculation on this queue.

E1: Frame differential (in eighths of a multiframe) = (frame differential + FRAME_REMAINDER) / 2

T1: Frame differential (in eighths of a multiframe) = (frame differential + FRAME_REMAINDER) / 3

Frame differential = (9 + 1) / 3 = 10, or three-eighths of a multiframe, remainder 1.

The CSD writes the remainder of this division into the FRAME_REMAINDER location for use in the next calculation on this queue.

9) The CSD calculates the signaling credit adjustment by multiplying the frame differential expressed in eighths of a multiframe by the number of signaling bytes in a structure.

Number of signaling bytes in the structure = 4 channels x 0.5 bytes per channel = 2 bytes per multiframe

Signaling adjustment = three eighths x 2 = 0.75 bytes

10) Then the CSD adds the signaling credit adjustment to the total and writes the result to memory, in preparation for the next service on this queue.

QUEUE_CREDITS = 48.5 + 0.75 = 49.25 bytes

Unstructured lines use a different procedure. In the case of unstructured lines, a cell will be sent every time 47 bytes are received. This assumes that no partial cells are used for UDF mode.

For DBCES the algorithm is similar. The main change is that any time a channel is activated or deactivated, the scheduling of the next bitmask cell has to dynamically update to account for the change in the number of active channels. If no channels are active a cell with an Inactive structure will be sent every 144 ms in T1 mode and 192 ms in E1 mode. DBCES is only supported for full cells.

11.2.1.2.1 Transmit CDV

The ideal minimum transmit CDV for all queue configurations is as follows. In each case, the frame rate is assumed to be 125us.

- UDF-ML/LOW_CDV bit set: 0 us.
- SDF-FR/Single-DS0-with-no-pointer: 0 us.

- SDF-FR Partial cell configurations with bytes_per_cell/num_chan = an integer: 0 us.

- All other configurations: 125 us.

The following items affect transmit CDV:

- Cell scheduling
- Contention with other cells scheduled at same time
- Actual cell build time
- UTOPIA contention
- OAM cell generation

1) The scheduler has a resolution of 125 µs. In other words, it works off a frame-based clock to determine whether or not a cell should be sent during the current frame. Therefore, if the ideal rate of cell transmission is not a multiple of 125 µs, there will be 125 µs of CDV. The scheduler will never add more than 125 µs of CDV.

For example, a single DS0 queue with no signaling and using full cells, will need to build a cell every 47 frames. Therefore, a cell will be scheduled every 47 frames, and the scheduler will add no CDV. Also in UDF mode all cells are sent every time 47 bytes are received so no CDV is added.

However, if signaling were added to the single DS0 queue, the extra byte that occurs every 24 bytes (assuming T1 mode) requires compensation. In this case, a cell will be sent every 46 or 47 frames. Therefore, there will be up to 125 µs of CDV due to the scheduler.

Note that for UDF lines there is a LOW_CDV bit which can be set in the LIN_STR_MODE memory register which will cause cells to be scheduled every 47 bytes instead of frame based. This eliminates the CDV caused by the scheduler. This mode can only be used in UDF-ML mode when BYTES_PER_CELL is 47. In High Speed mode cells are always scheduled every 47 bytes which assumes that partial cells are never used in HS mode.

2) Only one cell can be built at a time. Thus if multiple queues are scheduled to send cells during the same frame, additional delay will be incurred. If queues are activated and deactivated so that the number of queues scheduled ahead of a specific queue in the same frame changes, the resulting change in delay translates to CDV. The scheduling of multiple cells at the same time is known as

clumping. It takes approximately 8 µs to build a cell normally but can take up to 15 us under worst case traffic and processor activity (Note this assumes a 38.88 MHz SYS_CLK). Therefore, each cell that is waited for could add up to 15 µs of delay. When multiple queues are scheduled to send cells at the same time, the cells will be built in sequential order, starting with 0 and going to 256. Therefore, in a system that will be adding and dropping queues, the higher number queues will experience more CDV than the lower number queues, depending on how many queues are active at the time, and are scheduled within the same frame. The AAL1gator-32 minimizes the effects of clumping by offsetting the schedule point of each line by $1/8^{th}$ of a frame. Also when queues are added, an offset field can be supplied which will force multiple cells on the same line to be scheduled at different times. See Add Queue FIFO section in the Processor Interface section for more details.

3) For configurations that will require sending a cell every n frames where n is an integer divisor of 128 (for E1) or 96 (for T1), the cells will always be scheduled in the same frame unless the offset field is set differently for each cell.

4) The actual build time of a cell depends on microprocessor activity and contention with other internal state machines for the AAL1gator-32 memory bus. Therefore there will be some minor CDV that is added on a per cell basis, based on current microprocessor/memory traffic. This CDV is usually less than 4 µs and is not very noticeable.

6) If there is backpressure on the UTOPIA bus, cells will not be able to be sent which also causes CDV.

7) Since OAM cells have higher priority than data cells the transmission of OAM cells should be distributed. An OAM cell can add up to 8 us of CDV assuming a 38.88 MHz SYS_CLK under worst case processor load.

11.2.1.3 Transmit Adaptation Layer Processor (TALP)

11.2.1.3.1 OAM Cell Generation

When an OAM cell transmission is requested, it is sent at the first available opportunity. Transmit OAM cells have higher priority than cells scheduled by the CSD circuit. Because of this, care should be taken to ensure that OAM cells do not overwhelm the transmitter to such an extent that data cells are starved of adequate opportunities. The rate of OAM cells must be limited for the AAL1gator-32 to maintain its maximum CSD data rate.

To send an OAM cell, the microprocessor writes OAM cells into one of two dedicated cell buffers located in external memory. When the cell is assembled in the buffer, the microprocessor must set the appropriate bit in the Command register The TALP sends the cell as soon as possible, then clears the appropriate attention bit to indicate the requested cell has been sent. If requests for both OAM cells are active at the time the command register is read by the AAL1gator-32, OAM cell 0 will always be sent because it is assigned a higher priority. Therefore, to control the order of OAM cell transmission, the microprocessor should set only one OAM attention bit at a time and wait until it is cleared before setting the other attention bit.

OAM cells can optionally have the 48-byte OAM payload CRC-10 protected. This is accomplished by a CRC circuit that monitors the OAM cell as it is sent to the TUTOPIA and computes the CRC on the fly. It then substitutes the 10 bit resultant CRC, preceded by six 0s, for the last two bytes of the cell. The CRC generation is enabled by setting Bit 0 in Word 2 of the T_OAM_CELL.

11.2.1.3.2 Data Cell Generation

If the TALP receives a request to send a CSD-scheduled data cell and there are no OAM cell requests pending, it will do so as soon as it is free. It will look up the predefined ATM header from the T_QUEUE_TBL (refer to section 7.6.8 “T_QUEUE_TBL” on page 122). It will then obtain a sequence number for that queue from memory, and a structure pointer if necessary. After these bytes are written to the TUTOPIA interface, the TALP will then go to the data and the signaling frame buffers, locate the data bytes for the correct channels, and write them in the correct order to the UTOPIA interface. This cell building process is described in more detail in the following section.

11.2.1.3.2.1 Header Construction

The entire header is fixed per queue. Headers are maintained in the memory, one per queue. These headers include a Header Error Check (HEC) character for the fifth byte. The queue should be deactivated during header replacement to prevent cells from being constructed with incorrect header values. A queue can be paused by setting the SUPPRESS_TRANSMISSION bit in TX_CONFIG register. Emissions are still scheduled, just the transmissions are suppressed. For any cells that are suppressed, the T_SUPPRESSED_CELL_CNT is incremented.

11.2.1.3.2.2 Payload Construction

Payload construction is the most complex task the TALP circuit performs. The AAL1 requirements define much of the process, which is as follows:

1) The first byte of the payload is provided by a lookup into the T_QUEUE_TBL. This first byte consists of the CSI bit, a 3-bit sequence number, and a 4-bit sequence number protection field. The CSI bit is set depending on SRTS and pointer requirements. The sequence number is incremented every time a new cell is sent for the same VPI/VCI. If the queue has been configured for AAL0 mode, this step is not done and an additional data byte is loaded instead.

2) If the line is in one of the two structured modes, a structure pointer is needed in one of the even-numbered cells. The TALP inserts structure pointers according to the following rules:

- Only one pointer is inserted in each 8-cell sequence.
- A pointer is inserted in the first possible even-numbered cell of every 8-cell sequence.
- A pointer value of 0 is inserted when the structure starts in the byte directly after the pointer itself.
- A pointer value of 93 is inserted when the end of the structure coincides with the end of the 93-octet block of AAL-user information.
- A dummy pointer value of 127 is inserted in cell number six if no start-of-structure or end-of-structure occurs within the 8-cell sequence.

3) This algorithm supplies a constant number of structure pointers and, therefore, data bytes, regardless of the structure size. The pointer is inserted in the seventh byte location of the cell. To force the TALP to build a structure consisting of a

single DS0 with no signaling nibble and no pointer, set T_CHAN_UNSTRUCT = 1 in the QUEUE_CONFIG word of the T_QUEUE_TBL.

4) The TALP fills the rest of the cell payload with data and/or signaling information. The T_CHAN_ALLOC table in the transmit queue table determines which channels are dedicated to which queue. If a bit is set, the channel represented by that bit is assigned to that queue. The TALP successively writes the data from the marked channels into the UTOPIA interface. If the LOOPBACK_ENABLE bit is set in the TX_CONFIG register then the cell is written into a separate FIFO to be looped back to RALP. A queue-based parameter, BYTES_PER_CELL, decides when enough payload bytes have been obtained. If this number is fewer than 47, then the remaining bytes for the cell are loaded with P_FILL_CHAR. This implies that because of the presence of the structure pointer, the number of fill bytes will not be constant for structured data queues.

DBCES mode requires some additional adjustments. A bitmask must be placed at the beginning of a structure that is pointed to by a structure pointer. This bitmask can be one to four bytes in length. Also, the active status of the channels must be factored in. Only if a bit is set in the T_CHAN_ALLOC table and the corresponding channel is active does the TALP write data from the channel into the UTOPIA interface. If none of the channels is active, the cell will be filled with the null bitmask.

5) The structure used for signaling is determined by the mode of the line and the value of E1_WITH_T1_SIG. Normally the signaling structure will follow the mode of the line. However, if the line is in E1 mode and E1_WITH_T1_SIG is set, then a T1 signaling structure is used. This means that for a single DS0, signaling is inserted after 24 data bytes instead of after 16 data bytes. If data is to be sent from the data queue, this process continues byte-by-byte while updating pointers and counters until one of the following occurs:

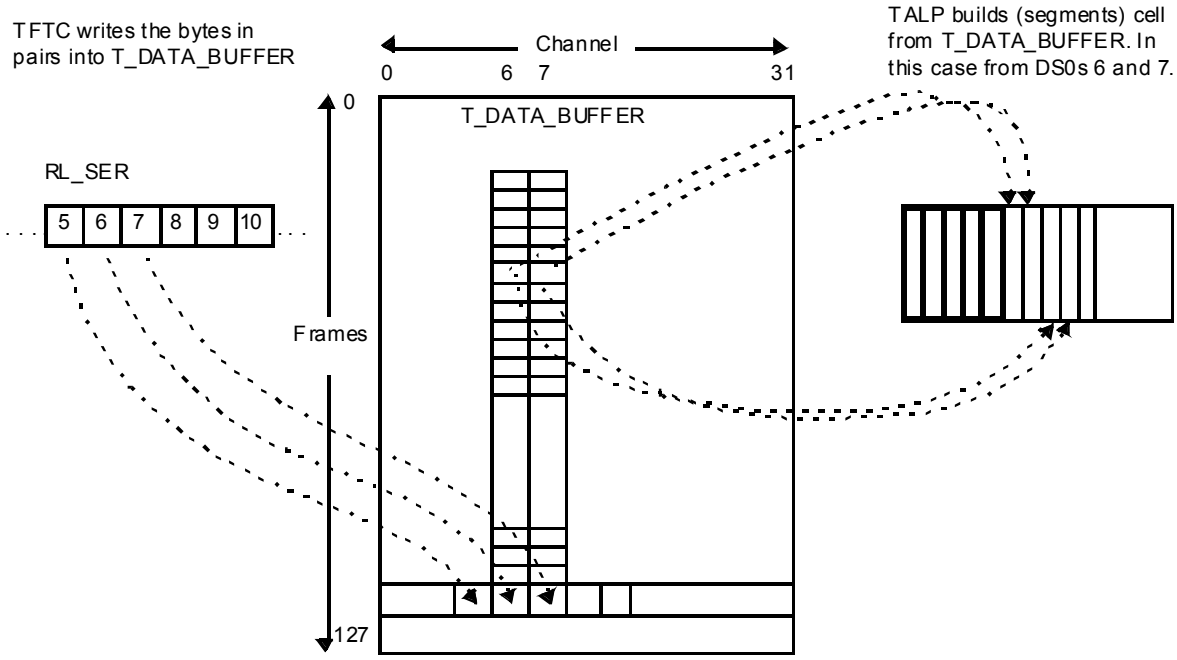
- The cell is complete.
- The last data byte for the last frame of the multiframe has been set.

6) When signaling information is to be sent, data is obtained from the signaling locations of the multiframe, with the help of the channel allocation table (T_CHAN_ALLOC). This process proceeds byte-by-byte until one of the following occurs:

- The cell is complete.
- All signaling nibbles for all channels assigned to the queue have been sent.

Figure 34 shows an example of the payload generation process.

Figure 34 Payload Generation



For AAL0 mode the cell build process takes 48 bytes of line data and does not add any AAL1 overhead bytes.

For DBCES mode, anytime a pointer is generated, the subsequent start of structure will contain the bitmask field with the number of channels currently active. Changes in the bitmask can only occur at this time.

11.2.1.3.3 Peak Cell Rates (PCRs)

For purposes of discussion, the following PCR information is assumed:

- Full cells are used,
- The PCR numbers are per line, and
- The SYS_CLK is 38.88 MHz.

11.2.1.3.3.1 Peak Cell Rates (PCRs) for Structured Cell Formats

- For structured connection without CAS, PCR <= 171 x n cells per second per connection where 1 <= n <= 32 (assuming completely filled cells).

- For structured connection with CAS, PCR <= 182 x n cells per second per connection where 1 <= n <= 32 (assuming completely filled cells).

- Each AAL1 cell is either 46 or 47 bytes, depending upon whether or not the cell contains a structure pointer.

11.2.1.3.3.2 Peak Cell Rates (PCRs) for Unstructured Cell Formats

- PCR <= 4,107 cells per second for T1 (assuming 47 bytes for each AAL1 cell).

- PCR <= 5,447 cells per second for E1 (assuming 47 bytes for each AAL1 cell).

- PCR <= 118,980 cells per second for T3 (assuming 47 bytes for each AAL1 cell).

- PCR <= 91,405 cells per second for E3 (assuming 47 bytes for each AAL1 cell).

- If all eight lines in an A1SP are at the same rate, totaling 20 Mbps throughput for the A1SP, then the aggregate A1SP PCR <= 53,191 cells per second for multiple-line unstructured data format (assuming 47 bytes for each AAL1 cell). If all lines are not at the same rate, the aggregate A1SP PCR <= 46,542.

- PCR <= 1,000 cells per second per A1SP for OAM cells. This rate of OAM cells is calculated on the basis of up to four cells per second per VC. Transmitting and receiving OAM cells at this rate consumes 20% of the microprocessor accesses.

11.2.1.3.3.3 Peak Cell Rates and Partial Cells

Partial Cells can be used to minimize the amount of delay required to assemble a cell. However, the amount of overhead required for the same amount of TDM data increases when partial cells are used. This overhead increases as the number of data bytes per cell decreases. The following table shows the minimum partial cell sizes that can be supported for the different modes of operation if all connections are active on the device. If a smaller partial cell size is desired, either some time slots/links must not be used or only a subset of the connections must use that partial cell size. In any case the total PCR of the device should not exceed 400,000 cells per second with a 38.88 MHz SYSCLK or 450,000 cells per second with a 45 MHz SYSCLK.

Table 5 Minimum Partial Cell Size Permitted If All Connections Are Active

MODE	SYS_CLK=38.88 MHz	SYS_CLK = 45 MHz
T1 SDF-FR	16	14

MODE	SYS_CLK=38.88 MHz	SYS_CLK = 45 MHz
T1 SDF-MF	16	15
E1 SDF-FR	21	19
E1 SDF-MF	22	20

11.2.1.4 Transmit FIFO (TALP_FIFO)

This block contains an 8 cell FIFO which buffers cells going out to the UTOPIA Interface block.

The UTOPIAI polls the A1SP blocks in a round robin fashion to determine which A1SP has cells available in their TALP_FIFO. If the TUTOPIA FIFO has room for a cell, the A1SP block is selected, a cell is read from the selected A1SP's TALP FIFO, and moved into the TUTOPIA FIFO.

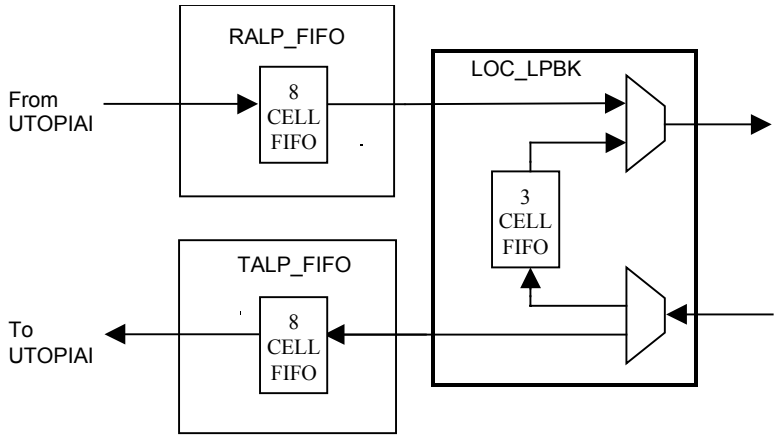
11.2.1.5 Local Loopback Block(LOC_LPBK)

The AAL1gator-32 supports local loopback of cells. Local loopback is when a cell generated by TALP is looped back to RALP instead of being sent out to the UTOPIA bus via the TALP FIFO.

When LOOPBACK_ENABLE is set in the TRANSMIT_CONFIG register, cells for that queue will be looped back to the RALP block instead of being transmitted toward the UTOPIA bus. Since this bit is configurable on a per queue basis it can be used during normal operation to loop back individual time slots, groups of time slots on a given line or even entire lines. The cells, which are looped back, are stored in a separate 3 cell FIFO that RALP can read. The loopback FIFO will have higher priority to prevent the transmitter from backing up. See Figure 35 for architecture of the local loopback.

Note: Remote loopback is also supported in the AAL1gator-32, but it is performed in the UTOPIAI block.

Figure 35 Local Loopback



11.2.2 AAL1 SAR Receive Side (RxA1SP)

11.2.2.1 RALP FIFO (RFIFO)

This block contains an 8 cell FIFO, which buffers cells received from the UTOPIA Interface block. Each A1SP has it's own 8 cell FIFO to allow the RALP to process the bytes from the cell at it's own pace without any interaction with the other A1SPs.

Once the RUTOPIA FIFO in the UTOPIAI block has a cell to send, the UTOPIAI polls the A1SP associated with the cell to determine if the RALP FIFO has room for the cell. Once the RALP FIFO has room for the cell, the UTOPIAI reads the cell out of the RUTOPIAI FIFO and places it in the RALP FIFO.

11.2.2.2 Receive Adaptation Layer Processor (RALP)

The RALP receives ATM cells from the RALP_FIFO and removes and checks the AAL1 encapsulation as well as providing a VCI to queue mapping. It performs pointer searches, detects SN mismatches, and extracts the line data from the cells and stores it in the receive buffers located in external memory. It also checks for overrun and underrun conditions, processes DBCES bitmasks and checks for OAM cells The RALP does not verify the HEC because it expects a PHY layer device to verify the HEC before presenting the cell to the AAL1gator-32.

11.2.2.2.1 VCI Mapping

The AAL1gator-32 supports two options for VCI to queue mapping. Eleven bits of the VCI are always used. The eleven-bit field can be either located in the least significant bits of the VCI or shifted up 4 bits to occupy bits 14 down to 4 of the VCI field. The two most significant bits in the field are used to determine which A1SP processor receives the cell.

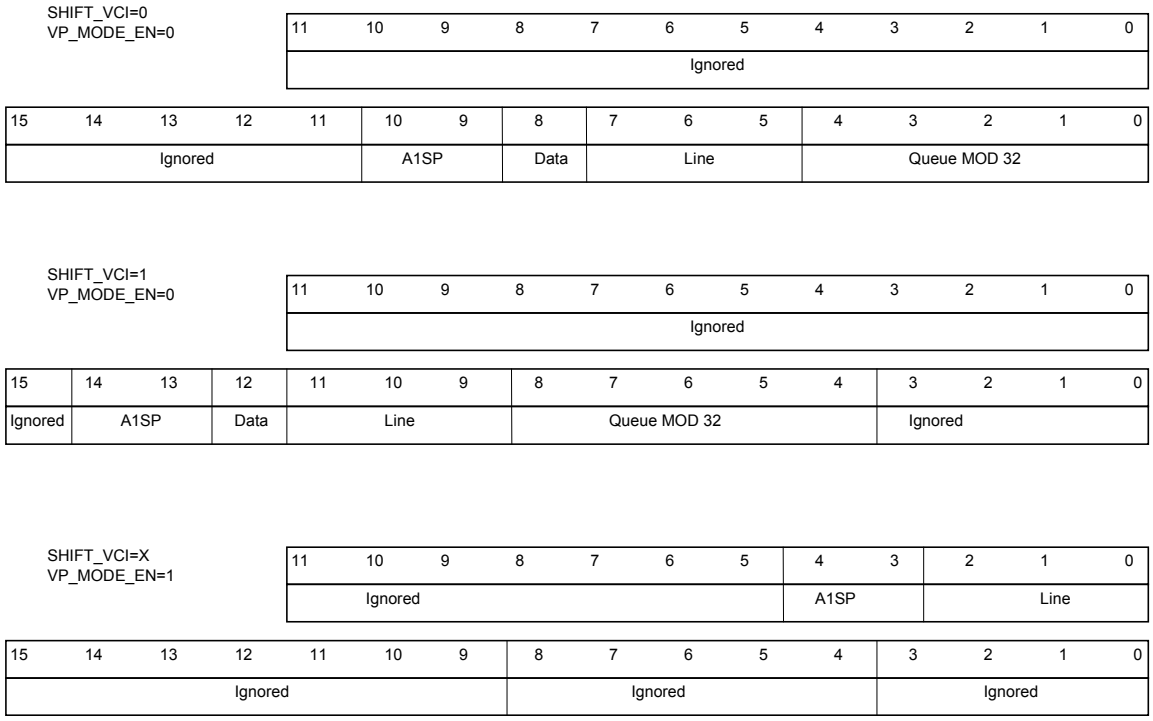
There are two methods of detecting a cell to be redirected to the OAM queue. The first method uses a data indicator bit in the header. The second method uses the customary PTI field. If the data indicator is not set (VCI(8) = 0 when SHIFT_VCI=0) or Payload Type Indicator (PTI) = 4 to 7, the cells are sent to the OAM queue and are stored using the pointers located in the OAM receive queue table. The head pointer is the address to the first cell received for each queue, and is usually maintained by the microprocessor. The tail pointer is the address of the last cell of the queue, and is maintained by the RALP. The RALP updates the tail pointer upon each OAM cell arrival.

If the cell is a data cell (cells received with VCI(8) = 1 when SHIFT_VCI=0 and PTI = 0 to 3), the cell is sent to the queue with VCI(7:0). Bits 7:5 determine the line, and bits 4:0 determine the queue. The A1SPn block ignores VCI(15:9) and VPI(11:0). If SHIFT_VCI = 1, the interpretation of the VCI bits is shifted by four bits.

There is also a special mode when VP_MODE_EN is set in the UI_COMN_CFG register. When set the VPI field is used to select the A1SP and Line number. Queue 0 is always assumed. So this mode is only available if all lines are configured with one queue. Normally this mode can only be used if all lines are configured in UDF-ML mode. In this mode the VCI field is used to detect an OAM cell. If the VCI is less than or equal to 31 then the cell will go to the OAM cell buffer.

Figure 36 shows the interpretation of the incoming cell header once the cell arrives at the A1SP.

Figure 36 Cell Header Interpretation



11.2.2.2.2 Sequence Number Processing

When the cell is a data cell, the RALP verifies the SN CRC-3 code and parity bit. If the SN CRC-3 and parity do not verify, the RALP attempts to correct it as specified in ITU-T Recommendation I.363.1 and increments the R_INCORRECT_SNP counter. If the RALP cannot correct the SN byte, it identifies the cell as invalid.

If the DISABLE_SN bit in the R_SN_CONFIG register has been set for this queue, then no further sequence number processing occurs. There are two modes of sequence number processing supported: Fast Sequence Number processing algorithm on all types of connections and Robust Sequence Number processing on Unstructured Data Format (UDF) connections. Both the Fast and Robust algorithms are done in accordance with the algorithms described in ITU-T Recommendation I.363.1.

For the “Fast SN Algorithm”, the RALP acts upon the current cell based on the value of the current SN and SNP and the previous SN and SNP. The RALP either accepts the current cell, drops the current cell, or accepts the current cell and inserts cells.

For the “Robust SN Algorithm”, the RALP makes decisions on the previous cell based on the value of the current SN and SNP and the previous SN and SNP. The RALP either accepts the previous cell, drops the previous cell, or inserts cells and accepts the previous cell. The Robust SN algorithm adds delay by requiring the following cell to arrive prior to accepting the previous cell. When a cell is accepted, it means that the contents of the cell are available to the RFTC for transmission onto the line. This requires that two write pointers be maintained, one that is at the end of the last accepted cell and one at the end of the last stored/received cell.

Inserted cells have the following properties:

- If the queue is in an unstructured mode or if the queue is in structured mode and the inserted cell should not contain a pointer, each inserted cell contains the number of payload bytes as determined by R_BYTES_CELL field in R_MP_CONFIG.

- If the queue is in a structured mode and the inserted cell should contain a pointer and R_BYTES_CELL is equal to 47, then the inserted cell contains 46 payload bytes.

- If the queue is in a structured mode and the inserted cell should contain a pointer and R_BYTES_CELL is less than 47, then the inserted cell contains R_BYTES_CELL payload bytes.

- If the queue is in DBCES mode and a bitmask is expected in the missing cell, then the payload bytes will be adjusted to take into account the missing bitmask bytes. The bitmask will be assumed to keep the same value. That is the cell structure will be processed as if the number of active channels did not change.

- The determination on whether or not the inserted cell should contain a pointer is based on the pointer generation rules defined by ITU-T Recommendation I.363.1. A pointer will be assumed if the queue is structured and the following conditions are met:

    - The SN is even AND there has been no other pointer in the group of eight cells so far AND (the sequence number = 6 OR the structure ends within the current inserted cell or next cell).

- If the queue is in SDF-MF mode and the inserted cell should contain signaling data, the number of payload bytes is adjusted but the signaling information is copied from the last valid signaling and is written to the signaling buffer. Therefore, signaling information is frozen during the playout of the frame with invalid signaling information.

Note: In SDF-FR mode, the CES specification states that if the queue contains data for only one DS0, no pointer is used. If a queue is configured in this manner, set R_CHAN_UNSTRUCT in the R_BUF_CFG receive queue table memory register..

- The value of the payload data depends on the value of INSERT_DATA in the R_SN_CONFIG receive queue table memory register.. The default is to load the value of 0xFF. Other options are to use conditioned data as defined by R_CONDQ_DATA, old data, or pseudorandom data. If old data is chosen, no data will be written and the old data in the receive buffer will be used. The receive buffer write pointer will be adjusted to the correct location. If the pseudorandom data option is chosen, the data played out will be the value of R_CONDQ_DATA with the MSB replaced by the current value of the pseudorandom number generator $x^{18} + x^7 + 1$.

  Notes:

  - All DS0s within the replaced cell will use the same algorithm. To minimize the overhead of generating the inserted cells, use the old data option whenever possible. The old data option still needs to do internal processing on a byte-by-byte basis, but since it does not have to write any data, it is about twice as fast as the other options.

  - The “Fast SN Algorithm” will, under certain situations, allow bad cells to pass through. When this occurs the cells are marked as potentially bad. Any cells marked as potentially bad will not have pointer verification done on them and any signaling data or SRTS information they contain will not be written. However, if these cells should contain pointers or signaling data, adjustments are made to the amount of payload data written so bit integrity is maintained.

  - The pseudorandom option is not available for UDF-HS mode.

The RALP will maintain bit integrity if there is no more than one consecutive errored cell, or if there are up to six lost. If the receive buffer underruns and BITI_UNDERRUN is set, then the amount of time that has passed since the last cell is checked and if it appears that less than 6 cell times have passed then cells will be inserted if no more than MAX_INSERT cells are detected as lost. Otherwise the queue will resync and realign with the structure if one exists.

MAX_INSERT controls the maximum number of cells that will be inserted when cells are lost. If more cells than MAX_INSERT are lost, then the queue will be forced into an underrun condition unless the lost cells are a result of coming out of underrun. The default value of MAX_INSERT, “000”, is equivalent to “111”

which means that up to seven cells will be inserted. MAX_INSERT is in the R_SN_CONFIG register and can be specified on a per-queue basis.

If MAX_INSERT and R_MAX_BUF are configured such that inserting the maximum number of cells into a buffer which is near it's R_MAX_BUF limit will cause the number of frames in the buffer to exceed 1FE, the "Robust SN Algorithm" may cause overruns due to misinserted cells and failure of the SN error protection mechanism. This will occur when a cell arrives which causes the SN processing to go to the OUT OF SEQUENCE state and the buffer is too full to allow the cell to be treated as a lost cell and stored as if cells had been lost.

If seven cells are lost, this will appear as a misinserted cell and will not be handled correctly. Likewise, if more than seven cells are lost, it will appear as if fewer than seven cells were lost because the SN repeats every eight cells. If seven or more cells were lost, there is high probability that the queue will underrun. If the queue has not underrun, the RALP takes the following steps to minimize the impact:

- Any time cells are inserted, if the next received pointer mismatches, it will immediately create a forced underrun to realign to the structure instead of waiting for two consecutive mismatches.
- No signaling information will be updated until a valid and correct pointer is received.

If the DISABLE_SN bit in the R_SN_CONFIG receive queue table memory register (refer to "DISABLE_SN" on page 242) is set, then both the Fast and Robust Sequence Number Processing algorithms are disabled.. That is, RALP will neither insert nor drop cells due to sequence number errors, but will update both the R_INCORRECT_SN and R_INCORRECT_SNP counters.

If R_AAL0_MODE bit is set in the R_MP_CONFIG register then no SN processing is done and byte six of the cell is handled as a data byte.

11.2.2.2.3 Fast Sequence Number processing State Machine

The RALP sequence number processing state machine begins in the START state. Once a cell is received with a valid SN, the OUT_OF_SYNC state is entered. Any cells received while in the START state, including one with a valid SN are dropped, unless NODROP_IN_START, in the R_SN_CONFIG receive queue table memory register, is set. If this bit is set and the cell has a valid SN, the cell will be accepted.

Note: If it is important to not dump the first cell received, make sure NODROP_IN_START is set.

While in the OUT_OF_SYNC state, if another cell is received with a valid SN and in the correct sequence, then the SYNC state is entered and the cell is accepted. Otherwise, if a cell with an invalid SN is received, then the START state is re-entered and the cell is dropped. Otherwise, if the cell has a valid SN but is in the incorrect sequence, then the cell is dropped and the RALP remains in the OUT_OF_SYNC state.

While in the SYNC state, if another cell is received with a valid and correct SN, the RALP remains in the SYNC state. Otherwise, if an invalid SN is received, the RALP goes to the INVALID state; or if a valid but incorrect SN is received, the OUT_OF_SEQUENCE state is entered. All cells received while in the SYNC state are accepted.

While in the INVALID state, four possibilities can occur.

1. If the cell received has an invalid SN, the START state is re-entered and the cell is dropped.

2. If the cell received has a valid SN and is in sequence with the last valid SN, then a misinserted cell is detected and RALP returns to the SYNC state, but the cell is dropped to keep SN integrity because the previous cell has already been sent.

3. If the cell received has a valid SN and is equal to SN + 2 with respect to the last valid SN, then the RALP returns to the SYNC state and the cell is accepted.

4. Otherwise, if the SN is valid but does not meet any of the previous criteria, then the cell is dropped and the OUT_OF_SYNC state is entered.

While in the OUT_OF_SEQUENCE state, five possibilities can occur.

1. If the cell received has an invalid SN, the START state is re-entered and the cell is dropped.

2. If the SN is valid and in sequence with the last valid, in sequence SN, then a misinserted cell is detected and RALP returns to the SYNC state, but the cell is dropped to keep SN integrity because the previous cell has already been sent.

3. If the SN is valid and the SN is in sequence with the SN of the previous cell, the RALP assumes cells were lost; it inserts a number of dummy cells identical to the number of lost cells, accepts the cell and returns to SYNC. If the number of lost cells is greater than MAX_INSERT, then no cells are inserted and a forced underrun occurs. If an underrun occurred when cells were lost, no cells are inserted unless bit integrity through underrun is

enabled.

Note that if lost cells are detected, but the queue is currently in underrun and BITI_UNDERRUN is not set, then lost cells will not be inserted and a normal underrun recovery will be executed. If lost cells are detected, the queue is in underrun, and BITI_UNDERRUN is set, and less then six cells were lost, then lost cell will be inserted and processing will occur as if the underrun did not happen.

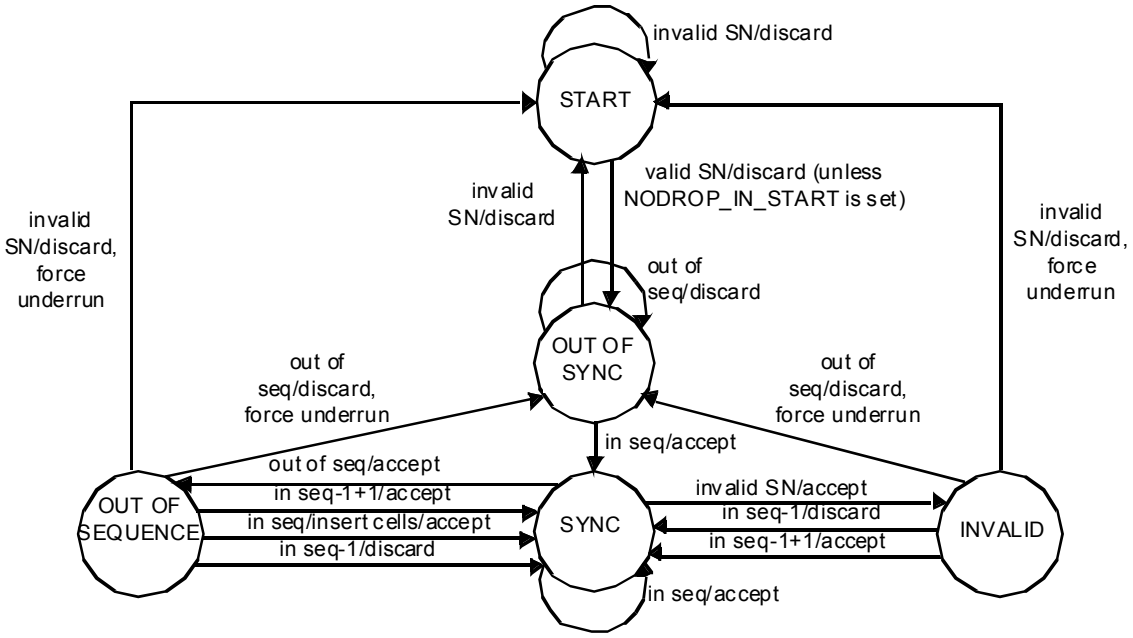
4. If the received SN is valid and the SN has a value equal to SN + 2 with respect to the last SN received in sequence, then the cell is accepted and the RALP returns to the SYNC state.

5. And finally, if the SN is valid but does not meet any of the previous criteria, then the cell is dropped and the RALP enters the OUT_OF_SYNC state.

See Figure 37 on page 107 for the flow of the “Fast SN Algorithm”.

Anytime a cell is dropped, the R_DROPPED_CELLS , in the receive queue table memory register,is incremented and the SN_CELL_DROP sticky bit is set. Anytime a cell is detected lost, the R_LOST_CELLS is incremented by the number of lost cells. Anytime the SN state machine transitions from the SYNC state to the OUT_OF_SEQUENCE state, the R_SEQUENCE_ERR counter is incremented. Anytime a misinserted cell is detected the R_MISINSERTED counter is incremented. In all these cases the RCVN_STAT_FIFO will be written with the error that occurred and the queue number for which the error occurred, if the corresponding mask bit is not set or this is not the first unmasked sticky bit to be set for this queue since the sticky bit register was last cleared.

Figure 37 Fast SN Algorithm

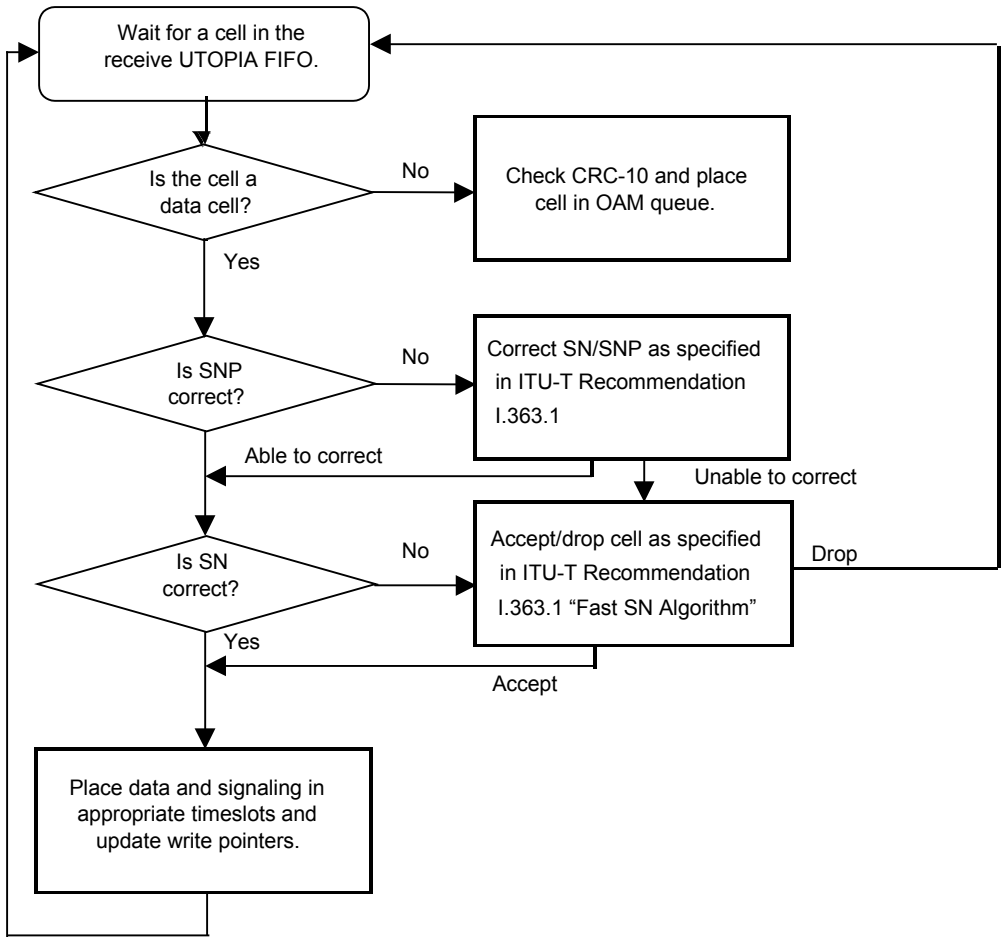


All cells received while in the SYNC state are accepted whether or not they are good. Any errored cells received while in the SYNC state are marked as potentially bad cells. These marked cells will not have their pointers checked, or bitmask checked, if they contain one; or if they contain signaling data, the signaling data will not be written to memory.

If the cell is accepted, the RALP then transfers the cell to the external memory using the R_CHAN_ALLOC fields in the R_QUEUE_TBL. Figure 38shows this receive cell process. If a valid cell is not received in time, the queue may enter an underrun condition.

Note that during an underrun, the RALP ‘Fast’ sequence number processing state machine freezes in its current state. For example, if the state machine is in the SYNC state when underrun occurs, when the next cell arrives, potentially some time later, the state machine will still be in the SYNC state.

Figure 38 Receive Cell Processing for Fast SN



11.2.2.2.4 Robust Sequence Number processing State Machine

The Robust SN processing State Machine is the same as the Fast SN processing state machine except that all actions are taken on the previous cell versus the current new cell. Robust Sequence Number processing is only supported in the UDF-ML and UDF-HS modes with full cells.

The RALP robust sequence number processing state machine begins in the START state. While in the START state two possibilities can occur:

1. If a cell is received with a valid SN, the OUT_OF_SYNC state is entered and the cell is stored.

2. If a cell is received with an invalid SN, the cell is dropped and the START state is maintained

While in the OUT_OF_SYNC state, three possibilities can occur.

1. If a cell is received with a valid SN and in the correct sequence, then the SYNC state is entered and the previously stored cell is accepted.

2. If a cell with an invalid SN is received, then the START state is re-entered and the previously stored cell is dropped.

3. If the cell has a valid SN but is in the incorrect sequence, then the previously stored cell is dropped, the new cell is stored and the RALP remains in the OUT_OF_SYNC state.

While in the SYNC state, three possibilities can occur.

1. If a cell is received with a valid and correct SN, the RALP remains in the SYNC state. The stored cell is accepted and the new cell is stored.

2. If an invalid SN is received, the RALP goes to the INVALID state. The stored cell is accepted and the new cell is stored

3. If a valid but incorrect SN is received, the OUT_OF_SEQUENCE state is entered. The stored cell is accepted and the new cell is stored as the next cell and also in the position it would reside if cells had been lost . If there is not physically enough space in the buffer to store the second copy of the cell an overrun condition is declared.

While in the INVALID state, four possibilities can occur.

1. If the cell received has an invalid SN, the START state is re-entered and the stored cell is dropped.

2. If the cell received has a valid SN and is in sequence with the last valid SN, then a misinserted cell is detected and RALP returns to the SYNC state, the stored cell with invalid SN is dropped and the new cell is stored.

3. If the cell received has a valid SN and is equal to SN + 2 with respect to the last valid SN, then the RALP returns to the SYNC state and the stored cell is accepted and the new cell is stored.

4. Otherwise, if the SN is valid but does not meet any of the previous criteria, then the stored cell is dropped, the new cell is stored and the OUT_OF_SYNC state is entered.

While in the OUT_OF_SEQUENCE state, five possibilities can occur.

1. If the cell received has an invalid SN, the START state is re-entered and the stored cell is dropped.

2. If the SN is valid and in sequence with the last valid, in sequence SN, then a misinserted cell is detected and the stored cell is dropped. The RALP returns to the SYNC state and stores the new cell.

3. If the SN is valid and the SN is in sequence with the SN of the previous cell, the RALP assumes cells were lost; it inserts a number of dummy cells identical to the number of lost cells, accepts the second copy of stored cell and returns to SYNC. If the number of lost cells is greater than MAX_INSERT, then no cells are inserted and a forced underrun occurs. If an underrun occurred when cells are lost, no cells are inserted unless bit integrity through underrun is enabled.

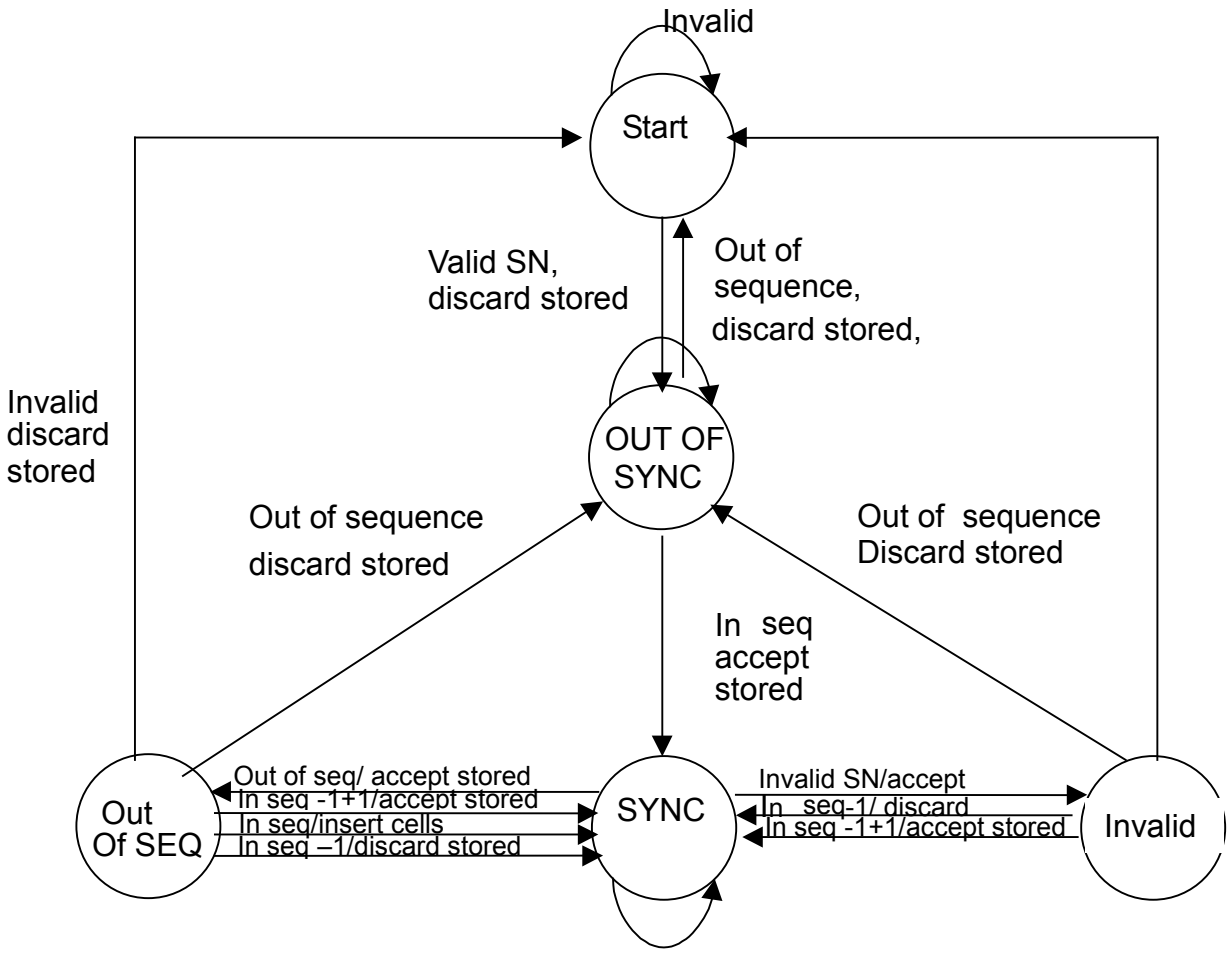
4. If the received SN is valid and the SN has a value equal to SN + 2 with respect to the last SN received in sequence, then the stored cell is accepted, the new cell is stored and the RALP returns to the SYNC state.

5. And finally, if the SN is valid but does not meet any of the previous criteria, then the stored cell is dropped, the new cell is stored and the RALP enters the OUT_OF_SYNC state.

See Figure 39 on page 111 for the flow of the “Robust SN Algorithm”.

Anytime a cell is dropped, the R_DROPPED_CELLS counter is incremented and the SN_CELL_DROP sticky bit is set. Anytime a cell is detected lost, the R_LOST_CELLS counter is incremented by the number of lost cells. Anytime the SN state machine transitions from the SYNC state to the OUT_OF_SEQUENCE state, the R_SEQUENCE_ERR counter is incremented. Anytime a misinserted cell is detected the R_MISINSERTED counter is incremented. In all these cases the RCV_STAT_FIFO will be written with the error that occurred and the queue number for which the error occurred, if the corresponding mask bit is not set or this is not the first unmasked sticky bit to be set for this queue since the sticky bit register was last cleared.

Figure 39 Robust SN Algorithm



If the cell is stored, the RALP then transfers the cell to the external memory using the R_CHAN_ALLOC fields in the R_QUEUE_TBL. If a cell is accepted, the wr_ptr for the RFTC is advanced to make the cell available to the RFTC. . If a valid cell is not received in time, the queue may enter an underrun condition.

The CDVT value allows the receiver to be configured to store a variable amount of data for that queue before data is emitted. This storage permits the cells to arrive with variable delays without causing errors on the line outputs. This CDVT value is used when the first cell is received after an underrun. The AAL1gator-32 also provides protection from buffer overrun and pointer misalignment.

Note that during an underrun, the RALP ‘Robust’ sequence number processing state machine freezes in its current state. For example, if the state machine is in

the SYNC state when underrun occurs, when the next cell arrives, potentially some time later, the state machine will still be in the SYNC state.

11.2.2.2.5 Line Data Storage

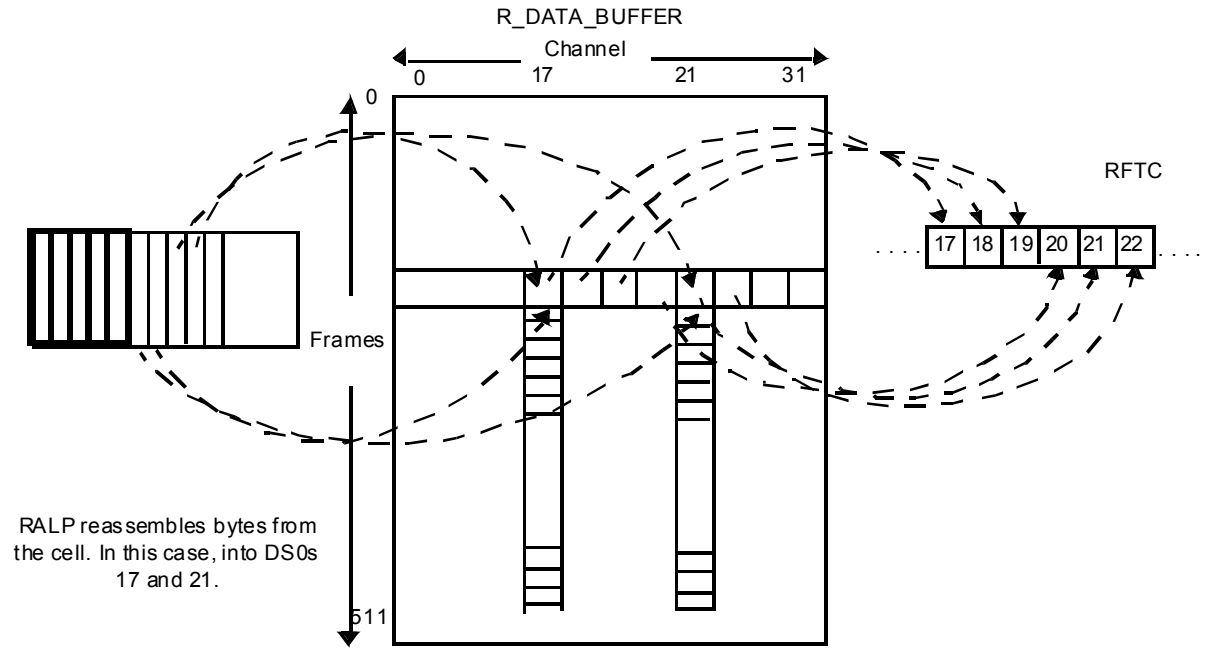
The RALP reads the ATM cell from the RALP_FIFO, verifies the header, and determines the queue to which the cell belongs. It then locates the data bytes of the cell and writes them into frame buffers provided for that line. Receive queues exist in the external memory, and 128 kBytes of external memory are allocated for receive queues for each A1SP. Since there are eight lines per A1SP, each line is allocated 16 kBytes of memory. This provides for 32 multiframes of E1 data (16 multiframes if T1 signaling is used) or 16 multiframes of T1. The queues are used in a wrap-around fashion. Read and write pointers are used at the frame and multiframe level to access the receive queue.

Figure 40 on page 113 shows cell reception. Read and write pointers are used at the frame and multiframe level to access the correct data byte location. The RALP writes sequential cell data bytes into successive DS0 locations assigned to that queue.

When the RALP encounters signaling bytes, it places them in the receive signaling buffer. The buffers are organized in a multiframe format. There is one signaling nibble per multiframe allocated to each DS0 channel. Therefore, 32 bytes of signaling storage are allocated for each multiframe worth of data buffer. The signalling bytes are not used for UDFmodes.

Figure 40 and the figures that follow illustrate these points and identify how different data formats are stored in the data and signaling buffers.

Figure 40 Cell Reception



The RALP determines channels by reading from the R_CHAN_ALLOC table and then storing the data in the corresponding timeslots of successive frame buffers in the R_DATA_BUFFER.

For DBCES mode, the CHNACT table is used instead of the R_CHAN_ALLOC to determine which channels are contained within the structure. Since the DBCES structure only has data for the active channels, only the active channels are written the frame buffers in the R_DATA_BUFFER.

Figure 41 shows the contents of the receive data buffer for ESF-formatted T1 data for lines in the SDF-MF mode. Only the first 24 bytes of each frame buffer and the first 24 frame buffers of every 32 are used. This provides storage for 384 frames or 16 multiframes of T1 data.

Figure 41 T1 ESF SDF-MF Format of the R_DATA_BUFFER

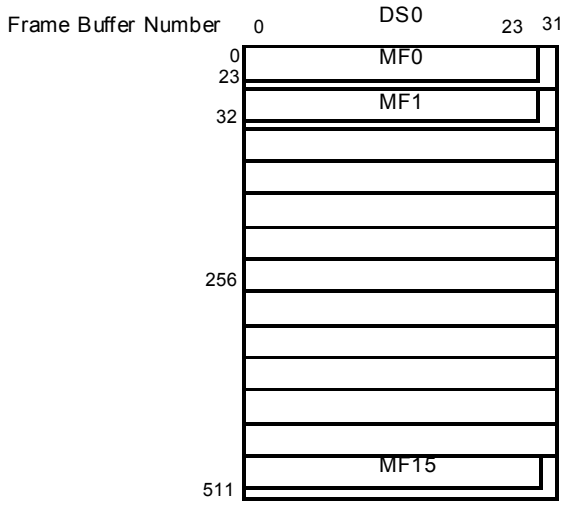


Figure 42 shows the contents of the receive data buffer for SF-formatted T1 data for lines in the SDF-MF mode. Only the first 24 bytes of each frame buffer and the first 24 frame buffers of every 32 are used. This provides storage for 384 frames of T1 data.

Figure 42 T1 SF SDF-MF Format of the R_DATA_BUFFER

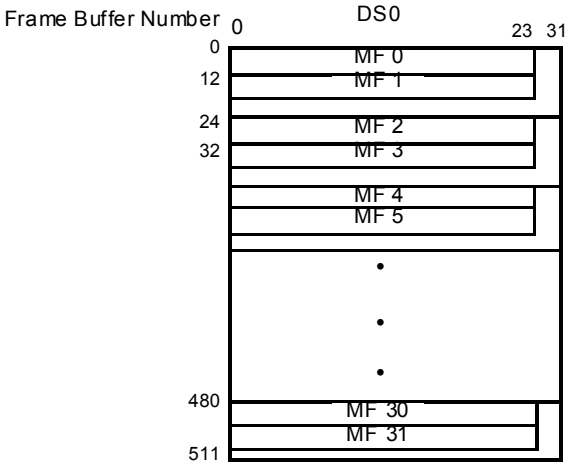


Figure 43 shows the contents of the receive buffer with T1 data for lines in the SDF-FR mode. Only the first 24 bytes of each frame buffer and the first 24 frame buffers of every 32 are used. This provides storage for 384 frames of T1 data.

Figure 43 T1 SDF-FR Format of the R_DATA_BUFFER

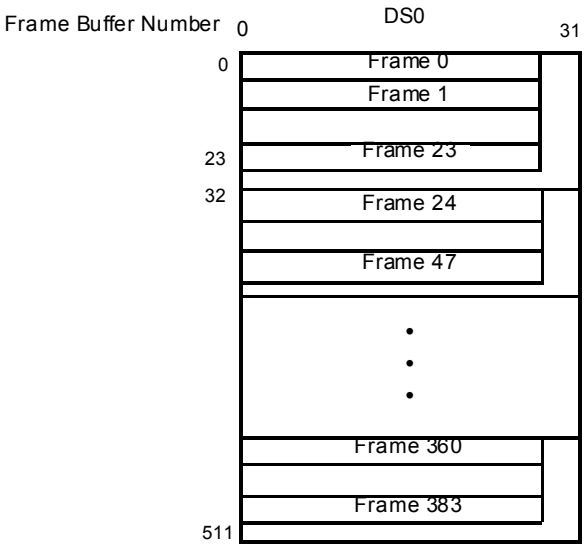


Figure 44 shows the contents of the receive buffer with E1 data for lines in the SDF-MF mode.

Figure 44 E1 SDF-MF Format of the R_DATA_BUFFER

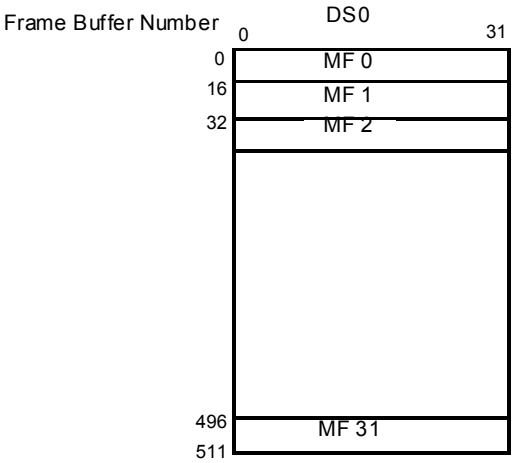


Figure 45 shows the contents of the receive data buffer for E1 SDF-MF data using T1 signaling. In this case a 24 frame multiframe is used.

Figure 45 E1 SDF-MF with T1 Signaling Format of the R_DATA_BUFFER

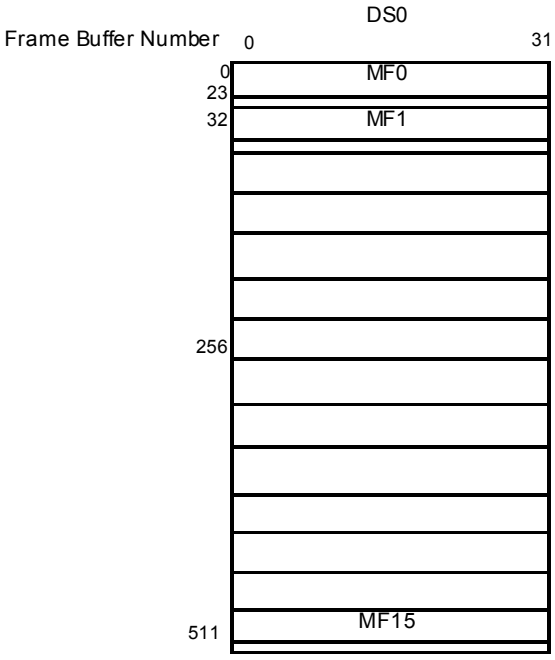


Figure 46 shows the contents of the receive data buffers with E1 data for lines in the SDF-FR mode.

Figure 46 E1 SDF-FR Format of the R_DATA_BUFFER

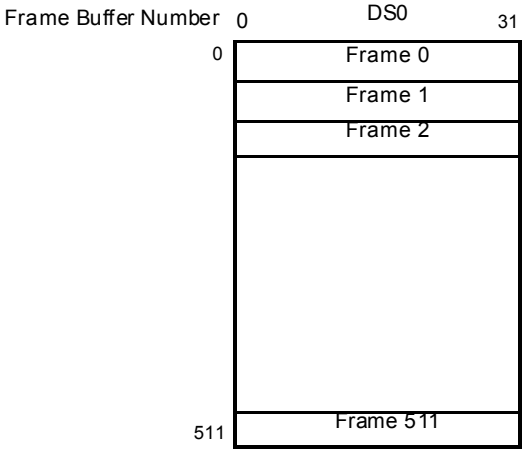


Figure 47 shows the contents of the receive data buffer for lines in the UDF-ML or UDF-HS mode, including T1 unstructured mode. In unstructured modes, each frame buffer contains 256 bits. In the case of unstructured T1, each frame of T1

data consists of 192 bits. Therefore, each frame buffer contains 1.33 frames of T1 data. This must be considered when determining CDVT numbers.

Figure 47 Unstructured Format of the R_DATA_BUFFER

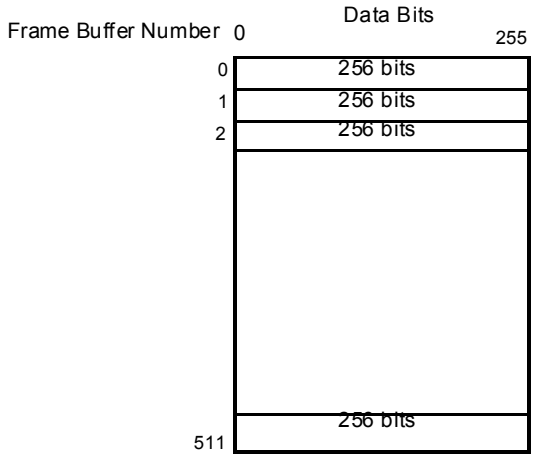


Figure 48 shows the contents of the receive signaling buffer with an ESF-formatted T1 line in the SDF-MF mode. Even channel bytes are stored in the low-byte end of the data words.

Figure 48 T1 ESF SDF-MF Format of the R_SIG_BUFFER

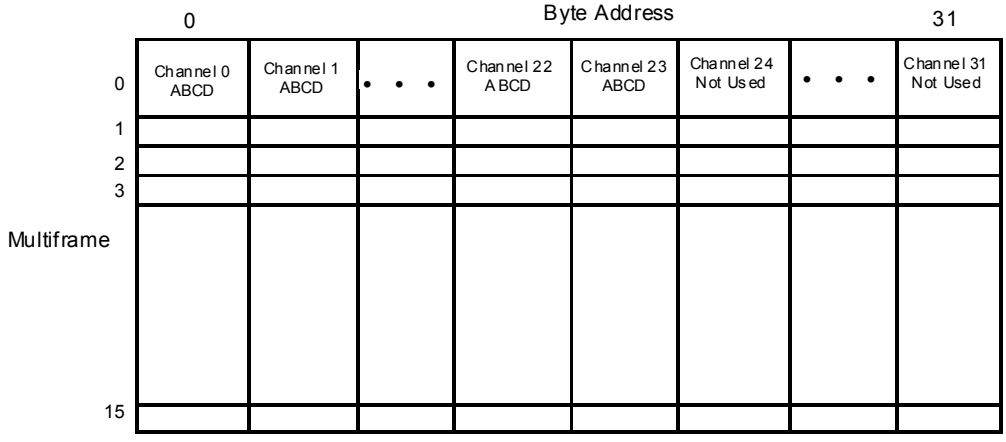


Figure 49 shows the contents of the receive signaling buffer with an SF-formatted T1 line in the SDF-MF mode. Even channel bytes are stored in the low-byte end of the data words.

Figure 49 T1 SF SDF-MF format of the R_SIG_BUFFER

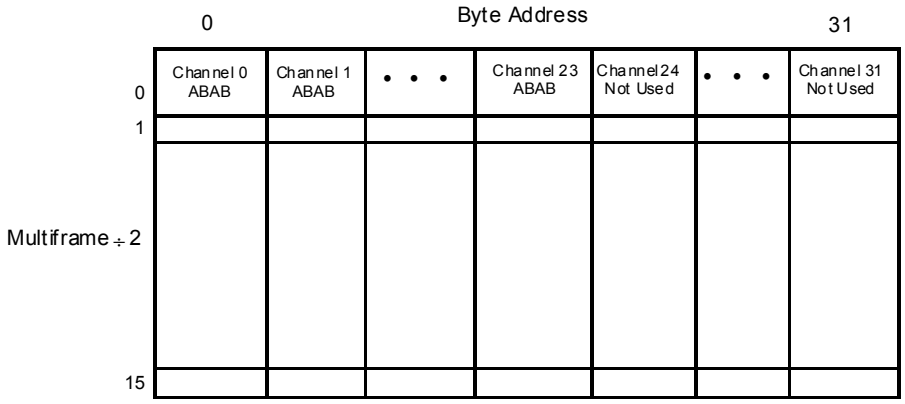


Figure 50 shows the contents of the receive signaling buffer with an E1 line in the SDF-MF mode. Even channel bytes are stored in the low-byte end of the data words.

Figure 50 E1 SDF-MF Format of the R_SIG_BUFFER

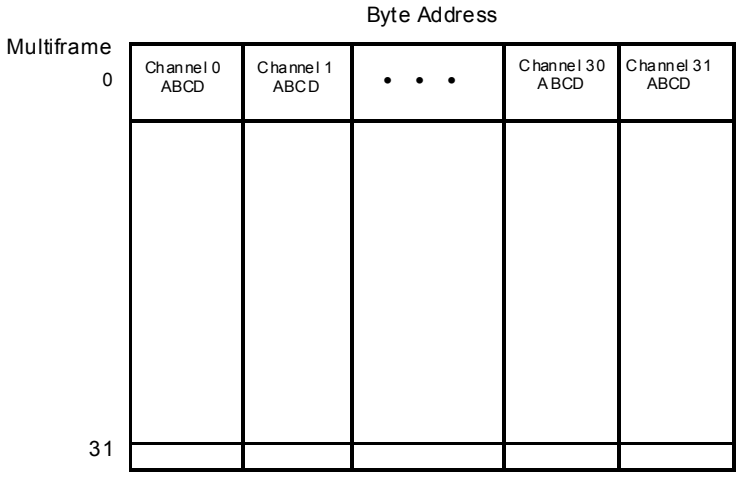
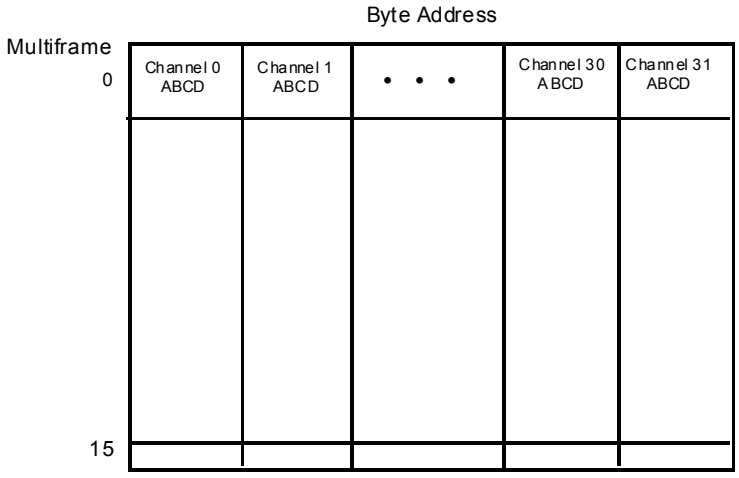


Figure 51 shows the contents of the receive signaling buffer with an E1 line in the SDF-MF mode with T1 signaling. Even channel bytes are stored in the low-byte end of the data words.

Figure 51 E1 SDF-MF with T1 Signaling Format of the R_SIG_BUFFER



11.2.2.2.6 Handling Data and Signaling Bytes in a Structure

A data structure consists of all of the data bytes for that structure, followed by all of the signaling bytes for that structure, if any. In order to locate the data and signaling bytes, the following memory structures are used:

- The R_TOT_SIZE structure provides the number of data and signaling bytes in a structure. An element can be a data byte or a signaling nibble. This value is initialized by the microprocessor.

- The R_TOT_LEFT structure provides the running count of the number of bytes remaining in the structure.

- The R_DATA_LAST structure identifies the last byte of the data structure. As the data bytes are stored in memory, the R_TOT_LEFT structure is decremented by one. When R_TOT_LEFT is equal to R_DATA_LAST, it indicates the end of the data structure. The remaining bytes are stored in the R_SIG_QUEUE. The signaling nibbles are written to the memory until the R_TOT_LEFT equals zero. Once R_TOT_LEFT is zero, R_TOT_SIZE is copied to R_TOT_LEFT and the storing of data and signaling structures is repeated. R_TOT_SIZE and R_DAT_LAST are initialized by the microprocessor.

- The structure used for signaling is determined by the mode of the line and the value of E1_WITH_T1_SIG in the LIN_STR_MODE register. Normally the signaling structure will follow the mode of the line. However if the line is in E1 mode and E1_WITH_T1_SIG is set, then a T1 signaling structure is used. This means that for a single DS0, signaling is updated after 24 data bytes instead of after 16 data bytes.

- If the line is in SDF-MF mode and R_CHAN_NO_SIG = 1 in the R_BUF_CFG word, then the queue is handled as if it is in SDF-FR mode. The structure should be adjusted from a multiframe structure to a frame structure. For example, a channel with two DS0s would have a structure size of two bytes when R_CHAN_NO_SIG is set and a structure size of 49 bytes (include two signaling nibbles) if in T1 mode when R_CHAN_NO_SIG is off.

11.2.2.2.7 Underrun

The AAL1gator-32 declares an underrun condition for a VC when no data is present in the VC receive buffer. When this situation occurs, the AAL1gator-32 plays out conditioned data and frozen signaling onto the timeslots assigned to the VC experiencing underrun. Timeslots generated by other VCs are unaffected. Each time a cell is received after a queue has entered an underrun condition, the UNDERRUN sticky bit is set. RALP does not know about an underrun until a cell is received for the queue that underflowed. To make sure that each underrun is counted only once, RALP will increment the R_UNDERRUN counter when exiting the underrun state and entering the resume state. The initial underrun caused by reset is not counted. Forced underruns due to other errors are not counted by the underrun counter.

Each time a queue enters or exits the Underrun state an entry will be made to the RCV_STAT_FIFO.

If the underrun counter is read and the queue is currently in underrun, the present underrun condition will not be accounted for until the queue exits underrun. To determine if the queue is in underrun, check the value of the R_UNDERRUN bit in the R_LINE_STATE register. When not in UDF-HS mode, the choice of conditioning data while in underrun depends on the value of RX_COND in the R_CH_QUEUE_TBL. Three choices for data exist:

1. Play out the data from R_COND_DATA (Default).

2. Play out pseudorandom data . (For applications that are sensitive to constant data.) The pseudorandom data option uses the data from R_COND_DATA and then replaces the most significant bit with the result of an 18^{th} order polynomial, specifically $x^{18} + x^{7} + 1$.

3. Play out old data . (Also for applications that are sensitive to constant data.) The old data option replays out the contents of the data in the receive buffer for that channel. Data is played out from the location of the read pointer. Therefore, the oldest data is played out first.

While in underrun, the signaling data will be frozen if signaling data was not already conditioned.

If in UDF-HS mode and in underrun, the data played out is the conditioned data defined for line 0 for the A1SP, channel 0 or DS3 AIS. There are no old data or pseudorandom data options available for UDF-HS mode.

Bit integrity may be maintained through an underrun condition if at least one cell is lost and less than 6 cells are lost if the BITI_UNDERRUN is set. In order to detect the amount of lost cells, whenever a cell is received, the value of the line rd_ptr is stored. When a cell is received with a SN error when the queue is in underrun, or cell is received when the SN state machine is in the OUT_OF_SEQUENCE state, the current value of the line rd_ptr is compared against the stored value; if the time that has passed since the last cell was received is less than time it should have taken to play out 6 cells worth of data on the line (stored rd_ptr – current rd_ptr < 7 * FRAMES_PER_CELL), less than 7 cells have been lost and the new incoming cell is processed normally as if an underrun never happened. If the new updated wrt_ptr is greater than the read_ptr, the resume bit is set to indicate to the RFTC that the underrun condition is ending and the end-of underrun is set at wrt_ptr. One problem is that underun can persist for a long time. To detect this, a R_LONG_UNDERRUN bit will be set by the RFTC whenever it detects that the rd_ptr is wrapping (rd_ptr+64= wr_ptr and the queue is already in underrun)

To exit an underrun condition, if the BITI_UNDERRUN is not set or more than 6 cells have been lost, the RALP queues up data for one CDVT worth of time before exiting the underrun condition. The R_UNDERRUN bit in R_LINE_STATE word of R_QUEUE_TBL indicates if the queue is in an underrun state. The UNDERRUN sticky bit is set each time a cell is received during the underrun condition. If the data is structured, the RALP searches for a new pointer, and finds the start of the structure. Cells received while the pointer and start of structure are being located are dropped and the POINTER_SEARCH sticky bit is set. The DROPPED_CELL counter is also incremented. If the underrun condition persists, the microprocessor should set the conditioned bits to derive both the data and the signaling from the conditioned areas. The RALP can optionally maintain MF alignment when exiting underrun, when this is done, the new data is written at the first multiframe boundary after the minimum CDVT buffering requirement. This option may add more delay.

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, ALIGN_MF should be set. If MF_ALIGN_MODE is set to ALIGN_MF 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 insure MF alignment across the ATM network but will add variable delay which could be between 0 and 3 ms(T1)/2 ms(E1).

If enabled, when a queue enters UNDERRUN or exits UNDERRUN an entry is made into the RCVN_STAT_FIFO indicating the QUEUE number and setting either the ENTER_UNDERRUN bit or the EXIT_UNDERRUN bit. These conditions can be disabled from causing an entry in the RCVN_STAT_FIFO. When an entry is made in the RCVN_STAT_FIFO, this event may cause the INTB line to go low if the A1SP interrupt is enabled and the RSTAT_FIFO_EMPB_EN is set and A1SPn_INTR_EN is setB.

Refer to line items 5 and 6 in Figure 53 on page 126 that show how a start up after an underrun occurs.

11.2.2.2.8 Pointer Processing

When an incoming cell has a cell pointer, the cell pointer is checked against the local pointer value maintained by the RALP. A single pointer mismatch causes no corrective actions. The pointer is ignored and the cell is used as if it contained valid data. If two consecutive pointer mismatches occur, then the RALP forces an underrun condition that causes the receiver to realign to the next incoming pointer. The proper R_COND_DATA and frozen signaling are played out until the new pointer is found and one CDVT worth of time has passed. The PTR_MISMATCH sticky bit is set when the pointer mismatch occurs. The FORCED_UNDERRUN sticky bit is set each time a cell is received and dropped during the forced underrun condition. Then, the underrun and pointer search bits are set, as described previously in section 11.2.2.2.7 “Underrun” on page 120.

Only one pointer is supposed to be received within a sequence of 8 cells (SN 0 – SN 7). If more than one pointer is received or if no pointers are received in an otherwise error free sequence of 8 cells, then the PTR_RULE_ERROR sticky bit will be set. Pointers are also supposed to be less than or equal to 93 or equal to the dummy value 127. If a pointer is received that does not fall within the legal range and no other SN error occurred the PTR_RULE_ERROR sticky bit will be set.

If the received cell is potentially bad as determined by SN processing, but should contain a pointer, the pointer is not checked against the local pointer. However, in this situation, and when cells are inserted, if the next received pointer

mismatches, then a PTR_MISMATCH error is reported and a forced underrun occurs. By not waiting for the second pointer mismatch in these situations where bit integrity may have been lost, a quicker detection and recovery will result.

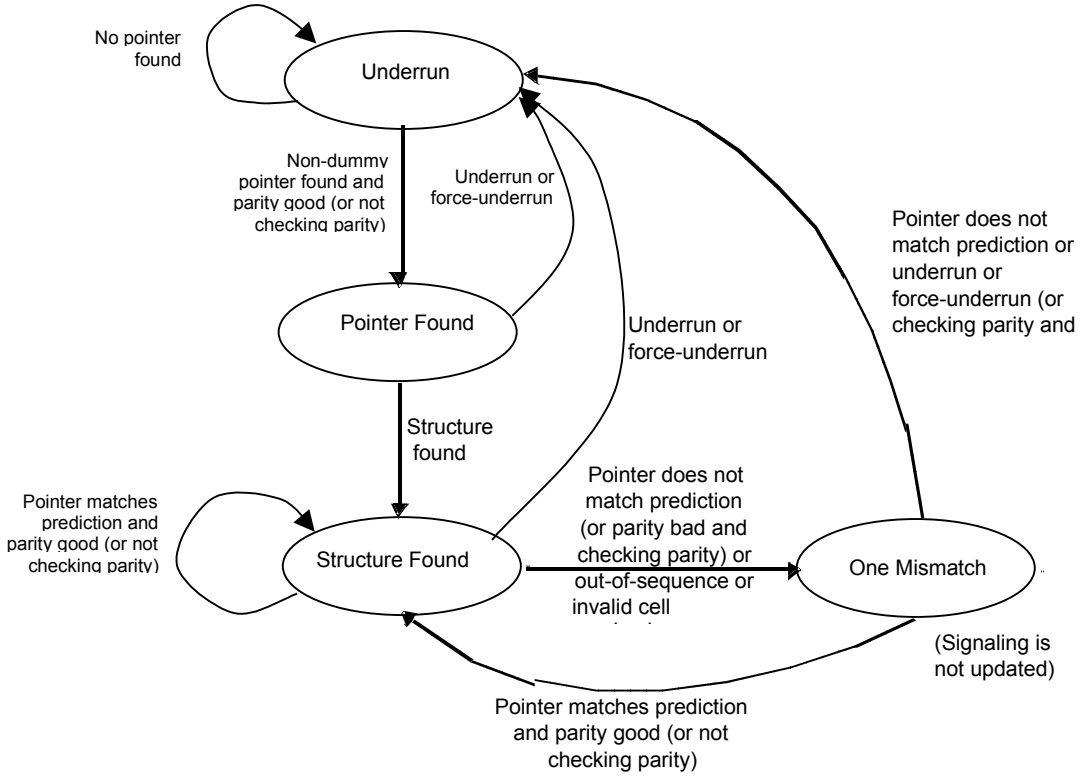
When a PTR_MISMATCH error occurs, the R_POINTER_REFRAMES counter is incremented. Also, when a single pointer mismatch occurs, signaling information will not be updated until a valid pointer is received to prevent corruption of the signaling information.

Parity checking is also performed on pointers if R_CHK_PARITY is set in R_MP_CONFIG. If a parity error is detected the PTR_PARITY_ERR sticky bit will be set and the R_PTR_PAR_ERR counter will increment. If two consecutive pointer parity errors occur, then the RALP forces an underrun condition and resynchronizes. This resynchronization will cause R_POINTER_REFRAMES to increment.

When any pointer error is detected an entry will be made into the RCVN_STAT_FIFO, unless the corresponding enable bit is not set or this is not the first enabled sticky bit to be set for this queue since the sticky bit register was last cleared.

Figure 52 shows the state machine that checks for valid pointers and structures.

Figure 52 Pointer/Structure State Machine



11.2.2.2.9 Overrun

Overrun occurs when the data in the buffer is removed at a slower rate than it is filled. However, because the AAL1gator-32 buffers are quite large, 16 Kbytes per line, by the time this happens, all data in the buffer can be quite old. Therefore, the buffer size is adjustable, which regulates how much data can be stored in the buffer before an overrun occurs. The R_MAX_BUF field in the R_BUF_CFG register controls the maximum size of the receive buffer. The value of R_MAX_BUF should be equal to or greater than two times CDVT, or CDVT plus two times the number of frames required per cell, whichever is greater. If MF_ALIGN is set, then extra margin should be added for the additional multi-frame of data that may be present. Therefore the value should be increased by 16 frames for E1 or 24 frames for T1.

The amount of data that is buffered during DBCES mode is different than the amount of data that is buffered in normal mode. In DBCES mode there must be enough data to handle the case where the structure changes from the maximum number of channels being active to only one channel being active without going

into underrun. In order to prevent the underrun, (47 - frames_per_cell) frames are buffered before data is sent out. This buffering when combined with the frames_per_cell + 1 that is buffered on the transmit side results in a total of 48 frames of buffering. The result is every DBCES connection has the same amount of delay in starting up regardless of the number of active DS0's. This delay is (48 + CDVT) * 125 us. The calculation to determine how much extra buffering is needed in DBCES mode is calculated by RALP every time the queue exits underrun or exits from an all idle state. This buffer adjustment guarantees that the delay for a given channel does not change regardless of how many other channels are active or inactive. This is due to the fact the total buffering between the transmit side and receive side is always equal to 48 frames. If there is only one channel active, 48 frames will be buffered on the transmit side and 0 frames (excluding CDVT) on the receive side. If 31 channels are active, then 3 frames will be buffered on the transmit side and 45 frames will be buffered.

The overrun condition is declared when the data in the receive buffer exceeds the maximum specified buffer. When a cell is received that causes the maximum buffer depth to be exceeded, the OVERRUN sticky bit is set and the AAL1gator-32 enters the forced underrun condition. The incoming cells for the queue are dropped until underrun occurs. Each time a cell is received and dropped in the forced underrun condition, the FORCED_UNDERRUN sticky bit is set. Once underrun occurs, the overrun flag is cleared and the same algorithm used in underrun is followed. Figure 53 on page 126 describes overrun detection, the underrun and recovery process.

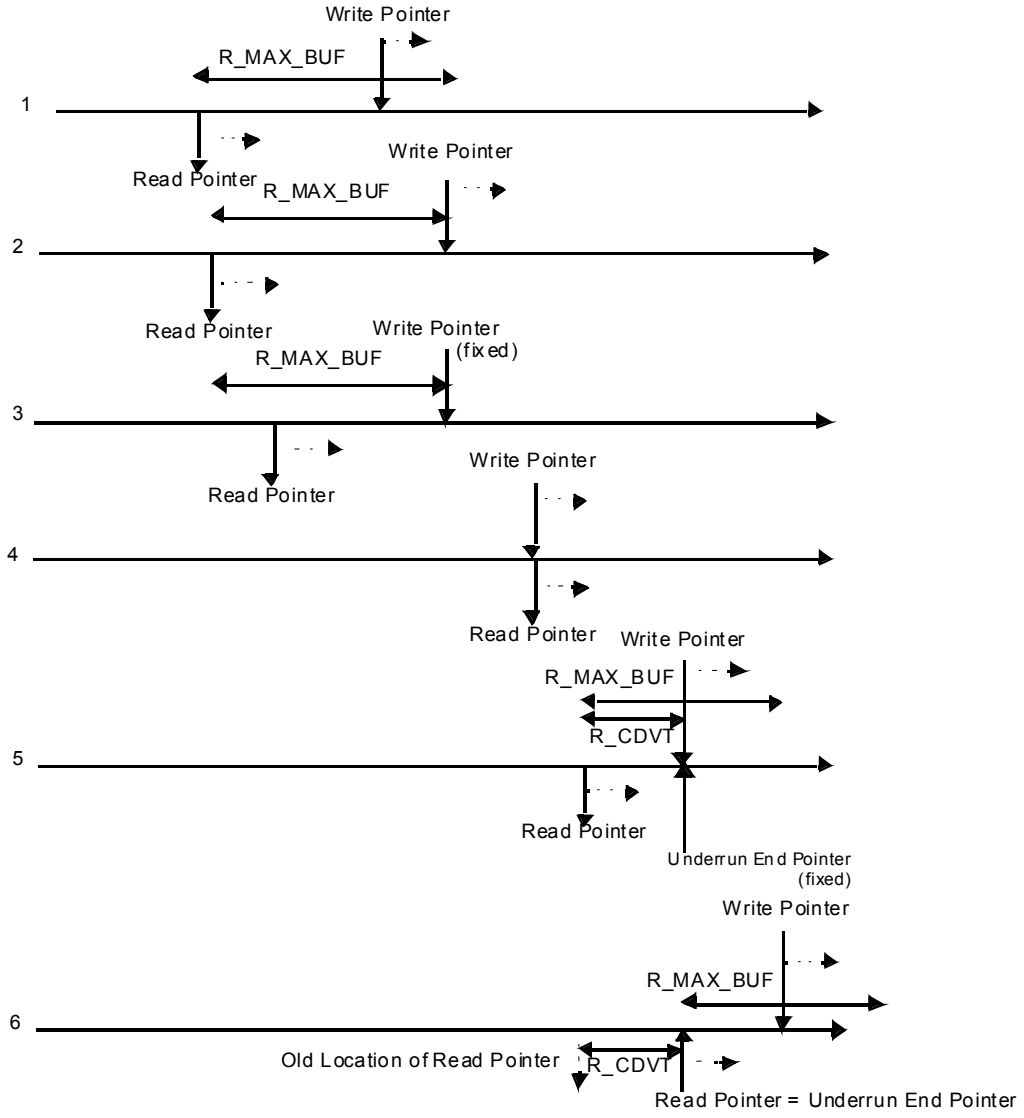
Overruns can also occur due to lost/misinserted cells when robust SN processing is done if the buffer is set too small. This is because when the SN processing detects a potentially lost cell event, the cell will be written into the buffer at the correct position assuming that cells have been lost. When Robust SN processing is enabled, the R_MAX_BUF should be equal to or greater than two times CDVT or CDVT plus 9 times the number of frames required per cell to allow for the processing performed on lost/misinserted cells.

Anytime an overflow occurs, the R_OVERRUNS counter is incremented. When an overrun is detected an entry will be made into the RCVN_STAT_FIFO, unless the corresponding enable bit is not set or this is not the first enabled sticky bit to be set for this queue since the sticky bit register was last cleared.

Note:

- Inserting cells can cause an overrun. The threshold is checked as each byte is written into memory. If an overflow occurs in the middle of a cell, the remainder of the cell will be dropped.

Figure 53 Overrun Detection



NOTES:
1. Normal operation.
2. If an overrun occurs, then the OVERRUN sticky bit is set and a forced underrun condition is set.
3. During the forced underrun condition, the write pointer is fixed, new data is dropped, and data in the buffer is played out, causing the read pointer to increment. Each time a cell is received and dropped, the FORCED_UNDERRUN sticky bit is set.
4. The read pointer catches up to the write pointer, indicating a forced underrun condition, and the underrun condition is set. Both pointers are advanced for each frame that the queue remains in the underrun condition. Conditioned underrun data is played out.
5. When the first valid cell comes in, the RESUME sticky bit is set. The write pointer and the underrun end pointer are set to the proper frame that is one R_CDVT ahead of the read pointer. Conditioned underrun data is played out.
6. Once the read pointer is equal to the underrun end pointer, then RESUME is complete and real data is now played out. Normal operation now takes place.

11.2.2.2.10 DBCES

Dynamic Bandwidth Circuit Emulation Service can be enabled to support the dynamic adding and dropping of individual DS0s within a VC. DBCES is not supported in conjunction with partially filled cells. When DBCES is enabled, the RALP processes the incoming bit masks and adjusts the saved structure sizes, and active connections.

When a channel changes its active state, an entry will be made into the RCVN_STAT_FIFO., if the corresponding enable bit is set. At this point the DBCES RX CHANNEL ACTIVE table, the DBCES_PENDING table and the DBCES_PTR_TABLE will also be updated.

The DBCES RX CHANNEL ACTIVE table provides the active status for all of the receive channels.

Using the bit mask, the number of active DS0s is calculated and the R_TOT_SIZE, LAST_CHAN, R_DATA_LAST; R_CHAN_ACTIVE; FRAMES_PER_CELL fields in the R_QUEUE_TABLE are updated.

If a queue implementing DBCES with active channels enters the underrun condition, the normal underrun processing will occur. As individual time slots go inactive, data will be played out as if the timeslot is in underrun.

If a DBCES queue has no active timeslots (all timeslots are inactive), the underrun status from the RFTC is ignored and not counted as an underrun to prevent a pointer search. When the queue transitions from all inactive to some active channels, the write pointer is calculated as if resuming from an underrun.

If a sequence number error is detected and cells are inserted, the normal cell insertion procedures apply except if the inserted cell should contain a bit mask. In this case, the number of inserted bytes is adjusted for the bit mask.

If the bit mask is errored (parity error) the bit mask processing will be bypassed, bit mask error sticky bit will be set and no changes made to the activity states of the channels. If the errored bit mask should have contained a change in the bit mask, all of the channels in that DBCES connection may be incorrectly mapped until the next bit mask is received.

11.2.2.2.10.1 DBCES Receive Side Buffering

The amount of data that is buffered during DBCES mode is different than the amount of data that is buffered in normal mode. In DBCES mode there must be enough data to handle the case where the structure changes from the maximum number of channels being active to only one channel being active without going

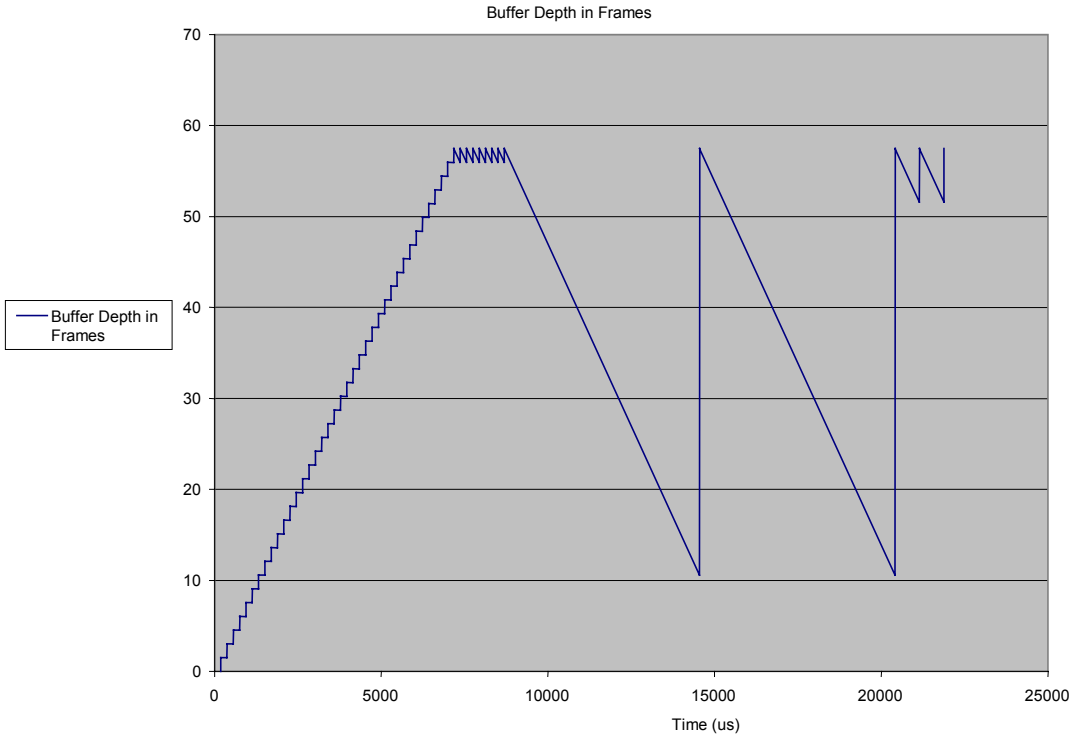
into underrun. In order to prevent the underrun, an additional (47 - frames_per_cell) frames are buffered before data is sent out. This buffering when combined with the frames_per_cell + 1 that is buffered on the transmit side results in a total of 48 frames of buffering. The result is every DBCES connection has the same amount of delay regardless of the number of active DS0's. This delay is (48 + CDVT) * 125 us. The calculation to determine how much extra buffering is needed in DBCES mode is calculated by RALP every time the queue exits underrun or exits from an all idle state. This buffer adjustment guarantees that the delay for a given channel does not change regardless of how many other channels are active or inactive. This is due to the fact the total buffering between the transmit side and receive side is always equal to 48 frames. If there is only one channel active, 48 frames will be buffered on the transmit side and 0 frames (excluding CDVT) on the receive side. If 31 channels are active, then 3 frames will be buffered on the transmit side and 45 frames will be buffered on the receive side.

Figure 54 below shows the DBCES receive side buffering as a function of time. The y-axis represents the buffer depth in frames and the x-axis is time in microseconds. The CDVT in this example is set for 1 ms (8 frames) and initially 31 channels are active. The first 6600 us are spent on the DBCES buffering and CDVT buffering (45 frames + 8 frames = 53 frames).

Once the buffering is complete, data playout begins. In Figure 54 below playout for 31 channels is represented by the first saw tooth wave. On average a cell is received every 188 us and it takes 188 us to play a cell's worth of data onto the line. The second saw tooth wave represents one active channel. The wave begins at 8700 us and a cell is received on average every 5858 us. The third saw tooth wave represents 8 active channels. The wave begins at 20413 us and on average a cell is received every 731 us.

Note that if the buffer adjustment was not made, the queue would have underun, when all channels but one went inactive.

Figure 54 DBCES Receive Side Buffering



11.2.2.2.11 Counters and Sticky Bits

The RALP sets sticky bits for overrun, underrun, pointer mismatch, resume, SRTS underrun, SRTS resume, and other conditions. As with most registers, the sticky bits are located in the external RAM. They are set by the AAL1gator-32 and must be cleared by the microprocessor. Sticky bits provide a history of events that have occurred. Since these bits can be set with every cell, it is better to use the counters for statistics gathering purposes. The AAL1gator-32 increments the counters for incorrect SNs, incorrect SNPs, cells received, underflows, overflows, dropped cells, misinserted cells, lost cells, pointer parity errors, and pointer mismatches. The transfer status bits can be used to detect that a clear was overwritten by toggling transfer bit 15 with each clear of the status bits and then reading the value immediately thereafter. Refer to “R_ERROR_STKY Word Format” for a description of sticky bits.

The first time that a sticky bit is set for a queue that has the corresponding enable bit set in the RCV_STAT_EN_REG register, an entry will be made in the RCV_STAT_FIFO containing the queue number and the sticky bit which was just

set. No new entries for sticky bits will be made into the RCVN_STAT_FIFO for this queue until the sticky bit register is cleared.

11.2.2.2.12 OAM Cells

When an OAM cell arrives, the RALP stores it in the OAM queue. The RALP notifies the microprocessor of the arrival of OAM cells by setting the OAM_INTR bit in the A1SPn_INTR_REG. The microprocessor reads the OAM cells from the OAM queue. The microprocessor maintains the OAM_HEAD value. The RALP maintains the OAM_TAIL value. The AAL1gator-32 also checks the CRC-10 of the cell and records the results in the receive buffer in the CRC_10_PASS parameter.

11.2.2.2.13 OAM Cell Interrupt Handling

The AAL1gator-32 has the capability to generate an interrupt on the receipt of an OAM cell.

At the end of an OAM cell, the RALP sets the OAM_INTR bit in the A1SPn_INTR_REG. If the OAM_INTR bit is enabled, the A1SPn interrupt bit will be set in the MSTR_INTR_REG which will generate an interrupt if enabled.

11.2.2.3 Receive Frame Transfer Controller (RFTC)

The RFTC moves data bytes from the receive frame buffer to the appropriate timeslot of the appropriate line. It must perform a timeslot-to-queue translation for each timeslot by reading the receive channel-to-queue table. The RFTC outputs data to the Line Interface Block. For structured data, the RFTC uses the frame pulses generated by the Line Interface Block or internally generated frame pulses to perform a parallel-to-serial conversion on the outgoing data that it reads from a multiframe buffer in the order in which it is needed

A rising edge on the appropriate frame pulse indicates the beginning of a frame or multiframe. The RFTC realigns when a rising edge is seen on these signals. It is not necessary to provide an edge every frame or multiframe. Signaling data is driven for all frames of any multiframe and will change only on multiframe boundaries. Signaling will not change when the queue is in underrun, or if a DBCES channel is not active. This is called frozen signaling. The RFTC also has the option of playing out conditioned signaling.

For T1 mode, signaling data may change every 24th frame. For E1 mode, signaling may change every 16th frame. The RFTC accommodates the T1 Super Frame (SF) mode by treating it like the Extended Super Frame mode(ESF)

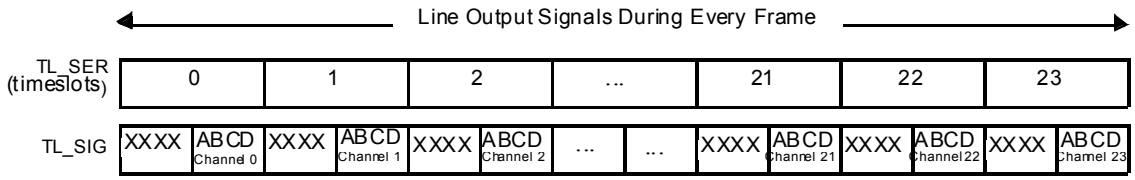
format. The RFTC generally ignores the multiframe pulses that occur on the 12th frame in the multiframe.

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

If GEN_SYNC is set in the LIN_STR_MODE memory register for this line then the RFTC will generate the frame pulses based on its internal counter.

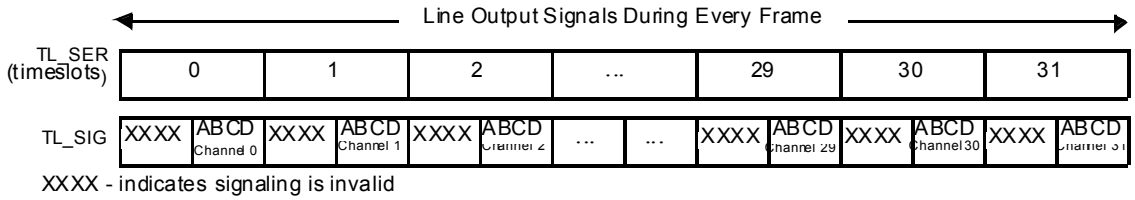
The signaling nibble is valid for each channel when the last nibble of each channel's data is being driven unless the SHIFT_CAS bit is set in the LIN_STR_MODE register. If the SHIFT_CAS bit is set the signaling nibble is valid for each channel when the first channel of each channel's data is being driven. See Figure 55 for an example of signaling bits in a T1 frame. See Figure 56 for an example of signaling bits in the E1 mode.

Figure 55 Output of T1 Signaling Bits (SHIFT_CAS=0)



XXXX - indicates signaling is invalid

Figure 56 Output of E1 Signaling Bits (SHIFT_CAS=0)



XXXX - indicates signaling is invalid

Note:

- The AAL1gator-32 treats all 32 timeslots identically. Although E1 data streams contain 30 timeslots of channel data and 2 timeslots of control (timeslots 0

and 16), data and signaling for all 32 timeslots are stored in memory and can be sent and received in cells.

Each line request is serviced by the main RFTC state machine using a priority encoder (line 0 has the highest priority). The line requests two bytes at a time. This means two channels have to be serviced by the RFTC state machine. When the RFTC state machine receives an attention request from a line, it services the request according to the following pseudocode (please note that the bits discussed are maintained in R_LINE_STATE in the queue tables and should not be confused with the externally accessible sticky bits):

- If the RX_COND_H (or RX_COND_L) = “10”, play out the data from the R_COND_DATA area and signaling from the R_COND_SIG area. This is the conditioned state.

- Else if either the R_UNDERRUN bit or the R_RESUME bit is set in R_LINE_STATE, then play out the data and signaling as determined by the value of RX_COND_H (or RX_COND_L). This is the frozen signaling state. The data played out can either be constant, pseudorandom, or old data.

- Else if DBCS_EN=’1’ and the channel is inactive; then play out the data and signaling as determined by the value of RX_COND_ (or RX_COND_L) as if in underrun.

- Else play out the data from the R_DATA_BUF and the signaling from the R_SIG_BUFFER. This is the normal operating state.

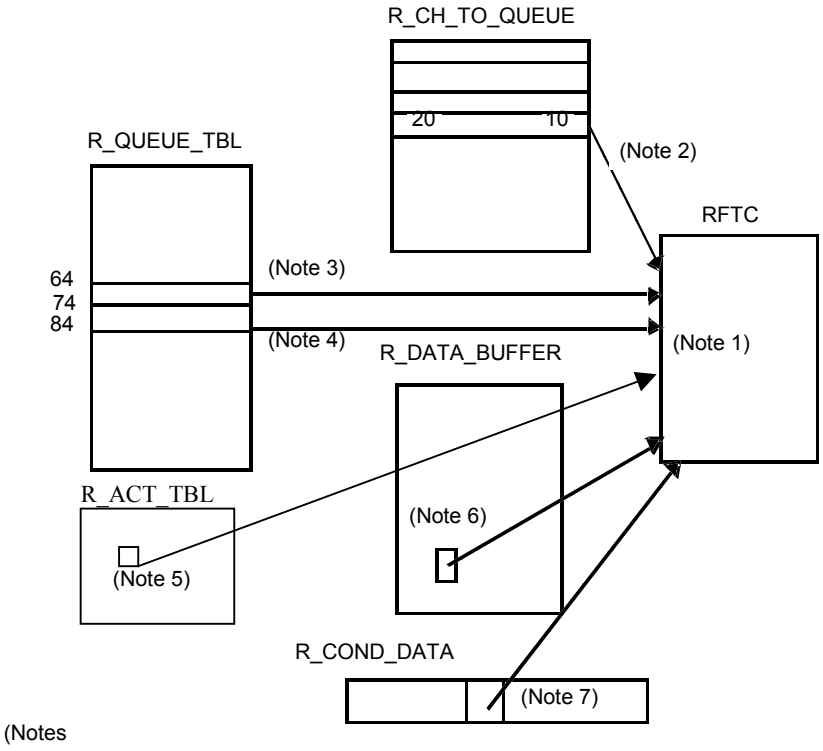
- In any of the above states, the conditioned signaling may be forces by setting the RX_SIG_CONDH(or RX_SIG_COND_L)

- When there is no data in the frame buffer, the RFTC sets an underrun bit, R_UNDERRUN. The RALP clears R_UNDERRUN and sets R_RESUME when it encounters the first valid cell in the receive buffer.

- If the R_RESUME bit is set and the R_RD_FR_PTR = R_END_UNDERRUN_PTR, then RFTC clears the R_RESUME bit. This occurs one CDVT time after the first valid cell arrives. If the line is in SDF_MF mode, then R_SIG_RESUME is set to indicate that signaling is not yet available. Once a multiframe has completed and signaling data is available, R_SIG_RESUME will be cleared.

Figure 57 shows the channel-to-queue table operation

Figure 57 Channel-to-Queue Table Operation



(Notes

1. RFTC fetches R_CH_TO_QUEUE entry for DS0s 6 and 7 for line 2.
2. DS0 6 is being serviced by Queue MOD 32 = 10
   Queue = line 2 x 32 + 10 = 74
   DS0 7 is being serviced by Queue MOD 32 = 20
   Queue = line 2 x 32 + 20 = 84
3. RFTC fetches the queue status and write pointer for queue 74. The queue is not in underrun.
4. RFTC fetches the queue status and write pointer for queue 84. The queue is in underrun.
5. RFTC fetches the active status for DS0 6 from R_ACT_TBL if DBCES is enabled
6. RFTC fetches the data for DS0 6 from the R_DATA_BUFFER.
7. RFTC fetches the data for DS0 7 from R_COND_DATA.

11.2.2.3.1 SRTS Support

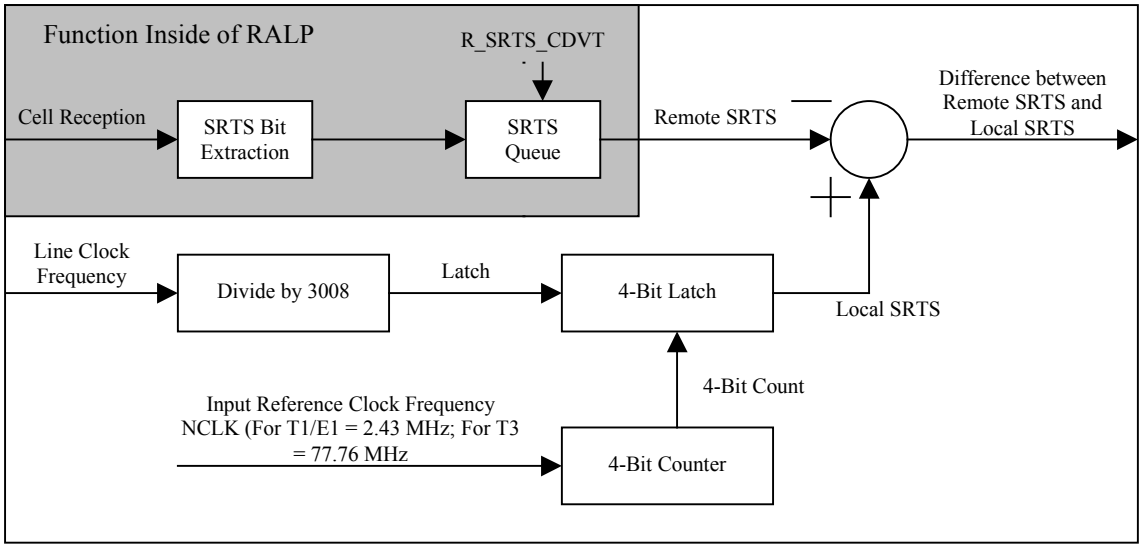
Figure 58 shows the process implemented for each UDF line enabled for SRTS. It queues the incoming SRTS values, and, when requested, passes this SRTS value to the AAL1 Clock Generation Control block. This value, along with a locally calculated SRTS value, is used by the AAL1 Clock Generation Control block to determine the appropriate line frequency to synthesize.

The RFTC supports SRTS only for unstructured data formats on a per-line basis. The SRTS_CDVT value in R_SRTS_CONFIG register must be configured

correctly so the time delay value of the SRTS data matches the time delay value of the signal data. The RFTC queues the SRTS nibbles and then fetches when requested by the AAL1 Clock Generation Control block. If the SRTS queue overruns or underruns, the value is indicated as invalid.

Note SRTS can be used with the internal clock synthesizers for E1 and T1 mode. If other frequencies are used including DS3 and E3, an external clock synthesizer needs to be used.

Figure 58 Receive Side SRTS Support



11.3 AAL1 Clock Generation Control

11.3.1 Description

The CGC block is responsible for generating the A1SP transmit line clocks. A given line clock is synthesized internally using a 38.88 MHz system clock (SYS_CLK). Only E1 or T1 clocks can be generated internally. Any other frequency clock must be generated externally and passed into the AAL1gator-32 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. Alternatively a nominal E1 or T1 frequency clock can be generated. 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. The CGC block also outputs status information for channel pairs through the external interface.

The CGC consists of four major blocks: External Interface, SRTS, Adaptive, and Frequency Synthesizer. The External Interface passes SRTS and adaptive information out to the user and allows the user to control the Frequency Synthesizer. The SRTS block receives SRTS values and uses the values to determine the frequency to be synthesized. The Adaptive block receives buffer depth information and uses this information to determine the frequency to be synthesized. 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 transmit line clock source is controlled by the CLK_SOURCE_TX field and the T1_MODE bit in the LIN_STR_MODE memory register. The eight possible options are:

000 Use external clock. (TL_CLK is an input).

001 LOOPED – Use RL_CLK as the clock source.

010 NOMINAL Synthesized – Generate a clock of the nominal (T1 or E1) frequency from SYS_CLK.

011 SRTS Synthesized- Generate a clock frequency based on the received SRTS values.

100 ADAPTIVE Synthesized- uses receive buffer depth to control TL_CLK

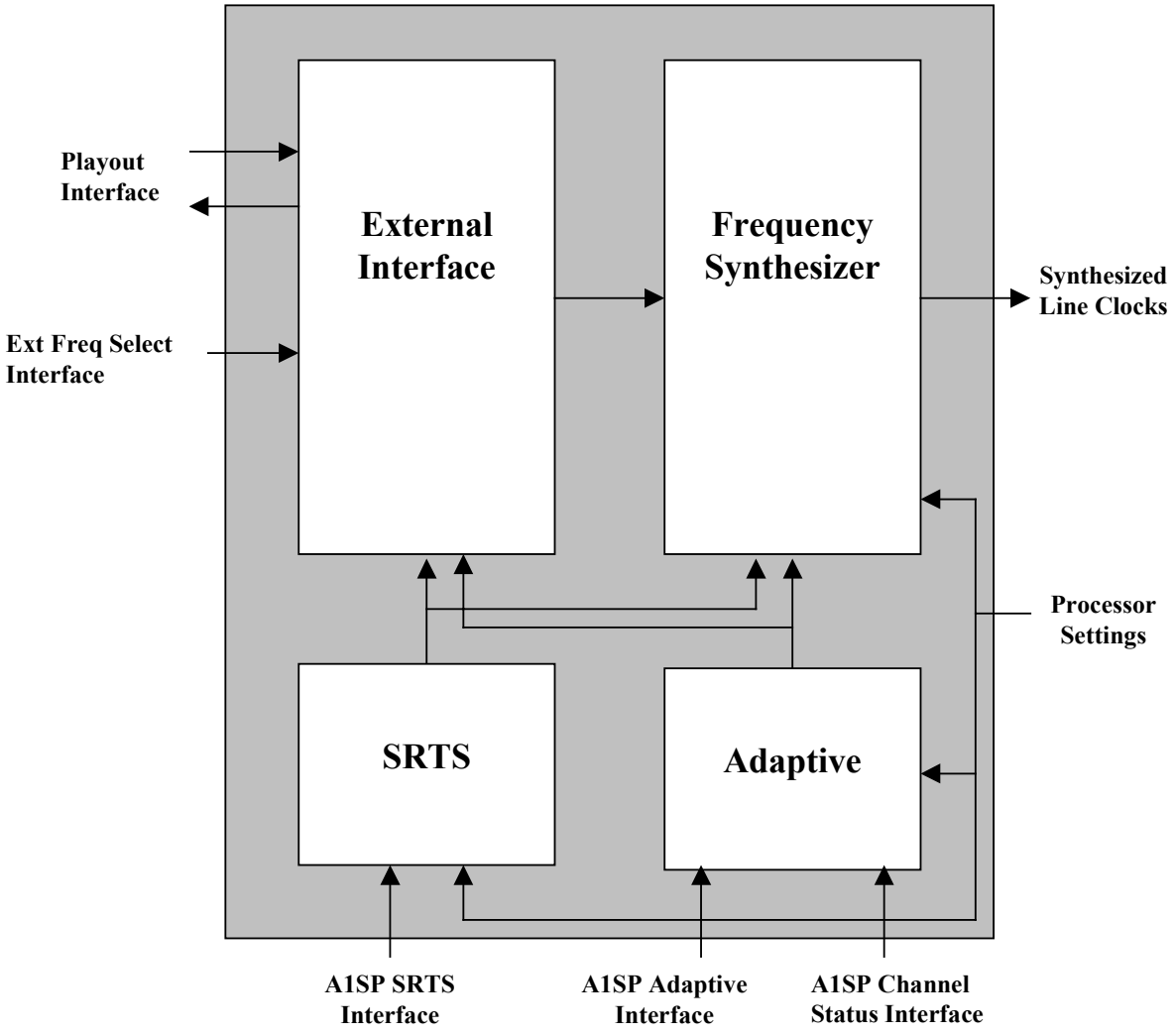
101 Externally controlled Synthesized: Generate a clock frequency based on the values provided by CGC_SER_D pin. This mode is used for external implementations of SRTS or Adaptive clocking.

110 Use common external clock (CTL_CLK) (Not valid in H-MVIP or SBI mode)

111 If in Direct Low Speed Mode, use common external clock (CTL_CLK) and drive TL_SIG data onto TL_CLK pin. If in SBI mode then use clock sourced by SBI. Not valid in other modes.

Options “010” through “101” involve the CGC block. Each line is independently controlled. When a particular line is not in one of these modes it is held in reset.

11.3.2 CGC BLOCK DIAGRAM



11.3.3 FUNCTIONAL DESCRIPTION

The CGC consists of four major blocks: External Interface, SRTS, Adaptive, and Frequency Synthesizer.

11.3.3.1 External Interface

The External Interface transmits and receives line clock related information that allows the line clocks to be controlled externally. It serves two functions:

1. external playout of Adaptive, SRTS, and Channel Status data

2. receipt of external frequency select data.

The External Interface block transmits either SRTS, Adaptive, or Channel Status information on a line basis which enables an external decision to made about altering the line clock frequency. The SRTS output data is the difference between the local and remote SRTS nibble. The adaptive output data is the averaged (using the adaptive algorithm described in section 11.3.3.7) relative buffer depth in units of bytes. If the adaptive weighting is set to 0 this value becomes the raw buffer depth in units of bytes.

Status information is played out on the External Clock Generation Control (CGC) Interface which includes the external output signals: CGC_DOUT[3:0], CGC_LINE[4:0], SRTS_STBH, and ADAP_STBH.

Information is played out for all chip modes (Direct Low Speed, SBI, H-MVIP and High Speed).

The interface clocking operates according to the following pseudocode:

- If a channel pair is being serviced, start a state machine to play out the channel status, as shown in Table 6, asserting ADAP_STBH as each state is played out.

- Else, if an SRTS difference value is ready, it is played out with SRTS_STBH asserted and ADAP_STBH deasserted.

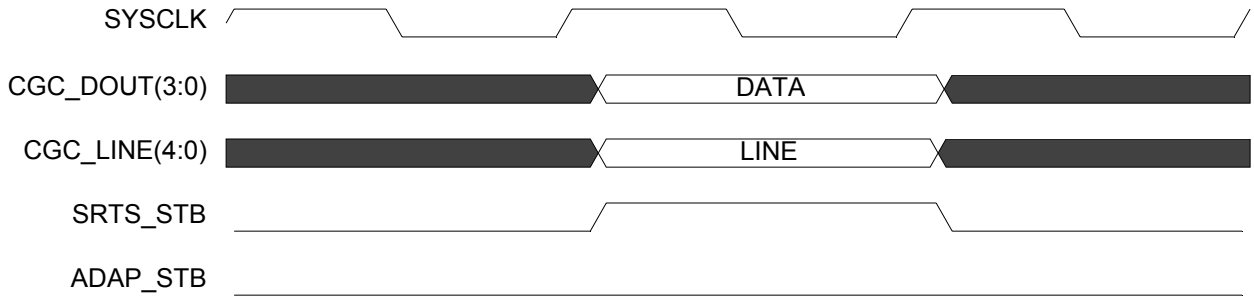
- Else if a cell is received and a valid averaged relative buffer depth can be computed, start a state machine to play out the averaged relative buffer depth and queue number, as shown in Table 7, asserting ADAP_STBH as each state is played out.

- Once playout of a certain data type has begun it will not be interrupted.

11.3.3.2 SRTS Data Output

In SRTS mode the CGC block puts out a 4 bit value which represents the difference between the local SRTS value and the received SRTS value. Figure 59 below shows a typical CGC output in SRTS mode.

Figure 59 SRTS Data



11.3.3.3 Channel Underrun Status Output

Figure 60 shows an example of the channel underrun status being reported on the CGC Interface. In this example for a structured line, the line number is 19 (A1SP = 2, local line number = 3) the channel pair is 7 (channels 14 and 15), and the high channel (channel 15) is in underrun while the low channel (channel 14) is in normal mode. For an unstructured line, there is no per-channel status, only per-line status; Channel(4:1) can be ignored, and both Underrun_h and Underrun_l will have the same value – either can be used. These values are encoded into the 4 data words following the format shown in Table 6.

Status is only reported when a channel enters or exits underrun.

Figure 60 Channel Status Functional Timing

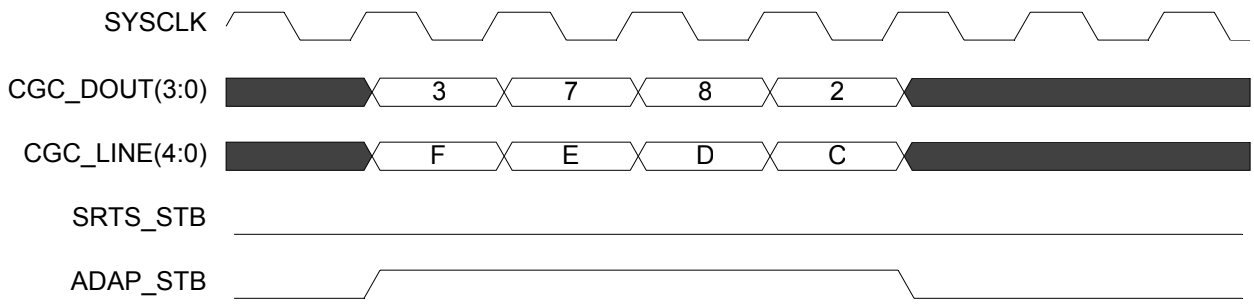


Table 6 Channel Status

CGC_LINE(4:0) Value	CGC_DOUT(3:0) Value			
	3	2	1	0
15	0	Line(3)	Line(2)	Line(1)
14	Channel(4)	Channel(3)	Channel(2)	Channel(1)
13	Underrun_h	0	underrun_l	0
12	0	0	A1SP(1)	A1SP(0)
	0	0	0	0

11.3.3.4 Adaptive Status Output

The AAL1gator-32 provides the average buffer depth in units of bytes for an external circuit to generate an adaptive clock. (If adap_filt_size is set to zero it will provide the current buffer depth with no averaging). The general mechanism is often termed “buffer centering”. A clock delta value is determined externally by subtracting the nominal buffer depth (value of CDVT) from the actual buffer depth. This delta value is then transformed into the frequency selection for an external frequency synthesizer. The closed-loop action of this circuit causes the delta value to find a center point. When the delta is above the center point, there is too much data buffered and the frequency must be increased. When the delta is below the center point, the frequency must be reduced.

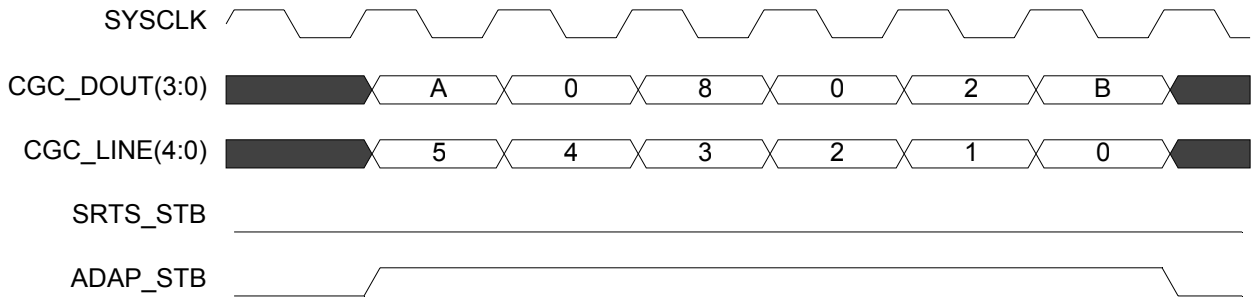
The AAL1gator-32 implements a programmable weighted moving average internally. However, if an alternative adaptive algorithm is desired then this information can be processed externally. In High Speed mode, since an E3 or DS3 clock cannot be internally synthesized, this information can be used for external synthesis of an E3 or DS3 clock.

Table 7 below shows an example of the CGC Interface for adaptive data. In this example the line number is 21, and the average buffer depth in units of bytes is 43. (For unstructured lines, the average buffer depth is given in units of bytes. For structured lines, the average buffer depth is given in units of frames, ie, buff_depth(4:0) should be ignored.) These values are encoded into the 6 data words following the format shown.

Table 7 Buffer Depth

CGC_LINE_OUT[4:0] Value	CGC_DOUT[3:0] Value			
	3	2	1	0
5	queue(7)	queue(6)	queue(5)	queue(4)
4	queue(3)	queue(2)	queue(1)	queue(0)
3	A1SP(1)	A1SP(0)	buff_depth(13)	buff_depth(12)
2	Buff_depth(11)	buff_depth(10)	buff_depth(9)	buff_depth(8)
1	Buff_depth(7)	buff_depth(6)	buff_depth(5)	buff_depth(4)
0	Buff_depth(3)	buff_depth(2)	buff_depth(1)	buff_depth(0)

Figure 61 Adaptive Data Functional Timing



11.3.3.5 Ext Freq Select Interface

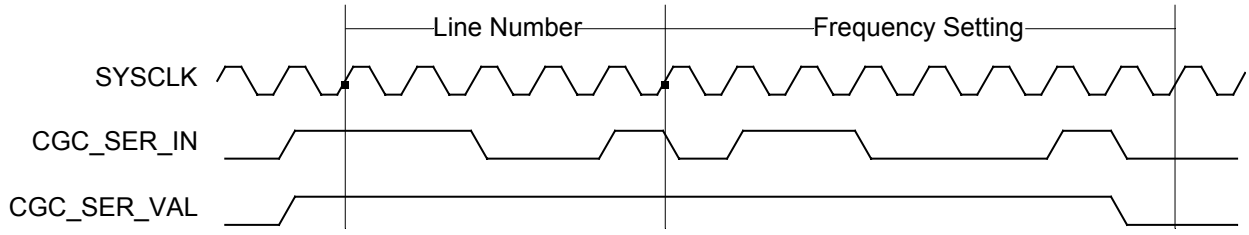
The Ext Freq Select Interface allows an external source to directly select the line clock frequency from any one of 171 T1 or 240 E1 frequencies centered around the nominal clock rate. For T1 the legal input values are –83 to 88. For E1 the legal input values are –128 to 111. Any values outside of this range will be clamped to these levels. These levels correspond to a +/-200 ppm T1 clock and a +/- 100 ppm E1 clock. See section 11.3.3.8 for more detailed information.

The External Interface block has two input ports that allow an external source to control the frequency synthesizers internal to the CGC. These two ports are CGC_SER_D and CGC_VALID. CGC_SER_D contains the data that selects one of the 171/240 frequencies and CGC_VALID indicates when this data is valid. There should be a rising edge of CGC_VALID at the start of each timing message. And CGC_VALID should go low at the end of each timing message.

CGC_VALID only needs to be deactivated for one cycle between timing messages. The CGC block decodes the incoming line number, and if the address matches one of its 32 line numbers, passes the data on to the appropriate frequency synthesizer.

Figure 62 below shows an example of where Line 19 is being programmed to run with the frequency setting of –79 (Two's complement). See the section on the Frequency Synthesizer block for a discussion of the different frequency settings which range from -83 to 88 in T1 mode and –128 to 111 in E1 mode.

Figure 62 Ext Freq Select Functional Timing



11.3.3.6 SRTS

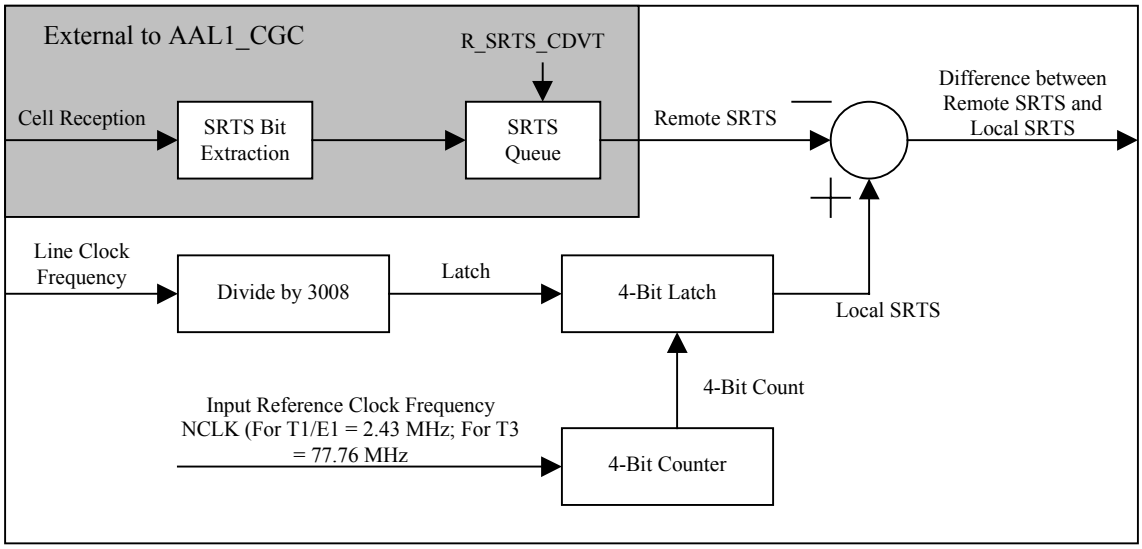
SRTS functionality is enabled by setting the SRTS_EN bit in the LIN_STR_MODE memory register. To disable SRTS processing on the chip via hardware, tie the NCLK/SRTS_DISB signal to ground.

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 is supported for unstructured data formats on a per-line basis. SRTS support requires an input reference clock (NCLK). The input reference frequency is defined as $155.52 / 2^n$ MHz, where n is chosen so the reference clock frequency is greater than the frequency being transmitted, but less than twice the frequency being transmitted (2 x TL_CLK > NCLK > TL_CLK). For T1 or E1 implementations, the input reference clock frequency is 2.43 MHz and must be synchronized to the ATM network. Figure 63 shows the process implemented for each UDF line enabled for SRTS. The CGC generates a local SRTS value from the network clock (NCLK) and the local TL_CLK. It makes a request to the A1SP to read a new remote SRTS value from the SRTS queue and, at the appropriate time, generates a 4-bit two's complement code that indicates the difference between the locally generated SRTS value and the incoming SRTS value. The

value of this code ranges from –8 (1000) to +7 (0111). A higher value than had been output previously indicates the remote clock is running faster than the local clock. A lower value than had been output previously indicates the remote clock is running slower than the local clock. The AAL1_CGC uses this value to synthesize the TL_CLK. Since the frequency synthesizers accept 8 bit input values, the lower 4 bits will be set to zero and the upper 4 bits will receive the SRTS nibble. The synthesizers should be set for low resolution mode by programming the HI_RES_SYNTH register bit in the LIN_STR_MODE memory register to zero.

Figure 63 Receive Side SRTS Support



11.3.3.7 Adaptive

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.

When CDV is accounted for, the buffer depth will not only vary due to differences in the line clock frequencies, but also due to differences in the interarrival time of cells. In order to truly measure the difference in the remote line clock frequency and the local line clock frequency the buffer depth needs to be filtered. The

method the Adaptive block uses is a weighted moving average algorithm where the amount of weighting is programmable. The following formula is used to determine the average buffer depth.

$$\frac{(\text{PrevAvgDepth} * (N-1)) + \text{CurBufDepth}}{N}$$

N = Weighting = 2^Adap_Filt_Size

PrevAvgDepth = Previous Average Buffer Depth

CurBufDepth = Current Buffer Depth

N must be a power of 2 and is controlled by the ADAP_FILT_SIZE field in the An_ADAP_CFG register. If ADAP_FILT_SIZE equals '0' then no filtering is done.

To enable quicker lock times, the window size will start small and will grow until it matches the ADAP_FILT_SIZE parameter. The average value will reset when either the line is disabled or it goes into underrun.

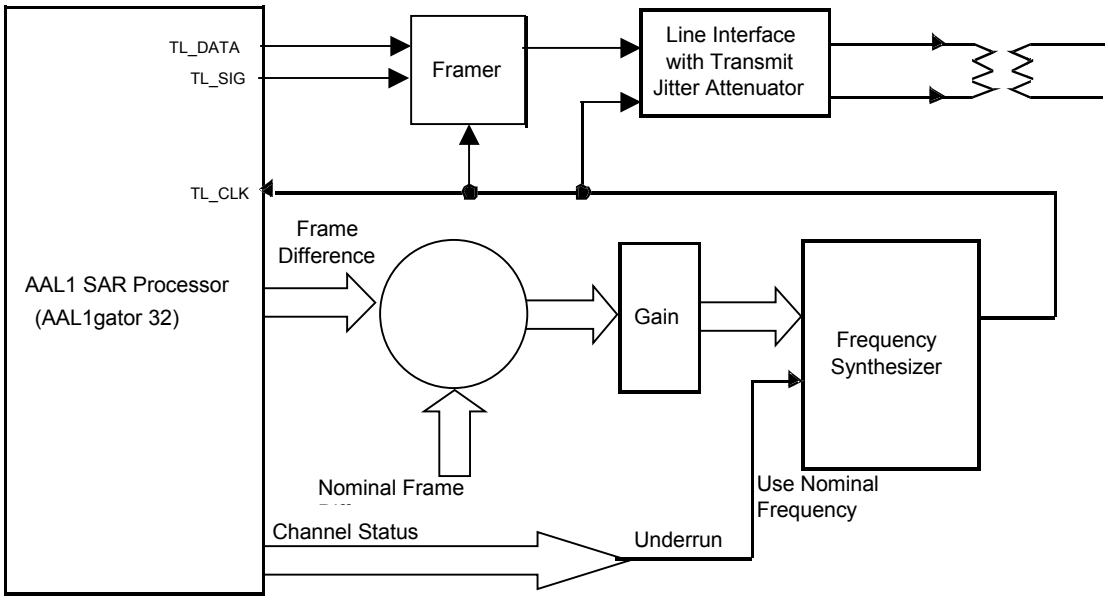
The value sent to the synthesizers is a truncated value of the Relative Buffer Depth. Relative Buffer Depth is the difference between the calculated average buffer depth and the CDVT setting (R_CDVT in the receive queue table) which represents the ideal buffer depth value. The Relative Buffer Depth is limited to represent a value of +/- 8 frames (ie, +/- 256 bytes) from the CDVT setting. Relative buffer depths that exceed this value are truncated to a max/min value of +/- 8 frames. Using a signed 8 bit data value, this corresponds to 1 bit representing 2 bytes of data.

The frequency synthesizer will further cap this value by limiting the frequency range to +/- 200 ppm for T1 and +/-100 ppm for E1.

Note that adaptive clocking, in general, is not well suited for voice applications since low frequency or DC changes of the CDV will pass through most filters and cause frame slips. 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. The buffer depth will be played out for each queue, but the information will be in frames. (the bottom 5 bits are invalid).

If an alternative algorithm is desired, this can be handled externally following an architecture similar to what is shown in Figure 64. Note that the internal synthesizers can be used via the CGC control port, so that only the adaptive algorithm has to be in external logic and not the synthesizers.

Figure 64 Direct Adaptive Clock Operation



11.3.3.8 Frequency Synthesizer

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.

In order for this block to work properly, the system clock must be 38.88 MHz. The accuracy of the synthesized clock is dependent on the accuracy of SYS_CLK. Therefore, if a 50 ppm T1 clock is desired, SYS_CLK needs to be a 38.88 MHz clock signal with 50 ppm accuracy. To lock the synthesized clock to a network clock, be sure SYS_CLK is derived from the network clock. This block can be accessed through the CGC serial data in port, and it is also used internally by the SRTS and adaptive algorithms

To minimize jitter, long and short cycles are distributed. The resultant synthesized clocks meet G.823 and G.824 specs for jitter. Jitter can be further reduced by running the TL_CLKs through a jitter attenuator in the framer or LIU.

The synthesizers can be set to operate in normal or high resolution mode. In normal mode only the 4 high order bits of the frequency setting value are

monitored to generate 1 of 16 frequencies. In high resolution mode all 8 bits are monitored to generate 1 of 256 frequencies. SRTS requires the use of normal mode, while high resolution mode is recommended for Adaptive clock recovery.

11.3.3.8.1 T1 Mode

In T1 mode the 1.544 MHz nominal frequency is synthesized by maintaining a certain ratio of long cycles to short cycles. The long cycle is comprised of 26 SYS_CLK periods (38.88 MHz). The short cycle is comprised of 25 SYS_CLK periods. For the nominal frequency in high resolution mode, the synthesis period is comprised of 3088 cycles. 560 of the 3088 cycles are long cycles and the remaining 2528 are short cycles. For the nominal frequency in low resolution mode, the synthesis period is comprised of 193 cycles, 35 of those 193 cycles are long cycles and the remaining 158 are short cycles. In order to alter the frequency the number of short cycles is increased or decreased according to the frequency select value. The resultant frequency is calculated using the following equation:

$$\frac{38.88 * (M + N)}{26M + 25N}$$

M = # long cycles

N = # short cycles

Table 8 shows results of this calculation for various values of frequency select. . The average increment in frequency is ~3.6 Hz.

Table 8 Frequency Select – T1 Mode

Freq Select	M	N	Total # Cycles	Frequency
-128	560	2432	2992	1.5436433121
-127	560	2432	2992	1.5436433121
-126	560	2432	2992	1.5436433121
-125	560	2432	2992	1.5436433121
…	…	…	…	…
-96	560	2432	2992	1.5436433121
-95	560	2433	2993	1.5436471447
-94	560	2434	2994	1.5436509747
-93	560	2435	2995	1.5436548021
-92	560	2436	2996	1.5436586271
…	…	…	…	…
-4	560	2524	3084	1.5439855782
-3	560	2525	3085	1.5439891871
-2	560	2526	3086	1.5439927937
-1	560	2527	3087	1.5439963980
0	560	2528	3088	1.5440000000
1	560	2529	3089	1.5440035997
2	560	2530	3090	1.5440071970
3	560	2531	3091	1.5440107921
4	560	2532	3092	1.5440143848
…	…	…	…	…
93	560	2621	3181	1.5443251545
94	560	2622	3182	1.5443285482
95	560	2623	3183	1.5443319399
96	560	2624	3184	1.5443353293
…	…	…	…	…
124	560	2624	3184	1.5443353293
125	560	2624	3184	1.5443353293
126	560	2624	3184	1.5443353293
127	560	2624	3184	1.5443353293

***** Note that the synthesizers are limited internally to a freq select range of –96 to 96 in order to maintain a +/- 230 ppm range.

11.3.3.8.2 E1 Mode

In E1 mode the 2.048 MHz nominal frequency is synthesized by maintaining a certain ratio of long cycles to short cycles. The long cycle is comprised of 19 SYS_CLK periods (38.88 MHz). The short cycle is comprised of 18 SYS_CLK periods. For the nominal frequency in high resolution mode, the synthesis period is comprised of 1024 cycles. 1008 of the 1024 cycles are long cycles and the

remaining 16 are short cycles. For the nominal frequency in low resolution mode, the synthesis period is comprised of 64 cycles, 63 of the cycles are long and the remaining 1 is short. In order to alter the frequency the number of long cycles is increased or decreased according to the frequency select value. The resultant frequency is calculated using the following equation:

$$\frac{38.88 * (M + N)}{19M + 18N}$$

M = # long cycles

N = # short cycles

Table 9 shows results of this calculation for various values of frequency select.

Table 9 Frequency Select – E1 Mode

Freq Select	M	N	Total # Cycles	Frequency
-128	1135	16	1151	2.0478140301
-127	1134	16	1150	2.0478153339
-126	1133	16	1149	2.0478166399
-125	1132	16	1148	2.0478179482
-124	1131	16	1147	2.0478192589
-123	1130	16	1146	2.0478205717
-122	1129	16	1145	2.0478218869
-121	1128	16	1144	2.0478232044
-120	1127	16	1143	2.0478245242
-119	1126	16	1142	2.0478258463
…	…	…	…	…
-4	1012	16	1028	2.0479934413
-3	1011	16	1027	2.0479950762
-2	1010	16	1026	2.0479967142
-1	1009	16	1025	2.0479983555
0	1008	16	1024	2.0480000000
1	1007	16	1023	2.0480016477
2	1006	16	1022	2.0480032986
3	1005	16	1021	2.0480049528
4	1004	16	1020	2.0480066102
…	…	…	…	…
109	899	16	915	2.0482008175
110	898	16	914	2.0482028818
111	897	16	913	2.0482049507
112	896	16	912	2.0482070240
…	…	…	…	…
124	896	16	912	2.0482070240
125	896	16	912	2.0482070240
126	896	16	912	2.0482070240
127	896	16	912	2.0482070240

* Note that the frequency synthesizers are limited internally to a freq select range of –128 to 112 in order to maintain a +/- 100 ppm range. The average increment in frequency is ~1.66 Hz.

11.4 Processor Interface Block (PROCI)

The microprocessor interface block provides normal and test mode registers, as well as memory mapped registers and the logic required to connect to the microprocessor interface. The normal mode registers and memory mapped registers are required for normal operation, and test mode registers are used to enhance the testability of the AAL1gator-32. A general memory map for the register set can be seen in the following four figures:

Figure 65 shows the memory region broken into five blocks. The first four blocks, A1SP0 – A1SP3, are the memory mapped registers which are mostly contained within external SSRAM. The four blocks are identical and are accessed by taking the 17-bit address shown in Figure 66 and appending the 2-bit A1SP identifier to the front. The fifth block, Internal Registers, is composed of configuration registers common to the entire chip that are contained internally to the chip.

Figure 65 Memory Map

00000 1FFFF	A1SP 0 SRAM
20000 3FFFF	A1SP 1 SRAM
40000 5FFFF	A1SP 2 SRAM
60000 7FFFF	A1SP 3 SRAM
80000 BFFFF	Internal Registers
C0000 FFFFF	Test Registers

Figure 66 shows the structure of an A1SP block within the external SSRAM. The addresses listed in the figure represent the relative location within the A1SP block, and a 2-bit A1SP identifier appended to the front of these addresses selects the particular A1SP block. The first block, Control Registers, contains registers which are common to both the transmit block and the receive block. The second and third blocks contain registers which are specific to the transmit block and the receive block respectively.

Figure 66 A1SP SRAM Memory Map

00000 0001F	Control Registers
00020 07FFF	Transmit Data Structures
08000 1FFFF	Receive Data Structures

Figure 67 shows the Control Registers block in more detail.

Figure 67 Control Registers Memory Map

00000	Reserved
00001	HS_LIN_REG
00002 00003	Unused
00004	P_FILL_CHAR
00005 0000F	Unused
00010	LIN_STR_MODE0
00011	LIN_STR_MODE1
00012	LIN_STR_MODE2
00013	LIN_STR_MODE3
00014	LIN_STR_MODE4
00015	LIN_STR_MODE5
00016	LIN_STR_MODE6
00017	LIN_STR_MODE7
00018 0001F	Unused

Figure 68 shows the format of the Transmit Data Structures block in more detail.

Figure 68 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)

Figure 69 shows the format of the Receive Data Structures block in more detail.

Figure 69 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)

Figure 70 below shows the format of the Normal Mode Registers block in more detail.

Figure 70 Normal Mode Registers Memory Map

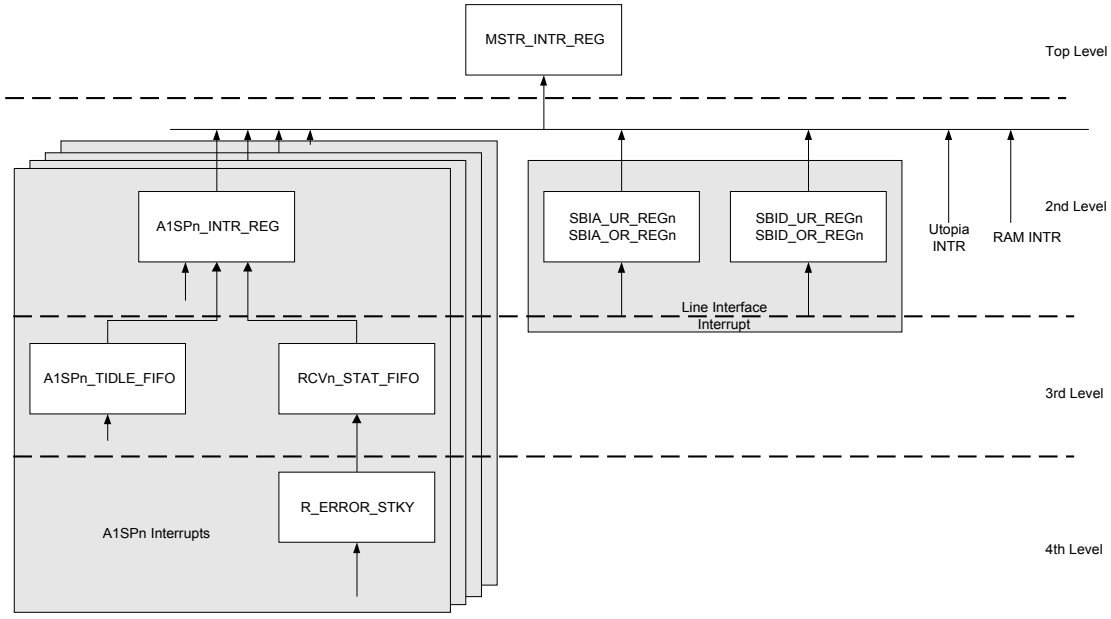
80000 800FF	Command Reigsters
80100 8011F	RAM Interface Registers
80120 801FF	UTOPIA Interface Registers
80200 80FFF	Line Interface Reigsters
81000 812FF	Interrupt and Status Registers
82000 82FFF	Idle Channel Configuration and Status Registers
84000 84FFF	DLL Control and Status Registers

11.4.1 Interrupt Driven Error/Status Reporting

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 A1SP blocks and the LINE Interface block. The top layer of the interrupt logic, the Master Interrupt Register, 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 71 shows the registers in the interrupt tree. The microprocessor traverses the tree based on the value of individuals bits within each register.

Figure 71 Interrupt Hierarchy



11.4.1.1 UTOPIA Interrupts

The UTOPIA block sources five interrupts directly to the Master Interrupt Register. 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.

11.4.1.2 RAM Interface Interrupts

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

11.4.1.3 A1SP Interrupts

The A1SP blocks each source an interrupt to the Master Interrupt Register. Within each A1SP block, the A1SPn 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: A1SPn Receive Status FIFO not empty, A1SPn Transmit Idle

State FIFO not empty, A1SPn 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 A1SPn Receive Status FIFO overflow, A1SP Transmit Idle State FIFO overflow, and Frame Advance FIFO overflow interrupts simply indicate that the FIFOs have overflowed. The A1Spn Receive Status FIFO not empty interrupt indicates the A1SPn Receive Status FIFO is not empty and the A1SPn 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 A1SPn_INTR_REG captures if the condition has occurred since the last time the register was read. The A1SPn_STAT_REG reflects the current status.

11.4.1.3.1 A1SPn 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 INTR_FIFO_EMPB bit in both the A1SPn_INTR_REG and the A1SPn_STAT_REG will go active. When there are no longer any entries in the FIFO, the INTR_FIFO_EMPB bit in the A1SPn_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. An error entry will only occur for the latter case if this is the first unmasked sticky bit error to occur for this queue since the last time the sticky bit memory register was cleared.

11.4.1.3.2 A1SPn 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 A1SPn_INTR_REG and the A1SPn_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 A1SPn_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. See the section on DBCES in the TX A1SP section which discusses the different DBCES words and the structure of the entries in the Idle State FIFO.

11.4.1.4 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.

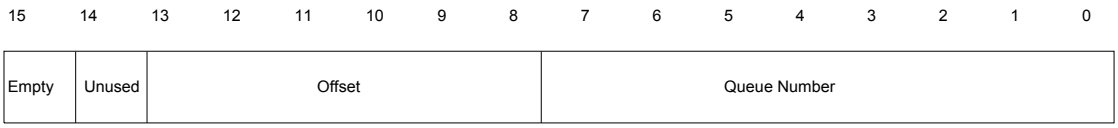
11.4.2 Add Queue FIFO

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 72. 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 (see section 11.2.1.2.1 on Transmit CDV).

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 72 ADDQ_FIFO Word Structure



It is necessary to understand the basics of the frame-based scheduling performed by the chip to best utilize the Offset field in the ADDQ_FIFO. Read Section 11.2.1.2 and pay particular attention to how the Transmit Calendar works and to Section 11.2.1.2.1 and the information on clumping. The purpose of the offset field in the ADDQ_FIFO is to reduce the amount of clumping.

In frame mode (SDF-FR), the first cell of a queue can be scheduled relative to a reference value. When REF_VAL_ENABLE=0 in the LIN_STR_MODE memory mapped register, the reference value is always frame 0. When REF_VAL_ENABLE =1, the reference value is based upon the configuration consisting of all 24 (T1) or 32 (E1) queues configured identically as single DS0 with no pointer, full cells, and no CAS connections which will be scheduled to build cells during frame 0, 47, 94, 13, etc.. The reference value increments by 47 frames every time the current t_data_buffer frame write pointer is equal to the current reference value. When the microprocessor wants to add a queue, it writes the queue number and an offset to the ADDQ_FIFO. This offset is then added to the current reference value, and the first cell is scheduled in the resulting frame. For example, if two single DS0 queues are scheduled one right after the other with offsets of 0 and 1 respectively, there will never be any clumping because the first queue will be scheduled during frames 0, 47, 94, 13, etc, and the second queue will be scheduled during frames 1, 48, 95, 14, etc..

Note that the reference value is optimized to the configuration consisting of all 24 (T1) or 32 (E1) queues configured identically as single DS0 with no pointer, full cells, and no CAS. This is the configuration that is most likely to experience clumping. By using the offset field, cells can be prevented from being scheduled within the same frame even if queues were added some time ago. Software can guarantee this by assigning an offset equal to queue number(4:0), or by keeping track of which offsets are available. For non-single DS0 queues, the offset is beneficial for initial offsets if all queues have the same configuration, but won't guarantee that a newly added queue won't overlap with a queue started some time ago. However, for non-single DS0 queues, there are less queues active per line, and therefore clumping is less of an issue.

In multiframe mode (SDF-MF), the REF_VAL_ENABLE bit should be set only for lines which have all queues configured identically as single DS0 queues with full cells. When enabled, the reference value is based upon a configuration consisting of all 24 (T1) or 32 (E1) queues configured identically as single DS0

full cells with CAS. The reference value increments by 45 frames for T1 connections, and for E1 connections the reference value increments by either 44 frames or 45 frames to give an average of 44 2/17ths frames. By using different offsets, clumping can be minimized, but not completely avoided as in the frame mode (SDF-FR) description above. However, when using offsets, generation of a 0 pointer cannot be guaranteed, because cells may start on non-MF boundaries.

In order to guarantee a zero pointer for single DS0 SDF-MF connections as well as all non-single DS0 SDF-MF connections, REF_VAL_ENABLE should be disabled and the offset field used should be set equal to the FRAMES_PER_CELL field in the QUEUE_CONFIG word of the Transmit Queue Table. This will add the queue in the frame which is FRAMES_PER_CELL past frame number 0. Note that doing this will cause all queues, that have the same FRAMES_PER_CELL value, that are added at the same time, to be scheduled in the same frame. To avoid the clumping of cells, Add-Queue operations can be spread out in time so that not all queues are added at the same time.

For SDF-MF DBCES queues, OFFSET must be set equal to FRAMES_PER_CELL.

In summary, when REF_VAL_EN is set, the generated CDV for the following two configurations will be minimized, more closely approaching the ideal minimum CDV for each case (REF_VAL_EN provides no added value for other configurations.):

- All 24(32) Queues SDF-FR, Single-DS0-with-no-pointer, full cells, Ideal minimum CDV = 0 us. Clumping is guaranteed not to occur.

- All 24(32) Queues SDF-MF, Single-DS0, full cells, Ideal minimum CDV = 125 us. Clumping is minimized.

IMPORTANT NOTE:

If a multiframe resync occurs on a line and REF_VAL_ENABLE is set, the relationship between different queues on that line will change. If this occurs, clumping may occur in the existing connections and in any future connections that are added. In H-MVIP mode, multiframe resyncs will never occur. In low speed mode, if REF_VAL_ENABLE is set it is recommended that MF_SYNC_MODE in LS_Ln_CFG_REG is disabled, as this will prevent multiframe resyncs. In SBI mode, multiframe resyncs cannot be prevented and may occur if a slip occurs on that tributary. A multiframe resync error will be reported if this occurs.

11.5 RAM Interface Block (RAMI)

The RAMI 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.

The AAL1gator-32 has a separate SSRAM and processor interface. Either one or two 256K x 16 or 256K x 18 SSRAMs are needed, depending on the mode of operation. If using the SBI interface, or the upper four 8 Mbps MVIP interfaces, or the upper two High speed interfaces, two rams are required

Either a pipelined synchronous SRAM with a single cycle deselect or a pipelined ZBT or ZBT-compatible synchronous SRAM can be used.

For most applications the pipelined single-cycle deselect SSRAM 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 SSRAM is recommended.

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

11.6 Line Interface Block (AAL1_LI)

11.6.1 Conventions

The following conventions are used in this document:

The 32 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. The AAL1gator-32 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.

11.6.2 Functional Description

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 options available are:

Direct Low Speed Mode

This is mainly a pass through mode between the external 16 lines and the lower 16 local links, which connect to the A1SP blocks. This mode is used to connect to any PMC E1 or T1 Framer or compatible device. This mode is also used to interface to devices, which support the MVIP-90 protocol. 16 lines are supported, including a clock, data, frame pulse, and signaling pin for each direction. In addition any low speed (< 2.5 MHz) clear channel data stream can be passed in this mode.

Note clocks rates up to 15 MHz are supported in this mode. However, the aggregate bandwidth cannot exceed 20 Mbps. per A1SP block. Therefore, if all 8 lines of the A1SP are used and are the same rate, 2.5 MHz is the highest rate supported.

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.

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 16 lines. Tri-stating of individual time slots is not supported. There is a common 4 MHz clock and a common framing reference signal.

Note that if a mix of MVIP-90 and non MVIP-90 lines are being used, line 0 must be MVIP-90.

H-MVIP Mode

This mode supports 16 separate 8 Mbps H-MVIP lines with 8 in the transmit direction and 8 in the receive direction. There is a common 16 MHz clock, a 4 MHz clock, and a common framing reference signal. Eight Separate data pins and signaling pins are provided in each direction. Individual time slots cannot be disabled.

Internally each 8 Mbps stream is converted into four 2 Mbps streams in a round-robin fashion.

SBI Mode

This mode is used to interface to the TEMUX chip or any other PHY layer chip that has a SBI Interface on it. In this mode, 32 T1 or E1 lines or two DS3 lines are supported. Also a combination of any two of the following: 16 E1 lines, 16 T1 lines, or 1 DS3 line is supported.

High Speed Mode

This mode is used to interface to a high speed clear channel data stream. Two high speed lines are supported. This is a pass-thru mode where external lines (0,2) are connected to internal links(0,16). Only clock and data are used.

The mode of the module is determined by the value of the LINE_MODE pins. The following encoding is used:

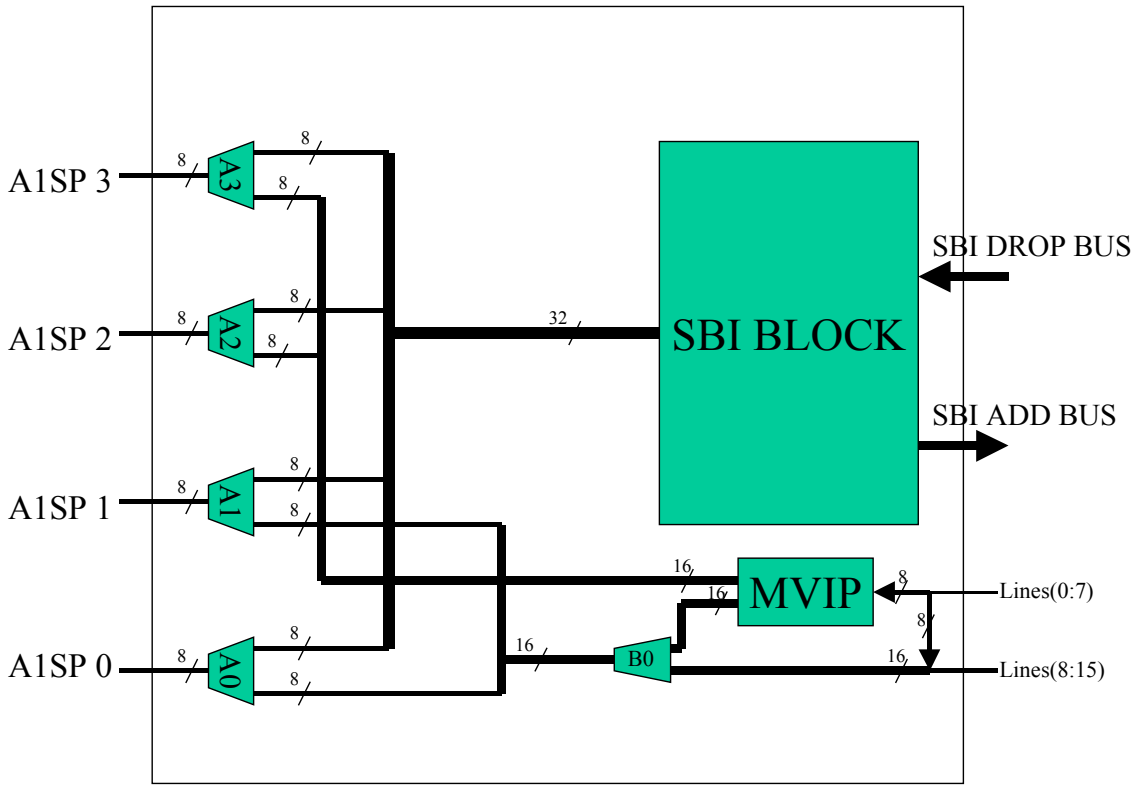
Table 10 LINE_MODE Encoding

LINE_MODE[1:0]	Line Interface Mode
“00”	Direct Low Speed
“01”	SBI
“10”	H-MVIP
“11”	High Speed

Figure 73 shows the block diagram for the Line Interface Block. The block consists of the SBI Block, a H-MVIP Block and mux/demux logic.

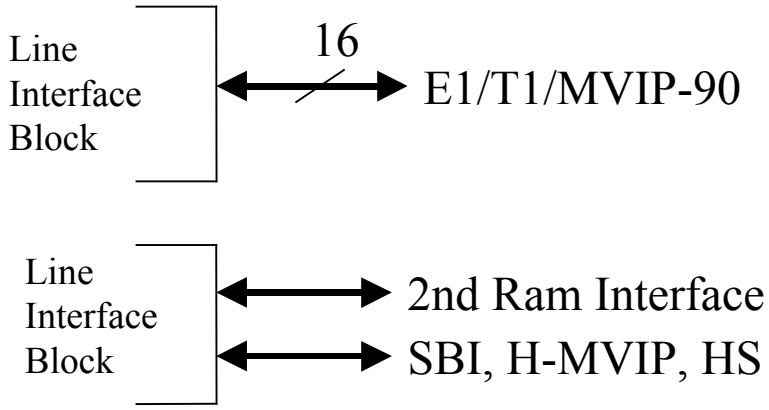
The A0 to A3 mux/demux logic selects between SBI links and non-SBI links. The B0 mux/demux logic selects between the H-MVIP data or external direct links. The upper 16 lines can only be used in H-MVIP, or SBI mode. In high speed mode, external lines 0, and 2 are used, but they are mapped to internal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

Figure 73 Line Interface Block Architecture



Internal to the chip, two of the A1SP blocks use one ram interface and two use the other ram interface. The second ram interface shares pins with lines [8:15]. The second ram is required when using SBI mode, the upper 4 H-MVIP lines or the 2nd high speed line. The second ram cannot be used and is not needed when using Direct Low Speed mode. See Figure 74 for figure of different options.

Figure 74 Line Interface and 2nd RAM Interface



The different modes of the Line Interface Block require the pin definitions to change depending on the mode you are in. See Section 9 for exact definitions.

The following sections describe each of the four modes

11.6.2.1 Direct Low Speed Mode

This section defines how the Line Interface functions when in Direct Low Speed Mode. The Line interface does no processing on the any of the line signals but just passes them between the lower 16 local links, which connect to A1SP 0 and 1, and the 16 external lines. In this mode the SBI Block and H-MVIP portions of the Line Interface Block are not used.

The type of line is selected based on the value of T1_MODE in the LIN_STR_MODE memory register for each line. For unstructured data format (UDF mode), the value of T1_MODE does not matter because data is interpreted as a clear channel bit stream. If MVIP-90 mode is being used (MVIP_EN bit is set in the LS_Ln_CFG_REG register for this line), then T1_MODE should be inactive (‘0’).

The frame structure for E1 and T1 lines can be unstructured-multiline (UDF-ML), Structured-Frame (SDF-FR) or Structured-Multi-Frame (SDF-MF) and is determined by the value of FR_STRUCT[1:0] in the LIN_STR_MODE memory register for each line.

00 Reserved

01 SDF-FR

10 UDF-ML

11 SDF-MF

SDF-FR mode is used when making a structured connection and CAS signaling is not being transported. SDF-MF mode is used when making a structured connection and CAS signaling is being transported. 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.

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.

In the receive direction, the CLK_SOURCE_RX bit has two possible options. If this bit is set then the line receives its clock from the CRL_CLK pin. If this bit is not set then the line receives its clock from the RL_CLKn pin associated with that line.

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

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 in the CGC Block as a nominal E1 or T1 clock based on SYS_CLK and the value of T1_MODE. The clock is output on TL_CLK[n] pin.

011 Clock is internally synthesized in the CGC Block based on SRTS. The clock is output on TL_CLK[n] pin.

100 Clock is internally synthesized in the CGC Block using the adaptive algorithm. The clock is output on TL_CLK[n] pin.

101 Clock is internally synthesized based on values received on the CGC interface. (externally controlled) The clock is output on TL_CLK[n] pin.The clock on CTL_CLK input pin is used.

110 The clock on CTL_CLK input pin is used and signaling data is output on TL_CLK[n] pin.

11.6.2.1.1 Receive Direction

The Line Interface Block accepts deframed data from the 16 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. (However the total aggregate bandwidth cannot exceed 20 Mbps). 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 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. 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 normal mode (MVIP_EN bit is low in the LS_Ln_CFG_REG register for this line), 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 (MVIP_EN bit is set in the LS_Ln_CFG_REG register for this line), 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.

It is not necessary to provide an edge at the beginning of every frame or multi-frame. However if a frame or multi-frame pulse is detected on a non-frame or non-multi-frame boundary, then the AAL1gator-32 will resync to the new boundary and cell generation will be suppressed for 12 – 32 ms.

In T1 mode a multi-frame can either be 12 or 24 frames depending on if the line is in Super Frame (SF) or Extended Super Frame (ESF) mode. An E1 multi-frame is 16 frames long. A special case of E1 mode exists that permits the use of T1 signaling with E1 framing. When E1_WITH_T1_SIG is set in LIN_STR_MODE and the line is in E1 mode, a multi-frame of 24 frames will be used.

Since signaling is accumulated by the framer over an entire multi-frame, signaling only has to be sampled once per multi-frame. The AAL1gator-32 reads signaling during the last frame of every multi-frame. If the framer supplies

constant signaling for an entire multi-frame, then the multi-frame sync signal is not required unless there is a desire to synchronize the multi-frame with the data across the network. In general this is not necessary.

The AAL1gator-32 reads the signaling nibble for each channel when it reads the last nibble of each channel's data. See Figure 75 for an example of a T1 frame. See Figure 76 for an example of an E1 frame.

Figure 75 Capture of T1 Signaling Bits

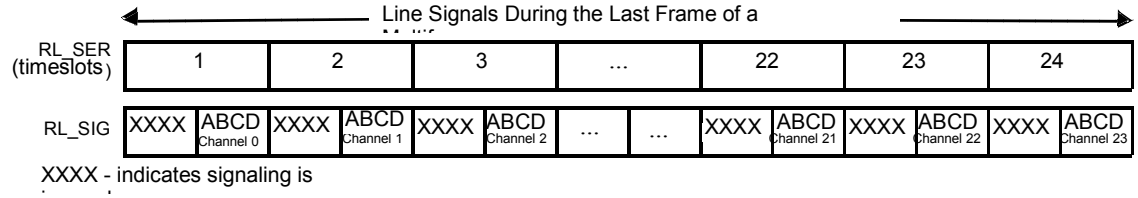
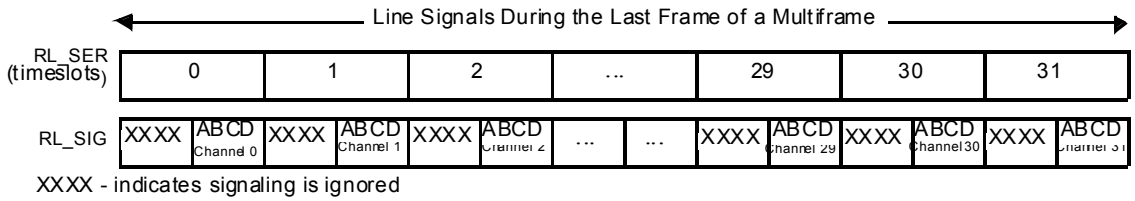


Figure 76 Capture of E1 Signaling Bits



Note:

AAL1gator-32 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.

11.6.3 Transmit Direction

In the line transmit direction, for structured data, the Line Interface Block may take the TL_SYNC input signal and depending on the value of MF_SYNC_MODE interpret the signal as either a frame pulse or multi-frame pulse. Alternatively if GEN_SYNC in LIN_STR_MODE memory register is high for that line, then the Line Interface Block will take the frame pulse or multi-frame pulse generated by the A1SP Block for the local link and output that signal to the TL_SYNC[n] pin on the external lines. Whether the Line Interface Block

generates 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.

In normal mode (MVIP_EN bit is low in the LS_Ln_CFG_REG register for this line), the first time TL_SYNC is sampled high after being low, indicates the beginning of a frame or multi-frame. In MVIP-90 mode (MVIP_EN bit is set in the LS_Ln_CFG_REG register for this line), the first time TL_SYNC is sampled low after being high, indicates the beginning of a frame or multi-frame. For T1 structured data a frame is completed every 193 bits. For E1 structured data a frame is completed every 32 bytes.

It is not necessary to provide an edge at the beginning of every frame or multi-frame. However if a frame or multi-frame pulse is detected on a non-frame or non-multi-frame boundary, then the AAL1gator-32 will resync to the new boundary and there will be a temporary disruption of data being played out on the line.

In T1 mode a multi-frame can either be 12 or 24 frames depending on if the line is in Super Frame (SF) or Extended Super Frame (ESF) mode. An E1 multi-frame is 16 frames long. A special case of E1 mode exists that permits the use of T1 signaling with E1 framing. When E1_WITH_T1_SIG is set in LIN_STR_MODE and the line is in E1 mode, a multi-frame of 24 frames will be used.

Signaling data is driven for all frames of any multi-frame and will change only on multi-frame boundaries. For T1 mode, signaling data may change every 24th frame. For E1 mode, signaling may change every 16th frame.

The signaling nibble is valid for each channel when the last nibble of each channel's data is being driven. See Figure 77 for an example of signaling bits in a T1 frame. See Figure 78 for an example of signaling bits in the E1 mode.

The line interface block just passes the signaling and data bits from the A1SP block to the external output pins.

Figure 77 Output of T1 Signaling Bits

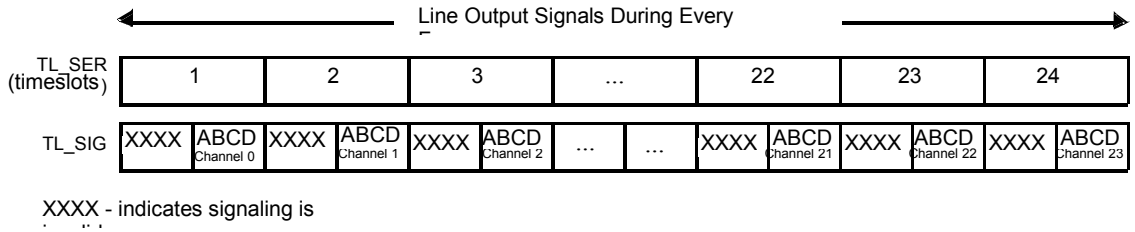
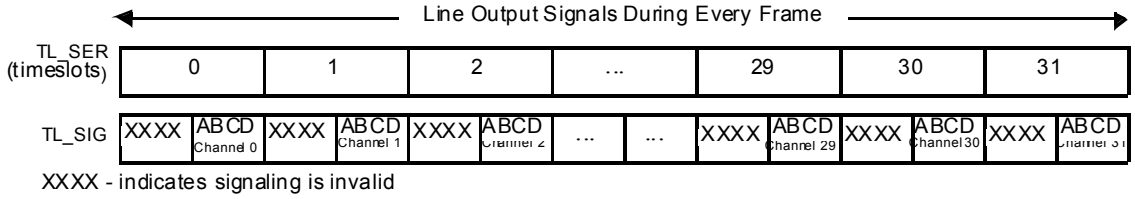


Figure 78 Output of E1 Signaling Bits



Note:

- The AAL1gator-32 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.

11.6.3.1 SBI Mode

11.6.3.1.1 SBI Overview

An overview of the SBI bus and the specific details which relate directly to the AAL1gator-32 will be given here.

11.6.3.1.2 Conventions

The interface where data flows from the Line Interface Block to a Physical Layer device is the ADD BUS Interface. The interface where data flows from the Physical Layer device to the Line Interface Block is the DROP BUS Interface.

The term “Link” refers to the link which is multiplexed onto the SBI bus. This is either a T1, E1, or DS3 signal which is being multiplexed or demultiplexed from the SBI bus. When a Link is multiplexed within the SBI then it is referred to as a tributary. Stated another way a link is internal to the AAL1gator while a tributary is external to the AAL1gator within the SBI bus.

In SBI mode, link and tributary numbering starts at ‘1’.

11.6.3.1.3 General

The Scalable Bandwidth Interconnect (SBI) is a high density 8 bit parallel bus recommended for interconnection of asynchronous and synchronous physical

interface devices with link layer devices. This allows system designers to flexibly interconnect high density multi-port physical layer devices in a standardized way with multi-channel and multi-function link layer devices. The Bus operates at a 19.44MHz +/-50 ppm clock rate. This clock is common to all devices connecting to a Scaleable Bandwidth Interconnect.

The Scaleable Bandwidth Interconnect is shared by multiple PHYs and multiple Link Layer devices such as the AAL1gator-32. The maximum number of devices sharing a bus or signal is limited only by the need to meet the AC timing requirements of the bus.

11.6.3.1.4 Timing Masters

The Scaleable Bandwidth Interconnect is a synchronous bus which is timed to a reference 19.44MHz clock and a 2KHz frame pulse (8KHz is easily derived from the 2KHz and 19.44MHz clock). All sources and sinks of data on this bus are timed to the reference clock and frame pulse. For the Line Interface Block, the SBI reference clock is REFCLK. The reference 2KHz multi-frame pulse is C1FP.

The data format on the data bus allows for compensating between clock differences on the PHY, SBI and Line Interface links. This is achieved by floating data structures within the SBI format, see section 11.6.3.1.8.

Timing is communicated across the Scaleable Bandwidth Interconnect by floating data structures within the SBI. Payload indicator signals in the SBI control the position of the floating data structure and therefore the timing. When sources are running faster than the SBI the floating payload structure is advanced by an octet be passing an extra octet in the V3 octet locations (H3 octet for DS3 mappings). When the source is slower than the SBI the floating payload is retarded by leaving the octet after the V3 or H3 octet unused. Both these rate adjustments are indicated by the SBI control signals.

On the DROP BUS all timing is sourced from the SBI and is passed to the AAL1gator by the arrival rate of data over the SBI. The only exception to this rule is when using DS3 mode and SRTS. See Section 11.6.3.1.5.

On the ADD BUS timing can be controlled by either the SBI or the AAL1gator by controlling the payload and by making justification requests. When the local transmit link is the timing master the external SBI PHY device gets it's transmit timing information from the arrival rate of data across the SBI. When the external PHY device is the timing master it signals the Line Interface Block to speed up or slow down with justification request signals. The external PHY timing master indicates a speedup request to the Line Interface Block by asserting the justification request signal (AJUST_REQ) high during the V3 or H3 octet. When this is detected by the Line Interface Block it will advance the channel by

inserting data in the next V3 or H3 octet as described above. The PHY timing master indicates a slowdown request to the Line Interface Block by asserting the justification request signal (AJUST_REQ) high during the octet after the V3 or H3 octet. When detected by the Line Interface Block it will retard the channel by leaving the octet following the next V3 or H3 octet unused. Both advance and retard rate adjustments take place in the frame or multi-frame following the justification request.

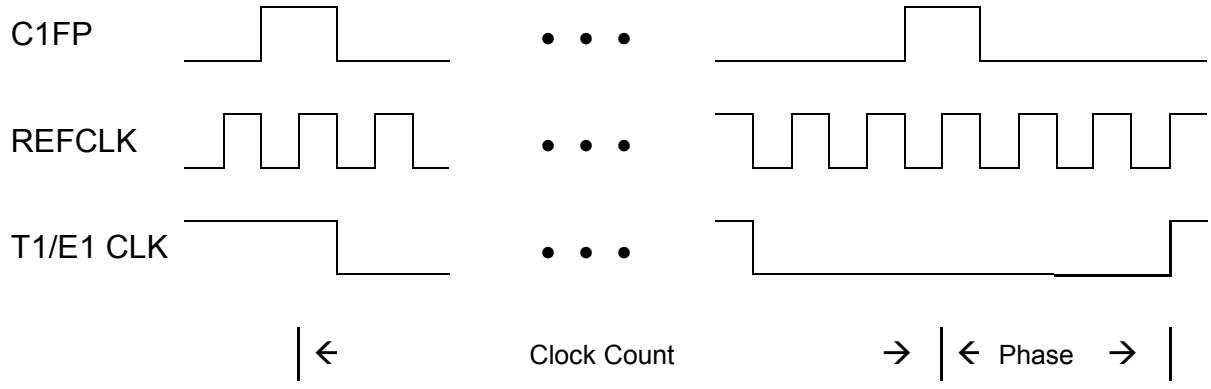
11.6.3.1.5 Jitter

The Scaleable Bandwidth Interconnect is a time division multiplexed bus and as such, introduces jitter into the transported signal. Although ideal for datacom applications, care needs to be taken when using the SBI for some jitter sensitive applications.

Link rate information can optionally be carried across the SBI on a per link basis. Two methods are specified, one for T1 and E1 channels and the second for DS3 channels. These methods use REFCLK and the C1FP frame synchronization signal to measure channel clock ticks and clock phase for transport across the bus.

The T1 and E1 method allows for a count of the number of T1 or E1 rising clock edges between two C1FP frame pulses. This count is encoded in ClkRate[1:0] to indicate that the nominal number of clocks, one more than nominal or one less than nominal should be generated during the C1FP period. This method also counts the number of REFCLK edges after sampling C1FP low to the next rising edge of the T1 or E1 clock, giving the ability to control the phase of the generated clock. The link rate information is passed across the SBI bus via the V4 octet and is shown in Figure 79. Table 11 shows the encoding of the clock count, ClkRate[1:0], passed in the link rate octet.

Figure 79 T1/E1 Link Rate Information



Link Rate Octet	Bit #	7	6	5:4	3:0
T1/E1 Format		ALM	0	ClkRate[1:0]	Phase[3:0]

Table 11 T1/E1 Clock Rate Encoding

ClkRate[1:0]	T1 Clocks / 2KHz	E1 Clocks / 2 KHz
“00” – Nominal	772	1024
“01” – Fast	773	1025
“1x” – Slow	771	1023

When supporting DS3 links, the RL_CLK0/2 and TL_CLK0/2 clocks must be used directly. In this case the AAL1gator should be configured as an SBI clock master and the CLK_SOURCE bits in LIN_STR_MODE memory register need to select the clock pins. That is CLK_SOURCE_RX=’1’ and CLK_SOURCE_TX=”000”. (Note the remote side on the other side of the SBI bus will also be using the same external DS3 clock as a master. Therefore both sides think they are the master and are ignoring the link rate information on the SBI. Since they are both using the same clock, there will not be a problem since the buffering of the SBI data is large enough to accommodate jitter caused directly by the SBI bus).

11.6.3.1.6 Alarms

The SBI specification provides a method for transferring alarm conditions across the SBI bus. This is optional on a per tributary basis and is valid for T1, E1, DS3 tributaries.

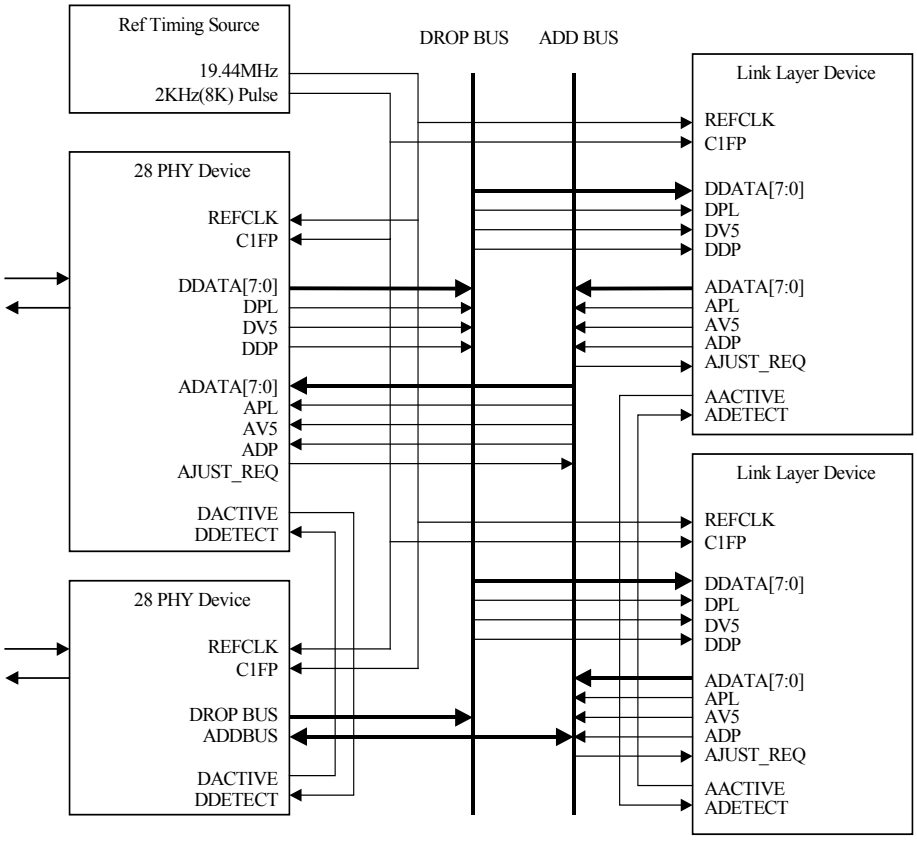
Figure 79 shows the alarm indication bit, ALM, as bit 7 of the Link Rate Octet. The presence of an alarm condition is indicated by the ALM bit set high in the Link Rate Octet. The absence of an alarm condition is indicated by the ALM bit set low in the Link Rate Octet. When SBI_ALRM_EN is set for a given link associated with a particular tributary and an alarm signal is detected on that tributary, the SBI_ALRM_DET bit will be set in the EXT_ALRM_REG register for the link associated with that tributary.

If the SBI_ALRM_INS bit is set in the INS_ALRM_REG register for a given link, then the ALM bit will be set in the tributary associated with that link.

11.6.3.1.7 Interface Example

Figure 80 illustrates how the Scaleable Bandwidth Interconnect interconnects multiple PHY devices with multiple link-layer Devices**.** The AAL1gator is a link layer device.

Figure 80 Multi-PHY to Multi-Link Layer Device Interface



This bus structure is intended to interconnect various physical layer devices with link layer devices. Therefore, the Interfaces that must be supported over this multiplexed bus are varied over a wide range of rates and requirements. Table 12 summarizes the links that are supported by the AAL1gator. The AAL1gator can switch links from one SPE to another so all the links being processed by the AAL1gator do not have to reside within the same SPE. However all links processed by either the upper 16 links or lower 16 links of the AAL1gator have to be of the same type. (Note that the 4 types of links are E1, T1, and DS3). Note also that if DS3 type is used, only the least significant link of that group of 16 links is used.)

Table 12 Supported Links

Link	Mapping	Timing Method	Links
DS3	Unchannelized DS3	Floating Payload	2
E1	Byte Synchronous	Floating Payload or Locked Payload	32
T1	Byte Synchronous	Floating Payload or Locked Payload	32

11.6.3.1.8 Scaleable Bandwidth Interface Multiplexing Structure

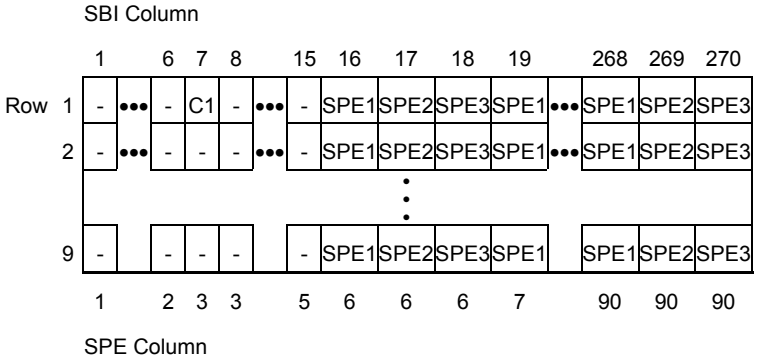
The time division multiplexed bus uses the SONET/SDH virtual tributary structure to carry T1 links, E1 links. Unchannelized DS3 payloads follow a byte synchronous structure modeled on the SONET/SDH format.

The SBI structure uses a locked SONET/SDH structure fixing the position of the TU-3 relative to the STS-3/STM-1. The SBI is also of fixed frequency and alignment as determined by the reference clock (REFCLK) and frame indicator signal (C1FP). Frequency deviations are compensated by adjusting the location of the T1/E1/DS3 channels using floating tributaries as determined by the V5 indicator and payload signals (DV5, AV5, DPL and APL). TVTs also allow for synchronous operation where SONET/SDH tributary pointers are carried within the SBI structure in place of the V5 indicator and payload signals (DV5, AV5, DPL and APL).

Table 13 shows the bus structure for carrying T1, E1, DS3 tributaries in a SDH STM-1 like format. Up to 84 T1s, 63 E1s, or 3 DS3s are carried within the octets labeled SPE1, SPE2 and SPE3 in columns 16-270. All other octets are unused and are of fixed position. The frame signal (C1FP) occurs during the octet labeled C1 in Row 1 column 7.

The multiplexed links are separated into three Synchronous Payload Envelopes called SPE1, SPE2 and SPE3. Each envelope carries up to 28 T1s, 21 E1, or a DS3. SPE1 carries the T1s numbered 1,1 through 1,28, E1s numbered 1,1 through 1,21, or DS3 number 1,1. SPE2 carries T1s numbered 2,1 through 2,28, E1s numbered 2,1 through 2,21, or DS3 number 2,1. SPE3 carries T1s numbered 3,1 through 3,28, E1s numbered 3,1 through 3,21, or DS3 number 3,1.

Table 13 Structure for Carrying Multiplexed Links



The mappings for each link type are rigidly defined, however the mix of links transported across the bus at any one time is flexible. Each synchronous payload envelope, comprising 85 columns numbered 6 through 90, operates independently allowing a mix of T1s, E1s, or DS3s. For example, SPE1 could transport a single DS3, SPE2 could transport 28 T1s and SPE3 could transport 21 E1s. Each SPE is restricted to carrying a single tributary type. SBI columns 16-18 are unused for T1, and E1 tributaries.

Tributary numbering for T1 and E1 uses the SPE number, followed by the Tributary number within that SPE and are numbered sequentially. Table 14 and Table 15 show the T1 and E1 column numbering and relates the tributary number to the SPE column numbers and overall SBI column structure. Numbering for DS3 follows the same naming convention even though there is only one DS3 per SPE.

Table 14 T1 Tributary Column Numbering

T1#	SPE1 Column	SPE2 Column	SPE3 Column	SBI Column
1,1	7,35,63			19,103,187
2,1		7,35,63		20,104,188
3,1			7,35,63	21,105,189
1,2	8,36,64			22,106,190
2,2		8,36,64		23,107,191
•••				
1,28	34,62,90			100,184,268
2,28		34,62,90		101,185,269
3,28			34,62,90	102,186,270

Table 15 E1 Tributary Column Numbering

E1#	SPE1 Column	SPE2 Column	SPE3 Column	SBI Column
1,1	7,28,49,70			19,82,145,208
2,1		7,28,49,70		20,83,146,209
3,1			7,28,49,70	21,84,147,210
1,2	8,29,50,71			22,85,148,211
2,2		8,29,50,71		23,86,149,212
•••				
1,21	27,48,69,90			79,142,205,268
2,21		27,48,69,90		80,143,206,269
3,21			27,48,69,90	81,144,207,270

11.6.3.1.9 AAL1gator-32 SBI Operation

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 type of link is controlled by the setting of LINK_TYPH and LINK_TYPL in SBI_LINK_CFG_REG. On the SBI bus all the tributaries within an SPE must be the same type (E1, T1, or DS3). The type of tributaries carried by an SPE is defined in the SPEn_TYP field in SBI_BUS_CFG_REG. Any tributary within any SPE of the SBI can be mapped to any link within the Line Interface Block. Program the INS_TRIB_MAP ram and EXT_TRIB_MAP ram to identify which A1SP local link should be mapped to each tributary. Since there are more SBI tributaries then Line Interface links, any tributaries, which are not mapped should be disabled.

If LINK_TYPn 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. In other words 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.

If the LINK_TYPn is set to DS3 mode then the UDF_HS mode bit should be set in the HS_LIN_REG in either A1SP0 or A1SP2.

Within an SPE that is configured for E1 or T1, the individual tributaries can be Structured with CAS, Structured without CAS, or Unstructured. Configuration of each tributary is controlled by setting the correct values in the TRIB_TYP field of the INS_TRIB_CTL and EXT_TRIB_CTL registers. If Structured with CAS mode is used, the SYNCH_TRIB must be set in EXT_TRIB_CTL register for that

tributary and the corresponding SYNC_LINK bit must be set in the SBI_SYNC_LINK register.

The link that is mapped to each tributary must also be configured so that the value of FR_STRUCT in LIN_STR_MODE memory register for that link is the same as the TRIB_TYP field for its corresponding tributary. SDF-FR mode is used when making a structured connection and CAS signaling is not being transported. SDF-MF mode is used when making a structured connection and CAS signaling is being transported. 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.

To control whether a particular tributary floats within its SPE or is locked, set the SYNCH_TRIB bit in the INS_TRIB_CTL/EXT_TRIB_CTL register to the appropriate value. This bit must be set if CAS is being transported.

Each link may either receive its timing from the SBI bus or control the tributary clock associated with that link. The CLK_MSTR bit in the Extract and Insert Tributary Control registers and the CLK_MODE bits in the Extract Tributary Control registers determine if how clocking is done with respect to the SBI for each tributary and consequently for the link associated with it.

In addition to whether or not the link is a clock master on the SBI, 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.

In the receive direction, the CLK_SOURCE_RX bit has two possible options. The function of this bit is a little different than in DLS mode. If this bit is set in SBI mode then the line receives its clock from the RL_CLK[n] pin. If this bit is not set then the line receives its clock from the SBI. This bit should only be set when supporting DS3 tributaries.

In the transmit direction, eight possible options exist and are controlled by the value of CLK_SOURCE_TX bits in the LIN_STR_MODE memory register for each line. The function of this field is a little different than in DLS mode. The eight options are:

000 Clock is an input on pin TL_CLK[n].

001 The receive clock is used. (loop timing mode).

010 Clock is internally synthesized in the CGC Block as a nominal E1 or T1 clock based on SYS_CLK and the value of T1_MODE.

011 Clock is internally synthesized in the CGC Block based on SRTS.

100 Clock is internally synthesized in the CGC Block using the adaptive algorithm.

101 Clock is internally synthesized based on values received on the CGC interface. (externally controlled)

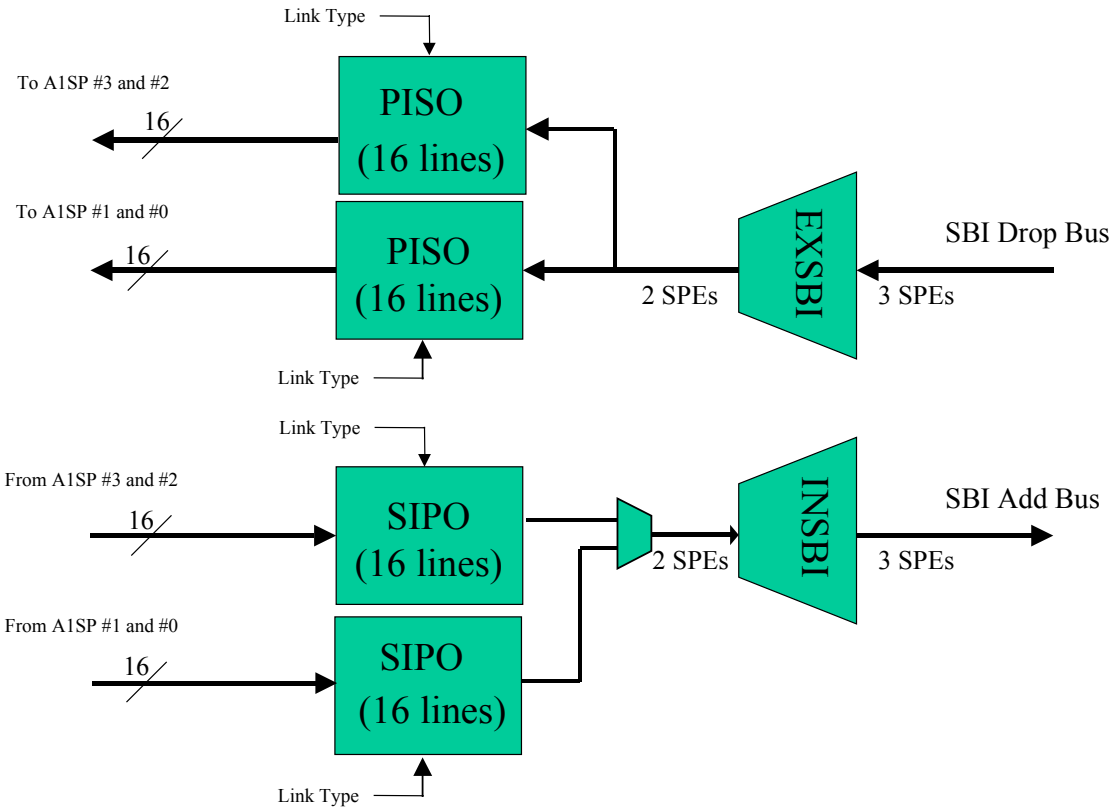
110 Not Valid in SBI mode

111 SBI is clock master, use SBI clock.

All options but the last require that the AAL1gator is the SBI clock master for that line in the Add direction.

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 81 for SBI Block architecture.

Figure 81 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.

11.6.3.1.9.1 Parallel In to Serial Out Converter (PISO)

The Parallel In to Serial Out Converter (PISO) serializes up to 16 T1, E1 links or 1 DS3 link which have been demapped from the SBI. The mode of this block is determined by LINK_TYPH and LINK_TYPL in SBI_LNK_CFG_REG. The SER_ENBL bit in each Extract Tributary Control Register enables the corresponding link within the PISO block. This block also performs the desynchronizer function to provide a low jitter T1, E1 serial clock and data for the A1SP blocks as well as generating a frame sync or multi-frame sync pulse and correct alignment of the signaling as the data is passed to the local links. The PISO can also take the RL_CLK serial clock as an input clock in DS3 mode.

11.6.3.1.9.1.1 Desynchronizer

11.6.3.1.9.1.1.1 E1/T1 Clock Generation

The Desynchronizer uses a combination of two clock generation techniques to desynchronize the demapped T1s and E1s. Incoming bit stuff events cause an extra bit of data to be generated or removed from the generated serial stream over the following 2KHz multi-frame. Pointer justifications are spread out by advancing or retarding the generated T1 or E1 clock phase.

REFCLK is used to generate an internal nominal 1.544Mb/s or 2.048Mb/s clock over the 2KHz frame as is indicated by the C1FP pin. A nominal T1 rate consists of 772 clocks in 500us. A nominal E1 rate consists of 1024 clocks in 500us. Stuff

events, as indicated by the SBI block, are compensated within the desynchronizer by generating three separate clocks to construct the faster or slower rate as shown in Table 16.

A mixture of T1 clock cycles is generated using 12 REFCLK cycles (Fast T1 Cycles) and 13 REFCLK cycles (Slow T1 Cycles) to produce an overall rate of 1.544MHz over the 500us period. A mixture of E1 clock cycles is generated using 9 REFCLK cycles (Fast E1 cycles) and 10 REFCLK cycles (Slow E1 cycles) to produce an overall rate of 2.048MHz over the 500us period. Table 16 shows the number of fast and slow cycles required to generate all three T1 and E1 rates.

Table 16 Desynchronizer E1/T1 Clock Generation Algorithm

Clock Rate	Fast T1 Cycles	Slow T1 Cycles	Overall T1 Cycles	Fast E1 Cycles	Slow E1 Cycles	Overall E1 Cycles
Slow	303	468	771	510	513	1023
Nominal	316	456	772	520	504	1024
Fast	329	444	773	530	495	1025

SBI Pointer justification events, are compensated within the desynchronizer by advancing or retarding the phase of the generated fast, slow and nominal clocks during the 2KHz period. Because pointer justification have a limited frequency of occurrence the phase adjustments are leaked out slowly. Twelve phase adjustments will remove or add an entire T1 clock whereas nine phase adjustments will remove or add an entire E1 clock. The number of phase adjustments needed per pointer justification is on average 89.077 for T1 or 65.829 for E1. These pointer adjustments are spread out over a 1 second period. Using the phase adjustments minimizes the jitter that is introduced.

11.6.3.1.9.2 Serial In to Parallel Out Converter (SIPO)

The Serial In to Parallel Out Converter (SIPO) accepts serial data from up to 16 T1/E1 sources or 1 DS3 source originating from the local links and converts these streams to byte serial format. The bytes are passed to the INSBI for insertion onto the SBI. SIPO can generate frame sync and multi-frame syncs as well as receive them. It also can generate a serial clock derived from the SBI or take in a serial clock synthesized by the A1SP block. When deriving a clock from the SBI timing this block performs a desynchronizer function to provide a low jitter serial clock. The desynchronizer circuit is the same as the one in the PISO logic.

11.6.3.1.9.3 Extract Scaleable Bandwidth Interconnect Block (EXSBI)

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 type of link is defined by the TRIB_TYP field in the Extract Tributary Control Register. The links, which the EXSBI processes can originate from any SPE but all links within the SPE must be of the same type. The SPE field in the SBI_BUS_CFG_REG and LINK fields in the Extract Tributary Control Register define which link is associated with that tributary. Note that link/tributary numbering starts from '1' in SBI mode.

All links extracted from the SBI bus can be timed from the SBI source or from the local link side, but in general timing is provided from the SBI source. When timing is from the SBI source (CLK_MSTR = '0') the EXSBI controls the clock by monitoring the SBI FIFO depth or from timing link rate adjustments provided from the source and carried with the links over the SBI bus. The method chosen is dependent on the value of CLK_MODE in the Extract Tributary Control Register.

When the local link is the clock master for a link (CLK_MSTR = '1'), clocks for the link are controlled by the local link clock. Based on buffer fill levels, the EXSBI sends link rate adjustment commands to the source indicating that it should send one additional or one fewer bytes of data during the next 500 uS interval. However this case is used only in DS3 mode, where the source side and the link side of the SBI are both using the same clock, no adjustments will need to be made other than synchronizing the data across the SBI bus

If clock timing is not being handled correctly, overruns or underruns will result. If an underrun or overrun occurs a maskable interrupt is generated and the related bit is set in the Extract Overrun or Extract Underrun Register. 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. In addition a maskable interrupt can be generated if a parity error is detected on the SBI bus. Status for this error is reported in the SBI Parity Error Interrupt Reason Register. Any EXSBI errors will cause a bit to be set in the SBI Interrupt Register which also causes a bit to be set in the Master Interrupt Register.

Also reported are C1FP slips and buffer depth errors.

11.6.3.1.9.4 Insert Scaleable Bandwidth Interconnect Block (INSBI)

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 type of link is defined by the TRIB_TYP field in the Insert Tributary Control Register. 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. The SPE field in the SBI_BUS_CFG_REG and LINK fields in the Insert Tributary Control Register define which link is associated with that tributary. Note that link/tributary numbering starts from '1' in SBI mode.

All links inserted into the SBI bus can be timed from the SBI or from the local links. For synchronous services timing is usually obtained from the SBI. For unstructured services where adaptive clocking or SRTS is used, timing is controlled by the local links. When timing is from the SBI (CLK_MSTR = '0') the INSBI controls the clock by monitoring the SBI FIFO depth.

When the local link is the clock master for a link (CLK_MSTR = '1'), clocks for the link are controlled by the SIPO. Based on buffer fill levels, the INSBI will speed up or slow down the rate at which data is played out onto the SBI by adding or deleting an extra byte of data over a 500 uS interval. The INSBI also sends link rate adjustment information across the SBI. This information is used by the sink side to create a recovered link clock which is based on small clock phase adjustments signaled by the source.

If clock timing is not being handled correctly, overruns or underruns will result. If an underrun or overrun occurs a maskable interrupt is generated and the related bit is set in the Insert Overrun or Insert Underrun Register. 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. Any INSBI errors will cause a bit to be set in the SBI Interrupt Register which also causes a bit to be set in the Master Interrupt Register.

11.6.3.2 H-MVIP Block

This section defines how the Line Interface functions in H-MVIP Mode.

The H-MVIP block supports 8 lines in each direction of 8Mbps H-MVIP formatted data.

In this mode, the H-MVIP block takes 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 (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 blocks.

A common frame pulse F0B is also used for all external lines. Individual local link frame pulses are derived from F0B. This signal is always an input.

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.

In this mode the SBI portions of the Line Interface Block are not used.

The type of line is selected based on the value of T1_MODE in the LIN_STR_MODE memory register for each line. For H-MVIP mode T1_MODE must be off.

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

00 Reserved

01 SDF-FR

10 not valid in H-MVIP mode

11 SDF-MF

SDF-FR mode is used when making a structured connection and CAS signaling is not being transported. SDF-MF mode is used when making a structured connection and CAS signaling is being transported. 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.

Because H-MVIP lines all run off the same clock, there is only one mode of clocking 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.

11.6.3.3 High Speed Mode

In high speed mode external lines 0, and 2 are connected to local links 0 and 16 respectively. The Line Interface Block just passes the data and clock between

the external lines and the corresponding local link. The data is passed as a clear channel bit stream and data rates are supported up to 45 Mbps.

Note that if a rate > 45 MHz is used, the SYS_CLK needs to be faster than 38.88 MHz. If the line rate is 52 MHz, SYS_CLK must be 45 MHz.

A1SP0 and A1SP2 should be configured in UDF-HS mode in the HS_LIN_REG memory register.

11.7 JTAG Test Access Port

The JTAG Test Access Port block provides JTAG support for boundary scan. The standard JTAG EXTEST, SAMPLE, BYPASS, IDCODE and STCTEST instructions are supported. The AAL1gator-32 identification code is 3122 hexadecimal.

12 MEMORY MAPPED REGISTER DESCRIPTION

The Chip SW_RESET state is automatically entered after a hardware reset is removed, or it can be asserted by setting the SW_RESET bit in the DEV_ID_REG. (Note memory cannot be accessed while the chip SW_RESET bit is set.) The A1SP SW_RESET state is automatically entered after a hardware reset, or chip SW_RESET, or it can be asserted by setting the An_SW_RESET bit in the An_CMD_REG for that A1SP.

The initial programming for the AAL1gator-32 is performed by loading the external memory with specified information while the An_SW_RESET bit is set, in the An_CMD_REG. (and the Chip SW_RESET is inactive) There is a separate register for each A1SP. After the memory is initialized, the CMD_REG_ATTN bit should be set so the configuration data can be read. Then An_SW_RESET can be removed. The device then reads the data structures from memory and enters the correct operating mode.

Word data structures have the first byte located at the low-byte end of the bus, which is also the location of the even data bytes (little endian implementation). The AAL1gator-32 supports 4 times the data as earlier versions of the AAL1gator. To facilitate existing software and to permit the use of smaller rams if not all lines are being used, the memory is arranged so that all data structures associated with lines 0-7 are grouped in the bottom quadrant of RAM, the next 8 lines are grouped in the next quadrant and so on.

Most AAL1gator control structures and all data buffers are stored in the external SSRAM. SSRAM accesses by the processor require the A[19] input to be equal to ‘0’. R_STATE_1, and R_LINE_STATE registers in the receive queue table are mapped to the memory address space but are actually located inside the chip to improve performance. 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.

The remaining registers which are located inside the chip are accessed when A[19] is high and A[18] is low. See the Normal Mode Register section for descriptions of all the internal registers.

The address map is as follows:

Table 17 AAL1gator-32 Memory Map

A[19:17]	Description
"000"	A1SP0 memory registers
"001"	A1SP1 memory registers
"010"	A1SP2 memory registers
"011"	A1SP3 memory registers
"10X"	Internal Normal Mode Registers
"11X"	Test Mode Registers

This section lists all the memory mapped registers and defines the bit fields of each register. The memory mapping for one A1SP is shown. The memory mapping for each A1SP is identical. The only difference is the value of A[18:17] which is used to select each A1SP memory space.

A[18:17] = "00" A1SP0

A[18:17] = "01" A1SP1

A[18:17] = "10" A1SP2

A[18:17] = "11" A1SP3

12.1 Initialization

The memory must be initialized to 0 (unless otherwise indicated) before the A1SP software reset is released, but the global software reset must be cleared before writes to memory can take place. A number of data structures used by the device in reserved areas depends on this initialization.

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.

12.2 A1SP and Line Configuration Structures

Table 18 A1SP and Line Configuration Structures Summary

Note the addresses listed below are the offsets within each A1SP address space.

Addr	Name	R/W	Org	Size	Description
0001_h	HS_LIN_REG	R/W	1 word	2 bytes	The High Speed Line Register provides overall mode information.
0010_h	LIN_STR_MODE_0	R/W	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.
0011_h	LIN_STR_MODE_1	R/W	1 word	2 bytes	
0012_h	LIN_STR_MODE_2	R/W	1 word	2 bytes	
0013_h	LIN_STR_MODE_3	R/W	1 word	2 bytes	
0014_h	LIN_STR_MODE_4	R/W	1 word	2 bytes	
0015_h	LIN_STR_MODE_5	R/W	1 word	2 bytes	
0016_h	LIN_STR_MODE_6	R/W	1 word	2 bytes	
0017_h	LIN_STR_MODE_7	R/W	1 word	2 bytes	

12.2.1 HS_LIN_REG

Organization: 1 word

Base address within A1SP: 1_h

Type: Read/Write

Function: Stores the configuration of the high speed line. Writes to this register will not take affect until the CMD_ATTN bit is set in the CMD_REG.

Format: Refer to the following table.

Field (Bits)	Description
Reserved (15:14)	Initialize to 0.
Not used (13:6)	Write with a 0 to maintain future software compatibility.
HS_GEN_DS3_AIS (5)	When the bit is set, Send cells with a DS3 framed “1010” pattern which is a DS3 AIS signal. (line 0 only).
HS_TX_COND (4)	Send cells with all 1s when in high-speed mode (line 0 only).
HS_RX_COND (3)	Fetch receive conditioning data from the R_COND_DATA buffer for line 0, channels 0 and 1. High-speed mode (line 0 only).
UDF_HS (2)	Line 0 is in the UDF-HS mode. 0 Disables the UDF-HS (T3/E3) mode. 1 Enables the UDF-HS (T3/E3) mode. If this mode is selected, the T_QUEUE_TBL and R_QUEUE_TBL entry index 0 are used. Initialize to the proper value. This bit works in conjunction with the T1_MODE bit in LIN_STR_MODE to enable playing out of framed DS3 AIS when in underrun.)
Unused (1:0)	Write with a 0 to maintain future software compatibility

12.2.1.1 LIN_STR_MODE

Organization: Eight 16-bit words.

Base address within A1SP: 10_h

Index: 1_h

Type: Read/Write

Function: Stores the per-line configuration. Writes to this memory register will not take affect until the CMD_ATTN bit is set in the CMD_REG.

Format: Refer to the following table.

Offset	Name	Description
0_h	LIN_STR_MODE_0	Line structure mode for line 0.
1_h	LIN_STR_MODE_1	Line structure mode for line 1.
2_h	LIN_STR_MODE_2	Line structure mode for line 2.
3_h	LIN_STR_MODE_3	Line structure mode for line 3.
4_h	LIN_STR_MODE_4	Line structure mode for line 4.
5_h	LIN_STR_MODE_5	Line structure mode for line 5.
6_h	LIN_STR_MODE_6	Line structure mode for line 6.
7_h	LIN_STR_MODE_7	Line structure mode for line 7.

LIN_STR_MODE Word Format

Field (Bits)	Description
LOW_CDV (15)	This bit is used in unstructured mode to decrease the CDV. If this bit is set then the cells are scheduled every 47 bytes, otherwise the scheduling is frame based. This mode cannot be used when the queue is configured for partial cells or for AAL0 mode. By default, a high speed queue is always configured for low CDV and this bit is ignored. 1 Unstructured line is in low CDV mode (byte based scheduling) 0 Unstructured line is not in low CDV mode (frame based scheduling)
REF_VAL_ENABLE (14)	This bit enables the generation of the reference value for this line when adding queues.

Field (Bits)	Description
T1_MODE (13)	Determines mode of the line 1 Line is in T1 mode. If HS_UDF is also set, the highspeed line will play out of framed DS3 AIS when in underrun. 0 Line is in E1 mode. Note: When using the SBI interface, T1/E1 granularity is at the DA1SP level, as chosen by the SBI LINK_TYPL and LINK_TYPH fields. The T1_MODE bits must be set to match these SBI fields.
E1_WITH_T1_SIG (12)	Enables T1 signaling while in E1 mode for this line. Signaling is updated every 24 frames instead of every 16 frames. This bit is valid only when the line is in E1 SDF-MF mode. AAL1 cell structures contain a signaling nibble every 25 bytes instead of every 17 bytes per single DS0. 1 Use T1 signaling. 0 Use E1 signaling. Note: E1_WITH_T1_SIG is not supported with the SBI interface.
HI_RES_SYNTH (11)	When this bit is set the clock synthesizer for this line is in high resolution mode. In high resolution mode the synthesizer has 256 distinct frequencies to use when recovering an E1 or T1 clock. When high resolution mode is disabled the synthesizer has 16 distinct frequencies. This bit should be set for adaptive clocking and must be off for SRTS.
Reserved (10)	Reserved
MF_ALIGN_EN (9)	When operating in SDF-MF mode, MF_ALIGN_MODE controls the MF alignment options: 0) only the CDVT value is taken into account when determining when to play out data. Normally this mode is used 1) MF_ALIGN_EN, The CDVT and the multiframe boundaries are taken into account, the data is aligned with the next MF boundary after CDVT.

Field (Bits)	Description
SHIFT_CAS (8)	Causes the CAS nibble to be shifted so as to be coincident with the upper nibble of data instead of with the lower nibble of data. 0) CAS nibble is coincident with the lower nibble of data 1) CAS nibble is coincident with the upper nibble of data
GEN_SYNC (7)	Enables the AAL1gator to generate TL_SYNC instead of receiving it as an input. Only valid in Direct Low Speed mode. 1) TL_SYNC is an output for this line 0) TL_SYNC is an input for this line

Field (Bits)	Description
CLK_SOURCE_TX (6:4)	Selects TL_CLK source. This value will override the setting defined by the TLCLK_OUTPUT_EN input. If switching from an external to an internal clock or visa versa, make sure there are not two clocks driving simultaneously. In Direct Low Speed mode all options are valid. In SBI mode all options except “110” are valid. In high speed mode the only valid options are “000” and “001”. The field is ignored in H-MVIP mode. 000 Use external clock. (TL_CLK is an input). 001 LOOPED – Use RL_CLK as the clock source. (not supported for DS3 over SBI) 010 NOMINAL Synthesized – Generate a clock of the nominal (T1 or E1) frequency from SYS_CLK. 011 SRTS Synthesized- Generate a T1/E1 clock frequency based on the received SRTS values. 100) ADAPTIVE Synthesized- uses receive buffer depth to generate a T1/E1 clock. 101) Externally controlled Synthesized: Generate a T1/E1 clock frequency based on the values provided by CGC_SER_D pin. This mode is used for external implementations of SRTS or Adaptive clocking. 110) Use common external clock (CTL_CLK) (only valid in low speed mode) 111) If in Direct Low Speed Mode, use common external clock (CTL_CLK) and drive TL_SIG data onto TL_CLK pin. If in SBI mode then use the nominal T1/E1 clock sourced by the SBI SIPO. This is required when this line sources data to an SBI INSBI Synchronous Tributary. Not valid in other modes. Note for Dual DS3 mode see important configuration note in High Speed Operations Section 14.2.4.4

Field (Bits)	Description
CLK_SOURCE_RX (3)	Selects RL_CLK source. 0 Use external clock. (RL_CLK is an input) or if in SBI mode then use clock derived from the SBI ref clock. 1 Use common external clock (CRL_CLK) as the clock source. If in SBI mode use RL_CLK from pins. (Only used when passing DS3 over SBI)
SRTS_EN (2)	Enable SRTS for this line. (Assert only for UDF-ML or UDF-HS). 1 The insertion of the transmit SRTS bits is enabled for this line and the received SRTS bits are accumulated.0 The CSI bits of the odd transmit AAL1 cells are set to 0 and the received SRTS bits are ignored.
FR_STRUCT (1:0)	Frame structure. 11 Enables SDF-MF. Signaling information is preserved. 01 Enables SDF-FR. No signaling information is preserved. 10 Enables UDF, clear channel, no channelization, no signaling. If this mode (UDF-ML) is selected, the T_QUEUE_TBL and R_QUEUE_TBL entries used are found at index = 32 x line. For example, line 3 uses T_QUEUE_TBL and R_QUEUE_TBL entry 96. 00 Not used. Initialize to the proper value.

12.3 Transmit Structures Summary

Table 19 Transmit Structures Summary

Note the addresses listed below are the offsets within each A1SP address space.

Note “A” in the Addr in the following table means A1SP digit:-(000x=A1SP0, 011x=A1SP3)

Name	R/W	Org	Size	Addr	Description
P_FILL_CHAR	R/W	1 word	2 bytes	0004_h	The empty bytes in a partially filled cell are filled with P_FILL_CHAR.
Reserved(AQ)	R/W	16 words	32 bytes	0030_h – $003F_h$	(Reserved (AQ))
T_SEQNUM_TBL	R/W	16 words	32 bytes	0020_h – $002F_h$	The Transmit Sequence Number Table is initialized according to a table.
T_COND_SIG	R/W	32 bytes x 8 lines	256 bytes	A 0400_h - A $047F_h$	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	0480_h- $04FF_h$	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	0700_h- $07FF_h$	Reserved (Frame Advance FIFO).
Reserved	R/W	8 x 128 x 2 words	4 Kbytes	0800_h- $0FFF_h$	Reserved (Transmit Calendar).
Reserved	R/W	8 x 256 bytes	2 Kbytes	1000_h- $13FF_h$	Reserved (Transmit Signaling Buffer).
T_OAM_QUEUE	R/W	2 x 32 words	128 bytes	1400_h- $143F_h$	The Transmit OAM Queue contains the OAM cells to be transmitted.
T_QUEUE_TBL	R/W	256 x 32 words	16 Kbytes	2000_h- $3FFF_h$	The Transmit Queue Table contains all pointers and variables that are queue-dependent.
Reserved	R/W	8 x 2 K words	32 Kbytes	4000_h- $7FFF_h$	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.

12.3.1 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.

12.3.2 T_SEQNUM_TBL

Organization: 16 words

Base address within A1SP: 20_h

Index: 1_h

Type: Read/Write

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

Initialization: Initialize to the values in the following table

Offset	Data Value
0_h	0000_h
1_h	0017_h
2_h	$002D_h$
3_h	$003A_h$
4_h	$004E_h$
5_h	0059_h
6_h	0063_h
7_h	0074_h
8_h	$008B_h$
9_h	$009C_h$
A_h	$00A6_h$
B_h	$00B1_h$
C_h	$00C5_h$
D_h	$00D2_h$
E_h	$00E8_h$
F_h	$00FF_h$

12.3.3 T_COND_SIG

Organization: 32 bytes x 8 lines

Base address within A1SP: 400_h

Index: 10_h

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
00000_h	T_COND_SIG_0	Transmit conditioned signaling for line 0.
00010_h	T_COND_SIG_1	Transmit conditioned signaling for line 1.
00020_h	T_COND_SIG_2	Transmit conditioned signaling for line 2.
00030_h	T_COND_SIG_3	Transmit conditioned signaling for line 3.
00040_h	T_COND_SIG_4	Transmit conditioned signaling for line 4.
00050_h	T_COND_SIG_5	Transmit conditioned signaling for line 5.
00060_h	T_COND_SIG_6	Transmit conditioned signaling for line 6.
00070_h	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 = ((channe– -1) / 2) + line x 16.
T_COND_SIG_B_H (10)	Transmit conditioned B bit for: Offset = ((channe– -1) / 2) + line x 16.
T_COND_SIG_C_H (9)	Transmit conditioned C bit or A bit if T1 SF for: Offset = ((channe– -1) / 2) + line x 16.
T_COND_SIG_D_H (8)	Transmit conditioned D bit or B bit if T1 SF for: Offset = ((channe– -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.

Field (Bits)	Description
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.

12.3.4 T_COND_DATA

Organization: 32 bytes x 8 lines

Base address within A1SP: 480_h

Index: 10_h

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 FF_h 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
00000_h	T_COND_DATA_0	Transmit conditioned data for line 0.
00010_h	T_COND_DATA_1	Transmit conditioned data for line 1.
00020_h	T_COND_DATA_2	Transmit conditioned data for line 2.
00030_h	T_COND_DATA_3	Transmit conditioned data for line 3.
00040_h	T_COND_DATA_4	Transmit conditioned data for line 4.
00050_h	T_COND_DATA_5	Transmit conditioned data for line 5.
00060_h	T_COND_DATA_6	Transmit conditioned data for line 6.
00070_h	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.

12.3.5 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: 01000_h

Index: 80_h

Type: Read/Write

Function: Stores the outgoing signaling data.

Figure 82 SDF-MF Format of the T_SIGNALING BUFFER

	Bit 15	Bit 0
Word 0	1	0
1	3	2
2	5	4
3	7	6
4	9	8
5	11	10
6	13	12
7	15	14
8	17	16
9	19	18
10	21	20
11	23	22
12	25	24
13	27	26
14	29	28
15	31	30

The upper nibble of each byte is 0.

12.3.6 T_OAM_QUEUE

Organization: 2 cells x 32 words

Base address within A1SP: 01400_h

Index: 20_h

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
01400_h	T_OAM_CELL_1	Transmit OAM cell 1.
01420_h	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.	

12.3.7 T_QUEUE_TBL

Organization: 256 x 32 words

Base address within A1SP: 2000_h

Index: 20_h

Type: Read/Write

Function: Configures the VCs.

Format: Each queue will be allocated 32 consecutive words.

Offset	Name	Description
0_h	Reserved	(Data pointer.) Initialize to FFFF each time this queue is initialized.
1_h	Not used	Initialize to 0 each time this queue is initialized to maintain future software compatibility.
2_h	T_COND_CELL_CNT	A 16-bit rollover count of conditioned cells transmitted.
3_h	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.
4_h	Not used	Initialize to 0 each time this queue is initialized to maintain future software compatibility.
5_h	Reserved	(Sequence number.) Initialize to 0 each time this queue is initialized.
6_h	QUEUE_CONFIG	The configuration of the current queue. Initialize to the proper value.
7_h	T_CELL_CNT	A 16-bit count of the cells transmitted.
8_h	TX_HEAD(1:2)	Header byte 1 in bits 15:8, header byte 2 in bits 7:0.
9_h	TX_HEAD(3:4)	Header byte 3 in bits 15:8, header byte 4 in bits 7:0.
A_h	TX_HEAD(5)	Header byte 5 (pre-calculated HEC) in bits 15:8.
B_h	QUE_CREDITS	A 10-bit quantity representing the number of byte credits accumulated for the queue.
C_h	CSD_CONFIG	Stores the average number of bytes in each cell, and carries the number of DS0s for this queue.
D_h	Not used	Initialize to 0 each time this queue is initialized to maintain future software compatibility.
E_h	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.
F_h	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.

Offset	Name	Description
10_h	T_CHAN_LEFT(15:0)	Initialize to the same value as T_CHAN_ALLOC(15:0).
11_h	T_CHAN_LEFT(31:16)	Initialize to the same value as T_CHAN_ALLOC(31:16).
12_h	TRANSMIT_CONFIG	Controls transmission of data.
13_h	T_CUR_ACT_CHAN (15:0)	(T_CUR_ACT_CHAN(15:0)). For DBCES mode, indicates which of the lower 16 channels is currently active. Initialize to 0 each time this queue is initialized.
14_h	T_CUR_ACT_CHAN (31:16)	(T_CUR_ACT_CHAN(31:16)). For DBCES mode, indicates which of the upper 16 channels is currently active. Initialize to 0 each time this queue is initialized.
15_h	T_NEW_ACT_CHAN (15:0)	(T_NEW_ACT_CHAN(15:0)). For DBCES mode, indicates which of the lower 16 channels is currently active in the new structure. Initialize to 0 each time this queue is initialized.
16_h	T_NEW_ACT_CHAN (31:16)	(T_NEW_ACT_CHAN(31:16)). For DBCES mode, indicates which of the upper 16 channels is currently active in the new structure. Initialize to 0 each time this queue is initialized.
17_h	CSD_BYTES_LEFT	(CSD_BYTES_LEFT) Only used in DBCES mode. This number is used to determine how many bytes are left in the structure after the first cell.
18_h-$1F_h$	Not used	Initialize to 0 each time this queue is initialized.

T_COND_CELL_CNT Word Format (02_H)

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 once, each time a cell with conditioned data is sent. If only signaling is conditioned this counter will not increment.

T_SUPPRESS_CNT Word Format (03_H)

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 (06_H)

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)	When set, this bit enables this queue. To enable a connection for this queue: • **1)Assert this bit.** • **2)Add this queue to the ADDQ_FIFO Register.** To disable a connection on this queue, 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, T1 SDF-FR, and E1_w_T1_sig modes, 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 AAL0 mode this field is set to 48 (in T1 mode this is encoded as 1 MF and 24 frames: 111000_B). 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 T1 SDF-MF single DS0 queues, encode the value 48 as one multiframe and 24 frames: 0x38. • 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. • FRAMES_PER_CELL is ignored for lines with the LOW_CDV bit set in the LIN_STR_MODE location.

Field (Bits)	Description
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. This bit can be ignored for AAL0 mode.
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. For all available queues configured as single-DS0, the minimum BYTES_PER_CELL is TBD.

T_CELL_CNT Word Format (07_H)

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 (08_H)

This word is maintained by the microprocessor. . Note: in UDF-HS mode, TALP reads this word only once, before it generates the first cell of the connection; as a result, any writes to this word after TALP generates the first cell will have no affect. TALP simply reads the TX_HEAD locations, then writes them into the UTOPIA FIFO during cell construction.

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 (09_H)

This word is maintained by the microprocessor. . Note: in UDF-HS mode, TALP reads this word only once, before it generates the first cell of the connection; as a result, any writes to this word after TALP generates the first cell will have no affect.

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 ($0A_H$)

This word is maintained by the microprocessor. . Note: in UDF-HS mode, TALP reads this word only once, before it generates the first cell of the connection; as

a result, any writes to this word after TALP generates the first cell will have no affect.

Field (Bits)	Description
TX_HEAD(5) (15:8)	Fifth header byte that contains the precalculated HEC word. TALP neither calculates nor verifies the HEC. Initialize to the proper value.
Not used (7:0)	Write with a 0 to maintain compatibility with future software versions.

QUE_CREDITS Word Format ($0B_H$)

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 00_b except in DBCES mode where this value must be calculated. The frame remainder field must be initialized correctly. The equation is as follows: Remainder = frame_diff MOD N (where N=2 for E1; N=3 for T1)
STR_PTR_SENT (13)	Indicates a structure pointer has been sent in the current set of 8 cells. Initialize to “0”.
TEST_SPDUP_NULL (12)	A test mode bit which speeds up the frequency of the generation of Null cells in DBCES mode to less than 4 ms in between cells. Initialize to “0”.
BITMASK_SENT (11)	Indicates the bitmask has been sent in the current set of 8 cells. Initialize to “0”.
FIRST_CELL_SCHED (10)	Indicates the first cell has been scheduled for this queue. Initialize to 0.

Field (Bits)	Description
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 the following in non-DBCES mode: UDF-ML: 178_H (47 x 8) SDF-FR, single DS0, no pointer: 178_H (47 x 8) SDF-FR (except above): 177_H (46.875 x 8) AAL0: 180_H (48 x 8) Partial Cells: #Bytes per cell x 8 SDF-MF, E1, single DS0: 187_H ((46.875 + 2) * 8) SDF-MF T1, single DS0: $17F_H$ ((46.875 + 1) * 8) SDF-MF E1, two DS0s: $17F_H$ ((46.875 + 1) * 8) SDF-MF (except above): 177_H (46.875 * 8) UDF-HS and Low_CDV mode: NOT USED For DBCES mode the credits written during setup must be accurate. The equations are as follows: Frame_diff = ROUNDUP(47/num_chan) Data_credits = frame_diff * num_chan Sig_per_1/8th = [ROUNDDOWN((num_chan+1)/2)] / 8 Sig_credits = ROUNDDOWN[(frame_diff/N) * sig_per_1/8th] (where N=2 for E1; N=3 for T1) Credit_reg = data_credits + sig_credits + 1 + bitmask_size (the 1 accounts for the structure pointer)

CSD_CONFIG Word Format ($0C_H$)

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.
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 non-DBCES SDF-FR or SDF-MF mode. When in DBCES mode initialize to 47. Initialize to 47 (0101111.000) for full cells when in UDF-ML mode. For AAL0 cells, this value is 48 (0110000.000). 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 ($0E_H$)

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 $FFFF_h$ for UDF-ML and UDF-HS modes.

T_CHANNEL_ALLOC(31:16) Word Format ($0F_H$)

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 $FFFF_h$ for UDF-ML and UDF-HS modes.

T_CHANNEL_LEFT(15:0) Word Format (10_H)

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 (11_H)

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 (12_H)

This word is maintained by the microprocessor. Note: in UDF-HS mode, TALP reads this word only once, before it generates the first cell of the connection; as a result, any writes to this word after TALP generates the first cell will have no affect.

Field (Bits)	Description
SUPPRESS_XMT (15)	Set to 1 to suppress the generation of cells for this queue. Cells are scheduled but not transmitted. Note that this bit is invalid in UDF-HS mode. A UDF-HS queue needs to be stopped by clearing the TX_ACTIVE bit and restarted by adding the queue.
LOOPBACK_ENABLE (14)	Set to 1 to loopback cell to receive side. Set VPI/VCI to corresponding receive queue number. Note to maximize throughput, this register is only read once in UDF-HS mode, when building the first cell. Therefore if loopback is desired for UDF-HS mode, this bit must be set first and cannot be changed after the queue is already running.

Field (Bits)	Description
AAL0_MODE_ENABLE (13)	Set to 1 to build AAL0 cells instead of AAL1. Use QUEUE_CREDITS=x0180, AVG_SUB_VALU=x0180, and BYTES_PER_CELL=x30 for full AAL0 cells.
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.

CSD BYTES LEFT Word Format (17_H)

This word is initialized by the microprocessor and then maintained by CSD. This register is only used for DBCES. This register should not be written after queue is started.

Field (Bits)	Description
BITMASK_CELL (15)	Set to 1 to build a cell with a bitmask. This bit must be initialized to 1 when starting a DBCES queue.

Field (Bits)	Description
SEQ_NUM (14:12)	The Sequence number of the next cell, used by CSD. Initialize to “000”.
NULL_BITMASK_CELL (11)	Set to 1 by CSD to send null cell. Initialize to “0”.
SCH_BUT_DONT_SEND_NULL (10)	Used internally by CSD to spread NULL cells. Initialize to 0.
CSD_BYTES_LEFT (9:0)	Only used by CSD 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.

12.3.8 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 line– - 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 = 2304 bytes). 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: 4000_h

Index(line): 800_h

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 + (channe– - 1) / 2. T1 offset = line x 2048 + multiframe x 512 + frame x 16 + (channe– - 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.

12.4 Receive Data Structures Summary

Table 25 lists the data structures unique to the receive side of the AAL1gator-32.

Note the addresses listed below are the offsets within each A1SP address space.

Name	Org	Size	Addr	Description
R_OAM_QUEUE_TBL	2 words	4 bytes	8000_h–8001_h	Receive OAM head and tail pointers.
R_OAM_CELL_CNT	1 word	2 bytes	8002_h	Count of received OAM cells.
R_DROP_OAM_CELL	1 word	2 bytes	8003_h	Count of dropped OAM cells.
Reserved	16 words	32 bytes	8020_h-$802F_h$	Reserved (SRTS Queue Pointers).
R_SRTS_CONFIG	2 bytes x 8 lines	16 bytes	8038_h-$803F_h$	Receive SRTS configuration.

Name	Org	Size	Addr	Description
R_CRC_SYNDROME	128 words	256 bytes	8080_h-$80FF_h$	Mask of bits. Initialized from a table.
R_CH_TO_QUE_TBL	128 words	256 bytes	8200_h-$827F_h$	Receive channel to queue table.
R_COND_SIG	16 x 8 bytes	256 bytes	8400_h-$847F_h$	Receive signaling conditioning values.
R_COND_DATA	32 x 8 bytes	256 bytes	8480_h-$84FF_h$	Receive data conditioning values.
Reserved	8 x 256 words	4 Kbytes	8800_h-$8FFF_h$	Reserved (Receive SRTS Queue).
Reserved	8 x 32 x 16 words	8 Kbytes	9000_h-$9FFF_h$	Reserved (Receive Signaling Buffer).
R_QUEUE_TBL	256 x 32 words	16 Kbytes	$A000_h$-$BFFF_h$	Receive queue table.
R_OAM_QUEUE	256 x 64 bytes	16 Kbytes	$E000_h$-$FFFF_h$	Receive OAM queue.
Reserved	8 x 512 x 32 bytes	128 Kbytes	$1\ 0000_h$-$1\ FFFF_h$	Reserved (Receive Data Buffer).

This section describes the structures used by the receive side of the AAL1gator-32.

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.

12.4.1 R_OAM_QUEUE_TBL

Organization: 2 words

Base address within A1SP: 8000_h

Index: 1_h

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 2^{56}th 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.

12.4.2 R_OAM_CELL_CNT

Organization: 1 word

Base address within A1SP: 8002_h

Index: 1_h

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.

12.4.3 R_DROP_OAM_CELL

Organization: 1 word

Base address within A1SP: 8003_h

Index: 1_h

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.

12.4.4 R_SRTS_CONFIG

Organization: 2 bytes x 8 lines

Base address within A1SP: 8038_h

Index: 1_h

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

Offset	Name	Description
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 (5:0)	Receive SRTS CDVT

12.4.5 R_CRC_SYNDROME

Organization: 128 words

Base address within A1SP: 8080_h

Index: 1_h

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 83 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 20 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
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

Sequence Number	Offset	Data (Hex)		Sequence Number	Offset	Data (Hex)
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
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

Sequence Number	Offset	Data (Hex)		Sequence Number	Offset	Data (Hex)
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

12.4.6 R_CH_TO_QUEUE_TBL

Organization: 128 words (8 lines x 32 DS0s)

Base address within A1SP: 8200_h

Index: 1_h

Type: Read/Write

Hardware Reset Value: 8080_h (this table is located inside device)

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-32 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-32 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.

Field(Bits)	Description
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 = (channe– - 1) / 2 + line x 16. For unstructured lines, set to the receive queue number mod 32.
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

Field(Bits)	Description
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.

12.4.7 R_COND_SIG

Organization: 16 words x 8

Base address within A1SP: 8400_h

Index: 10_h

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
00000_h	R_COND_SIG_0	Receive conditioned signaling for line 0.
00010_h	R_COND_SIG_1	Receive conditioned signaling for line 1.
00020_h	R_COND_SIG_2	Receive conditioned signaling for line 2.
00030_h	R_COND_SIG_3	Receive conditioned signaling for line 3.
00040_h	R_COND_SIG_4	Receive conditioned signaling for line 4.
00050_h	R_COND_SIG_5	Receive conditioned signaling for line 5.
00060_h	R_COND_SIG_6	Receive conditioned signaling for line 6.
00070_h	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 = (channe– - 1) / 2 + line x 16.
R_COND_B_H (10)	Receive conditioned B signaling bit for: Offset = (channe– - 1) / 2 + line x 16.
R_COND_C_H (9)	Receive conditioned C signaling bit or A bit if T1 SF for: Offset = (channe– - 1) / 2 + line x 16.
R_COND_D_H (8)	Receive conditioned D signaling bit or B bit if T1 SF for: Offset = (channe– - 1) / 2 + line x 16.
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.

12.4.8 R_COND_DATA

Organization: 16 words x 8

Base address within A1SP: 8480_h

Index: 10_h

Type: Read/Write

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

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 FF 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
00000_h	R_COND_DATA_0	Receive conditioned data for line 0.
00010_h	R_COND_DATA_1	Receive conditioned data for line 1.
00020_h	R_COND_DATA_2	Receive conditioned data for line 2.
00030_h	R_COND_DATA_3	Receive conditioned data for line 3.
00040_h	R_COND_DATA_4	Receive conditioned data for line 4.
00050_h	R_COND_DATA_5	Receive conditioned data for line 5.
00060_h	R_COND_DATA_6	Receive conditioned data for line 6.
00070_h	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 = (channe– - 1) / 2 + line x 16.
R_COND_DATA_L (7:0)	Receive conditioned data for: Offset = channel / 2 + line x 16.

12.4.9 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: 8800_h

Index: 100_h

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.

12.4.10 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 receive 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: 9000_h

Index (line): 200_h

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 + (channe– - 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.

Field (Bits)	Description
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.

12.4.11 R_QUEUE_TBL

Organization: 256 x 32 words

Base address within A1SP: $A000_h$

Index: 20_h

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 21R_QUEUE_TBL Format

Offset	Name	Description
0_h	R_STATE_0	Cell receiver state 0.
1_h	R_MP_CONFIG	Bytes per cell and CDVT constant.
2_h	R_STATE_1	Cell receiver state 1.
3_h	R_LINE_STATE	Line state.
4_h	R_BUF_CFG	Receive maximum buffer size.
5_h	R_SEQUENCE_ERR	16-bit rollover count of SN errors.
6_h	R_INCORRECT_SNP	16-bit rollover count of cells with incorrect SNP.
7_h	R_CELL_CNT	16-bit rollover count of played out cells.
8_h	R_ERROR_STKY	Receive sticky bits.
9_h	R_TOT_SIZE	Total bytes in structure.
A_h	R_DATA_LAST	Number of signaling bytes in structure.

Offset	Name	Description
B_h	R_TOT_LEFT	Number of bytes remaining in the structure. Initialize to 0 each time this queue is initialized.
C_h	Not used	Initialize to 0 each time this queue is initialized.
D_h	R_SN_CONFIG	Configures sequence number processing algorithm.
E_h	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.
F_h	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.
10_h	Reserved (R_CHAN_LEFTL)	Initialize to 0 each time this queue is initialized.
11_h	Reserved (RCHAN_LEFTH)	Initialize to 0 each time this queue is initialized.
12_h	R_DROPPED_CELLS	16-bit rollover count of cells that were received but dropped. Initialize to 0.
13_h	R_UNDERRUNS	16-bit rollover count of the occurrences of underrun on this queue. Initialize to 0.
14_h	R_LOST_CELLS	16-bit rollover count of the number of lost cells for this queue. Initialize to 0.
15_h	R_OVERRUNS	16-bit rollover count of the occurrences of overrun on this queue. Initialize to 0.
16_h	R_PTR_REFRAMES	16-bit rollover count of the occurrences of pointer reframes. Initialize to 0.
17_h	R_PTR_PAR_ERR	16-bit rollover count of the occurrences of pointer parity errors. Initialize to 0.
18_h	R_MISINSERTED	16-bit rollover count of the occurrences of misinserted cells. Initialize to 0.
19_h	R_ROBUST_SN	Write pointer for robust SN processing
$1A_h$	Reserved (R_CHAN_ACTL)	Initialize to 0 each time this queue is initialized.
$1B_h$	Reserved (R_CHAN_ACTH)	Initialize to 0 each time this queue is initialized.
$1C_h$	R_RD_PTR_LAST	Read pointer for bit integrity through underrun
$1D_h$-$1F_h$	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 (00_H)

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

Field (Bits)	Description
R_DBCES_BM_INACT (15)	Indicates that currently there is an inactive bitmask on the queue. Initialize to 0.
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 (01_H)

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. Use R_BYTES_CELL=x30 for full AAL0 cells.
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-FR, T1 SDF-MF, and E1_WITH_T1_SIG, modes, 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 us periods. For Robust SN Processing, this field represents the R_CDVT desired plus the number of frames stored in the cell that is conditionally stored. The minimum recommended value is R_CDVT=2.

R_STATE_1 Word Format (02_H)

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.

Field (Bits)	Description
Reserved (PTRMMST) (13)	Initialize to 0 to maintain future software compatibility.
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 (03_H)

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 “9000”. This register is not used in UDF-HS mode.

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. Initialize to 1.
Reserved (11:9)	Initialize to 0 to maintain future software compatibility.

Field (Bits)	Description
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_BUF_CFG Word Format (04_H)

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. This bit can be ignored for AAL0 mode.
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 count integrity through underrun. Set to 0 for normal operation. This mode is not supported for UDF-HS mode.
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. The value of R_MAX_BUF should be equal to or greater than two times R_CDVT, or R_CDVT plus two times the number of frames required per cell, whichever is greater. If MF_ALIGN_EN is set in LIN_STR_MODE, then extra margin should be added for the additional multi-frame of data that may be present. Therefore, the value should be increased by 16 frames for E1 or 24 frames for T1. Also if DBCES is being used then 48 frames should be added to account for the DBCES buffer adjustment.

R_SEQUENCE ERROR Word Format (05_H)

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 (06_H)

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 (07_H)

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 (08_H)

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-32 receives a cell. This feature allows the processor to determine if the AAL1gator-32 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-32 has had its sticky word update interrupted.
CELL_RECEIVED (14)	A cell was received.

Field (Bits)	Description
DBCES_BIT_MASK_ERR (13)	There was a parity error in the DBCES Bit Mask.
PTR_RULE_ERROR (12)	There was a violation of a structure pointer generation rule. An eight-cell sequence begins with an SN=0 cell and ends with an SN=7 cell. This bit will be set if no structure pointer was received, more than one structure pointer was received, or an 'out of range' structure pointer was received in the sequence. This condition will be checked only in modes where a structure 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 in the R_BUF_CFG memory register was asserted.
POINTER_SEARCH (10)	A cell was dropped because a valid structure 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 an overrun, consecutive structure pointer mismatches, and consecutive sequence number errors.
SN_CELL_DROP (8)	A cell was dropped in accordance with an SN Algorithm (as specified in ITU-T Recommendation I.363.1). If Fast SN processing is used and the line is not in high speed mode, this bit will always be set for the first cell if NODROP_IN_START = 0. This bit will also be set for the first cell if ROBUST SN processing is enabled.
POINTER_RECEIVED (7)	A structure pointer was received.
PTR_PARITY_ERR (6)	A cell was received with a structure pointer parity error.
SRTS_RESUME (5)	An SRTS resume has occurred. A valid SRTS value was received and stored in the R_SRTS_QUEUE.

Field (Bits)	Description
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. The data carried in this cell will be played out after R_CDVT frames.
PTR_MISMATCH (2)	A cell was dropped because of a structure pointer mismatch. This event causes a forced underrun condition.
OVERRUN (1)	A cell was dropped due to overrun. The receive buffer depth exceeded R_MAX_BUF. This event causes a forced underrun condition.
UNDERRUN (0)	A cell was received while the queue was in underrun.

R_TOT_SIZE Word Format (09_H)

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 used only if DBCES_EN =1 or BITI_UNDERRUN=1.
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 DS0– - 1 For T1 SDF-MF or E1 with T1 Signaling: 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 ($0A_H$)

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 memory registers.
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 in the structure, minus one. (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, UDF-HS, and SDF-FR modes. R_DATA_LAST = RoundUp[# of DS0's/2] –1.

R_TOT_LEFT Word Format ($0B_H$)

This word is read-only and is maintained by RALP

Field (Bits)	Description
Not used (15:14)	Driven with a 0. Mask on reads to maintain future software compatibility.
R_DBCES_BM_IN_NEXT (13)	Indicates that a bitmask is expected in the next structure. Initialize to "0".
R_DBCES_BM_LEFT (12:11)	Total unprocessed bytes remaining in bit mask structure. Initialize to "0".

Field (Bits)	Description
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. Initialize to “0”.
R_TOT_LEFT (9:0)	Total bytes minus one remaining in the structure. Not used in UDF-ML or UDF-HS mode. Initialize to “0”.

R_SN_CONFIG Word Format ($0D_H$)

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”. Note: Do not set if DISABLE_SN is set. Only one of these bits can be set.
INSERT_DATA (6:5)	Controls the format of the data inserted for lost cells: $\mathbf{00_b}$ Insert xFF. $\mathbf{01_b}$ Insert data from R_CONDQ_DATA. $\mathbf{10_b}$ Insert old data from receive buffer. $\mathbf{11_b}$ 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, both “Fast SN Algorithm” and “Robust SN Algorithm” are disabled. RALP will neither drop nor insert cells due to SN errors, but will maintain both R_INCORRECT_SNP and R_INCORRECT_SN counters. This bit should be set to 0 for AAL0 mode.

Field (Bits)	Description
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 “000”$_b$ any received cell will be dropped. 1 When SN_STATE equals “000”$_b$ any received cell with valid SNP will be accepted. Note that in Robust SN processing this bit has no effect.
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 000_b, it is interpreted the same as 111_b, which means that up to seven cells will be inserted.

R_CHAN_ALLOC(15:0) Word Format ($0E_H$)

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 $FFFF_h$. (DS0 15 is in bit 15).

R_CHAN_ALLOC(31:16) Word Format ($0F_H$)

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 $FFFF_h$. (DS0 31 is in bit 15).

(10_H)

(11_H)

R_DROPPED_CELLS Word Format (12_H)

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. NOTE: This counter will always increment for the first cell received, if: a) FAST SN processing and NODROP_IN_START is set. b) ROBUST SN processing is enabled.

R_UNDERRUNS Word Format (13_H)

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. This counter does not count underruns which are the result of a FORCED_UNDERRUN.

R_LOST_CELLS Word Format (14_H)

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 (15_H)

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 (16_H)

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 (17_H)

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 (18_H)

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 (19_H)

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.
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.

($1A_H$)

($1B_H$)

R_RD_PTR_LAST Word Format ($1C_H$)

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.

Field (Bits)	Description
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.

12.4.12 R_OAM_QUEUE

Organization: 256 cells x 64 bytes

Base address within A1SP: $E000_h$

Index: 20_h

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
00000_h	R_OAM_CELL_0	Receive OAM cell 0
00010_h	R_OAM_CELL_1	Receive OAM cell 1
.	.	.
.	.	.
.	.	.
$01FFF_h$	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

Offset	Bits (15:8)	Bits (7:0)
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.

12.4.13 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: 10000_h

Index (line): 2000_h

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.

13 NORMAL MODE REGISTER DESCRIPTION

Normal mode registers are used to configure and monitor the operation of the AAL1gator-32. Internal Registers are selected when A[19] is high. Normal mode registers are selected when A[18] is low. Test registers are accessed when both A[19] and A[18] are high.

Notes on Normal Mode Register Bits:

1. 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.

2. 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.

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

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

5. 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.

The register memory map is arranged as follows:

Table 22 Register Memory Map

Address	Register Description
0x8000X	Command Registers
0x8010X	Ram Interface Registers
0x8012X	UTOPIA Interface Registers
0x80200-0x80FFF	Line Interface Registers
0x81000-0x812FF	Interrupt and Status Registers
0x82000-0x82FFF	Idle Channel Configuration and Status Registers
0x84000-0x84FFF	DLL Control and Status Registers

13.1 Command Registers

These registers provide the means to update line and chip configuration from memory, reset unused lines, and send OAM cells. The register to add queues is also located here. There is one register per A1SP block.

Table 23 Command Register Memory Map

Address	Register Description	Register Mnemonic
0x80000	Reset and Device ID	DEV_ID_REG
0x80010	Command Register for A1SP 0	A0_CMD_REG
0x80011	Command Register for A1SP 1	A1_CMD_REG
0x80012	Command Register for A1SP 2	A2_CMD_REG
0x80013	Command Register for A1SP 3	A3_CMD_REG
0x80020	Add Queue FIFO Register for A1SP 0	A0_ADDQ_FIFO
0x80021	Add Queue FIFO Register for A1SP 1	A1_ADDQ_FIFO
0x80022	Add Queue FIFO Register for A1SP 2	A2_ADDQ_FIFO
0x80023	Add Queue FIFO Register for A1SP 3	A3_ADDQ_FIFO
0x80030	Clock Configuration for A1SP 0	A0_CLK_CFG
0x80031	Clock Configuration for A1SP 1	A1_CLK_CFG
0x80032	Clock Configuration for A1SP 2	A2_CLK_CFG
0x80033	Clock Configuration for A1SP 3	A3_CLK_CFG

Register 0x80000: Reset and Device ID Register (DEV_ID_REG)

Bit	Type	Function	Default
15	R/W	SW_RESET	1
14:7	Rsvd	Unused	X
6:4	R	DEV_TYPE[2:0]	011
3:0	R	DEV_ID[3:0]	0001

DEV_ID[3:0]

The ID bits can be read to provide a binary number indicating the AAL1gator-32 feature version. These bits are incremented only if features are added in a revision of the chip.

DEV_TYPE[2:0]

The TYPE bits can be read to distinguish the AAL1gator-32 from the other members of the AAL1gator family of devices.

SW_RESET

When set, causes all of the device to be held in reset including all other registers. While set, the external SSRAM may not be accessed. This is a chip software reset.
0) Chip is active
1) Chip is in reset

Notes:

1) Software should ensure that the RUN bit in DLL_STAT_REG reads back a 1 before releasing the AAL1gator-32 from software reset.
2) Software should wait 2 clock periods of the slowest clock before attempting to write to any other register. Exception to this rule is the DLL register port.

Register 0x80010, … 13: A1SPn Command Register (An_CMD_REG)

Bit	Type	Function	Default
Bit 15	R	Unused	X
Bit 14	R	Unused	X
Bit 13	R	Unused	X
Bit 12	R	Unused	X
Bit 11	R	Unused	X
Bit 10	R	Unused	X
Bit 9	R	Unused	X
Bit 8	R	Unused	X
Bit 7	R	Reserved	0
Bit 6	R	Reserved	0
Bit 5	R/W	An_SW_RESET	1
Bit 4	R	Reserved	0
Bit 3	R/W	An_CMDREG_ATTN	0
Bit 2	R/W	An_SEND_OAM_A1	0
Bit 1	R/W	An_SEND_OAM_A0	0
Bit 0	R	Reserved	0

An_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.

An_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.

An_CMDREG_ATTN

When written with a 1, causes the device to read the HS_LIN_REG and the LIN_STR_MODE memory registers for the corresponding A1SP. Reads as a 0 when the operation is complete.

An_SW_RESET

When set, causes the corresponding A1SP to be held in reset. This bit also functions as a Queue reset in High Speed mode.

Register 0x80020, … 23 : A1SPn Add Queue FIFO Register (An_ADDQ_FIFO)

Bit	Type	Function	Default
15	RO	EMPTY	1
14	Rsvd	Unused	X
13:8	WO	OFFSET	XX_H
7:0	WO	QUEUE_NUM	XX_H

This register is the write port of a 64 word FIFO that is used to add a queue on a first come first serve basis.

QUEUE_NUM

The number of the queue being added. Note that this field is invalid on reads.

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 (See Add Queue description in Processor Interface section . Note that this field is invalid on reads.)

Note: In SDF-MF mode, OFFSET must be set equal to FRAMES_PER_CELL in QUEUE_CONFIG in order to insure that the first pointer generated has a value of 0.

Note: For SDF-MF DBCES queues, OFFSET must be set equal to FRAMES_PER_CELL.

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.

Register 0x80030, … 33 : A1SPn Clock Configuration Register (An_CLK_CFG)

Bit	Type	Function	Default
15:10	R	Unused	X
9	R/W	NCLK_DIV_EN	0
8:5	R/W	NCLK_DIV	0000_B
4:0	R/W	ADAP_FILT_SIZE	00000_B

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 (ie, $2^{ADAP\_FILT\_SIZE}$). 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"$_B$ or 16.

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). For instance if this field is 000 and NCLK_DIV_EN = '1' then the NCLK is divided by 2: If this field is "1111" then NCLK is divided by 32.

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.

13.2 RAM Interface Registers

This register controls the configuration of the RAM interface.

Table 24 RAM Interface Registers Memory Map

Address	Register Description	Register Mnemonic
0x80100	RAM Configuration Register	RAM_CFG_REG

Register 0x80100: RAM Configuration Register (RAM_CFG_REG)

Bit	Type	Function	Default
15:2	R/W	Unused	X
1	R/W	RAM_EVEN_PAR	0
0	R/W	SSRAM_ZBT_MODE	0

This register controls the configuration of RAM1 and, if it is used, RAM 2.

SSRAM_ZBT_MODE

When set to 0, the pipelined single-cycle deselect SSRAM protocol is used on the RAM interfaces allowing glueless connection to pipelined single-cycle deselect SSRAMs. When set to a 1, the pipelined ZBT SSRAM protocol is used allowing glueless connection to pipelined ZBT SSRAMS.

RAM_EVEN_PAR

The RAM_EVEN_PAR bit selects even or odd parity for the RAM I/F. When set to a 1, even parity is generated and checked. When set to a 0, odd parity is used.

13.3 UTOPIA Interface Registers

These registers control the configuration of the UTOPIA interface.

Table 25 UTOPIA Interface Registers Memory Map

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 Config Register	UI_SRC_ADD_CFG
0x80124	UTOPIA Sink Address Config Register	UI_SNK_ADD_CFG
0x80125	UTOPIA to UTOPIA Loopback VCI Register	UI_U2U_LOOP_VCI
0x80126	UTOPIA Source Polling Priority List Register	UI_SRC_POLL_LIST

Register 0x80120: UI Common Configuration Register (UI_COMN_CFG)

Bit	Type	Function	Default
15:5	R	Unused	X
4	R/W	VP_MODE_EN	0
3	R/W	SHIFT_VCI	0
2	R/W	VCI_U2U _LOOP	0
1	R/W	U2U_LOOP	0
0	R/W	UI_EN	0

This register controls the general configuration of the Utopia Interface

UI_EN

When set, enables the UTOPIA Interface in both directions. When this bit is cleared (disabled) all the UI logic is held in reset and all the UTOPIA FIFO's are cleared and all the UTOPIA outputs are tristated. The AAL1gator-32 will not respond in the UTOPIA interface when this bit is disabled. The registers are not affected by this bit.
0) UI is disabled
1) UI is enabled

Note: UI_EN must only be set AFTER all other UI interface registers are configured.

U2U_LOOP

When set, all cells received by UI are sent back out to the UI (regardless of single or multi-addressing mode).
0) UI in normal mode
1) UI in remote loopback mode

VCI_U2U_LOOP

When set, all cells received by UI with a VCI which matches the VCI programmed in the U2U_LOOP_VCI register are sent back out by Utopia. To avoid any momentary corruption of data, this bit should only be changed when UI_EN is disabled.
0) UI in normal mode
1) UI in VCI based remote loopback mode

SHIFT_VCI

Selects the VCI Address range used for mapping to queue numbers. This bit only controls the reception of cells. This field is not used if VP_MODE_EN is set.

0) Will use VCI(10:9) to select A1SP and VCI(7:0) as the queue number if VCI(8) = 1.

2) Will use VCI(14:13) to select A1SP and VCI(11:4) as the queue number if VCI(12) = 1.

VP_MODE_EN

When set uses the VPI field for A1SP and Line selection and uses VCI field for OAM cell detection. This bit only controls the reception of cells. This bit can only be set if all lines are in UDF mode. In particular:

1) VPI[4:3] are used to select the A1SP if in single port mode.

2) VPI[2:0] selects line within an A1SP

3) If VCI <= 31 then interpret cell as OAM cell and place in OAM buffer.

4) Queue 0 will be assumed and no VCI bits need to be used to indicate queue number

Note that VP_MODE_EN should only be set in the Utopia 1 or Utopia 2 single address modes.

Register 0x80121: UI Source Config Reg (UI_SRC_CFG)

Bit	Type	Function	Default
15:6	R	Unused	X
5	R/W	CS_MODE_EN	0
4	R/W	16_BIT_MODE	0
3	R/W	EVEN_PAR	0
2	R/W	ANY-PHY_EN	0
1:0	R/W	UTOP_MODE	00

This register controls the source side configuration of the Utopia Interface

UTOP_MODE(1:0)

Selects the UTOPIA operating mode for the source side interface:
00 Utopia-1 Master
01 Utopia-1 Slave
10 Utopia-2 Single Address Slave
11 Reserved

ANY-PHY_EN

Enables Any-PHY mode for the source side interface:
0) UTOPIA mode. (Use UTOP_MODE for UTOPIA type)
1) Any-PHY mode.

EVEN_PAR

Determines the calculated parity across data bytes/words sent out of the source interface.
0) Odd parity
1) Even parity.

16_BIT_MODE

When set, UI source side interface operates in 16-bit mode.
0) 8-bit mode
1) 16-bit mode

CS_MODE_EN

When set, RPHY_ADDR(3)/RCSB input pin is used as a chip select (RCSB) for the source side interface, when clear RPHY_ADDR(3)/RCSB is used as

an address bit (RPHY_ADDR(3)). This bit should only be set in Any-PHY mode.

0) RPHY_ADDR(3)/RCSB input is used as RPHY_ADDR(3).
1) RPHY_ADDR(3)/RCSB input is used as RCSB

Register 0x80122: UI Sink Config Reg (UI_SNK_CFG)

Bit	Type	Function	Default
15:6	R	Unused	X
5	R/W	CS_MODE_EN	0
4	R/W	16_BIT_MODE	0
3	R/W	EVEN_PAR	0
2	R/W	ANY-PHY_EN	0
1:0	R/W	UTOP_MODE	00

This register controls the sink side configuration of the Utopia Interface

UTOP_MODE(1:0)

Selects the operating mode for the sink side interface:
00 Utopia-1 Master
01 Utopia-1 Slave
10 Utopia-2 Single Address Slave
11 Utopia-2 Multi-Address Slave

ANY-PHY_EN

Enables Any-PHY mode for sink side interface.
0) UTOPIA mode. (Use UTOP_MODE for UTOPIA type)
1) Any-PHY mode.

EVEN_PAR

Determines the checked parity across data bytes/words received by the sink interface.
0) Odd parity
1) Even parity.

16_BIT_MODE

When set, UI sink side interface operates in 16-bit mode.
0) 8-bit mode
1) 16-bit mode

CS_MODE_EN

When set, TPHY_ADDR(3)/TCSB input pin is used as a chip select (TCSB) for the sink side interface, when clear TPHY_ADDR(3)/TCSB is used as a

regular address bit (TPHY_ADDR(3)). This bit should only be set in Any-PHY mode.

0) TPHY_ADDR(3)/TCSB input is used as TPHY_ADDR(3)
1) TPHY_ADDR(3)/TCSB input is used as TCSB

Register 0x80123: Slave Source Address Config Register (UI_SRC_ADD_CFG)

Bit	Type	Function	Default
15:0	R/W	CFG_ADDR	0000_H

CFG_ADDR(15:0)

These bits contain the configured slave address used for Utopia-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 Table 27 for details. In the source direction the AAL1gator is always one address.

Table 26 CFG_ADDR and PHY_ADDR Bit Usage in SRC direction

	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 UTOPIA-2 Multi-Addr mode is only an option in the SNK direction.
- 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='1' then CFG_ADDR[4:3] = "00".
- In Any-PHY mode, if CS_MODE_EN='0' then CFG_ADDR[4]="0".

Register 0x80124: Slave Sink Address Config Register (UI_SNK_ADD_CFG)

Bit	Type	Function	Default
15:0	R/W	CFG_ADDR	0000_H

CFG_ADDR(15:0)

These bits contain the configured slave address used for Utopia-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 Table 27 for details.

Table 27 CFG_ADDR and PHY_ADDR Bit Usage in SNK direction

	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 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 addr is prepended	[15:2]=device

Notes:

- In Any-PHY mode, if CS_MODE_EN='1' then CFG_ADDR[4:3] = "00".else if CS_MODE_EN='0' then CFG_ADDR[4]="0".
- 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.

Register 0x80125: UI to UI Loopback VCI (U2U_LOOP_VCI)

Bit	Type	Function	Default
15:0	R/W	U2U_LOOP_VCI	0000_H

U2U_LOOP_VCI(15:0)

If VCI_U2U_LOOP is set in the UI_COMN_CFG_REG, any cell received from the UI bus, with a VCI which matches this programmed VCI, will be sent back out to the UI bus. To avoid any momentary corruption of incoming cell data, the value of this register should only be changed only when VCI_U2U_LOOP in UTO_CFG_REG is disabled and UI_EN in UTO_CFG_REG s disabled.

Register 0x80126: UI Source Polling Priority List Register (UI_SRC_POLL_LIST)

Bit	Type	Function	Default
15	R	unused	0
14:12	R/W	Priority 4 Init Value	100_B
11:9	R/W	Priority 3 Init Value	011_B
8:6	R/W	Priority 2 Init Value	010_B
5:3	R/W	Priority 1 Init Value	001_B
2:0	R/W	Priority 0 Init Value	000_B

This register is to be configured while the UI interface is held in reset.

The register is used to assign an initial priority to a particular A1SP or the loopback FIFO for the source side polling mechanism. However, this is only the initial priority for the first cell transfer and the subsequent priority is determined by a Least Recently Serviced algorithm. The LRS algorithm rotates through the assigned sources defined in this register, thus by changing the default IDs and entering a particular source entry twice, for example, one source may be given a greater opportunity for a source side cell transfer than another source (ie, 2/5 vs 1/5) as long as another third source is left out. This allows an A1SP configured for high bandwidth operation to get more frequent servicing than lower bandwidth A1SPs assuming an idle A1SP is left out. **Note that if a source does not have a source ID configured in this register, it will never be serviced.** If all four A1SP sources and the loop back FIFO are to be used, this register can be left at it's default value.

Each 3-bit Priority Init sub-field is configured with a value that corresponds to a particular A1SP source or the Loop Back FIFO source according to the following table:

Priority Init Value ID	Corresponding Source
000	A1SP0
001	A1SP1
010	A1SP2
011	A1SP3
100	Loop Back FIFO

13.4 Line Interface Registers

These registers control the configuration and report status for the Line Interface. Gaps within the address space are reserved and should not be read or written. The Line interface registers are broken into the following sections:

Table 28 Line Interface Register Memory Map Summary

Address	Register Description
0x802XX	General Line Configuration Registers
0x803XX	General SBI Registers
0x804XX	SBI Extract Registers
0x805XX	SBI Insert Registers

13.5 Direct Low Speed Mode Registers

These registers are selected when the address = 0x802XX.

Table 29 Direct Low Speed Mode Register Memory Map

Offset	Register Description	Register Mnemonic
0x00 – 0x0F	Low Speed Line Configuration Registers	LS_Ln_CFG_REG
0x10	Line Mode Register	LINE_MODE_REG

Register 0x80200H, 01H … 0FH: Low Speed Line n Configuration Registers(LS_Ln_CFG_REG)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	0
Bit 6	Rsvd	Unused	0
Bit 5	Rsvd	Unused	0
Bit 4	Rsvd	Unused	0
Bit 3	Rsvd	Unused	0
Bit 2	Rsvd	Unused	0
Bit 1	R/W	MVIP_EN	0
Bit 0	R/W	MF_SYNC_MODE	0

This register configures how independent lines are handled by the Line Interface Block in Direct Low Speed Mode..

MF_SYNC_MODE

Controls if the line sync signal (RL_SYNC and TL_SYNC) function as frame sync signals or multi-frame sync signals.
1) Sync signals are multi-frame sync signals
0) Sync signals are frame sync signals

MVIP_EN

When set selects the MVIP-90 format for external line data instead of the normal format. On read:
1) This line is in MVIP-90 mode
0) This line is in normal mode

Register 0x80210H: Line Mode Register(LINE_MODE_REG)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	0
Bit 6	Rsvd	Unused	0
Bit 5	Rsvd	Unused	0
Bit 4	Rsvd	Unused	0
Bit 3	Rsvd	Unused	0
Bit 2	Rsvd	Unused	0
Bit 1:0	RO	LINE_MODE	Depends on value of LINE_MODE pins

This register indicates the line mode of the device...

LINE_MODE

This field shows the values of the LINE_MODE pins. It is read only.

00) Direct Low Speed Mode
01) SBI Mode
10) H-MVIP Mode
11) High Speed Mode

13.6 SBI Mode Registers

13.6.1 General SBI Registers

Table 30 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

Register 0x80300H: SBI Bus Configuration Register(SBI_BUS_CFG_REG)

Bit	Type	Function	Default
Bit 15	R/W	BUSMASTER	0
Bit 14	R/W	TWO_C1FP_EN	0
Bit 13	R/W	TS_EN	0
Bit 12	R/W	CLK_MSTR	0
Bit 11	R/W	SPE3_SYNCH	0
Bit 10	R/W	SPE2_SYNCH	0
Bit 9	R/W	SPE1_SYNCH	0
Bit 8	R/W	SPE3_ENBL	0
Bit 7	R/W	SPE2_ENBL	0
Bit 6	R/W	SPE1_ENBL	0
Bit 5	R/W	SPE3_TYP[1]	0
Bit 4	R/W	SPE3_TYP[0]	0
Bit 3	R/W	SPE2_TYP[1]	0
Bit 2	R/W	SPE2_TYP[0]	0
Bit 1	R/W	SPE1_TYP[1]	0
Bit 0	R/W	SPE1_TYP[0]	0

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

Note that changing the values of any of these bits will cause a configuration reset on all affected tributaries.

SPEn_TYP[1:0]

The SPEn_TYP fields identify the payload type of each SPE. The encoding for SPEn_TYPE is:

Payload Type	SPEn_TYP Value
T1	b”00”
E1	b”01”
DS3	b”10”
Reserved	b”11”

SPEn_ENBL

The SPEn_ENBL field is used to enable or disable an entire SPE on the SBI. When high the SPE is enabled. All SPEs default to being disabled. Important Note: An SPE must not be enabled without any tributaries enabled on it. When initially activating an SPE, enable the tributaries first and then enable the SPE. When deactivating an SPE, disable the SPE first and then deactivate the tributaries.

SPEn_SYNCH

The SPEn_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. When SPE1_SYNCH is ‘1’ all tributaries in SPE1 will be forced to be in synchronous mode. When ‘0’ the value of the TRIB_CTL settings will be used.

CLK_MSTR

The CLK_MSTR field is used as an override mechanism for configuring all INSBI tributaries to be clock masters irrespective of the INSBI CLK_MSTR settings. When ‘1’ all INSBI tributaries will be forced to be clock masters. When ‘0’ the value of the INSBI CLK_MSTR settings will be used. These default so that all INSBI tributaries are clock slaves.

TS_EN

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.

If this bit is a zero, then the 16 internal links of A1SP0 and A1SP1 are mapped to the lowest 16 tributaries of SPE1 on the SBI bus, and the 16 internal links of A1SP2 and A1SP3 are mapped to the lowest 16 tributaries of SPE2 on the SBI bus.

If this bit is a one, then the values of the TS_EN bits in the EXT_CTL and INS_CTL registers will be used.

TWO_C1FP_EN

The TWO_C1FP_EN allows a separate C1FP for the SBI Add bus and Drop bus. This can be used when the two buses are offset from each other. When TWO_C1FP_EN is set then the C1FP pin is used as the C1FP pulse for the Drop bus, and the C1FP_ADD pin is used as the C1FP pulse for the Add bus. When TWO_C1FP_EN is low, then the C1FP pin is used as the C1FP pulse for both the Add and Drop bus.

BUSMASTER

The BUSMASTER bit controls whether or not the AAL1gator32 drives the SBI bus when no one else is driving the bus. This includes SBI overhead bytes and disabled tributaries. If BUSMASTER is set and the DETECT signal is low (inactive), then it means no one is driving the SBI bus and the AAL1gator32 will drive the bus. This will prevent parity errors from being detected due to a tristated bus.

Note: ACTIVE is only driven for enabled tributaries and therefore is not driven for bytes that would only be driven when BUSMASTER is set. Only one link layer device on the SBI bus should have this feature enabled. If more than one device has this feature enabled, then bus collisions can occur, which would result in parity errors.

Register 0x80301H: SBI Link Configuration Register(SBI_LNK_CFG_REG)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	R/W	CKOUT_KILLH	0
Bit 11	R/W	Reserved	0
Bit 10	R/W	Reserved	0
Bit 9	R/W	LINK_TYPH[1]	0
Bit 8	R/W	LINK_TYPH[0]	0
Bit 7	Rsvd	Unused	0
Bit 6	Rsvd	Unused	0
Bit 5	Rsvd	Unused	0
Bit 4	R/W	CKOUT_KILLL	0
Bit 3	R/W	Reserved	0
Bit 2	R/W	Reserved	0
Bit 1	R/W	LINK_TYPL[1]	0
Bit 0	R/W	LINK_TYPL[0]	0

The AAL1gator III 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.

This register 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.

LINK_TYPL

The LINK_TYPL field defines the link type for the lower group of links. The encoding of this field is as follows.

Payload Type	LINK_TYP Value
T1	b”00”
E1	b”01”
DS3	b”10”
Reserved	b”11”

CKOUT_KILLL

The CKOUT_KILLL field is used to disable the clock on all the lower 16 links. Setting this bit to a one disables the clocks.

LINK_TYPH

The LINK_TYPH field defines the link type for the upper internal SBI processor. The encoding of this field is the same as LINK_TYPL.

CKOUT_KILLH

The CKOUT_KILLH field is used to disable the clock on all the upper 16 links. Setting this bit to a one disables the clocks.

Register 0x80302H: SBI Link Disable Register Low(SBI_LINK_DIS_REGL)

Bit	Type	Function	Default
Bit 15	R/W	LINK_DIS[15]	0
Bit 14	R/W	LINK_DIS[14]	0
Bit 13	R/W	LINK_DIS[13]	0
Bit 12	R/W	LINK_DIS[12]	0
Bit 11	R/W	LINK_DIS[11]	0
Bit 10	R/W	LINK_DIS[10]	0
Bit 9	R/W	LINK_DIS[9]	0
Bit 8	R/W	LINK_DIS[8]	0
Bit 7	R/W	LINK_DIS[7]	0
Bit 6	R/W	LINK_DIS[6]	0
Bit 5	R/W	LINK_DIS[5]	0
Bit 4	R/W	LINK_DIS[4]	0
Bit 3	R/W	LINK_DIS[3]	0
Bit 2	R/W	LINK_DIS[2]	0
Bit 1	R/W	LINK_DIS[1]	0
Bit 0	R/W	LINK_DIS[0]	0

LINK_DISn

The Link Disable n (LINK_DISn) bit allows individual serial SBI local links to be disabled. When set the link is disabled. When clear the link is enabled. All serial links default to being enabled.

Register 0x80303H: SBI Link Disable Register High (SBI_LINK_DIS_REGH)

Bit	Type	Function	Default
Bit 15	R/W	LINK_DIS[31]	0
Bit 14	R/W	LINK_DIS[30]	0
Bit 13	R/W	LINK_DIS[29]	0
Bit 12	R/W	LINK_DIS[28]	0
Bit 11	R/W	LINK_DIS[27]	0
Bit 10	R/W	LINK_DIS[26]	0
Bit 9	R/W	LINK_DIS[25]	0
Bit 8	R/W	LINK_DIS[24]	0
Bit 7	R/W	LINK_DIS[23]	0
Bit 6	R/W	LINK_DIS[22]	0
Bit 5	R/W	LINK_DIS[21]	0
Bit 4	R/W	LINK_DIS[20]	0
Bit 3	R/W	LINK_DIS[19]	0
Bit 2	R/W	LINK_DIS[18]	0
Bit 1	R/W	LINK_DIS[17]	0
Bit 0	R/W	LINK_DIS[16]	0

LINK_DISn

The Link Disable n (LINK_DISn) bit allows individual serial SBI local links to be disabled. When set the link is disabled. When clear the link is enabled. All serial links default to being enabled.

Register 0x80304H: SBI SYNC Link Register Low(SBI_SYNC_LINK_REGL)

Bit	Type	Function	Default
Bit 15	R/W	SYNC_LINK[15]	0
Bit 14	R/W	SYNC_LINK[14]	0
Bit 13	R/W	SYNC_LINK[13]	0
Bit 12	R/W	SYNC_LINK[12]	0
Bit 11	R/W	SYNC_LINK[11]	0
Bit 10	R/W	SYNC_LINK[10]	0
Bit 9	R/W	SYNC_LINK[9]	0
Bit 8	R/W	SYNC_LINK[8]	0
Bit 7	R/W	SYNC_LINK[7]	0
Bit 6	R/W	SYNC_LINK[6]	0
Bit 5	R/W	SYNC_LINK[5]	0
Bit 4	R/W	SYNC_LINK[4]	0
Bit 3	R/W	SYNC_LINK[3]	0
Bit 2	R/W	SYNC_LINK[2]	0
Bit 1	R/W	SYNC_LINK[1]	0
Bit 0	R/W	SYNC_LINK[0]	0

SYNC_LINKn

The SYNC Link n (SYNC_LINKn) 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 SPEn_SYNCH is set in the SBI_BUS_CFG register or SYNCH_TRIB is set in the corresponding Insert Tributary Control Register. This bit must be set if CAS is carried on this link.

Register 0x80305H: SBI SYNC Link High Register(SBI_SYNC LINKH_REG)

Bit	Type	Function	Default
Bit 15	R/W	SYNC_LINK[31]	0
Bit 14	R/W	SYNC_LINK[30]	0
Bit 13	R/W	SYNC_LINK[29]	0
Bit 12	R/W	SYNC_LINK[28]	0
Bit 11	R/W	SYNC_LINK[27]	0
Bit 10	R/W	SYNC_LINK[26]	0
Bit 9	R/W	SYNC_LINK[25]	0
Bit 8	R/W	SYNC_LINK[24]	0
Bit 7	R/W	SYNC_LINK[23]	0
Bit 6	R/W	SYNC_LINK[22]	0
Bit 5	R/W	SYNC_LINK[21]	0
Bit 4	R/W	SYNC_LINK[20]	0
Bit 3	R/W	SYNC_LINK[19]	0
Bit 2	R/W	SYNC_LINK[18]	0
Bit 1	R/W	SYNC_LINK[17]	0
Bit 0	R/W	SYNC_LINK[16]	0

SYNC_LINKn

The SYNC Link n (SYNC_LINKn) 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 SPEn_SYNCH is set in the SBI_BUS_CFG register or SYNCH_ TRIB is set in the corresponding Insert Tributary Control Register. . This bit must be set if CAS is carried on this link.

Register 0x80308H: SBI Extract Bus Alarm Interrupt Register Low (EXT_ALRM_INTL)

Bit	Type	Function	Default
Bit 15	R2C	SBI_ALRM_INT[15]	0
Bit 14	R2C	SBI_ALRM_INT [14]	0
Bit 13	R2C	SBI_ALRM_INT [13]	0
Bit 12	R2C	SBI_ALRM_INT [12]	0
Bit 11	R2C	SBI_ALRM_INT [11]	0
Bit 10	R2C	SBI_ALRM_INT [10]	0
Bit 9	R2C	SBI_ALRM_INT [9]	0
Bit 8	R2C	SBI_ALRM_INT [8]	0
Bit 7	R2C	SBI_ALRM_INT [7]	0
Bit 6	R2C	SBI_ALRM_INT [6]	0
Bit 5	R2C	SBI_ALRM_INT [5]	0
Bit 4	R2C	SBI_ALRM_INT [4]	0
Bit 3	R2C	SBI_ALRM_INT [3]	0
Bit 2	R2C	SBI_ALRM_INT [2]	0
Bit 1	R2C	SBI_ALRM_INT [1]	0
Bit 0	R2C	SBI_ALRM_INT [0]	0

SBI_ALRM_INTn

When set, SBI_ALRM_INTn 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.

Register 0x80309H: SBI Extract Bus Alarm Interrupt Register High (EXT_ALRM_INTH)

Bit	Type	Function	Default
Bit 15	R2C	SBI_ALRM_INT[31]	0
Bit 14	R2C	SBI_ALRM_INT [30]	0
Bit 13	R2C	SBI_ALRM_INT [29]	0
Bit 12	R2C	SBI_ALRM_INT [28]	0
Bit 11	R2C	SBI_ALRM_INT [27]	0
Bit 10	R2C	SBI_ALRM_INT [26]	0
Bit 9	R2C	SBI_ALRM_INT [25]	0
Bit 8	R2C	SBI_ALRM_INT [24]	0
Bit 7	R2C	SBI_ALRM_INT [23]	0
Bit 6	R2C	SBI_ALRM_INT [22]	0
Bit 5	R2C	SBI_ALRM_INT [21]	0
Bit 4	R2C	SBI_ALRM_INT [20]	0
Bit 3	R2C	SBI_ALRM_INT [19]	0
Bit 2	R2C	SBI_ALRM_INT [18]	0
Bit 1	R2C	SBI_ALRM_INT [17]	0
Bit 0	R2C	SBI_ALRM_INT [16]	0

SBI_ALRM_INTn

When set, SBI_ALRM_INTn 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_STATH 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.

Register 0x8030AH: SBI Extract Bus Alarm Status Register Low (EXT_ALRM_STAT_REGL)

Bit	Type	Function	Default
Bit 15	RO	SBI_ALRM_STAT [15]	0
Bit 14	RO	SBI_ALRM_STAT[14]	0
Bit 13	RO	SBI_ALRM_STAT[13]	0
Bit 12	RO	SBI_ALRM_STAT [12]	0
Bit 11	RO	SBI_ALRM_STAT [11]	0
Bit 10	RO	SBI_ALRM_STAT [10]	0
Bit 9	RO	SBI_ALRM_STAT [9]	0
Bit 8	RO	SBI_ALRM_STAT [8]	0
Bit 7	RO	SBI_ALRM_STAT [7]	0
Bit 6	RO	SBI_ALRM_STAT [6]	0
Bit 5	RO	SBI_ALRM_STAT [5]	0
Bit 4	RO	SBI_ALRM_STAT [4]	0
Bit 3	RO	SBI_ALRM_STAT [3]	0
Bit 2	RO	SBI_ALRM_STAT [2]	0
Bit 1	RO	SBI_ALRM_STAT [1]	0
Bit 0	RO	SBI_ALRM_STAT [0]	0

SBI_ALRM_STATn

SBI_ALRM_STATn indicates the current state of the Extract SBI ALARM signal on the SBI tributary mapped to this link. If this bit is set the ALARM signal is active. If not set then the ALARM signal is not active. The SBI_ALRM_INTn bit will be set in the SBI_ALRM_INTL register whenever this bit changes.

Register 0x8030BH: SBI Extract Bus Alarm Status Register High (EXT_ALRM_STAT_REGH)

Bit	Type	Function	Default
Bit 15	RO	SBI_ALRM_STAT[31]	0
Bit 14	RO	SBI_ALRM_STAT [30]	0
Bit 13	RO	SBI_ALRM_STAT [29]	0
Bit 12	RO	SBI_ALRM_STAT [28]	0
Bit 11	RO	SBI_ALRM_STAT [27]	0
Bit 10	RO	SBI_ALRM_STAT [26]	0
Bit 9	RO	SBI_ALRM_STAT [25]	0
Bit 8	RO	SBI_ALRM_STAT [24]	0
Bit 7	RO	SBI_ALRM_STAT [23]	0
Bit 6	RO	SBI_ALRM_STAT [22]	0
Bit 5	RO	SBI_ALRM_STAT [21]	0
Bit 4	RO	SBI_ALRM_STAT [20]	0
Bit 3	RO	SBI_ALRM_STAT [19]	0
Bit 2	RO	SBI_ALRM_STAT [18]	0
Bit 1	RO	SBI_ALRM_STAT [17]	0
Bit 0	RO	SBI_ALRM_STAT [16]	0

SBI_ALRM_STATn

SBI_ALRM_STATn indicates the current state of the Extract SBI ALARM signal on the SBI tributary mapped to this link. If this bit is set the ALARM signal is active. If not set then the ALARM signal is not active. SBI_ALRM_INTn in the SBI_ALRM_INT register will be set whenever this bit changes.

Register 0x8030CH: SBI Insert Bus Alarm Insert Register Low (INS_ALRM_REGL)

Bit	Type	Function	Default
Bit 15	R/W	SBI_ALRM_INS[15]	0
Bit 14	R/W	SBI_ALRM_INS[14]	0
Bit 13	R/W	SBI_ALRM_INS[13]	0
Bit 12	R/W	SBI_ALRM_INS[12]	0
Bit 11	R/W	SBI_ALRM_INS[11]	0
Bit 10	R/W	SBI_ALRM_INS[10]	0
Bit 9	R/W	SBI_ALRM_INS[9]	0
Bit 8	R/W	SBI_ALRM_INS[8]	0
Bit 7	R/W	SBI_ALRM_INS[7]	0
Bit 6	R/W	SBI_ALRM_INS[6]	0
Bit 5	R/W	SBI_ALRM_INS[5]	0
Bit 4	R/W	SBI_ALRM_INS[4]	0
Bit 3	R/W	SBI_ALRM_INS[3]	0
Bit 2	R/W	SBI_ALRM_INS[2]	0
Bit 1	R/W	SBI_ALRM_INS[1]	0
Bit 0	R/W	SBI_ALRM_INS[0]	0

SBI_ALRM_INSn

When set this bit will activate the Insert SBI Alarm indicator for the tributary mapped to this link.

Register 0x8030DH: SBI Insert Bus Alarm Insert Register High (INS_ALRM_REGH)

Bit	Type	Function	Default
Bit 15	R/W	SBI_ALRM_INS[31]	0
Bit 14	R/W	SBI_ALRM_INS[30]	0
Bit 13	R/W	SBI_ALRM_INS[29]	0
Bit 12	R/W	SBI_ALRM_INS[28]	0
Bit 11	R/W	SBI_ALRM_INS[27]	0
Bit 10	R/W	SBI_ALRM_INS[26]	0
Bit 9	R/W	SBI_ALRM_INS[25]	0
Bit 8	R/W	SBI_ALRM_INS[24]	0
Bit 7	R/W	SBI_ALRM_INS[23]	0
Bit 6	R/W	SBI_ALRM_INS[22]	0
Bit 5	R/W	SBI_ALRM_INS[21]	0
Bit 4	R/W	SBI_ALRM_INS[20]	0
Bit 3	R/W	SBI_ALRM_INS[19]	0
Bit 2	R/W	SBI_ALRM_INS[18]	0
Bit 1	R/W	SBI_ALRM_INS[17]	0
Bit 0	R/W	SBI_ALRM_INS[16]	0

SBI_ALRM_INSn

When set this bit will activate the Insert SBI Alarm indicator for the tributary mapped to this link.

13.6.2 EXSBI Registers

Table 31 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

Register 0x80400H: Extract Control Register (EXT_CTL)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	APAGE	0
Bit 6	R/W	DC_EN	1
Bit 5	R/W	SYNC_INT_EN	0
Bit 4	R/W	FIFO_OVR_EN	0
Bit 3	R/W	FIFO_UDR_EN	0
Bit 2	R/W	TS_EN	0
Bit 1	R/W	SBI_PERR_EN	0
Bit 0	R/W	SBI_PAR_CTL	1

SBI_PAR_CTL

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

When SBI_PAR_CTL is ’0’ parity will be even.
When SBI_PAR_CTL is ‘1’ parity will be odd

SBI_PERR_EN

The SBI_PERR_EN bit is used to enable the SBI Parity Error interrupt generation.

When SBI_PERR_EN is ‘0’ SBI Parity Error Interrupts will be disabled
When SBI_PERR_EN is ‘1’ SBI Parity Error Interrupts will be enabled

In both cases the SBI Parity checker logic will update the SBI Parity Error Interrupt Reason Register.

TS_EN

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.

When TS_EN is '0' mapping will be fixed to a one to one mapping and will not be programmable.

When TS_EN is '1' SBI tributary to local link mapping is enabled and is specified by the contents of the Extract Tributary Mapping Register RAM

FIFO_UDR_EN

This bit is set to enable the generation of an interrupt when a FIFO under-run is detected.

FIFO_OVR_EN

This bit is set to enable the generation of an interrupt when a FIFO over-run is detected.

SYNC_INT_EN

This bit is set to enable the generation of an interrupt when a Depth Check sync error, C1FP sync error, or SBIIP sync error is detected.

DC_EN

This bit enables the Depth Check Logic. When asserted high the depth check 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. A discrepancy will normally occur any time the SBI Extract FIFO overruns or underruns.

If this bit is not set, and an SBI Extract FIFO overrun or underrun is reported, the tributary should be reset by writing to the EXT_TRIB_CTL register.

APAGE

The tributary mapping RAMs active page select bit (APAGE) controls the selection of one of two pages in the tributary Mapping RAMs to be the active page. When APAGE is set high, the configuration in page 1 of the tributary mapping RAMs is used to associate incoming tributaries to AAL1gator links. When APAGE is set low, the configuration in page 0 of the tributary mapping RAMs is used to associate incoming tributaries to AAL1gator links. Changes

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

Note that only the Mapping RAMs have two pages. The Control RAMs have only one page.

Register 0x80401H: Extract FIFO Under Run Interrupt Status Register (EXT_FI_URI)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	LINK_GRP_HIGH	0
Bit 6	RO	LINK_GRP_LOW	1
Bit 5	RO	LINK[4]	0
Bit 4	RO	LINK[3]	0
Bit 3	RO	LINK[2]	0
Bit 2	RO	LINK[1]	0
Bit 1	RO	LINK[0]	0
Bit 0	R2C	FIFO_UDRI	0

Back to back reads of this register must be at least 250 ns apart.

FIFO_UDRI

This bit is set when a FIFO under-run is detected. If this bit is set and DC_EN is not set for that tributary, then the tributary mapped to the link specified by LINK below should be reset by writing to the EXT_TRIB_CTL register. If DC_EN is set, then the tributary will be automatically reset.

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP_HIGH/LOW fields are used to specify which link of which Link Group was associated with the FIFO buffer in which the under-run was detected.

Note that 5 bits are needed to specify LINK[4:0] as the sixteen links are numbered from 1 to 16, not 0 to 15.

Register 0x80402H: Extract FIFO Over Run Interrupt Status Register (EXT_FI_ORI)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	LINK_GRP_HIGH	0
Bit 6	RO	LINK_GRP_LOW	1
Bit 5	RO	LINK[4]	0
Bit 4	RO	LINK[3]	0
Bit 3	RO	LINK[2]	0
Bit 2	RO	LINK[1]	0
Bit 1	RO	LINK[0]	0
Bit 0	R2C	FIFO_OVRI	0

Back to back reads of this register must be at least 250 ns apart.

FIFO_OVRI

This bit is set when a FIFO over-run is detected. If this bit is set and DC_EN is not set for that tributary, then the tributary mapped to the link specified by LINK below should be reset by writing to the EXT_TRIB_CTL register. If DC_EN is set, then the tributary will be automatically reset.

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP_HIGH/LOW fields are used to specify which link of which Link Group was associated with the FIFO buffer in which the overrun was detected.

Register 0x80403H: Extract Tributary RAM Indirect Access Address Register (EXT_TRIAD)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MAP_REG	0
Bit 6	R/W	SPE[1]	0
Bit 5	R/W	SPE[0]	0
Bit 4	R/W	TRIB[4]	0
Bit 3	R/W	TRIB[3]	0
Bit 2	R/W	TRIB[2]	0
Bit 1	R/W	TRIB[1]	0
Bit 0	R/W	TRIB[0]	0

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] fields are used to fully specify which SBI tributary the Mapping or Control register write or read operation will apply.

TRIB[4:0] specifies the SBI tributary number within the SBI SPE as specified by the SPE[1:0] field. Legal values for TRIB[4:0] are b’00001’ through b‘11100’. Legal values for SPE[1:0] are b’01’ through b‘11’.

MAP_REG

MAP_REG specifies whether the Extract Tributary Mapping Registers or Extract Tributary Control Registers are to be addressed by the SPE[1:0] and TRIB[4:0] fields.

When MAP_REG = ‘1’ the Extract Tributary Mapping Registers are addressed

by the SPE[1:0] and TRIB[4:0] fields.
When MAP_REG = ‘0’ the Extract Tributary Control Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

Register 0x80404H: Extract Tributary RAM Indirect Access Control Register (EXT_TRIAC)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	BUSY	0
Bit 6	R2C	HST_ADDR_ERR	0
Bit 5	Rsvd	Unused	X
Bit 4	Rsvd	Unused	X
Bit 3	Rsvd	Unused	X
Bit 2	Rsvd	Unused	X
Bit 1	R/W	RWB	0
Bit 0	R/W	PAGE	0

PAGE

The indirect page select bit (PAGE) selects between the two pages in the tributary mapping RAMs. For Control RAM accesses, PAGE is a don't care.

RWB

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the tributary mapping or control configuration RAM. Writing a '0' to RWB triggers an indirect write operation. Data to be written is taken from the Extract Tributary Mapping Indirect Access Data Register or Extract Tributary Control Indirect Access Data Register. Writing a '1' to RWB triggers an indirect read operation. The data read can be found in the Extract Tributary Mapping Indirect Access Data Register or Extract Tributary Control Indirect Access Data Register.

HST_ADDR_ERR

When set following a host read this bit 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.

BUSY

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when a write to the Extract Tributary Register Indirect Access Control Register triggers an indirect access and will stay high until the access is complete. This register should be polled to determine when data from an indirect read operation is available in the Indirect Tributary Data register or to determine when a new indirect write operation may commence. The BUSY bit is asserted for up to 4.32us after either Control register or Mapping Register access. Due to either Control or Mapping Register access, Configuration Reset will follow, and the reset tributary will not go active again for up to 600 us

Note that it is possible for this bit to clear in the middle of a read.

Register 0x80405H: Extract Tributary Mapping RAM Indirect Access Data Register (EXT_TRIB_MAP)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	X
Bit 6	R/W	LINK_GRP_HIGH	See Note below
Bit 5	R/W	LINK_GRP_LOW	See Note below
Bit 4	R/W	LINK[4]	See Note below
Bit 3	R/W	LINK[3]	See Note below
Bit 2	R/W	LINK[2]	See Note below
Bit 1	R/W	LINK[1]	See Note below
Bit 0	R/W	LINK[0]	See Note below

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP fields are used to specify which link of which Link Group with the AAL1gator is mapped to the SBI tributary.

LINK[4:0] specifies the link number within the Link Group as specified by the Link Group field that should be used as the destination for data received from the SBI tributary associated with the Extract Tributary Control and Status Register. Note Link numbering starts from 1.

Note: The default mapping is straight through ie, 1:1. Therefore, Link Group LOW, LINK 1 inside the AAL1gator 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.

Register 0x80406H: Extract Tributary Control RAM Indirect Access Data Register (EXT_TRIB_CTL)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	X
Bit 6	R/W	CLK_MODE[1]	0
Bit 5	R/W	CLK_MODE[0]	0
Bit 4	R/W	CLK_MSTR	0
Bit 3	R/W	TRIB_TYP[1]	0
Bit 2	R/W	TRIB_TYP[0]	0
Bit 1	R/W	Reserved	X
Bit 0	R/W	TRIB_ENBL	0

TRIB_ENBL

The TRIB_ENBL bit is used to enable the Tributary. Writing to an Extract Tributary Control and Status Register with the ENBL bit set enables the EXSBI to take tributary data from an SBI tributary and transmit that data to the local link mapped to that tributary.

TRIB_TYP[1:0]

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in Table 32. 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.

Table 32 TRIB_TYP Encoding

CAS Enabled	Framed	Transparent VT	TRIB_TYP	Description
True	True	False	00	Structured with CAS
False	True	False	01	Structured without CAS
False	False	False	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.

CLK_MSTR

The CLK_MSTR bit is used to specify whether the Extract block functions as a clock master or a clock slave. When this bit is a ‘1’ the Extract block is a clock master.

The CLK_MSTR configuration bits are OR’d with the CLK_MSTR bit in the SBI_BUS_CFG_REG to allow the chip level to force master mode for all tributaries. The default state of these bits will be clock slave.

CLK_MODE[1:0]

The CLK_MODE[1:0] field is used to choose the method of tributary clock synthesis. After receiving tributary bytes from the SBI bus, PISO synthesizes serial clocks for the A1SP TFTC’s using one of the following methods. The Phase field method synthesizes a clock which most accurately reproduces the original source clock. When implementing T1/E1 SRTS across the SBI bus, the Phase method is required.

CLK_MODE[1:0]	MEANING
00	Use the rate of data flow across the SBI Drop bus to synthesize serial clocks for the A1SP TFTC's.
01	Use only ClkRate field of the V4 Link Rate octet to synthesize serial clocks for the A1SP TFTC's.
10	Use only Phase field of the V4 Link Rate octet to synthesize serial clocks for the A1SP TFTC's.
11	Reserved

Note: 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.

Register 0x80407H: SBI Parity Error Interrupt Status Register (SBI_PERR)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	SPE[1]	0
Bit 6	RO	SPE[0]	1
Bit 5	RO	TRIB[4]	0
Bit 4	RO	TRIB[3]	0
Bit 3	RO	TRIB[2]	0
Bit 2	RO	TRIB[1]	0
Bit 1	RO	TRIB[0]	0
Bit 0	R2C	PERRI	0

Back to back reads of this register must be at least 250 ns apart.

PERRI

When set PERRI indicates that an SBI parity error has been detected.

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] field are used to specify the SBI tributary for which a parity error was detected. These fields are only valid only when PERRI is set.

Register 0x80408H: MIN_DEPTH for T1 and E1 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_DEP_E1[3]	0
Bit 6	R/W	MIN_DEP_E1[2]	1
Bit 5	R/W	MIN_DEP_E1[1]	1
Bit 4	R/W	MIN_DEP_E1[0]	1
Bit 3	R/W	MIN_DEP_T1[3]	0
Bit 2	R/W	MIN_DEP_T1[2]	1
Bit 1	R/W	MIN_DEP_T1[1]	1
Bit 0	R/W	MIN_DEP_T1[0]	1

MIN_DEP_E1[3:0]

Used to modify the MIN_DEPTH for E1 tributaries. This value should be left at the default setting.

MIN_DEP_T1[3:0]

Used to modify the MIN_DEPTH for T1 tributaries. This value should be left at the default setting.

Register 0x80409H: MIN_DEPTH for DS3 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Reserved	1
Bit 6	Rsvd	Reserved	1
Bit 5	Rsvd	Reserved	1
Bit 4	Rsvd	Reserved	1
Bit 3	R/W	MIN_DEP_DS3[3]	1
Bit 2	R/W	MIN_DEP_DS3[2]	1
Bit 1	R/W	MIN_DEP_DS3[1]	1
Bit 0	R/W	MIN_DEP_DS3[0]	0

MIN_DEP_DS3[3:0]

Used to modify the MIN_DEPTH for DS3 tributaries. This value should be left at the default setting.

Register 0x8040AH: T1 Threshold Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_T1[3]	0
Bit 6	R/W	MIN_THR_T1[2]	0
Bit 5	R/W	MIN_THR_T1[1]	1
Bit 4	R/W	MIN_THR_T1[0]	0
Bit 3	R/W	MAX_THR_T1[3]	1
Bit 2	R/W	MAX_THR_T1[2]	1
Bit 1	R/W	MAX_THR_T1[1]	0
Bit 0	R/W	MAX_THR_T1[0]	1

MAX_TRH_T1[3:0]

Used to modify the Maximum Threshold for T1 tributaries. This value should be left at the default setting.

MIN_TRH_T1[3:0]

Used to modify the Minimum Threshold for T1 tributaries. This value should be left at the default setting.

Register 0x8040BH: E1 Threshold Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_E1[3]	0
Bit 6	R/W	MIN_THR_E1[2]	0
Bit 5	R/W	MIN_THR_E1[1]	1
Bit 4	R/W	MIN_THR_E1[0]	0
Bit 3	R/W	MAX_THR_E1[3]	1
Bit 2	R/W	MAX_THR_E1[2]	1
Bit 1	R/W	MAX_THR_E1[1]	0
Bit 0	R/W	MAX_THR_E1[0]	1

MAX_TRH_E1[3:0]

Used to modify the Maximum Threshold for E1 tributaries. This value should be left at the default setting.

MIN_TRH_E1[3:0]

Used to modify the Minimum Threshold for E1 tributaries. This value should be left at the default setting.

Register 0x8040CH: DS3 Threshold Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_DS3[3]	0
Bit 6	R/W	MIN_THR_DS3[2]	1
Bit 5	R/W	MIN_THR_DS3[1]	0
Bit 4	R/W	MIN_THR_DS3[0]	0
Bit 3	R/W	MAX_THR_DS3[3]	1
Bit 2	R/W	MAX_THR_DS3[2]	1
Bit 1	R/W	MAX_THR_DS3[1]	0
Bit 0	R/W	MAX_THR_DS3[0]	0

MAX_TRH_DS3[3:0]

Used to modify the Maximum Threshold for DS3 tributaries. This value should be left at the default setting.

MIN_TRH_DS3[3:0]

Used to modify the Minimum Threshold for DS3 tributaries. This value should be left at the default setting.

Register 0x8040EH: Extract Depth Check Interrupt Status Register (EXT_DCR_INT)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	SPE[1]	0
Bit 6	RO	SPE[0]	1
Bit 5	RO	SBI_TRIBUTARY[4]	0
Bit 4	RO	SBI_TRIBUTARY[3]	0
Bit 3	RO	SBI_TRIBUTARY[2]	0
Bit 2	RO	SBI_TRIBUTARY[1]	0
Bit 1	RO	SBI_TRIBUTARY[0]	0
Bit 0	R2C	DCR_INTI	0

Back to back reads of this register must be at least 250 ns apart.

DCR_INTI

This bit is set when a Depth Check error is detected. This error is detected when the internal FIFO pointers don't match the expected internal FIFO depth.

Values in these fields should only be looked at when DCR_INTI is a '1'.

SPE[1:0] and SBI_TRIBUTARY

The SPE and SBI_TRIBUTARY fields are used to specify which Tributary was associated with the Depth Check error. Note that if mapping is enabled the SBI Tributary is different than the internal link. Also note that SPEs are numbered 1,2,3.

Register 0x8040FH: Extract Master Interrupt Register (EXT_MSTR_INT)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RSVD	Reserved	0
Bit 6	RSVD	Unused	X
Bit 5	RO	EXT_DC_INT	0
Bit 4	RO	EXT_SBI_PERR_INT	0
Bit 3	RO	EXT_FIFO_UDR_INT	0
Bit 2	RO	EXT_FIFO_OVR_INT	0
Bit 1	R2C	EXT_SYNC_INT	0
Bit 0	R2C	EXT_C1FP_INT	0

EXT_C1FP_INT

This bit is set when a C1FP realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the Extract Control Register.

EXT_SYNC_INT

This bit is set when a SBIIP_SYNC realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the Extract Control Register.

EXT_FIFO_OVR_INT

This bit is set when a FIFO Overrun Interrupt is pending. Read Extract FIFO Overrun Interrupt Status Register to determine on which link, the error occurred. This bit will not be set if FIFO_OVR_EN is low in the Extract Control Register.

EXT_FIFO_UDR_INT

This bit is set when a FIFO Underrun Interrupt is pending. Read Extract FIFO Underrun Interrupt Status Register to determine on which link, the error occurred. This bit will not be set if FIFO_UDR_EN is low in the Extract Control Register.

EXT_SBI_PERR_INT

This bit is set when a SBI Parity Error Interrupt is pending. Read the SBI Parity Error Reason Register to determine on which link, the error occurred. This bit will not be set if SBI_PERR_EN is low in the Extract Control Register.

EXT_DC_INT

This bit is set when an Extract Depth Check Interrupt is pending. Read Extract Depth Check Interrupt Status Register to determine on which SBI Tributary, the error occurred. This bit will not be set if SYNC_INT_EN is low in the Extract Control Register.

Note that if mapping is enabled the SBI tributary will not be the same as the internal link number.

13.6.3 INSBI Registers

Table 33 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

Register 0x80500H: Insert Control Register (INS_CTL)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	APAGE	0
Bit 6	R/W	DC_EN	1
Bit 5	Rsvd	SYNC_INT_EN	0
Bit 4	R/W	FIFO_OVR_EN	0
Bit 3	R/W	FIFO_UDR_EN	0
Bit 2	R/W	TS_EN	0
Bit 1	Rsvd	Unused	X
Bit 0	R/W	SBI_PAR_CTL	1

SBI_PAR_CTL

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

When SBI_PAR_CTL is a ‘0’ parity will be even.
When SBI_PAR_CTL is a ‘1’ parity will be odd.

TS_EN

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.

When TS_EN is a ‘0’ mapping will be fixed to a one to one mapping and will not be programmable.

When TS_EN is a '1' SBI tributary to AAL1gator link mapping is enabled and is specified by the contents of the Insert Tributary Mapping Register RAM

FIFO_UDR_EN

The FIFO_UDR_EN bit is used to enable/disable the generation of an interrupt when a FIFO underrun is detected.

When FIFO_UDR_EN is a '0' underrun interrupt generation is disabled.
When FIFO_UDR_EN is a '1' underrun interrupt generation is enabled.

FIFO_OVR_EN

The FIFO_OVR_EN bit is used to enable/disable the generation of an interrupt when a FIFO overrun is detected.

When FIFO_OVR_EN is a '0' overrun interrupt generation is disabled.
When FIFO_OVR_EN is a '1' overrun interrupt generation is enabled.

SYNC_INT_EN

This bit is set to enable the generation of an interrupt when a Depth Check sync error, C1FP sync error, or SBIIP sync error is detected.

DC_EN

This bit enables the Depth Check logic. When asserted high the depth checker logic will periodically monitor the Data/Framing FIFO Depth and compare it against the read and write pointers. If there is a discrepancy the tributary is reset by the depth checker. A discrepancy will normally occur any time the SBI Insert FIFO overruns or underruns.
This function is disabled anytime any Insert Tributary has the SYNCH_TRIB bit set.

If this bit is not set, or any Insert Tributary has the SYNCH_TRIB bit set, and an SBI Insert FIFO overrun or underrun is reported, the tributary should be reset by writing to the INS_TRIB_CTL register.

APAGE

The tributary mapping and control configuration RAMs active page select bit (APAGE) controls the selection of one of two pages in the tributary mapping RAMs to be the active page. When APAGE is set high, the configuration in page 1 of the tributary mapping RAMs is used to associate incoming tributaries to AAL1gator links. When APAGE is set low, the configuration in page 0 of the tributary mapping RAMs is used to associate incoming tributaries to Aal1gator links. Changes of the active page as a result of write

accesses to APAGE will be synchronized to SBI multi-frame boundaries at C1FP.

Note that only the Mapping RAMs have two pages. The Control RAMs have only one page.

Register 0x80501H: Insert FIFO Underrun Interrupt Status Register (INS_FI_URI)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	LINK_GRP_HIGH	0
Bit 6	RO	LINK_GRP_LOW	1
Bit 5	RO	LINK[4]	0
Bit 4	RO	LINK[3]	0
Bit 3	RO	LINK[2]	0
Bit 2	RO	LINK[1]	0
Bit 1	RO	LINK[0]	0
Bit 0	R2C	FIFO_UDRI	0

Back to back reads of this register must be at least 250 ns apart.

FIFO_UDRI

This bit is set when a FIFO underrun is detected. 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. If this bit is set and DC_EN is not set for that tributary and SYNCH_TRIB is not set for any tributary, then the tributary mapped to the link specified by LINK below should be reset by writing to the EXT_TRIB_CTL register. If DC_EN is set and SYNCH_TRIB is not set for any tributaries, then the tributary will be automatically reset.

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP fields are used to specify which link in which AAL1gator Link Group is associated with the FIFO buffer in which the underrun was detected.

LINK[4:0] and LINK_GRP are invalid unless FIFO_UDRI is set.

Register 0x80502H: Insert FIFO Overrun Interrupt Status Register (INS_FI_ORI)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	LINK_GRP_HIGH	0
Bit 6	RO	LINK_GRP_LOW	1
Bit 5	RO	LINK[4]	0
Bit 4	RO	LINK[3]	0
Bit 3	RO	LINK[2]	0
Bit 2	RO	LINK[1]	0
Bit 1	RO	LINK[0]	0
Bit 0	R2C	FIFO_OVRI	0

Back to back reads of this register must be at least 250 ns apart.

FIFO_OVRI

This bit is set when a FIFO overrun is detected. 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. If this bit is set and DC_EN is not set for that tributary and SYNCH_TRIB is not set for any tributary, then the tributary mapped to the link specified by LINK below should be reset by writing to the EXT_TRIB_CTL register. If DC_EN is set and SYNCH_TRIB is not set for any tributaries, then the tributary will be automatically reset.

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP fields are used to specify which link of which AAL1gator Link Group is associated with the FIFO buffer in which the overrun was detected.

LINK[4:0] and LINK_GRP are invalid unless FIFO_OVRI is set.

Register 0x80503H: Insert Tributary Register Indirect Access Address Register(INS_TRIAD)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MAP_REG	0
Bit 6	R/W	SPE[1]	0
Bit 5	R/W	SPE[0]	0
Bit 4	R/W	TRIB[4]	0
Bit 3	R/W	TRIB[3]	0
Bit 2	R/W	TRIB[2]	0
Bit 1	R/W	TRIB[1]	0
Bit 0	R/W	TRIB[0]	0

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] fields are used to fully specify for which SBI tributary the Mapping or Control register write or read operation will apply.

TRIB[4:0] specifies the SBI tributary number within the SBI SPE as specified by the SPE[1:0] field. Legal values for TRIB[4:0] are b'00001' through b'11100'. Legal values for SPE[1:0] are b'01' through b'11'.

MAP_REG

MAP_REG specifies whether the Insert Tributary Mapping Registers or Insert Tributary Control Registers are to be addressed by the SPE[1:0] and TRIB[4:0] fields.

When MAP_REG = '1' the Insert Tributary Mapping Registers are addressed

by the SPE[1:0] and TRIB[4:0] fields.
When MAP_REG = ‘0’ the Insert Tributary Control Registers are addressed by the SPE[1:0] and TRIB[4:0] fields.

Register 0x80504H: Insert Tributary Register Indirect Access Control Register (INS_TRIAC)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	BUSY	0
Bit 6	R2C	HST_ADDR_ERR	0
Bit 5	Rsvd	Unused	X
Bit 4	Rsvd	Unused	X
Bit 3	Rsvd	Unused	X
Bit 2	Rsvd	Unused	X
Bit 1	R/W	RWB	0
Bit 0	R/W	PAGE	0

PAGE

The indirect page select bit (PAGE) selects between the two pages in the tributary mapping RAMs. For Control RAM accesses, PAGE is a don't care.

RWB

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the tributary mapping or control configuration RAM. Writing a '0' to RWB triggers an indirect write operation. Data to be written is taken from the Insert Tributary Mapping Indirect Access Data Register or Insert Tributary Control Indirect Access Data Register. Writing a '1' to RWB triggers an indirect read operation. The data read can be found in the Insert Tributary Mapping Indirect Access Data Register or Insert Tributary Control Indirect Access Data Register.

HST_ADDR_ERR

When set following a host read this bit indicates that an illegal host access was attempted. An illegal host access occurs when an attempt is mage to access an out of range tributary.

BUSY

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when a write to the Insert Tributary Register Indirect Access Control Register triggers an indirect access and will stay high until the access is complete. This register should be polled to determine when data from an indirect read operation is available in the Indirect Tributary Data register or to determine when a new indirect write operation may commence. The BUSY bit is asserted for up to 4.32us after either Control register or Mapping Register access. Due to either Control or Mapping Register access, Configuration Reset will follow, and the reset tributary will not go active again for up to 600 us

Note that it is possible for this bit to clear in the middle of a read.

Register 0x80505H: Insert Tributary Mapping Indirect Access Data Register (INS_TRIB_MAP

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	0
Bit 6	R/W	LINK_GRP_HIGH	See Note below
Bit 5	R/W	LINK_GRP_LOW	See Note below
Bit 4	R/W	LINK[4]	See Note below
Bit 3	R/W	LINK[3]	See Note below
Bit 2	R/W	LINK[2]	See Note below
Bit 1	R/W	LINK[1]	See Note below
Bit 0	R/W	LINK[0]	See Note below

LINK[4:0] and LINK_GRP_HIGH/LOW

The LINK[4:0] and LINK_GRP fields are used to specify the AAL1gator link to SBI tributary mapping.

LINK[4:0] specifies the AAL1gator link number within a Link Group as specified by the LINK_GRP field that should be used as the source for data transmitted to the SBI tributary associated with the Insert Tributary Control and Status Register

Note: The default mapping is straight through ie, 1:1. Therefore, LINK_GRP_LOW, LINK 1 in the AAL1gator side will be mapped by default to SPE1, LINK 1 on the SBI side and so on up to SPE1, LINK 16. LINK_GRP_HIGH, LINK 1 in the AAL1gator side will be mapped by default to SPE2, LINK 1 on the SBI side and so on up to SPE2, LINK 16.

The mapping of more than one tributary to a link or more than one link to a tributary is not allowed.

Register 0x80506H: Insert Tributary Control Indirect Access Data Register (INS_TRIB_CTL)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Unused	0
Bit 6	Rsvd	Unused	0
Bit 5	R/W	SYNCH_TRIB	0
Bit 4	R/W	CLK_MSTR	0
Bit 3	R/W	TRIB_TYP[1]	0
Bit 2	R/W	TRIB_TYP[0]	0
Bit 1	R/W	Reserved	X
Bit 0	R/W	TRIB_ENBL	0

TRIB_ENBL

The TRIB_ENBL bit is used to enable the Tributary. Writing to an Insert Tributary Control and Status Register 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.

CLK_MSTR

The CLK_MSTR bit is used to specify whether the Insert block functions as a clock master or a clock slave. When this bit is a ‘1’ the Insert block is a clock master.

The CLK_MSTR configuration bits are OR’d with the CLK_MSTR bit in the SBI_BUS_CFG_REG to allow the chip level to force master mode for all tributaries. The default state of these bits will be clock slave.

TRIB_TYP[1:0]

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in the following table. It 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 setup.

Table 34 TRIB_TYP Encoding

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

Notes:

- CAS can only be enabled for a framed tributary.
- "Framed" means framing information available – may be channelized or unchannelized.
- "Unframed" means no framing information available.
- If Structured with CAS mode is used, SYNCH_TRIB must also be set.

SYNCH_TRIB

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. This bit will default to off.

Note: 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.

Register 0x80507H: MIN_DEPTH for T1 and E1 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_DEP_E1[3]	0
Bit 6	R/W	MIN_DEP_E1[2]	1
Bit 5	R/W	MIN_DEP_E1[1]	1
Bit 4	R/W	MIN_DEP_E1[0]	1
Bit 3	R/W	MIN_DEP_T1[3]	0
Bit 2	R/W	MIN_DEP_T1[2]	1
Bit 1	R/W	MIN_DEP_T1[1]	1
Bit 0	R/W	MIN_DEP_T1[0]	1

MIN_DEP_E1[3:0]

Used to modify the MIN_DEPTH for E1 tributaries. This value should remain at its default setting.

MIN_DEP_T1[3:0]

Used to modify the MIN_DEPTH for T1 tributaries. This value should remain at its default setting.

Register 0x80508H: MIN_DEPTH for DS3 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	Rsvd	Reserved	1
Bit 6	Rsvd	Reserved	1
Bit 5	Rsvd	Reserved	0
Bit 4	Rsvd	Reserved	1
Bit 3	R/W	MIN_DEP_DS3[3]	1
Bit 2	R/W	MIN_DEP_DS3[2]	1
Bit 1	R/W	MIN_DEP_DS3[1]	0
Bit 0	R/W	MIN_DEP_DS3[0]	0

MIN_DEP_DS3[3:0]

Used to modify the MIN_DEPTH for DS3 tributaries. This value should remain at its default setting.

Register 0x80509H: MIN_THR and MAX_THR for T1 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_T1[3]	0
Bit 6	R/W	MIN_THR_T1[2]	1
Bit 5	R/W	MIN_THR_T1[1]	1
Bit 4	R/W	MIN_THR_T1[0]	0
Bit 3	R/W	MAX_THR_T1[3]	1
Bit 2	R/W	MAX_THR_T1[2]	1
Bit 1	R/W	MAX_THR_T1[1]	1
Bit 0	R/W	MAX_THR_T1[0]	0

MIN_THR_T1[3:0]

Used to modify the Minimum Threshold for T1 tributaries. This value should remain at its default setting.

MAX_THR_T1[3:0]

Used to modify the Maximum Threshold for T1 tributaries. This value should be changed to “1010” when AAL1gator is clock master.

Register 0x8050AH: MIN_THR and MAX_THR for E1 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_E1[3]	0
Bit 6	R/W	MIN_THR_E1[2]	0
Bit 5	R/W	MIN_THR_E1[1]	1
Bit 4	R/W	MIN_THR_E1[0]	0
Bit 3	R/W	MAX_THR_E1[3]	1
Bit 2	R/W	MAX_THR_E1[2]	1
Bit 1	R/W	MAX_THR_E1[1]	1
Bit 0	R/W	MAX_THR_E1[0]	0

MIN_THR_E1[3:0]

Used to modify the Minimum Threshold for E1 tributaries. This value should remain at its default setting.

MAX_THR_E1[3:0]

Used to modify the Maximum Threshold for E1 tributaries. This value should be changed to “1010”, when AAL1gator is clock master.

Register 0x8050BH: MIN_THR and MAX_THR for DS3 Register

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	R/W	MIN_THR_DS3[3]	1
Bit 6	R/W	MIN_THR_DS32]	0
Bit 5	R/W	MIN_THR_DS31]	0
Bit 4	R/W	MIN_THR_DS30]	0
Bit 3	R/W	MAX_THR_DS3[3]	0
Bit 2	R/W	MAX_THR_DS3[2]	1
Bit 1	R/W	MAX_THR_DS3[1]	1
Bit 0	R/W	MAX_THR_DS3[0]	0

MIN_THR_DS3[3:0]

Used to modify the Minimum Threshold for DS3 tributaries. This value should remain at its default setting.

MAX_THR_DS3[3:0]

Used to modify the Maximum Threshold for DS3 tributaries. This value should remain at its default setting.

Register 0x80511H: Insert Depth Check Interrupt Status Register (INS_DVR_INT)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RO	SPE[1]	0
Bit 6	RO	SPE[0]	1
Bit 5	RO	SBI_TRIBUTARY[4]	0
Bit 4	RO	SBI_TRIBUTARY[3]	0
Bit 3	RO	SBI_TRIBUTARY[2]	0
Bit 2	RO	SBI_TRIBUTARY[1]	0
Bit 1	RO	SBI_TRIBUTATY[0]	0
Bit 0	R2C	DCR_INTI	0

Back to back reads of this register must be at least 250 ns apart.

DCR_INTI

This bit is set when a Depth Check error is detected. This error is detected when the internal FIFO pointers don't match the expected internal FIFO depth.
Values in these fields are only relevant when DCR_INTI is a '1'.

SPE[1:0] and SBI_TRIBUTARY[4:0]

The SPE and SBI_TRIBUTARY fields are used to specify which Tributary was associated with the Depth Check error.

Note that if mapping is enabled the SBI tributary will not be the same as the internal link number. Also note that SPEs are numbered 1,2,3.

Register 0x80512H: Insert Master Interrupt Register (INS_MSTR_INT)

Bit	Type	Function	Default
Bit 15	Rsvd	Unused	0
Bit 14	Rsvd	Unused	0
Bit 13	Rsvd	Unused	0
Bit 12	Rsvd	Unused	0
Bit 11	Rsvd	Unused	0
Bit 10	Rsvd	Unused	0
Bit 9	Rsvd	Unused	0
Bit 8	Rsvd	Unused	0
Bit 7	RSVD	Unused	0
Bit 6	RSVD	Unused	0
Bit 5	RO	INS_DC_INT	0
Bit 4	RSVD	Unused	X
Bit 3	RO	INS_FIFO_UDR_INT	0
Bit 2	RO	INS_FIFO_OVR_INT	0
Bit 1	R2C	INS_SYNC_INT	0
Bit 0	R2C	INS_C1FP_INT	0

INS_C1FP_INT

This bit is set when a C1FP realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the Insert Control Register.

INS_SYNC_INT

This bit is set when a SBIIP_SYNC realignment has been detected. This bit will not be set if SYNC_INT_EN is low in the Insert Control Register.

INS_FIFO_OVR_INT

This bit is set when a FIFO Overrun Interrupt is pending. Read Insert FIFO Over Run Interrupt Status Register to determine on which link, the error occurred. This bit will not be set if FIFO_OVR_EN is low in the Insert Control Register.

INS_FIFO_UDR_INT

This bit is set when a FIFO Underrun Interrupt is pending. Read Insert FIFO Inder Run Interrupt Status Register to determine on which link, the error occurred. This bit will not be set if FIFO_UDR_EN is low in the Insert Control Register.

INS_DC_INT

This bit is set when an Insert Depth Check Interrupt is pending. Read Insert Depth Check Interrupt Status Register to determine on which tributary, the error occurred. This bit will not be set if SYNC_INT_EN is low in the Insert Control Register.

Note that if mapping is enabled the SBI tributary will not be the same as the internal link number.

13.7 Interrupt and Status Registers

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

Table 35 Interrupt and Status Registers Memory Map

Address	Register Description	Register Mnemonic
0x81000	Master Interrupt Register	MSTR_INTR_REG
0x81010	A1SP0 Interrupt Register	A1SP0_INTR_REG
0x81011	A1SP1 Interrupt Register	A1SP1_INTR_REG
0x81012	A1SP2 Interrupt Register	A1SP2_INTR_REG
0x81013	A1SP2 Interrupt Register	A1SP3_INTR_REG
0x81020	A1SP0 Status Register	A1SP0_STAT_REG
0x81021	A1SP1 Status Register	A1SP1_STAT_REG
0x81022	A1SP2 Status Register	A1SP2_STAT_REG
0x81023	A1SP3 Status Register	A1SP3_STAT_REG
0x81030	A1SP0 Transmit Idle State FIFO	A1SP0_TIDLE_FIFO
0x81031	A1SP1 Transmit Idle State FIFO	A1SP1_TIDLE_FIFO
0x81032	A1SP2 Transmit Idle State FIFO	A1SP2_TIDLE_FIFO
0x81033	A1SP3 Transmit Idle State FIFO	A1SP3_TIDLE_FIFO
0x81040	A1SP0 Receive Status FIFO	A1SP0_RSTAT_FIFO
0x81041	A1SP1 Receive Status FIFO	A1SP1_RSTAT_FIFO
0x81042	A1SP2 Receive Status FIFO	A1SP2_RSTAT_FIFO
0x81043	A1SP3 Receive Status FIFO	A1SP3_RSTAT_FIFO
0x81100	Master Interrupt Enable Register	MSTR_INTR_REG
0x81110	A1SP0 Interrupt Enable Register	A1SP0_INTR_EN
0x81111	A1SP1 Interrupt Enable Register	A1SP1_INTR_EN
0x81112	A1SP2 Interrupt Enable Register	A1SP2_INTR_EN
0x81113	A1SP2 Interrupt Enable Register	A1SP3_INTR_EN
0x81140	A1SP0 Receive Status FIFO Enable Register	A1SP0_RSTAT_EN
0x81141	A1SP1 Receive Status FIFO Enable Register	A1SP1_RSTAT_EN
0x81142	A1SP2 Receive Status FIFO Enable Register	A1SP2_RSTAT_EN

Address	Register Description	Register Mnemonic
0x81143	A1SP3 Receive Status FIFO Enable Register	A1SP3_RSTAT_EN
0x81150	A1SP0 Receive Queue Error Enable	A1SP0_RCV_Q_ERR_EN
0x81151	A1SP1 Receive Queue Error Enable	A1SP1_RCV_Q_ERR_EN
0x81152	A1SP2 Receive Queue Error Enable	A1SP2_RCV_Q_ERR_EN
0x81153	A1SP3 Receive Queue Error Enable	A1SP3_RCV_Q_ERR_EN

Register 0x81000: Master Interrupt Register (MSTR_INTR_REG)

Bit	Type	Function	Default
15	RO	Unused	X
14	RO	Unused	X
13	RO	SBI_ALARM	0
12	RO	SBI_DROP_INTR	0
11	RO	SBI_ADD_INTR	0
10	R2C	R_UTOP_RUNT_CL	0
9	R2C	UTOP_LFIFO_FULL	0
8	R2C	T_UTOP_XFR_ERR	0
7	R2C	T_UTOP_FULL	0
6	R2C	UTOP_PAR_ERR	0
5	R2C	RAM2_PAR_ERR	0
4	R2C	RAM1_PAR_ERR	0
3	RO	A1SP3_INTR	0
2	RO	A1SP2_INTR	0
1	RO	A1SP1_INTR	0
0	RO	A1SP0_INTR	0

This register is 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.

A1SP0_INTR

When set, there is an interrupt pending from the A1SP0 block. Read the A1SP0_INTR_REG to determine the cause of the interrupt. This bit indicates current status and will clear only when no interrupt conditions remain in A1SP0_INTR_REG. On read:

0) No interrupt pending from the A1SP0 block
1) Interrupt pending from the A1SP0 block

A1SP1_INTR

When set, there is an interrupt pending from the A1SP1 block. Read the A1SP1_INTR_REG to determine the cause of the interrupt. This bit indicates current status and will clear only when no interrupt conditions remain in A1SP1_INTR_REG.On read:

0) No interrupt pending from the A1SP1 block
1) Interrupt pending from the A1SP1 block

A1SP2_INTR

When set, there is an interrupt pending from the A1SP2 block. Read the A1SP2_INTR_REG to determine the cause of the interrupt. This bit indicates current status and will clear only when no interrupt conditions remain in A1SP2_INTR_REG.On read:

0) No interrupt pending from the A1SP2 block
1) Interrupt pending from the A1SP2 block

A1SP3_INTR

When set, there is an interrupt pending from the A1SP3 block. Read the A1SP3_INTR_REG to determine the cause of the interrupt. This bit indicates current status and will clear only when no interrupt conditions remain in A1SP3_INTR_REG.On read:

0) No interrupt pending from the A1SP3 block
1) Interrupt pending from the A1SP3 block

RAM1_PAR_ERR

When set, indicates there was a parity error encountered in the RAM1 interface. This bit is cleared on read. On read:

0) No parity error encountered in RAM1 interface
1) Parity error encountered in RAM1 interface

RAM2_PAR_ERR

When set, indicates there was a parity error encountered in the RAM2 interface. This bit is cleared on read. On read:

0) No parity error encountered in RAM2 interface
1) Parity error encountered in RAM2 interface

UTOP_PAR_ERR

When set, indicates there was a parity error encountered in the UTOPIA interface. This bit is cleared on read. On read:

0) No parity error encountered in UTOPIA interface
1) Parity error encountered in UTOPIA interface

T_UTOP_FULL

When set, indicates the Transmit UTOPIA FIFO was full. This bit is cleared on read. If the Transmit UTOPIA FIFO is still full, the bit will be set again. On read:

0) Transmit UTOPIA FIFO is not full
1) Transmit UTOPIA FIFO is full

T_UTOP_XFR_ERR

When set indicates that the Transmit UTOPIA Interface was requested to send a cell when it did not have one available. This bit is cleared on read.

0) No transfer error occurred
1) A UTOPIA transfer error occurred

UTOP_LFIFO_FULL

When set indicates that the UTOPIA Loopback FIFO went full. This bit is cleared on read. If the Loopback FIFIO is still full, the bit will be set again.

0) UTOPIA loopback FIFO is not full.
1) UTOPIA loopback FIFO went full.

R_UTOP_RUNT_CL

When set indicates that a short cell (less than 53 bytes) was received. This bit is cleared on read.

0) No runt cell was received
1) A runt cell was received

SBI_ADD_INTR

When set, indicates that 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. This bit indicates current status and will clear only when no interrupt conditions remain in the Insert Master Interrupt Status Register. On read:

0) No SBI Add Bus interrupt pending
1) SBI Add Bus interrupt is pending

SBI_DROP_INTR

When set, indicates that 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. This bit indicates current status and will clear only when no interrupt conditions remain in the Extract Master Interrupt Status Register. On read:

0) No SBI Drop Bus interrupt pending
1) SBI Drop Bus interrupt is pending

SBI_ALARM

When set indicates that an interrupt is pending in either the SBI_ALARM_REGH or SBI_ALARM_REGL registers. This bit indicates current status, and will clear only when no interrupt conditions exist in either SBI alarm registers. On read:

0) No SBI alarm pending
1) SBI alarm pending

Register 0x81010, … 13: A1SPn Interrupt Register (A1SPn_INTR_REG)

Bit	Type	Function	Default
15	RO	Unused	X
14	RO	Unused	X
13	RO	Unused	X
12	RO	Unused	X
11	RO	Unused	X
10	RO	Unused	X
9	RO	Unused	X
8	RO	Unused	X
7	RO	Unused	X
6	R2C	TALP_FIFO_FULL	0
5	R2C	RSTAT_FIFO_FULL	0
4	R2C	RSTAT_FIFO_EMPB	0
3	R2C	TIDLE_FIFO_FULL	0
2	R2C	TIDLE_FIFO_EMPB	0
1	R2C	OAM_INTR	0
0	R2C	FR_ADV_FIFO_FULL	0

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 again. Read A1SPn_STAT_REG for current status. If any bit is set in this register and the corresponding enable bit is set in A1SPn_INTR_EN_REG, the A1SPn_INTR bit will be set in MSTR_INTR_REG. There is one register for each A1SP block.

FR_ADV_FIFO_FULL

When set indicates the Frame Advance FIFO has entered the full state since the last time this register was read. On read:

0) Frame Advance FIFO has not entered the full state
1) Frame Advance FIFO has entered the full state

OAM_INTR

When set, indicates the A1SPn block has received a new OAM cell. On read:

0) A1SPn has not received a new OAM cell
1) A1SPn has received a new OAM cell

TIDLE_FIFO_EMPB

When clear, indicates the Transmit Idle State FIFO has remained empty. On read:

0) Transmit Idle State FIFO has remained empty
1) Transmit Idle State FIFO has entered the not empty state

TIDLE_FIFO_FULL

When set, indicates the Transmit Idle State FIFO has entered the full state. On read:

0) Transmit Idle State FIFO has not entered the full state
1) Transmit Idle State FIFO has entered the full state

RSTAT_FIFO_EMPB

When clear, indicates the Receive Status FIFO has remained empty. On read:

0) Receive Status FIFO has remained empty
1) Receive Status FIFO has entered the not empty state

RSTAT_FIFO_FULL

When set, indicates the Receive Status FIFO has entered the full state. On read:

0) Receive Status FIFO has not entered the full state
1) Receive Status FIFO has entered the full state

TALP_FIFO_FULL

When set, indicates the TALP FIFO has entered the full state. On read:

0) TALP FIFO has not entered the full state
1) TALP FIFO has entered the full state

Register 0x81020, … 23: A1SPn Status Register (A1SPn_STAT_REG)

Bit	Type	Function	Default
15	RO	Unused	X
14	RO	Unused	X
13	RO	Unused	X
12	RO	Unused	X
11	RO	Unused	X
10	RO	Unused	X
9	RO	Unused	X
8	RO	Unused	X
7	RO	Unused	X
6	RO	TALP_FIFO_FULL	0
5	RO	RSTAT_FIFO_FULL	0
4	RO	RSTAT_FIFO_EMPB	0
3	RO	TIDLE_FIFO_FULL	0
2	RO	TIDLE_FIFO_EMPB	0
1	RO	OAM_INTR	0
0	RO	FR_ADV_FIFO_FULL	0

The bits in this register indicate current status and are not cleared on reads. There is one register for each A1SP block.

FR_ADV_FIFO_FULL

When set indicates the Frame Advance FIFO is full. On read:

0) Frame Advance FIFO is not full
1) Frame Advance FIFO is full

OAM_INTR

When set, indicates the A1SPn block has received a new OAM cell. On read:

0) A1SPn has not received a new OAM cell
1) A1SPn has received a new OAM cell

TIDLE_FIFO_EMPB

When clear, indicates the Transmit Idle State FIFO is empty. On read:

0) Transmit Idle State FIFO is empty
1) Transmit Idle State FIFO is not empty

TIDLE_FIFO_FULL

When set, indicates the Transmit Idle State FIFO is full. On read:

0) Transmit Idle State FIFO is not full
1) Transmit Idle State FIFO is full

RSTAT_FIFO_EMPB

When clear, indicates the Receive Status FIFO is empty. On read:

0) Receive Status FIFO is empty
1) Receive Status FIFO is not empty

RSTAT_FIFO_FULL

When set, indicates the Receive Status FIFO is full. On read:

0) Receive Status FIFO is not full
1) Receive Status FIFO is full

TALP_FIFO_FULL

When set, indicates the TALP FIFO is full. On read:

0) TALP FIFO is not full
1) TALP FIFO is full

Register 0x81030, …, 33: A1SPn Transmit Idle State FIFO (A1SPn_TIDLE_FIFO)

Bit	Type	Function	Default
15:0	RO	CHAN_STATUS See description below	X

This 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 A1SPn_INTR_REG. When this FIFO goes from an empty to a non-empty condition the TX_IDLE_FIFO_EMPB bit in the A1SPn_INTR_REG will be set. The presence of data in this FIFO will set the TX_IDLE_FIFO_EMPB bit in the A1SPn_STAT_REG. Read A1SPn_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.

CHAN_STATUS

This register structure 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_Cx for the respective line and channel number in the Idle Configuration Detection Table. The structure for automatic idle detection is shown first followed by the structure for processor idle detection.

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_Cx for that channel.

Bit	Type	Function	Default
15:8	RO	CHAN_NUM [7:0]	XX_H
7	RO	Unused	XX_H
6	RO	Unused	XX_H
5	RO	Unused	XX_H
4	RO	Unused	XX_H
3	RO	Unused	XX_H
2	RO	Unused	XX_H
1	RO	Unused	XX_H
0	RO	STATUS	X

STATUS

When set, indicates that the channel contained in the CHAN_NUM field is in the active state. On read:

0) Idle
1) Active

CHAN_NUM[7:0]

This field indicates the channel that encountered a change in activity status.

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 for a CAS change to be valid, it has to be stable for two consecutive samples. Any additional filtering must be done in the external framers.

Bit	Type	Function	Default
15:8	RO	CHAN_NUM	XX_{Hn}
7:4	RO	RX_CAS	X_H
3:0	RO	TX_CAS	X_H

TX_CAS

This field indicates the current value of the transmit CAS ABCD bits.

RX_CAS

This field indicates the current value of the receive CAS ABCD bits.

CHAN_NUM

This field indicates the channel that encountered a change in activity status.

Register 0x81040, …, 43: A1SPn Receive Status FIFO (A1SPn_RSTAT_FIFO)

Bit	Type	Function	Default
15:8	RO	QUEUE_NUMBER[7:0]	X
7:6	RO	Unused	X
5	RO	R_LINE_RESYNC	X
4	RO	T_LINE_RESYNC	X
3	RO	BITMASK_CHANGE	X
2	RO	EXIT_UNDERRUN	X
1	RO	ENTER_UNDERRUN	X
0	RO	RECEIVE_QUEUE_ERR	X

This register 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 A1SPn_INTR_REG. The presence of data in this FIFO will set the RSTAT_FIFO_EMPB bit in the A1SPn_INTR_REG. The RSTAT_EN_REG controls whether certain errors or status conditions cause an entry to be written into the RSTAT_FIFO. There is a separate RSTAT_ FIFO for each A1SP block.

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. The RCV_Q_ERR_EN register controls which Receive queue sticky bits will cause an entry into this FIFO.

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.

Register 0x81100: Master Interrupt Enable Register (MSTR_INTR_EN_REG)

Bit	Type	Function	Default
15	RO	Unused	X
14	RO	Unused	X
13	R/W	SBI_ALARM_EN	0
12	R/W	SBI_DROP_INTR_EN	0
11	R/W	SBI_ADD_INTR_EN	0
10	R/W	R_UTOP_RUNT_CL_EN	0
9	R/W	UTOP_LFIFO_FULL_EN	0
8	R/W	T_UTOP_XFR_ERR_EN	0
7	R/W	T_UTOP_FULL_EN	0
6	R/W	UTOP_PAR_ERR_EN	0
5	R/W	RAM2_PAR_ERR_EN	0
4	R/W	RAM1_PAR_ERR_EN	0
3	R/W	A1SP3_INTR_EN	0
2	R/W	A1SP2_INTR_EN	0
1	R/W	A1SP1_INTR_EN	0
0	R/W	A1SP0_INTR_EN	0

The above enable bits control the corresponding interrupt bits in the MSTR_INTR_REG. When an enable bit is set to a logic 1, the corresponding error event will cause INTB to go active.

Register 0x81110, … 13: A1SPn Interrupt Enable Register (A1SPn_EN_REG)

Bit	Type	Function	Default
15	RO	Unused	X
14	RO	Unused	X
13	RO	Unused	X
12	RO	Unused	X
11	RO	Unused	X
10	RO	Unused	X
9	RO	Unused	X
8	RO	Unused	X
7	RO	Unused	X
6	R/W	TALP_FIFO_FULL	0
5	R/W	RSTAT_FIFO_FULL	0
4	R/W	RSTAT_FIFO_EMPB	0
3	R/W	TIDLE_FIFO_FULL	0
2	R/W	TIDLE_FIFO_EMPB	0
1	R/W	OAM_INTR	0
0	R/W	FR_ADV_FIFO_FULL	0

The above enable bits control the corresponding interrupt bits in the A1SPn_INTR_REG. When an enable bit is set to a logic 1, the corresponding error event will cause the A1SPn_INTR bit to be set in the MSTR_INTR_REG.

Register 0x81140, …, 43: Receive(n) Status FIFO Enable Register (RSTAT_EN_REG)

Bit	Type	Function	Default
15:6	RO	Unused	X
5	R/W	R_LINE_RESYNC_EN	0
4	R/W	T_LINE_RESYNC_EN	0
3	R/W	BITMASK_CHANGE_EN	0
2	R/W	EXIT_UNDERRUN_EN	0
1	R/W	ENTER_UNDERRUN_EN	0
0	R/W	RECEIVE_QUEUE_ERR_EN	0

The above enable bits control the corresponding status bits in the RCVn_STAT_FIFO. When an enable bit is set to a logic 1, the corresponding receive status event will cause an entry to be made into the RCVn_STAT_FIFO. This mask applies to all receive queues.

Register 0x81150, …, 53: Receive(n) Queue Error Enable (RCV_Q_ERR_EN)

Bit	Type	Function	Default
15	R/W	Reserved	0
14	R/W	CELL_RECEIVED	0
13	R/W	DBCES_BM_ERR	0
12	R/W	PTR_RULE_ERROR	0
11	R/W	ALLOC_TBL_BLANK	0
10	R/W	POINTER_SEARCH	0
9	R/W	FORCED_UNDERRUN	0
8	R/W	SN_CELL_DROP	0
7	R/W	POINTER_RECEIVED	0
6	R/W	PTR_PARITY_ERR	0
5	R/W	SRTS_RESUME	0
4	R/W	SRTS_UNDERRUN	0
3	R/W	RESUME	0
2	R/W	PTR_MISMATCH	0
1	R/W	OVERRUN	0
0	R/W	UNDERRUN	0

The above enable bits 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.

13.8 Idle Channel Detection Configuration and Status Registers

These registers control how idle channel detection is configured for the AAL1gator-32 and indicate active/idle channel status for all channels on the chip.

Table 36 Idle Channel Detection Configuration and Status Registers Memory Map

Address	Register Description	Register Mnemonic
0x82000 – 0x8200F	A1SP 0 Receive Channel Active Table	RX_ACTIVE_TBL0
0x82010 – 0x8201F	A1SP 0 Receive Pending Channel Table	RX_PENDING_TBL0
0x82100 – 0x821FF	A1SP 0 Change Pointer Table	RX_CHANGE_PTR0
0x82200 – 0x8220F	A1SP 0 Transmit Channel Active Table	TX_ACTIVE_TBL0
0x82210 – 0x82217	A1SP 0 Pattern Matching Line Configuration	PAT_MTCH _CFG0
0x82220	A1SP 0 Idle Detection Configuration Table	IDLE_CFG_TBL0
0x82300 – 0x823FF	A1SP 0 CAS/Pattern Matching Configuration Table	CAS_P_CFG_TBL0
0x82400 – 0x8240F	A1SP 1 Receive Channel Active Table	RX_ACTIVE_TBL1
0x82410 – 0x8241F	A1SP 1 Receive Pending Channel Table	RX_PENDING_TBL1
0x82500 – 0x825FF	A1SP 1 Change Pointer Table	RX_CHANGE_PTR1
0x82600 – 0x8260F	A1SP 1 Transmit Channel Active Table	TX_ACTIVE_TBL1
0x82610 – 0x82617	A1SP 1 Pattern Matching Line Configuration	PAT_MTCH _CFG1
0x82620	A1SP 1 Idle Detection Configuration Table	IDLE_CFG_TBL1
0x82700 – 0x827FF	A1SP 1 CAS/Pattern Matching Configuration Table	CAS_P_CFG_TBL1
0x82800 –	A1SP 2 Receive Channel Active Table	RX_ACTIVE_TBL2

Address	Register Description	Register Mnemonic
0x8280F		
0x82810 – 0x8281F	A1SP 2 Receive Pending Channel Table	RX_PENDING_TBL2
0x82900 – 0x829FF	A1SP 2 Change Pointer Table	RX_CHANGE_PTR2
0x82A00 – 0x82A0F	A1SP 2 Transmit Channel Active Table	TX_ACTIVE_TBL2
0x82A10 – 0x82A17	A1SP 2 Pattern Matching Line Configuration	PAT_MTCH _CFG2
0x82A20	A1SP 2 Idle Detection Configuration Table	IDLE_CFG_TBL2
0x82B00 – 0x82BFF	A1SP 2 CAS/Pattern Matching Configuration Table	CAS_P_CFG_TBL2
0x82C00 – 0x82C0F	A1SP 3 Receive Channel Active Table	RX_ACTIVE_TBL3
0x82C10 – 0x82C1F	A1SP 3 Receive Pending Channel Table	RX_PENDING_TBL3
0x82D00 – 0x82DFF	A1SP 3 Change Pointer Table	RX_CHANGE_PTR3
0x82E00 – 0x82E0F	A1SP 3 Transmit Channel Active Table	TX_ACTIVE_TBL3
0x82E10 – 0x82E17	A1SP 3 Pattern Matching Line Configuration	PAT_MTCH _CFG3
0x82E20w	A1SP 3 Idle Detection Configuration Table	IDLE_CFG_TBL3
0x82F00 – 0x82FFF	A1SP 3 CAS/Pattern Matching Configuration Table	CAS_P_CFG_TBL3

Register 0x82000-0x8200F + 0x400*N (N=0-3): A1SP N RX Channel Active Table

This table contains the receive active channel status for A1SP N block which contains 8 lines of 32 channels (8 x 32 = 256 channels total). 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.

This table should be initialized to all zeros.

LINE = OFFSET (MOD 32) / 2

CHANNEL (ADDRESS) = OFFSET (MOD 2)

CHANNEL (BIT LOCATION) = CHANNEL (MOD 16)

ADDRESS	A1SP	LINE	CHANNEL
0x82000	0	0	RX_CHAN_ACTIVE[15:0]
0x82001	0	0	RX_CHAN_ACTIVE[31:16]
..	0	….	RX_CHAN_ACTIVE[15:0]
..	0	….	RX_CHAN_ACTIVE[31:16]
0x8200E	0	7	RX_CHAN_ACTIVE[15:0]
0x8200F	0	7	RX_CHAN_ACTIVE[31:16]

Bit	Type	Function	Default
15	RO	RX_CHAN_ACTIVE_15	X
14	RO	RX_CHAN_ACTIVE_14	X
13	RO	RX_CHAN_ACTIVE_13	X
12	RO	RX_CHAN_ACTIVE_12	X
11	RO	RX_CHAN_ACTIVE_11	X
10	RO	RX_CHAN_ACTIVE_10	X
9	RO	RX_CHAN_ACTIVE_9	X
8	RO	RX_CHAN_ACTIVE_8	X
7	RO	RX_CHAN_ACTIVE_7	X
6	RO	RX_CHAN_ACTIVE_6	X
5	RO	RX_CHAN_ACTIVE_5	X
4	RO	RX_CHAN_ACTIVE_4	X
3	RO	RX_CHAN_ACTIVE_3	X
2	RO	RX_CHAN_ACTIVE_2	X
1	RO	RX_CHAN_ACTIVE_1	X
0	RO	RX_CHAN_ACTIVE_0	X

RX_CHAN_ACTIVE_n

This one bit 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:

0) Inactive
1) Active

Register 0x82010-0x8201F + 0x400*N (N=0-3): A1SP N RX Pending Table

This table contains the receive pending channel status for A1SP N block which contains 8 lines of 32 channels (8 x 32 = 256 channels total). 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)

ADDRESS	LINE	CHANNEL
0x000	0	RX_PENDING[15:0]
0x001	0	RX_PENDING [31:16]
..	….	RX_PENDING [15:0]
..	….	RX_PENDING E[31:16]
0x00E	7	RX_PENDING [15:0]
0x00F	7	RX_PENDING E[31:16]

Bit	Type	Function	Default
15	RO	RX_PENDING_15	X
14	RO	RX_PENDING_14	X
13	RO	RX_PENDING_13	X
12	RO	RX_PENDING_12	X
11	RO	RX_PENDING_11	X
10	RO	RX_PENDING_10	X
9	RO	RX_PENDING_9	X
8	RO	RX_PENDING_8	X
7	RO	RX_PENDING_7	X
6	RO	RX_PENDING_6	X
5	RO	RX_PENDING_5	X
4	RO	RX_PENDING_4	X
3	RO	RX_PENDING_3	X
2	RO	RX_PENDING_2	X
1	RO	RX_PENDING_1	X
0	RO	RX_PENDING_0	X

RX_PENDING_n

This one bit 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

0) No change in state is pending for this channel
1) A state change is pending for this channel

Register 0x82100-0x821FF + 0x400*N (N=0-3): A1SP N RX Change Pointer Table (RX_CHG_PTR)

This 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.

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)

ADDRESS	A1SP	LINE	CHANNEL
0x82100	0	0	CHANGE_PTR_0
0x82101	0	0	CHANGE_PTR_1
..	0	….	….
..	0	….	….
0x821FE	0	7	CHANGE_PTR _30
0x821FF	0	7	CHANGE_PTR _31

CHANGE_PTR_n

Bit	Type	Function	Default
15:10	R/W	Unused	X
9	R/W	ACTIVE	X
8:0	R/W	FRAME_PTR	X

FRAME_PTR_

Indicates the value of frame pointer in which the active status of the channel should change to the value indicated in ACTIVE.

ACTIVE

Indicates the state which should become current at the frame indicated by FRAME_PTR

0) The channel should change to an inactive state.
1) The channel should change to an active state.

Register 0x82200-0x8220F + 0x400*N (N=0-3): A1SP N TX Channel Active Table

This table contains the transmit active channel status for A1SP N which contains 8 lines of 32 channels (4 x 8 x 32 = 1024 channels total). 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 zeros.

LINE = OFFSET (MOD 32) / 2

CHANNEL (ADDRESS) = OFFSET (MOD 2)

CHANNEL (BIT LOCATION) = CHANNEL (MOD 16)

ADDRESS	A1SP	LINE	CHANNEL
0x82200	0	0	TX_CHAN_ACTIVE[15:0]
0x82201	0	0	TX_CHAN_ACTIVE[31:16]
..	0	….	TX_CHAN_ACTIVE[15:0]
..	0	….	TX_CHAN_ACTIVE[31:16]
0x8220E	0	7	TX_CHAN_ACTIVE[15:0]
0x8220F	0	7	TX_CHAN_ACTIVE[31:16]

Bit	Type	Function	Default
15	R/W	TX_CHAN_ACTIVE_15	X
14	R/W	TX_CHAN_ACTIVE_14	X
13	R/W	TX_CHAN_ACTIVE_13	X
12	R/W	TX_CHAN_ACTIVE_12	X
11	R/W	TX_CHAN_ACTIVE_11	X
10	R/W	TX_CHAN_ACTIVE_10	X
9	R/W	TX_CHAN_ACTIVE_9	X
8	R/W	TX_CHAN_ACTIVE_8	X
7	R/W	TX_CHAN_ACTIVE_7	X
6	R/W	TX_CHAN_ACTIVE_6	X
5	R/W	TX_CHAN_ACTIVE_5	X
4	R/W	TX_CHAN_ACTIVE_4	X
3	R/W	TX_CHAN_ACTIVE_3	X
2	R/W	TX_CHAN_ACTIVE_2	X
1	R/W	TX_CHAN_ACTIVE_1	X
0	R/W	TX_CHAN_ACTIVE_0	X

TX_CHAN_ACTIVE_n

This one bit 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”b (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:

0) Inactive
1) 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.

Register 0x82210-0x82217 + 0x400*N (N=0-3): A1SP N Pattern Matching Line Configuration (PAT_MTCH_CFG0)

This 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.

LINE = OFFSET

ADDRESS	LINE	BIT	CHANNEL
0x82210	0	15:8	Reserved
0x82210	0	7:0	INTVL_LEN
..	….		….
..	….		….
0x82217	7	15:8	Reserved
0x82217	7	7:0	INTVL_LEN

INTVL_LEN(7:0)

This 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).

Register 0x82220 + 0x400*N (N=0-3): A1SP N Idle Detection Configuration Table

This table contains the idle detection configuration for the A1SP N block which contains 8 lines The configuration for each line is composed of a 2 bit field (IDLE_CONFIG) and therefore the register contains the configuration for 8 lines. The structure of the table is shown below.

ADDRESS	A1SP	LINE Configuration
0x82220	0	IDLE_CFG[7:0]

Bit	Type	Function	Default
15:14	R/W	IDLE_CFG_7	X
13:12	R/W	IDLE_CFG_6	X
11:10	R/W	IDLE_CFG_5	X
9:8	R/W	IDLE_CFG_ 4	X
7:6	R/W	IDLE_CFG_ 3	X
5:4	R/W	IDLE_CFG_ 2	X
3:2	R/W	IDLE_CFG_1	X
1:0	R/W	IDLE_CFG_0	X

IDLE_CFG_n

This two bit field defines the idle detection mode. There are four possible modes: idle detection disabled, processor controlled activation/deactivation of channels, automatic activation/deactivation of channels using CAS matching, and automatic activation/deactivation of channels using pattern matching.

00: Idle Detection Disabled
01: Processor
10: Automatic, CAS Matching
11: Automatic, Pattern Matching

Register 0x82300-0x823FF + 0x400*N (N=0-3): A1SP N CAS/Pattern Matching Configuration Table

This 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 8 lines of 32 channels (8 x 32 = 256 channels total per A1SP). 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)

ADDRESS	A1SP	LINE	CHANNEL
0x82300	0	0	AUTO_CONFIG_0/PROC_CONFIG_0
0x82301	0	0	AUTO_CONFIG_1/PROC_CONFIG_1
..	0	….	….
..	0	….	….
0x823FE	0	7	AUTO_CONFIG_30/PROC_CONFIG_30
0x823FF	0	7	AUTO_CONFIG_31/PROC_CONFIG_31

AUTO_CONFIG_n/PROC_CONFIG_n

AUTO_CONFIG 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 = “01” the field uses the PROC_CONFIG_n format which is different than the AUTO_CONFIG_n format.

AUTO_CONFIG/PROC_CONFIG should only be updated while IDLE_CFG = “00”. Once AUTO_CONFIG/PROC_CONFIG is configured correctly, then IDLE_CFG should be put into the desired mode.

The format follows the processor idle detection format shown below.

CAS Matching (IDLE_CFG_n = “10”)

Bit	Type	Function	Default
15:12	R/W	RX_MASK	X
11:8	R/W	TX_MASK	X
7:4	R/W	RX_CAS	X
3:0	R/W	TX_CAS	X

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.

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.

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.

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.

Pattern Matching (IDLE_CFG_n = “11”)

Bit	Type	Function	Default
15:8	R/W	PAT_MASK	X
7:0	R/W	IDLE_PATTERN	X

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.

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.

Processor Idle Detection (IDLE_CFG = “01”)

Bit	Type	Function	Default
15:12	R/W	RX_MASK	X
11:8	R/W	TX_MASK	X
7:4	R	Reserved	X
3:0	R	Reserved	X

Reserved

Reserved for internal use. Initialize to ‘0’.

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.

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.

13.9 DLL Control and Status Registers

These registers allow the DLL to be configured and indicate the current status of the DLL.

Table 37 DLL Control and Status Registers Memory Map

Address	Register Description	Register Mnemonic
0x84000	DLL Configuration Register	DLL_CFG_REG
0x84001	Reserved	
0x84002	DLL SW Reset Register	DLL_SW_RST_REG
0x84003	DLL Control Status Register	DLL_STAT_REG

Register 0x84000H: DLL Configuration Register (DLL_CFG_REG)

Bit	Type	Function	Default
15:6		Unused	X
Bit 5		Reserved	0
Bit 4	R/W	OVERRIDE	0
Bit 3		Unused	X
Bit 2		Reserved	0
Bit 1		Reserved	0
Bit 0		Reserved	0

The DLL Configuration Register controls the basic operation of the DLL.

OVERRIDE:

The override control (OVERRIDE) disables the DLL operation. When OVERRIDE is set low, the DLL generates the clock by delaying the SYS_CLK until the rising edge of internal SYS_CLK occurs at the same time as the rising edge of external SYS_CLK. When OVERRIDE is set high, the clock output is a buffered version of the SYS_CLK input.

Note that when clearing the OVERRIDE bit, the DLL should be reset so that it acquires sync cleanly. The RUN bit is only cleared by a DLL reset so it will give a false indication if the DLL is not reset.

Register 0x84002H: DLL SW Reset Register (DLL_SW_RST_REG)

Bit	Type	Function	Default
15:8		Unused	X
7:0	R	TAP[7:0]	X

Writing to this register performs a software reset of the DLL. A software reset requires a maximum of 24*256 SYS_CLK cycles for the DLL to regain lock. During this time the DLLCLK phase is adjusting from its current position to delay tap 0 and back to a lock position. Check the RUN bit to see when LOCK has occurred.

TAP[7:0]:

The tap status register bits (TAP[7:0]) specifies the delay line tap the DLL is using to generate the outgoing clock. When TAP[7:0] is logic zero, the DLL is using the delay line tap with minimum phase delay. When TAP[7:0] is equal to 255, the DLL is using the delay line tap with maximum phase delay.

Register 0x84003H: DLL Control Status Register (DLL_STAT_REG)

Bit	Type	Function	Default
Bit 7	R	SYS_CLKI	X
Bit 6	R	INT_SYS_CLKI	X
Bit 5	R	ERRORI	X
Bit 4	R	CHANGEI	X
Bit 3		Unused	X
Bit 2	R	ERROR	X
Bit 1	R	CHANGE	0
Bit 0	R	RUN	0

The DLL Control Status Register provides information of the DLL operation.

RUN:

The DLL lock status register bit (RUN) indicates the DLL found a delay line tap in which the phase difference between the rising edge of Internal SYS_CLK and the rising edge of external SYS_CLK is zero. After system reset, RUN is logic zero until the phase detector indicates an initial lock condition. When the phase detector indicates lock, RUN is set to logic 1. The RUN register bit is cleared only by a hardware reset or a DLL software reset (writing to DLL_SW_RST_REG). This bit should be polled when taking the chip out of reset. No other operations should take place until RUN is set.

CHANGE:

The delay line tap change register bit (CHANGE) indicates the DLL has moved to a new delay line tap. CHANGE is set high for eight SYS_CLK cycles when the DLL moves to a new delay line tap.

ERROR:

The delay line error register bit (ERROR) indicates the DLL has run out of dynamic range. When the DLL attempts to move beyond the end of the delay line, ERROR is set high. When ERROR is high, the DLL cannot generate a clock phase which causes the rising edge of internal SYS_CLK to be aligned to the rising edge of external SYS_CLK. ERROR is set low, when the DLL captures lock again.

CHANGEI:

The delay line tap change event register bit (CHANGEI) indicates the CHANGE register bit has changed value. When the CHANGE register changes from a logic zero to a logic one, the CHANGEI register bit is set to logic one. The CHANGEI register bit is cleared immediately after it is read, thus acknowledging the event has been recorded.

ERRORI:

The delay line error event register bit (ERRORI) indicates the ERROR register bit has gone high. When the ERROR register changes from a logic zero to a logic one, the ERRORI register bit is set to logic one. The ERRORI register bit is cleared immediately after it is read, thus acknowledging the event has been recorded.

INT_SYS_CLKI:

The reference clock event register bit INT_SYS_CLKI provides a method to monitor activity on the reference clock. When the internal SYS_CLK primary input changes from a logic zero to a logic one, the INT_SYS_CLKI register bit is set to logic one. The INT_SYS_CLKI register bit is cleared immediately after it is read, thus acknowledging the event has been recorded.

SYS_CLKI:

The system clock event register bit SYSLCKI provides a method to monitor activity on the system clock. When the SYS_CLK primary input changes from a logic zero to a logic one, the SYS_CLKI register bit is set to logic one. The SYS_CLKI register bit is cleared immediately after it is read, thus acknowledging the event has been recorded.

14 OPERATION

This section discusses procedures for setting up or configuring different functions of the AAL1gator-32.

14.1 Hardware Configuration

The AAL1gator-32 can be configured in several different modes. The line mode of operation needs to be setup from hardware reset and cannot be changed once the chip is powered up. The line mode is controlled by the LINE_MODE pins. These pins will determine whether the line interface supports 16 low speed lines, 2 high speed lines, 8 H-MVIP bi-directional lines or the SBI bus. The mode will also determine if a second ram interface is available. See the description of the Line interface for more details.

The UTOPIA interface can also be set up in different modes. Because different modes require different sides of the bus driving the control signals, the UTOPIA will power up with all outputs tri-stated. If it is desired at the system level to pull some of the control signal so that they default to one way or the other upon power-up, this can be done using weak pull-up or pull-down resistors. The UTOPIA interface will remain tri-state until the UI_EN bit in the UI_COMN_CFG register is set.

The AAL1gator-32 can either generate the transmit line clocks internally or use clocks supplied to its transmit line clock inputs. Because the device does not know upon power up which mode will be used, there is a TLCLK_OE signal which can be used to tell the chip to generate clocks or not generate clocks. If this pin is tied low the chip will not generate a clock until it is configured to source a clock. If this pin is tied high the chip will use the clock provided on its RL_CLK pin as it TL_CLK and will drive this clock externally. Note the option to drive clocks is only available in Direct Low Speed mode.

14.2 Start-Up

The AAL1gator-32 uses an internal DLL on SYS_CLK to maintain low skew on the ram interface. When the chip is taken out of hardware reset, the DLL will go into hunt mode and will adjust the internal SYS_CLK until it aligns with the external SYS_CLK. The microprocessor should poll the RUN bit in DLL_STAT_REG until this bit is set.

At this point the entire chip with the exception of the microprocessor interface and the DLL are in reset. Before any configuration can be done, including accessing the ram, the chip must be taken out of software reset by clearing the

SW_RESET bit in the DEV_ID_REG. Once taken out of reset RAM1 (and RAM2 if used) should be cleared to all zeros. At this point, the A1SP blocks are still in reset because their respective An_SW_RESET bits in their An_CMD_REG registers are still set. The UTOPIA interface is disabled and all UTOPIA outputs are tri-stated because the UI_EN bit in the UI_COMN_CFG register is not set. The line interface is configured in the mode indicated by the LINE_MODE pins but all internal registers are in their reset state. The Line Interface is out of reset at this point but will only be driving data as if all lines and/or queues are disabled.

14.2.1 Line Configuration

If in SBI mode the internal rams go through an internal initialization process. The BUSY bit in the Insert/Extract Tributary RAM Indirect Access Control Registers should be polled until low. Once low the SBI registers can be configured.

There are line interface registers for both Direct Low Speed Mode and SBI mode. These registers should be set up before the A1SPs are taken out of reset.

While the A1SPs are in reset the memory mapped registers which contain the line configuration (LIN_STR_MODE and HS_LIN_REG) can be initialized. Note that the R_CHAN_2_QUE_TBL registers and R_STATE_0 and R_LINE_STATE registers cannot be accessed because they are internal and are being held in reset.

Once the line is initialized and the LIN_STR_MODE and HS_LIN_REG memory registers are initialized the CMD_ATTN bit in the An_CMD_REG bit can be set so that the A1SPs can read their configuration. The An_SW_RESET bit should remain set. Note that each A1SP has a separate register.

See Line Configuration Details below for more details.

14.2.2 Queue Configuration

Once this is complete the An_SW_RESET bit in the An_CMD_REG can be cleared which will take each A1SP out of reset. The R_CHAN_2_QUE_TBL will then begin a 640 SYS_CLK cycle initialization, which reset each timeslot to playing out conditioned data. At this point the queues can be initialized as needed.

14.2.3 Adding Queues

Queues are added by writing to the ADDQ_FIFO with the number of the queue to be added. There is one add queue FIFO per A1SP. See Processor Interface section for more details

14.2.4 Line Configuration Details

This section is intended to be a guide for programmers.

14.2.4.1 Mode Selection

The mode of the Line Interface Block is controlled by the LINE_MODE input pins, which are also readable, by software via a read-only LINE_MODE register. These pins should be tied to a certain level through initial hardware reset and not be changed while out of the reset state. These pins also control the mode of the entire Line Interface block, so the entire block is either in Low Speed Direct mode, SBI mode, H-MVIP mode, or High Speed mode.

14.2.4.2 Direct Low Speed Mode

The options available in Direct Low Speed Mode are:

- Per line synchronization: Frame or multiframe basis, and internal control or externally controlled
- Per line format: PMC standard format or MVIP format.

14.2.4.2.1 Synchronization

Synchronization can be configured on a per line basis, and is controlled by the MF_SYNC_MODE bit in the Low Speed Configuration Register (LS_Ln_CFG_REG), and the GEN_SYNC bit in the LIN_STR_MODE memory register for each line.

Synchronization can either be done on a multi-frame basis or a frame basis. If multi-frame synchronization is required then MF_SYNC_MODE bit for that line in the LS_Ln_CFG_REG must be set. Otherwise, if frame synchronization or no synchronization is required then leave MF_SYNC_MODE bit clear as default.

These values should be configured before A1SP software reset is released.

In the receive direction, synchronization is always controlled by the external line interface. However, in the transmit direction, synchronization can be controlled from either the local link side or the external line side. Set GEN_SYNC if the local link side is controlling synchronization. Otherwise, if GEN_SYNC is low, then the external lines are controlling synchronization or no synchronization is required.

14.2.4.2.2 Line Format

There are two choices of line format: 1) PMC standard format, and 2) MVIP format.

The format can be controlled on a per line basis. If MVIP_EN is set in the Low Speed Configuration Register for that line (LS_Ln_CFG_REG), then the line is in MVIP-90 mode. Otherwise the line is in PMC standard format.

This value should be configured before A1SP software reset is released.

Note that if a mixture of MVIP-90 lines and non MVIP-90 lines are used then line 0 must be MVIP-90.

If lines are configured in MVIP-90 mode then TL_SYNC0 becomes the F0B input (125-us frame sync signal) and CRL_CLK becomes the C4B clock (4.096 MHz).

AAL1gator-32 samples F0B on a C4B falling edge, and expects F0B to be exactly one C4B clock cycle wide, but F0B need not mark every 125 us frame. AAL1gator-32 samples RL_DATA and RL_SIG at the 3/4 point in the MVIP-90 bit-period, which is a C4B rising edge. AAL1gator drives TL_DATA and TL_SIG at the C4B falling edge at the start of the MVIP-90 bit-period.

Set LIN_STR_MODE_n=0x0001 (no CAS) or LIN_STR_MODE_n=0x0003 (with CAS) for line 0 in A1SP 0, and any other MVIP-90 lines.

14.2.4.3 H-MVIP Mode

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 for all lines when this mode is in use.

When in H-MVIP mode, the line should be configured to be in normal mode, by clearing MVIP_EN bit in the corresponding LS_Ln_CFG_REG. In the LIN_STR_MODE register, the following bits should be disabled (value = ‘0’), LOW_CDV, E1_WITH_T1_SIG, T1_MODE, GEN_SYNC, CLK_SOURCE_TX, CLK_SOURCE_RX and SRTS_EN.

14.2.4.4 High Speed Mode

It is important to note that internal links 0 and 16 are connected to external lines 0, and 2 respectively.

The HS_LIN_REG in A1SP0 and A1SP2 control the High Speed functionality. The CLK_SOURCE_TX and CLK_SOURCE_RX fields in the LIN_STR_MODE memory register control the clock mode. In high speed mode only “000” (clock is an input), or “001” (loop timing) modes are permitted.

NOTE: Because internal link 16 is mapped to external line 2, the LIN_STR_MODE memory register for line 2 of A1SP0 and line 0 of A1SP2 must both be initialized and set to the same value.

The An_SW_RESET bit in the An_CMD_REG memory register functions as a queue reset signal in high speed mode. If the state of LOOPBACK_ENABLE in TRANSMIT_CONFIG is desired to be changed, the high speed queue must be reset using the An_SW_RESET bit.

Also when re-activating a highspeed queue, if it is required that the first sequence number of the new connection has to be 0, then the queue must be reset using the An_SW_RESET bit.

Otherwise, clearing and setting the TX_ACTIVE bit in QUEUE_CONFIG can be used to deactivate and activate the queue.

14.2.4.5 SBI Mode

In SBI mode the following items need to be properly configured:

- Link type for the upper and lower 16 links.

  > Link type can be E1, T1, DS3 and is configured by setting LINK_TYPH[1:0] or LINK_TYPHL[1:0] in the SBI Link Configuration Register (SBI_LNK_CFG_REG) to the appropriate value. All LINK_TYP default to T1 type.

- The SPE type for each of the 3 SBI SPEs.

  > SPE type can be E1, T1, or DS3 and is configured by setting SPEn_TYP bit in the SBI Bus Configuration Register (SBI_BUS_CFG_REG) to the appropriate value. All SPEs_TYP default to T1 type.

  > Each SPE must be enabled or disabled by writing appropriate value into SPEn_ENBL bit in the SBI Bus Configuration Register. All SPEs default to being enabled.

- Receive Serial Link Clock Selection

- Receive serial clock output to each link can be sourced from INSBI or from PISO, depending upon whether the each individual link is configured to be SYNC mode or ASYNC mode.

- Set SYNC-LINKn bit will configure a link to SYNC mode, in which case serial clock output to that link will be sourced from the Insert SBI Block. Note that SYNC mode must be set if CAS mode is used.

- Clearing SYNC-LINKn will configure a link to ASYNC mode, in which case the serial clock output to that link will be sourced from the PISO block.

  Note that this synchronous or asynchronous selection can be done on per link basis.

- Local Transmit Frame and Multi-frame Pulses Selection

  - In SYNC mode, SYNC-LINKn bit in the SBI_SYNC_LINKn registers must be set. In this mode, transmit frame/multi-frame pulses will be sourced by the INSBI block.

  - GEN_SYNC must = 0 in SBI mode.

- Tributary type for each tributary.

  - The tributary type can be configured to be Structured with CAS, Structured without CAS, or Unstructured. This is configured by setting TRIB_TYP to the appropriate value in the Extract/Insert. Tributary Control Register inside the INSBI/EXSBI for that tributary.

    The TRIB_TYP value inside INSBI/EXSBI is default to Structure with CAS. Note that if Structured with CAS mode is used, the tributary must also be in SYNC mode.

  - A tributary can float or be locked to the SBI frame structure. If the tributary is locked it is considered synchronous. A tributary is synchronous, if either SPEn_SYNCH in the SBI Bus Configuration Register (SBI_BUS_CFG_REG) or the SYNC_TRIB bit in the Insert Tributary Control RAM Indirect Access Data Register for this tributary is set.

    Setting SPEn_SYNCH bit in the SBI_BUS_CFG_REG will configure the entire SPE to be in synchronous mode. To configure individual link to in synchronous, SPEn_SYNCH bit must be clear and the corresponding SYNC_LINK bit must be set. Furthermore,

the SYNCH_TRIB bit in the Insert Tributary Control Indirect Access Data register must also be set accordingly.

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 CLK_MSTR bit in both Extract and Insert Control RAM Indirect Access Data registers

- Tributary Mapping or Disabled
    - Any tributary can be mapped to any link, unless the tributary is DS3. In that case the tributary can only be mapped to link 1 or link 17. Note links numbering starts at ‘1’.
    - The mapping of links to tributaries is controlled by the Extract/Insert Tributary Indirect Address Register and Extract/Insert Tributary Mapping Indirect Address Data Register for that tributary. The SPE field indicates either the SPE1 or SPE2 and upper or lower 16 links. The LINK field indicates which link in the group of 16 is mapped to that tributary.
    - The TS_EN bit in the SBI Bus Configuration Register (SBI_BUS_CFG_REG) must be set to globally enable the mapping. If this bit is set then mapping may occur on a tributary by tributary basis. If this bit is a zero, which is 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 Insert/Extract Control Register, which operates in a similar fashion. To map 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.
    - The TRIB_ENBL bit in the Insert/Extract Tributary Control Indirect Access Data Register for each tributary must be set to enable data to be extracted or inserted from/to that tributary. On power-up, each tributary is disabled by default.
    - The serial clock generated for each link can be disabled by setting the LINKn_DIS bit in the SBI_LINK_DIS_REGn register for that link. All clocks default to being active. All serial clocks can also be disabled on a group of 16 links by setting the CKOUT_KILL bit in the SBI Link Configuration Register (SBI_LNK_CFG_REG)

- DS3:
    - Use the DS3 clock directly and bypass the SBI bus. CLK_SOURCE_TX must be set to “000”.
- Clock Master or Slave Mode:
    - The CLK_MSTR bit determines which side controls the clock. If CLK_MSTR is ‘1’ then the local link side controls the clock, or clock master mode. If CLK_MSTR is ‘0’ then the local link is slave to the SBI, or clock slave mode
    - There is a global CLK_MSTR bit in the SBI Bus Configuration Register (SBI_BUS_CFG_REG), which if set will forces all links to be in clock master mode, and if clear will let it be decided on a per link basis.
    - There is a CLK_MSTR bit in the Extract/Insert Tributary Control Indirect Access Data Register for configuring clock mode for each tributary which is used when the global CLK_MSTR bit is ‘0’. In this case, setting the CLK_MSTR bit will configure that tributary to be a clock master and clearing the bit will configure it to be clock slave.

      When in clock slave mode, the CLK_MODE field in the Extract Tributary Control Indirect Access Data Register determines what type of clocking is used for a particular tributary.

      When in clock slave mode the clocking method can be either monitor SBI buffer depth (using EXT_CKCTL interface), use ClockRate field of EXT_LINKRATE, or use Phase field of EXT_LINKRATE value passed across SBI. Phase mode is recommended.
- SBI Error Checking and Reporting
    - The following errors can be detected on a per link basis: FIFO overrun, FIFO underrun, and SBI parity error. These all can be enabled/disabled in the Insert/Extract Control Register.

      Parity detection is done only on the drop bus, which is done by the EXSBI block. Parity can be configured to be odd or even, and it is default to be odd parity. Parity is checked at all times, including overhead and unused tributaries. It is recommended that one of the Physical Layer SBI devices (e.g. TEMUX) is enabled to drive good parity at all times. (set the equivalent to the BUSMASTER bit).

FIFO overrun/underrun errors are reported by indicating error status in the Extract/Insert FIFO Underrun/Overrun Interrupt Status Register with the failing link identified.

These errors result in either the SBI_DROP_INTR or SBI_ADD_INTR output being set.

- These interrupts can only be cleared by reading the FIFO overrun/underrun register with the failing link. 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.

- When an error is detected on a link, the link must be mapped back to a tributary to determine which tributary had an error.

- If a tributary has an underrun or overrun or depth check error, the tributary must be reset. The tributary will automatically be reset if the DC_EN bit is set AND the SYNC_LINK bit is not set on any tributaries in INSBI. Otherwise the tributary must be reset by the processor by writing the INS_TRIB_CTL or EXT_TRIB_CTL register for that tributary.

- If an SBI Alarm is detected to be changed on a particular link, then the SBI_ALARM output is set to indicate SBI alarm change. SBI_ALRM_STAT bit associated with that link indicates the current state of the Extract SBI ALARM signal on the SBI tributary mapped to this link. SBI_ALRM_INTn bit is also set to indicate that the SBI alarm state has changed on the SBI tributary mapped to this link.

  This SBI_ALRM_INTn interrupt can only be cleared by reading the SBI ALARM Interrupt register.

  The ALARM bit can be inserted into a tributary by setting the SBI_ALRM_INS bit in the SBI Insert Bus Alarm Insert Register associated with the link mapped to that tributary.

- Buffer Depths Control:

  - Buffer depths are controlled via the MIN_DEPTH for T1 and E1 registers, MIN_DEPTH for DS3 registers, and T1, E1, DS3 Threshold registers.

- In most cases, these values can be left at their default settings. However, for T1/E1 applications, when the AAL1gator-32 is a clock master, the Max_Threshold value should be changed to “A”.

- SBI Framing Offsets Control:

  - In the Insert direction the offset between the C1FP pulse and the start of an SBI frame is controlled by the C1FP_OFFSET for T1 Register and C1FP_OFFSET for E1 Register. This register should be left in its default state.

- Enabling Tributaries/SPEs

  - An SPE must not be enabled without any tributaries enabled on it. When initially activating an SPE, enable the tributaries first and then enable the SPE. When deactivating an SPE, disable the SPE first and then deactivate the tributaries.

14.2.4.5.1.1 Programming Sequence for SBI

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

General rules:

- All INSBI/EXSBI read and write accesses must wait for al least 500 us after chip SW_RESET is cleared, to allow time for the INSBI/EXSBI to complete self-initialization. Other general control register outside of INSBI/EXSBI can be configured at any time.

- At initialization, SBI tributary receiver should be enabled before the corresponding SBI tributary transmitter (ie, configuring EXSBI before INSBI)

- 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.

- Mapping of more than one tributary to a link or more than one link to a tributary is not allowed.

- If DC_EN 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 unless SYNCH_TRIB is set on any Insert tributary.

- CLK_MSTR pin TS_EN bits in the SBI_BUS_CFG_REG should be used for strapping purposes only. If programmability is required, these pins should be strapped to the non-overriding state and the register equivalents used to change the Clock Master and the Time Switch Enable states.

INSBI/EXSBI Programming Steps:

1. For each write access, wait until the BUSY bit in the Insert/Extract Tributary RAM Indirect Access Control Register is clear. Note that the BUSY bit might not be ready for up to 4.32 us after either a mapping page switch or a Control RAM or Mapping RAM access.

2. Once the BUSY bit is clear, write to the Insert/Extract Tributary RAM Indirect Access Address Register specifying the SPE and tributary on the SBI side that is about to be mapped onto the link side. If the access is to the Mapping RAM then set bit 7, if the access is to the Control RAM then leave bit 7 clear.

3. If access is to the Mapping RAM, then write into the Extract Tributary Mapping RAM Indirect Access Data register to specify the SPE and link that is to be mapped from the SBI side. If access is to the Control RAM, then write into the Extract Tributary Control RAM Indirect Access Data register to specify the desired control values for that tributary.

4. Next, write into the Extract Tributary RAM Indirect Access Control register to specify whether this is to be a write or a read access, by clearing or setting the WRB bit in this register.

   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.

Tributary Mapping Sequences:

1. Configuring all tributaries in the Mapping RAM in both EXSBI and INSBI first, following the above four steps.
2. Set the global TS_EN bit in the SBI_BUS_CFG_REG.
3. Set the global TS_EN bit in EXSBI/INSBI Control registers.
4. Configure all the Control RAM in both EXSBI and INSBI, following the above four steps.
5. Enable the SPEs last by setting the SPEn_ENBL bits in the SBI_BUS_CFG_REG.

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. For example, all 84 tributaries must be mapped if all 3 SPEs are T1, or all 63 tributaries must be mapped if all 3 SPEs are E1.

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 couple of important notes to be aware of:

1. If any of the 32 Insert links is configured to be in synchronous mode, then the DC_EN bit is ignored and SYNC_INT_EN in the Extract/Insert Control register should be disabled to prevent spurious interrupts. This must be done because Depth Check logic does not support synchronous mode.
2. If any link/tributary within an E1 SPE is in synchronous mode, the C1FP pulses **must not** be 500 us apart (2Khz rate) but 6 ms instead, which is once every 48 SBI frames.

14.3 UTOPIA Interface Configuration

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 enable bit should be set to enable normal operation. The UI_EN bit should be disabled by a chip, which is connected to the AAL1gator via the UTOPIA/AnyPHY bus, prior to its reset or reconfigured.

This disabling is required to prevent deactivation or reconfiguration in the middle of a cell transfer.

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 block can be configured to loop all received cells by setting the U2U_LOOP bit in the UI_COMN_CFG register. Alternately, only cells with a VCI that matches the VCI contained in U2U_LOOP_VCI register can be looped back while other cells proceed through normally. The VCI based UI to UI loopback should be disabled when writing to U2U_LOOP_VCI.

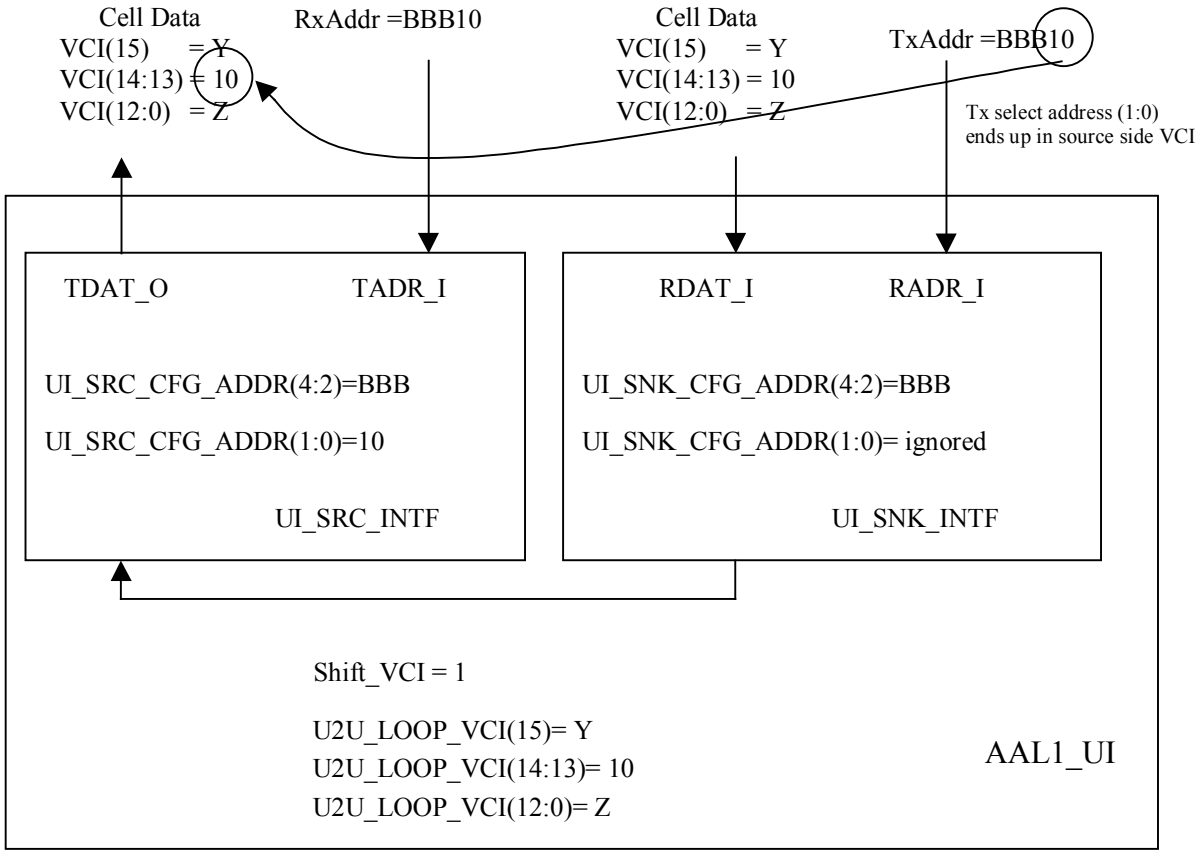
14.3.1 VCI Loopback Setup Example in Multi-Address mode

Internal routing circuitry in Utopia-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-2 multi-address mode is shown in Figure 84. 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 register.

To maintain correct functionality, the cell's VCI(14:13) should match that of the lower bits of the select address TxAddr, ie, 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 84 Utopia-2 Multi-Address Mode with VCI Based Loopback



14.4 Special Queue Configuration Modes

14.4.1 AAL0

AAL0 is a null ATM adaptation layer. In this mode, all 48 payload bytes of the ATM cell contain data. In this mode, there is no data structure, signaling or sequence number. However, AAL0 mode can be enabled for a queue, which is mapped to an entire link, or it can be mapped to a single DS0 queue, or it can be mapped to a group of DS0s.

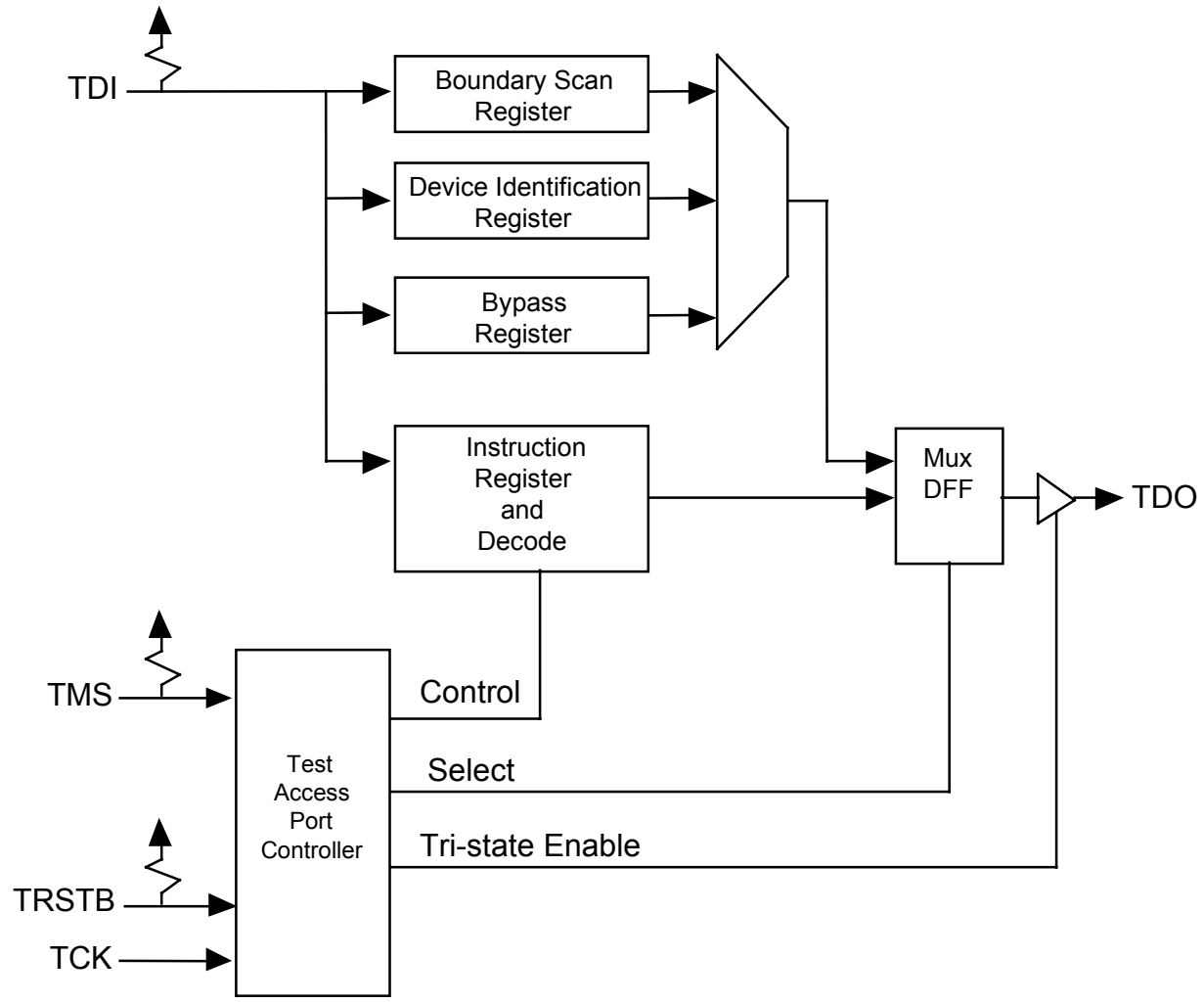
However, it is important to note that because there is no structure, the AAL0 queue will function as unstructured data. If the queue is a nxDS0 queue, the data will maintain byte alignment across the ATM network, but the bytes may not arrive in the same DS0s they were transmitted.

AAL0 mode is enabled by setting AAL0_MODE_ENABLE in the TRANSMIT_CONFIG memory register in the transmit queue table and by setting the R_AAL0_MODE bit in the R_MP_CONFIG memory register in the receive queue table.

14.5 JTAG Support

The AAL1gator-32 supports the IEEE Boundary Scan Specification as described in the IEEE 1149.1 standards. The Test Access Port (TAP) consists of the five standard pins, TRSTB, TCK, TMS, TDI and TDO used to control the TAP controller and the boundary scan registers. The TRSTB input is the active-low reset signal used to reset the TAP controller. TCK is the test clock used to sample data on input, TDI and to output data on output, TDO. The TMS input is used to direct the TAP controller through its states. The basic boundary scan architecture is shown below.

Figure 85 Boundary Scan Architecture



The boundary scan architecture consists of a TAP controller, an instruction register with instruction decode, a bypass register, a device identification register and a boundary scan register. The TAP controller interprets the TMS input and generates control signals to load the instruction and data registers. The instruction register with instruction decode block is used to select the test to be executed and/or the register to be accessed. The bypass register offers a single-bit delay from primary input, TDI to primary output, TDO. The device identification register contains the device identification code.

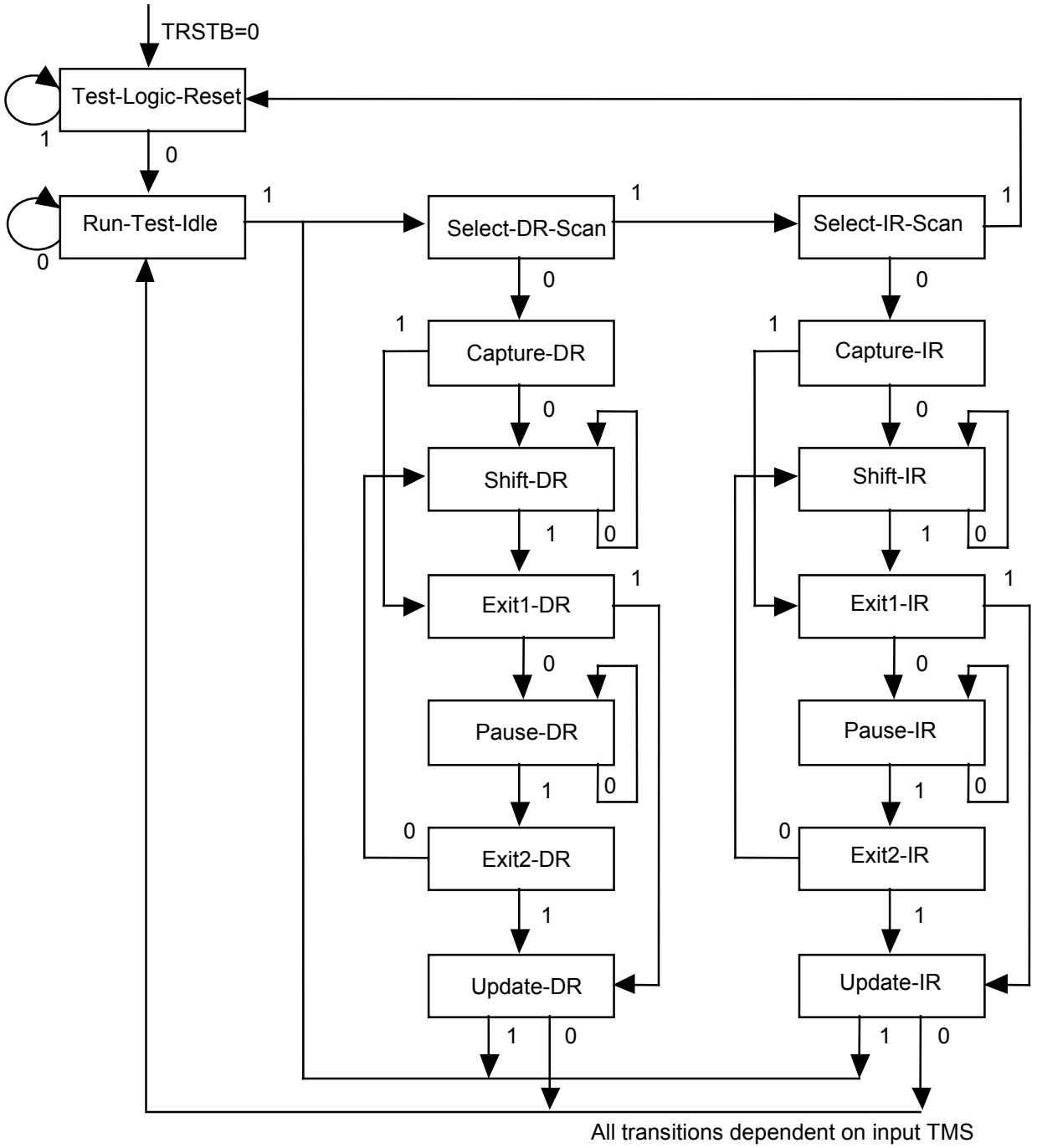
The boundary scan register allows testing of board inter-connectivity. The boundary scan register consists of a shift register place in series with device inputs and outputs. Using the boundary scan register, all digital inputs can be

sampled and shifted out on primary output, TDO. In addition, patterns can be shifted in on primary input, TDI and forced onto all digital outputs.

14.5.1 TAP Controller

The TAP controller is a synchronous finite state machine clocked by the rising edge of primary input, TCK. All state transitions are controlled using primary input, TMS. The finite state machine is described below.

Figure 86 TAP Controller Finite State Machine



Test-Logic-Reset

The test logic reset state is used to disable the TAP logic when the device is in normal mode operation. The state is entered asynchronously by asserting input, TRSTB. The state is entered synchronously regardless of the current TAP controller state by forcing input, TMS high for 5 TCK clock cycles. While in this state, the instruction register is set to the IDCODE instruction.

Run-Test-Idle

The run test/idle state is used to execute tests.

Capture-DR

The capture data register state is used to load parallel data into the test data registers selected by the current instruction. If the selected register does not allow parallel loads or no loading is required by the current instruction, the test register maintains its value. Loading occurs on the rising edge of TCK.

Shift-DR

The shift data register state is used to shift the selected test data registers by one stage. Shifting is from MSB to LSB and occurs on the rising edge of TCK.

Update-DR

The update data register state is used to load a test register's parallel output latch. In general, the output latches are used to control the device. For example, for the EXTEST instruction, the boundary scan test register's parallel output latches are used to control the device's outputs. The parallel output latches are updated on the falling edge of TCK.

Capture-IR

The capture instruction register state is used to load the instruction register with a fixed instruction. The load occurs on the rising edge of TCK.

Shift-IR

The shift instruction register state is used to shift both the instruction register and the selected test data registers by one stage. Shifting is from MSB to LSB and occurs on the rising edge of TCK.

Update-IR

The update instruction register state is used to load a new instruction into the instruction register. The new instruction must be scanned in using the Shift-IR state. The load occurs on the falling edge of TCK.

The Pause-DR and Pause-IR states are provided to allow shifting through the test data and/or instruction registers to be momentarily paused.

Boundary Scan Instructions

The following is an description of the standard instructions. Each instruction selects an serial test data register path between input, TDI and output, TDO.

BYPASS

The bypass instruction shifts data from input, TDI to output, TDO with one TCK clock period delay. The instruction is used to bypass the device.

EXTEST

The external test instruction allows testing of the interconnection to other devices. When the current instruction is the EXTEST instruction, the boundary scan register is place between input, TDI and output, TDO. Primary device inputs can be sampled by loading the boundary scan register using the Capture-DR state. The sampled values can then be viewed by shifting the boundary scan register using the Shift-DR state. Primary device outputs can be controlled by loading patterns shifted in through input TDI into the boundary scan register using the Update-DR state.

SAMPLE

The sample instruction samples all the device inputs and outputs. For this instruction, the boundary scan register is placed between TDI and TDO. Primary device inputs and outputs can be sampled by loading the boundary scan register using the Capture-DR state. The sampled values can then be viewed by shifting the boundary scan register using the Shift-DR state.

IDCODE

The identification instruction is used to connect the identification register between TDI and TDO. The device's identification code can then be shifted out using the Shift-DR state.

STCTEST

The single transport chain instruction is used to test out the TAP controller and the boundary scan register during production test. When this instruction is the current instruction, the boundary scan register is connected between TDI and TDO. During the Capture-DR state, the device identification code is loaded into the boundary scan register. The code can then be shifted out output, TDO using the Shift-DR state.

Boundary Scan Cells

In the following diagrams, CLOCK-DR is equal to TCK when the current controller state is SHIFT-DR or CAPTURE-DR, and unchanging otherwise. The multiplexer in the center of the diagram selects one of four inputs, depending on the status of select lines G1 and G2. The ID Code bit is as listed in the Boundary Scan Register table in the JTAG Test Port section 11.2.

Figure 87 Input Observation Cell (IN_CELL)

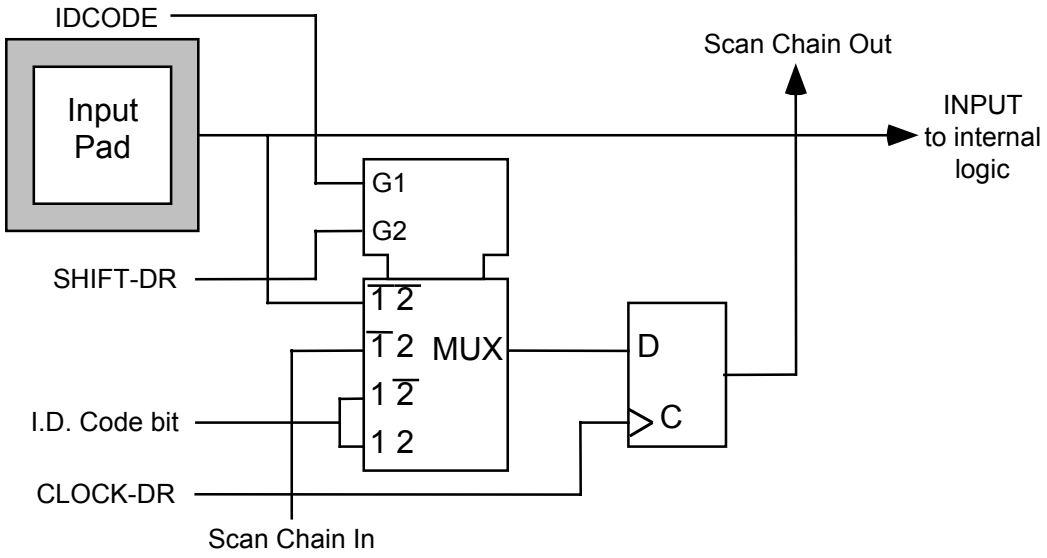


Figure 88 Output Cell (OUT_CELL)

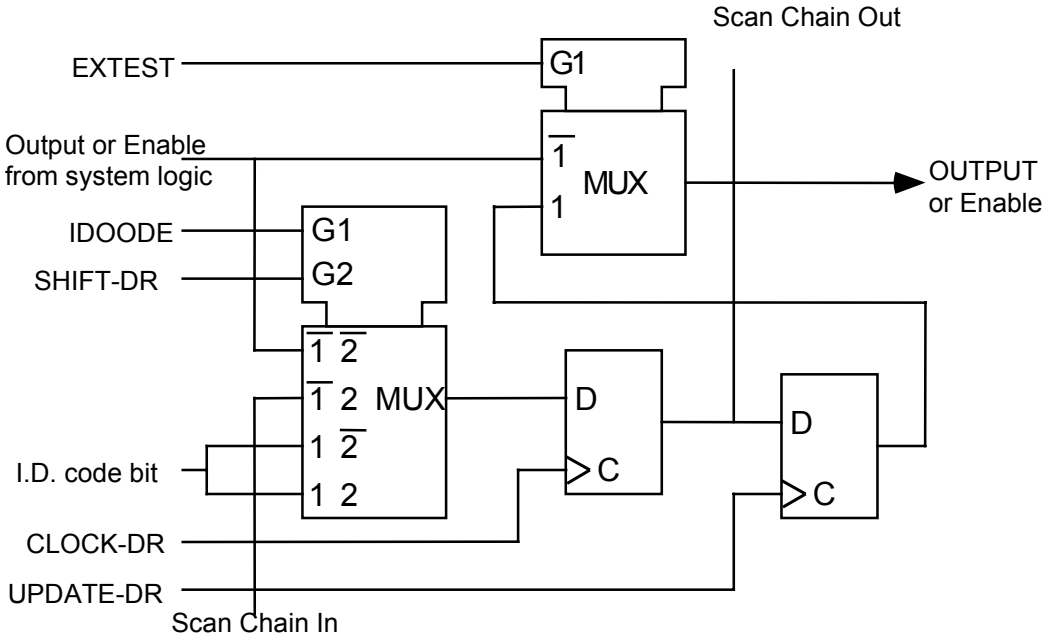


Figure 89 Bidirectional Cell (IO_CELL)

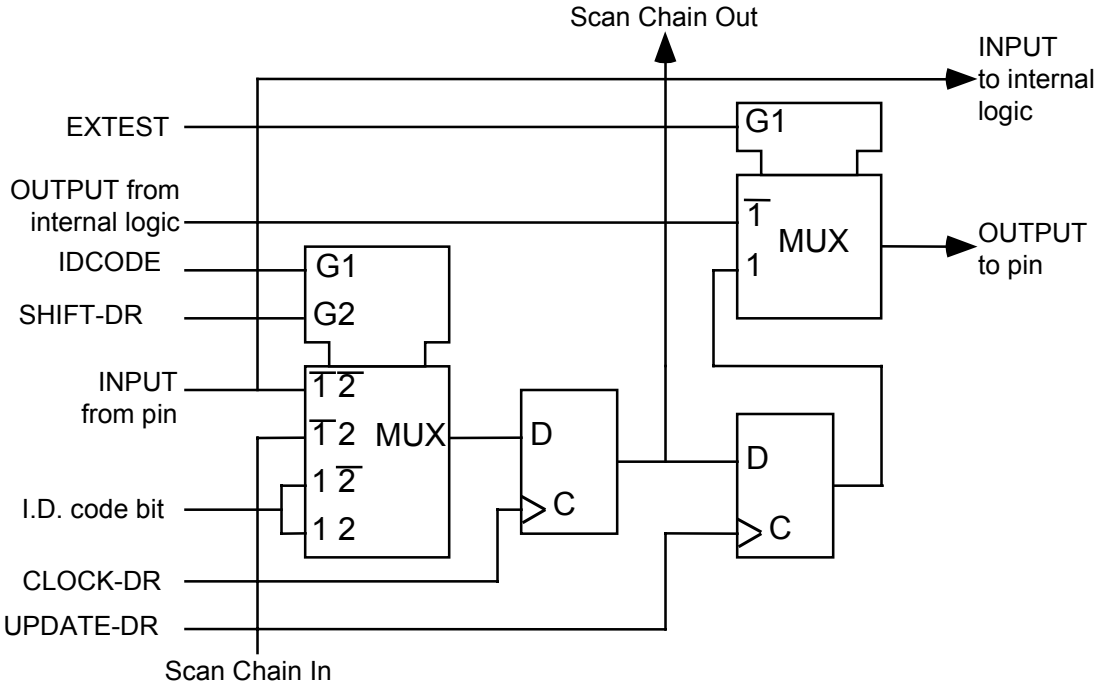
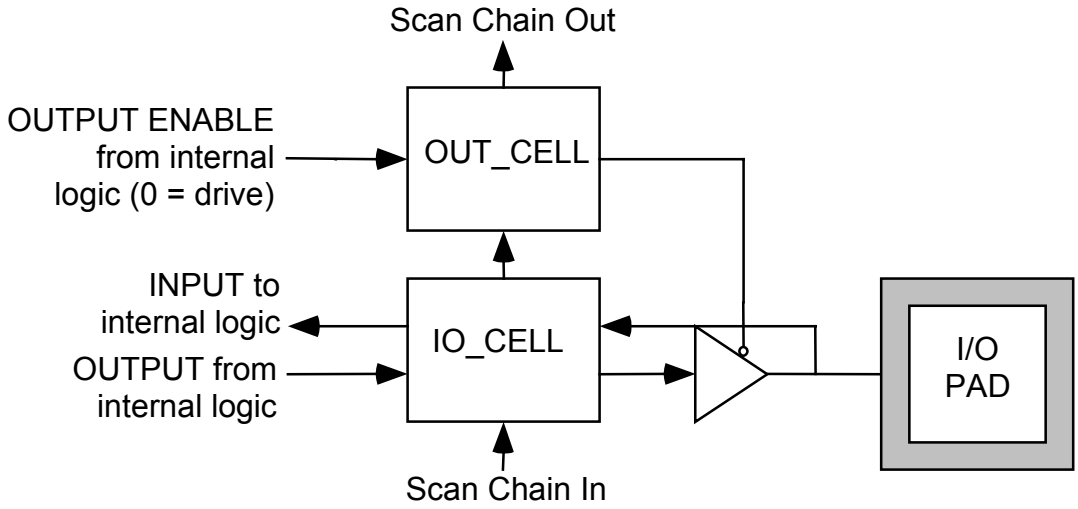


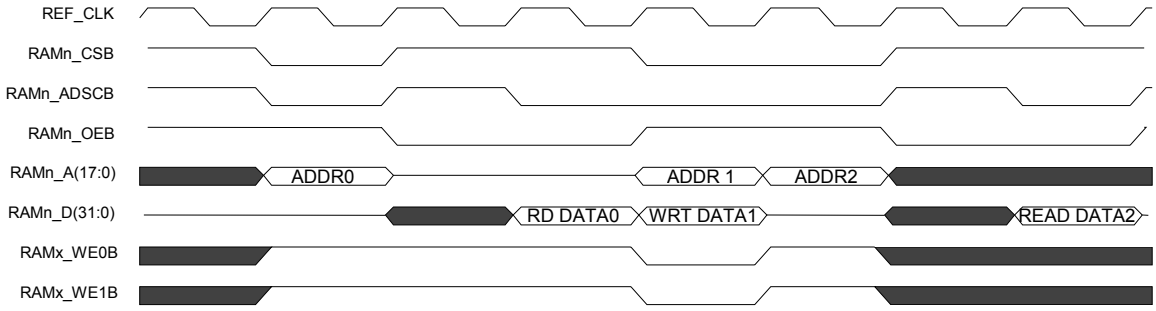
Figure 90 Layout of Output Enable and Bidirectional Cells



15 FUNCTIONAL TIMING

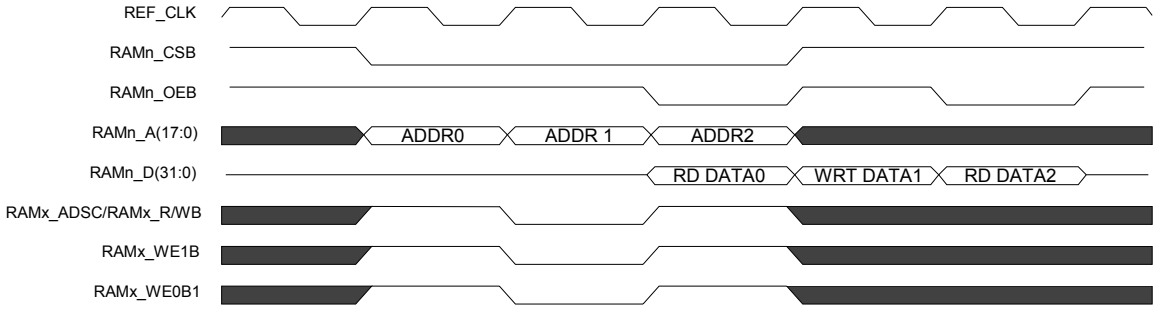
This section shows the functional relationship between inputs and outputs. No propagation delays are shown.

Figure 91 Pipelined Single-Cycle Deselect SSRAM



The diagram above illustrates the SSRAM accesses when it is configured to use a pipelined single-cycle deselect SSRAM. In this mode, ADSC is used and bursting is not used. A new address is presented for each access along with the associated control (CSB, ADSCB, OEB, WEB). The RAMx_ADSCB signal is pulsed during the last cycle of a read prior to a write to place the SSRAM into a deselect state. The deselect state of the SSRAM causes the SSRAM to tristate it's data bus after the rising edge of the clock. This operation reduces contention on the bus when switching from a read to a write operation. The RAMx_OEB is used to prevent bus contention when switching from a write to a read.

Figure 92 Pipelined ZBT SSRAM



The diagram above illustrates the SRAM accesses when it is configured to use the pipelined ZBT SSRAM for improved throughput. In this mode, RAMx_ADSC

is redefined as a RAMx_R/WB and bursting is not used. A new address is presented for each access along with the associated control (CSB, R/WB, OEB, WEB). The write data is placed on the bus 2 cycles after the write address and command The RAMx_OEB is used to prevent bus contention when switching from a write to a read. When switching from a read to a write, some bus contention may be present but is minimal since the ZBTs SSRAMs are quick to disable their buffers and the gator is slower to enable it's buffers.

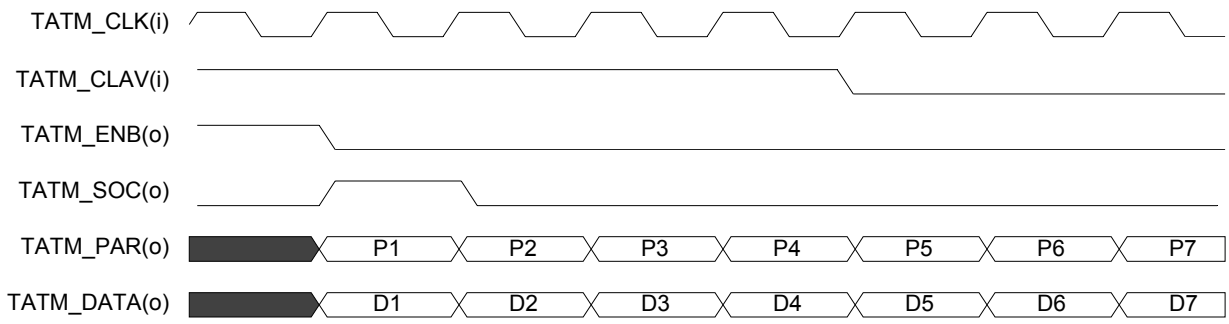
15.1 Source Utopia

In ATM master mode, the SRC_INTF block sources TATM_DATA, TATM_PAR, TATM_SOC, and TATM_ENB while receiving TATM_CLAV. The Start-Of-Cell (TATM_SOC) indication is generated coincident with the first word (8-bit or 16-bit) of each cell that is transmitted on TATM_DATA. TATM_DATA, TATM_PAR, and TATM_SOC are driven at all times. The write enable signal indicates which clock cycles contain valid data for the Utopia bus. The device will not assert the TATM_ENB signal until it has a full cell to send.

Note that during hardware/software reset, or when UI_EN in the UI_COMN_CFG register is deasserted low, all Utopia interface outputs are tri-stated.

In ATM master mode, the SRC_INTF responds to the TATM_CLAV by beginning a cell transfer as shown in Figure 93 below. If TATM_CLAV is asserted and the UI_SRC_INTF has data to send, it will do so by asserting TATM_TENB while driving data. Octet (byte) level handshaking and data transfer pausing is not supported, thus although TPA_I may go low during the middle of a transfer as shown in the Figure 93 example, the data transfer will still continue and TENB_O will not be deasserted.

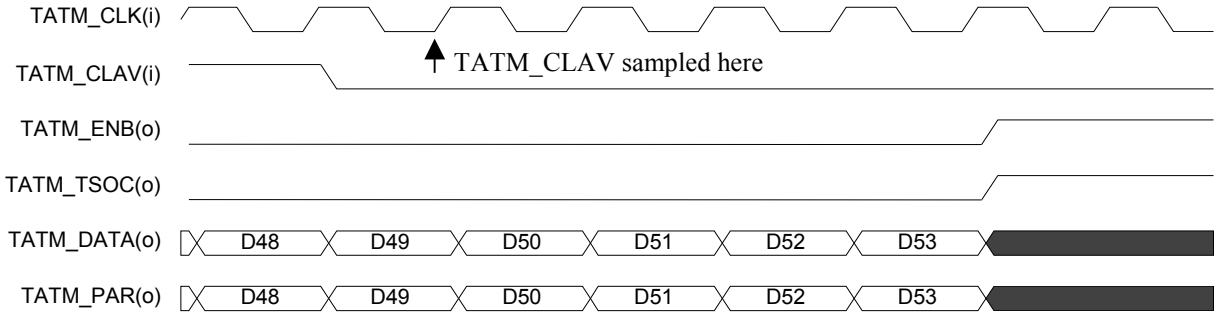
Figure 93 SRC_INTF Start of Transfer Timing (Utopia 1 ATM Mode)



The end of a transfer for the SRC_INTF in Utopia 1 ATM Master mode is shown in Figure 94. Per the Utopia 1 specification, TPA_I must be deasserted low by

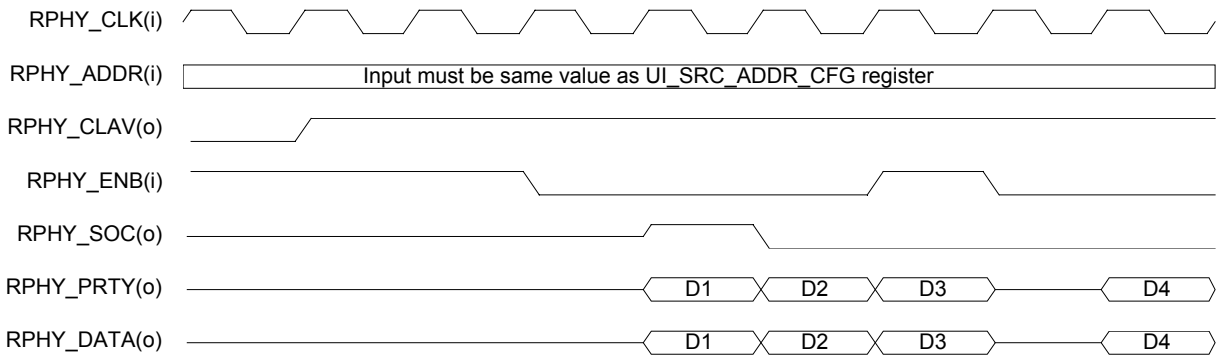
the PHY slave at least four cycles before the end of the cell if the PHY cannot accept another complete cell. Note that one additional cycle is used to allow the input TPA signal to be clocked into the master device, thus the PHY must assert TPA low along with data byte D49 and no later if it cannot accept another cell.

Figure 94 SRC_INTF End-of-Transfer Timing (Utopia 1 ATM Mode)



In PHY slave mode, the SRC_INTF block sources RPHY_DATA, RPHY_PAR, RPHY_SOC, and RPHY_CLAV, while receiving RPHY_ENB. The RPHY_ADDR(4:0) inputs are not used in Utopia 1 PHY slave mode but must be set to be the same value as the UI_SRC_ADDR_CFG register value for proper operation. The RPHY_SOC indication is generated coincident with the first word (8-bit or 16-bit) of each cell that is transmitted on RPHY_DATA. In PHY mode, RPHY_DATA, RPHY_SOC, and RPHY_PAR are driven only when valid data is being sent; otherwise they are tristated.

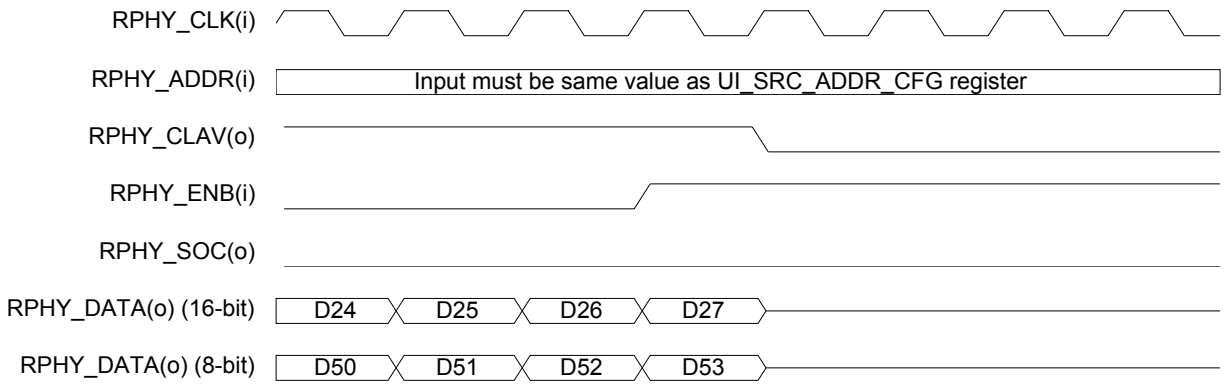
Figure 95 UI_SRC_INTF Start-of-Transfer Timing (Utopia 1 PHY Mode)



The cell available (RPHY_CLAV) signal indicates when the device has a complete cell to send. In Utopia 1 slave (SPHY) mode, RPHY_CLAV is always driven. Figure 95 above, shows a start-of-transfer for Utopia 1 PHY slave mode. If a cell is being read in Utopia 1 PHY mode, RPHY_CLAV will be asserted until

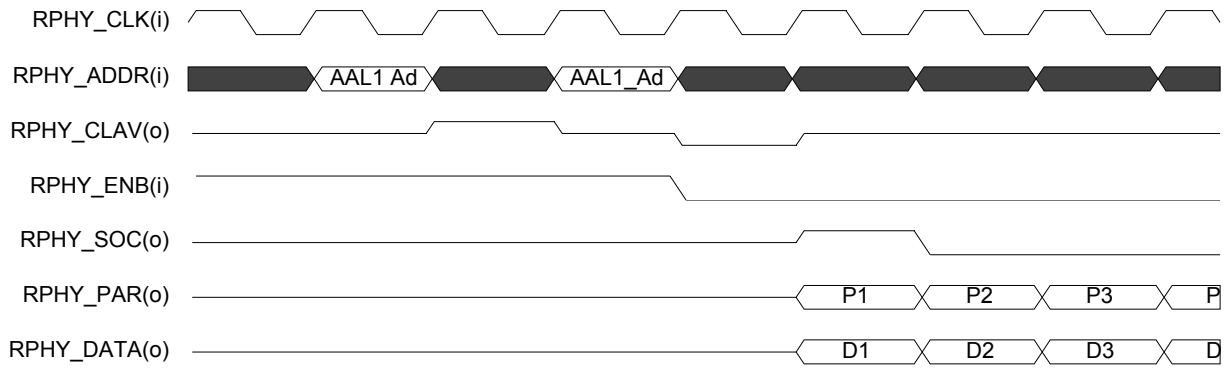
the cell has been read out of the source MCFF FIFO and there are no more cells to send. Data is placed on RPHY_DATA any cycle following one in which RPHY_ENB was asserted. In Utopia 1 PHY mode RPHY_ENB can be deasserted during the cell transfer for data transfer pausing. Figure 96 below shows the end of transfer behavior in PHY mode. Note that in Utopia 1 mode the status of RPHY_CLAV should reflect the current cell transfer, thus RPHY_CLAV remains asserted until the last byte/word of the cell.

Figure 96 UI_SRC_INTF End-of-Transfer (Utopia 1 PHY Mode)



In Utopia 2 PHY slave (MPHY) mode, RPHY_ADDR is used for device polling and selection. RPHY_CLAV is only driven during cycles following ones in which RPHY_ADDR[4:0] matches the CFG_ADDR[4:0] in the UI_SRC_ADD_CFG register. When a channel is being polled in Utopia 2 slave mode (MPHY), the value of RPHY_CLAV will be 1 until the cell has been read out of the FIFO and there are no more cells to send. In Utopia 2 mode, the SRC_INTF block has to be selected for data to be driven. This is done when RPHY_ADDR[4:0] matches source CFG_ADDR[4:0] in the UI_SRC_ADDR_CFG register the cycle before RPHY_ENB goes low. A start-of-transfer sequence for Utopia 2 PHY slave mode is shown below in Figure 97. Transfer pausing is supported in Utopia 2 PHY slave mode, thus RPHY_ENB can be deasserted during the cell transfer and the data driven by the SRC_INTF will stop one cycle later.

Figure 97 UI_SRC_INTF Start-of-Transfer Timing (Utopia 2 PHY Mode)



The end-of-transfer behavior is shown in Figure 98. As with Utopia 1 mode, the state of RPHY_CLAV reflects the current cell transfer status and so remains asserted until the last data byte/word. The SRC_INTF in this example does not have another cell to send, so RPHY_CLAV shows a low value after the transfer.

Figure 98 UI_SRC_INTF End-of-Transfer Timing (Utopia 2 PHY Mode)

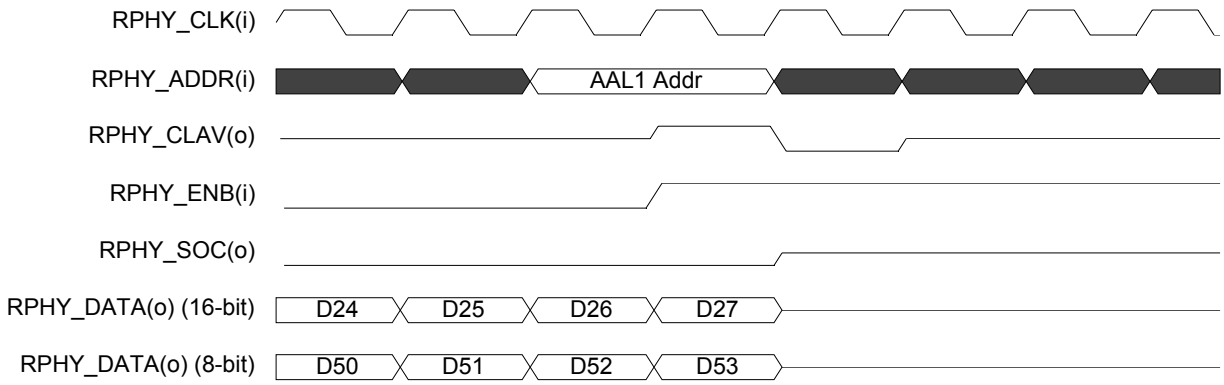


Figure 99 is a functional timing of the SRC_INTF for the start of a cell transfer when configured as an Any-PHY compliant receive slave. During a polling operation, when the SRC_INTF determines that the UI_SRC_ADDR_CFG address is on the bus it responds by driving RPHY_CLAV two cycles later. If CS_MODE_EN is set in the UI_SRC_CFG register then CSB must also be driven low one cycle after the RPHY_ADDR for proper response. If the SRC_INTF has a cell to send it will drive RPHY_CLAV high and, as a result, the master will activate RPHY_ENB to initiate a transfer. As with Utopia 2, the SRC_INTF is selected if its address is on RPHY_ADDR inputs during the cycle before RPHY_ENB goes low and in response transmits an entire cell. RPHY_RSX is driven high during the prepended byte address and RPHY_SOC is driven high during the first header byte of the ATM cell. Since data transfer pausing is not supported, once a transfer is initiated, RPHY_ENB should remain

asserted until the last data byte/word. Note that RPHY_CLAV will be deasserted along with the second data byte/word driven on the bus if the SRC_INTF does not have another cell to send, as in Figure 99. This is because, in Any-PHY mode, the RPHY_CLAV signal should always reflect the cell available status for the next possible future transfer rather than the current one as in Utopia mode.

Figure 99 UI_SRC_INTF Start-of-Transfer Timing (Any-PHY PHY Mode)

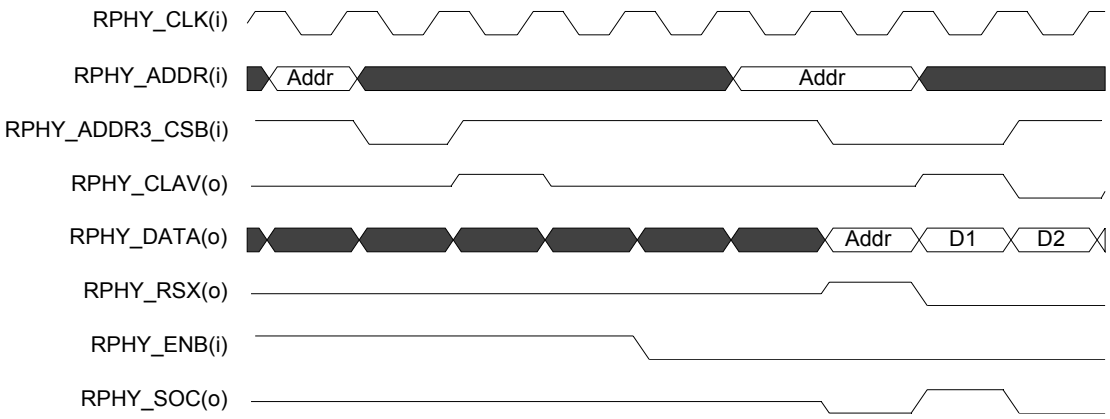
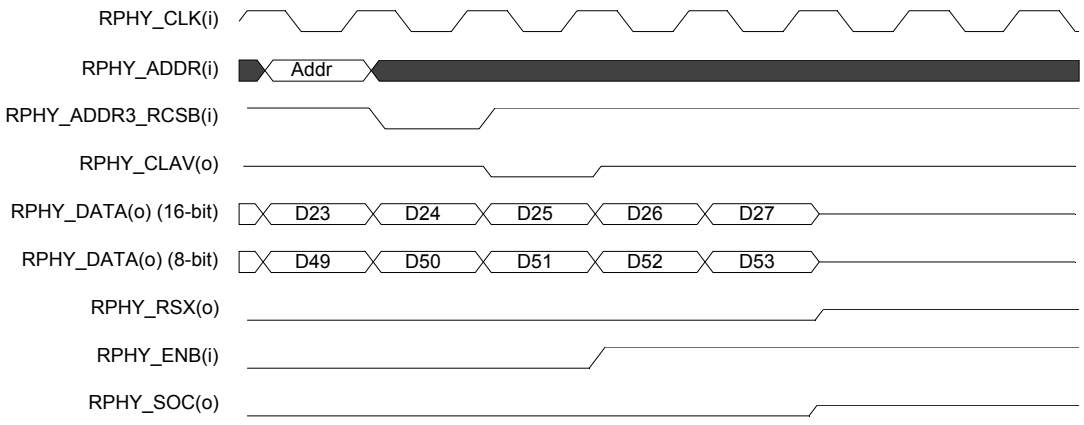


Figure 100 below shows an example of the end of a cell transfer for the SRC_INTF while in Any-PHY PHY slave mode. In this particular instance, the SRC_INTF still does not have another cell to send for this particular address, thus RPHY_CLAV is not asserted when polled.

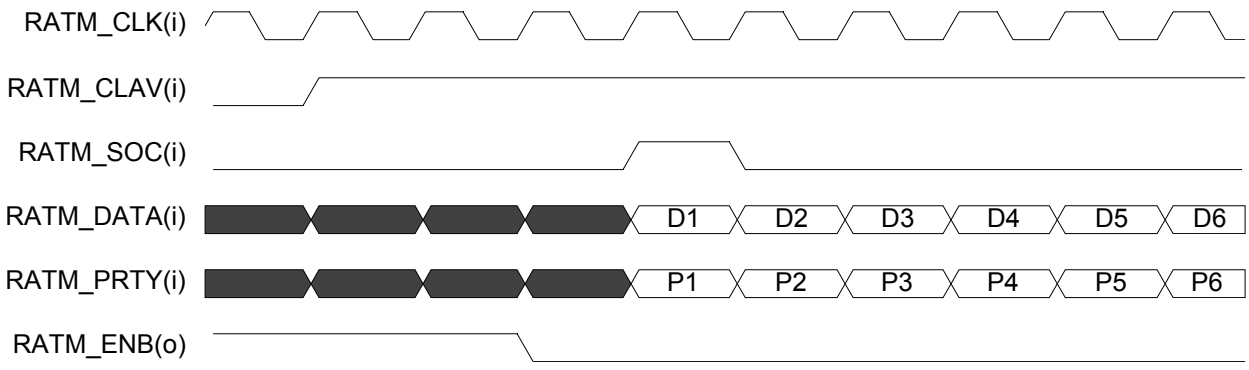
Figure 100 UI_SRC_INTF End-of-Transfer Timing (Any-PHY PHY mode)



15.2 Sink Utopia

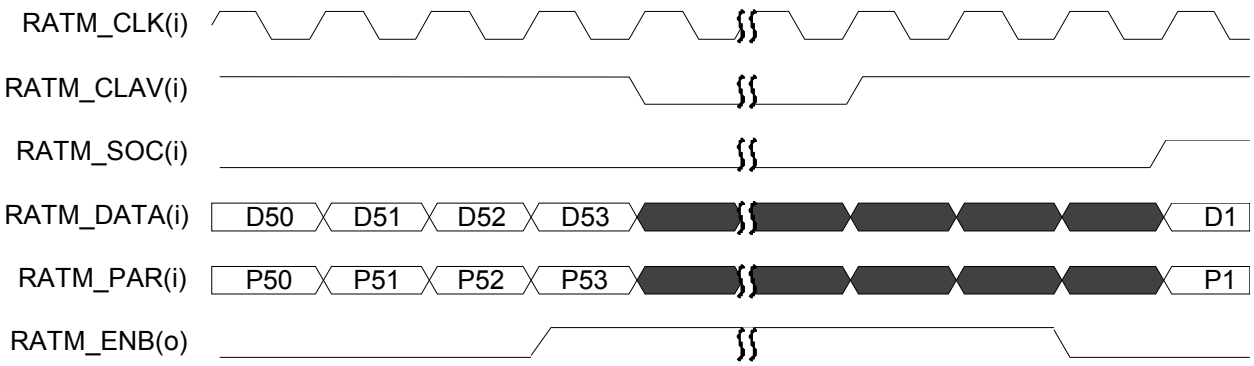
In ATM master mode, the SNK_INTF block receives RATM_DATA, RATM_PAR, RATM_SOC, and RATM_CLAV while driving RATM_ENB. During reset the SNK_INTF tristates the RATM_ENB (and all other SRC_INTF outputs). After the end of reset and after UI_EN in the UI_COMN_CFG register is set, if the RATM_CLAV input signal is not asserted the SNK_INTF waits and RATM_ENB is deasserted high. As shown in Figure 101, once the RATM_CLAV input signal is asserted, the RATM_RENB output signal is asserted low 2 cycles later and it is expected that the PHY slave will deliver the first data byte one cycle after RENB goes low. Typically a start-of-cell signal will be sent by the PHY slave along with the first byte of data, however RATM_SOC is optional and the RATM_SOC input could be tied low. Note however, not using an SOC eliminates the possibility of runt cell detection. Once RATM_ENB goes low, a counter is started, and 53 bytes are expected. If an SOC occurs within a cell, the counter reinitializes and the previous corrupted cell will be dropped and the second good cell will be received. The SNK_INTF block stores the ATM cell in the receive FIFO. The 4 cell MCFF FIFO allows the interface to accept data at the maximum rate and if the FIFO fills, the RATM_ENB signal will not be asserted again until RATM_CLAV is asserted and the device is ready to accept an entire cell. The maximum supported clock rate is 52 MHz.

Figure 101 SNK_INTF Start-of-Transfer Timing (Utopia 1 ATM Mode)



Note that in ATM master mode the SNK_INTF uses cell based handshaking, thus once a transfer has begun, RATM_ENB will be asserted continuously until the end of the cell regardless of the state of RATM_CLAV. Figure 102 shows the end of transfer signal behavior. In this example, since the PHY slave does not have another cell to deliver, the RATM_CLAV signal is deasserted after the last byte is delivered. Also, the AAL1GATOR acting as an ATM master always deasserts the RATM_ENB at the end of a cell transfer. RATM_CLAV is then sampled again on the following clock cycle, and if asserted, RATM_ENB will be asserted again for a new transfer.

Figure 102 SNK_INTF End-of-Transfer Timing (Utopia 1 ATM Mode)



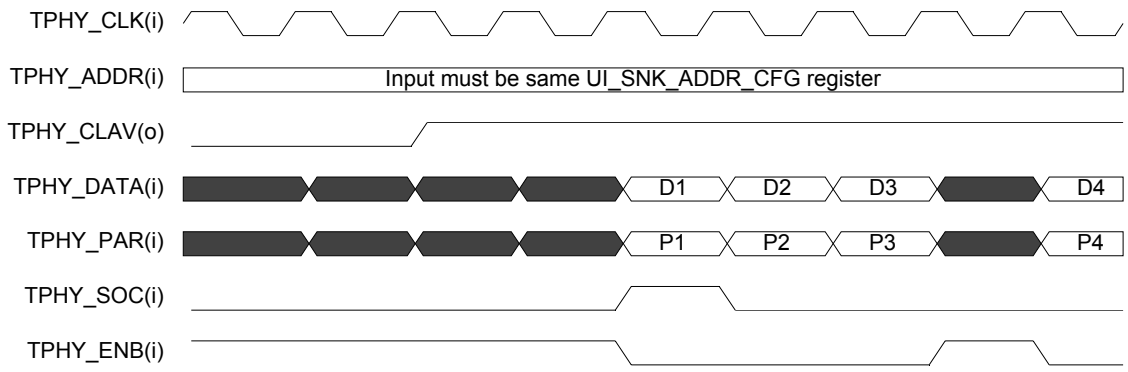
In PHY slave mode, the SNK_INTF block receives TPHY_DATA, TPHY_SOC, and TPHY_ENB while driving TPHY_CLAV. TPHY_CLAV indicates when the device is ready to receive a complete cell. In Utopia 1 mode, TPHY_CLAV is always driven. In Utopia 2 mode, TPHY_CLAV is only driven during cycles following ones in which TPHY_ADDR[4:0] matches the address CFG_ADDR[4:0] in the UI_SNK_ADDR_CFG register.

Figure 103 below shows the start of transfer timing for Utopia 1 Phy slave mode. At reset, and when UI_EN in the UI_COMN_CFG register is set low, the SNK_INTF block tristates TPHY_CLAV (not shown). While not in reset, the SNK_INTF asserts TPHY_CLAV if it can accept a cell and waits for TPHY_ENB to be asserted.

When TPHY_ENB is asserted, a counter is started and 53 bytes are received (27 words in 16-bit mode). If a new TPHY_SOC occurs within a cell, the counter reinitializes. This means that the corrupted cell will be dropped and the second good cell will be received.

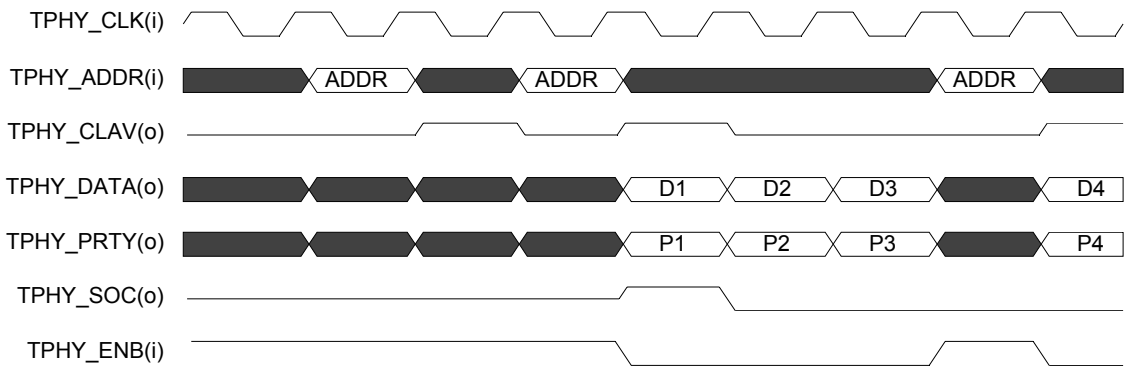
In Utopia 1 PHY mode, the SNK_INTF will accept data as long as TPHY_ENB is asserted and the sink MCFF FIFO has room. Data transfer pausing is supported in Utopia 1 PHY mode, thus if TPHY_ENB is deasserted, as in Figure 103, the data transfer is paused. The 4 cell MCFF FIFO allows the interface to accept data at the maximum rate. If the FIFO fills, the TPHY_CLAV signal will be deasserted by data transfer cycle D12 (D10 in 16-bit mode) and not be asserted again until the device is ready to accept an entire cell (This is not shown in a figure).

Figure 103 SNK_INTF Start-of-Transfer Timing (Utopia 1 PHY Mode)



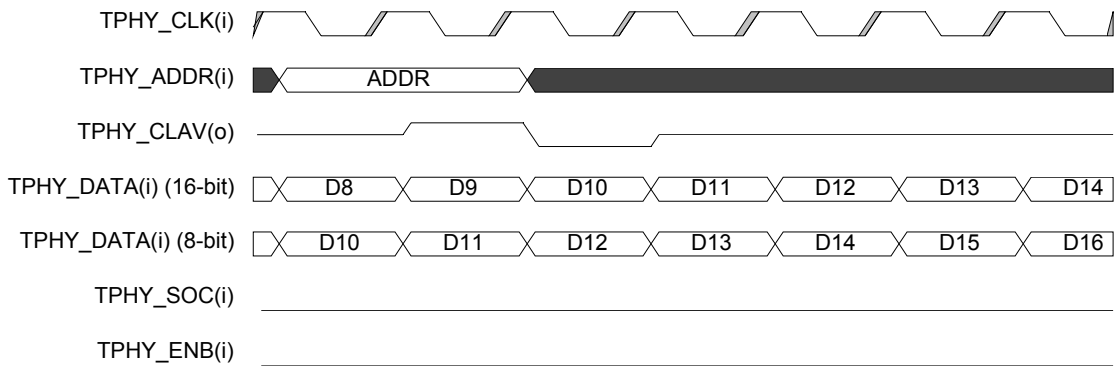
In Utopia 2 single address mode, a cell transfer is started as a result of an ATM master polling the SNK_INTF with an address matching the value in the UI_SNK_ADDR_CFG register. The SNK_INTF responds by asserting TPHY_CLAV the cycle after UI_SNK_RADR matches the UI_SNK_ADDR_CFG register value. Selection occurs when the value on the TPHY_ADDR matches the configured address during the cycle before UI_SNK_RENB is brought low. This is shown in Figure 104 below. The SNK_INTF will accept data as long as TPHY_ENB is asserted and it has room. Data transfer pausing is supported in Utopia 2 single address PHY mode, thus if TPHY_ENB is deasserted as in Figure 104 the data transfer is paused. Furthermore, in Utopia 2 single address mode, although a cell transfer has been started, additional polling may show TPHY_CLAV as being asserted if there is still room in the sink FIFO as shown at the end of Figure 104.

Figure 104 SNK_INTF Start-of-Transfer Utopia 2 (Single Address PHY Mode)



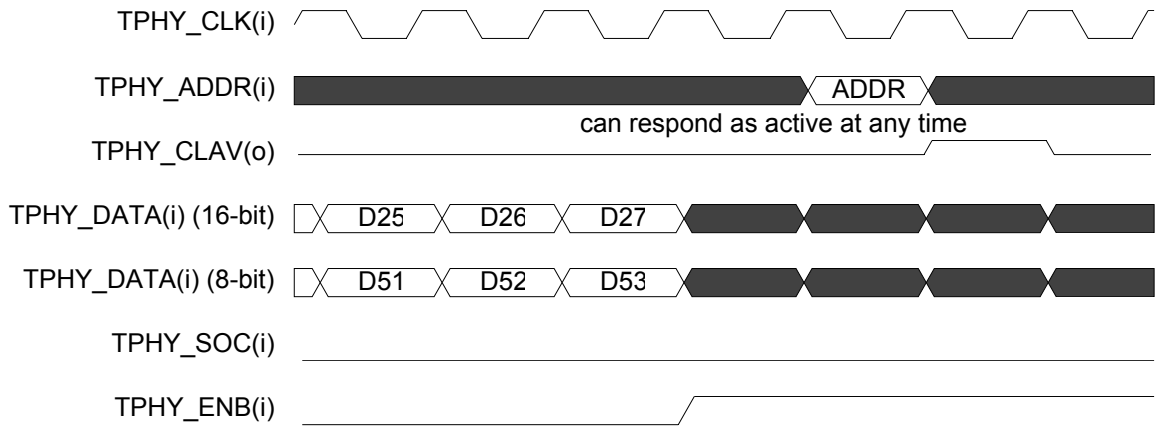
If the current cell being transferred will fill up the last of the four cell slots in the sink MCFF FIFO, then the TPHY_CLAV signal will be deasserted by the D10 or D12 data transfer cycle, in 16-bit and 8-bit modes respectively. This is shown in Figure 105. Note this transition will only be visible on TPHY_CLAV if the address is being polled as in the figure.

Figure 105 SNK_INTF CLAV Disable Utopia 2 (Single-Address PHY Mode)



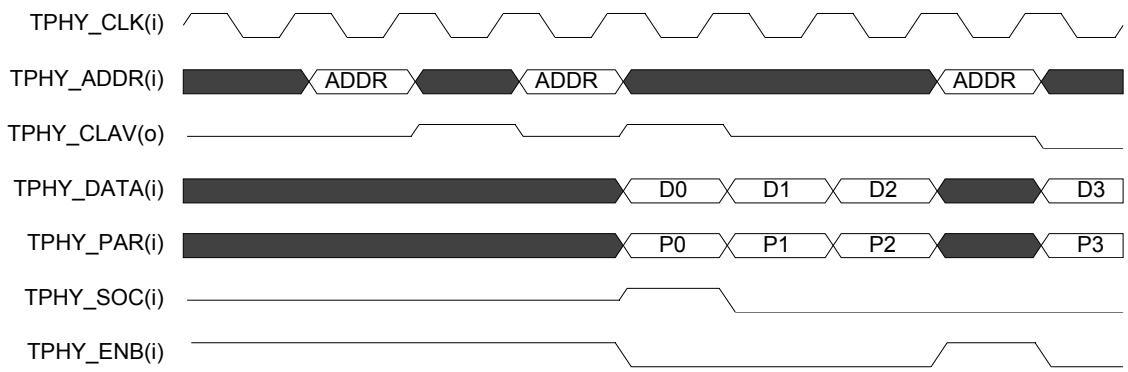
Since the Utopia 2 single address PHY mode utilizes the sink MCFF as a single four cell deep FIFO, the TPHY_CLAV signal can be asserted any time a cell slot becomes empty in the MCFF as is shown in Figure 106. The end of the transfer is coincident with the TPHY_ENB being deasserted, although it is possible for a master to keep TPHY_ENB asserted if the SNK_INTF has cell space available and the master has another cell to send. Note that once deasserted, TPHY_CLAV can become active at any time during a cell transfer if cell space becomes available in the sink MCFF FIFO.

Figure 106 SNK_INTF End-of-Transfer Utopia 2 (Single Address PHY Mode)



A start of transfer sequence for Utopia 2 multi-address mode is similar to single address mode, however because of the limitation of reserving one cell space in the sink MCFF for each of the four supported addresses, additional polling by the master will not show TPHY_CLAV being asserted immediately after the cell transfer is started. This is shown at the end of Figure 107. Data transfer pausing is supported, thus TPHY_ENB can be deasserted by the master during a transfer.

Figure 107 SNK_INTF Start-of-Transfer Utopia 2 (Multi-Address PHY Mode)



The TPHY_CLAV signal associated with a particular slave address (of the four) is held deasserted until after the cell sent to that address is read out of the sink MCFF. The time at which this occurs is determined by how often cells are read by the UMUX from the MCFF but is at least a minimum of 32 cycles after the end of a cell transfer assuming an empty MCFF with an immediate read by the UMUX. This is shown in Figure 108.

Figure 108 SNK_INTF End-of-Transfer Utopia 2 (Multi-Address PHY Mode)

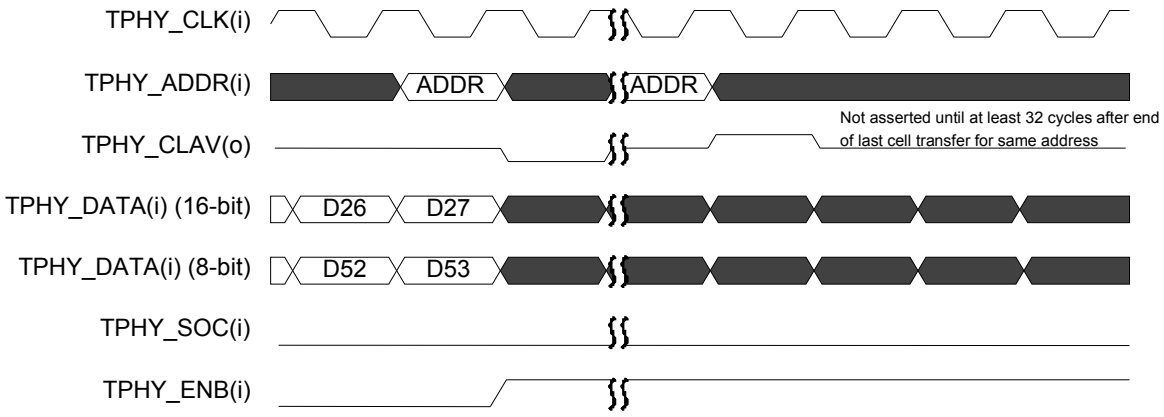
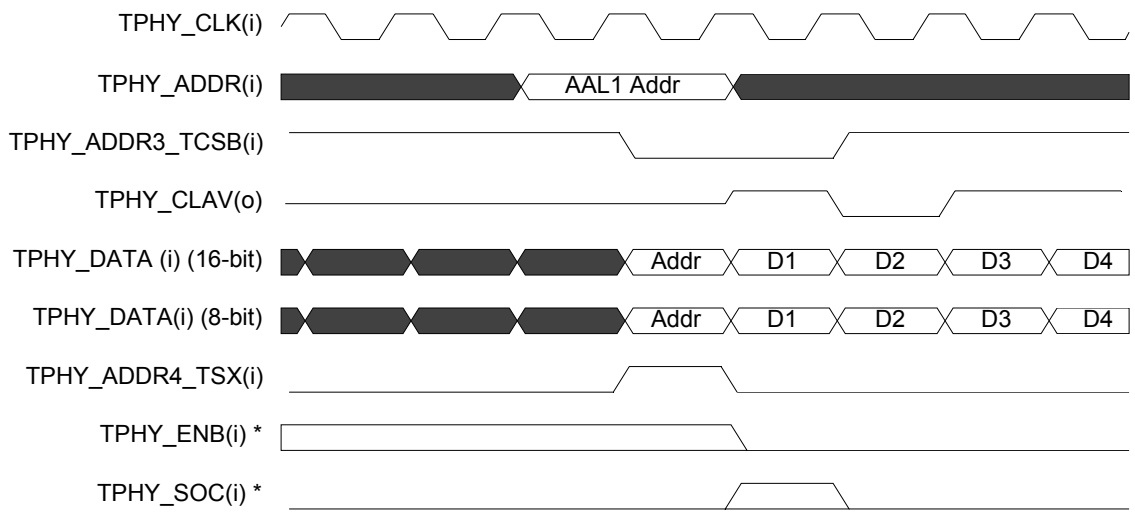


Figure 109 gives an example of the functional timing of the UI_SNK_INTF when configured as a Any-PHY compliant transmit slave. When the SNK_INTF has room for a cell and determines its address is on the bus, it will respond by driving TPHY_CLAV high **two** cycles later. If CS_MODE_EN is set in the UI_SNK_CFG register, then CSB should be driven low one cycle after the TPHY_ADDR. As a result, the master will activate TPHY_ENB to initiate a transfer. When TPHY_TSX is high the SNK_INTF will compare the prepended address with value stored in UI_SNK_ADDR_CFG register and if they match it will accept the data. TPHY_TSX should be driven high during the prepended byte address and TPHY_SOC should be driven high during the first header byte of the ATM cell.

Only cell level handshaking and no data transfer pausing is supported in Any-PHY mode, thus once transfer is initiated TPHY_ENB should remain asserted until the last data is transferred. Note that TPHY_CLAV is masked after the poll (when the address is applied) which is coincident with the assertion of TSX. Also note that, in Any-PHY mode both TPHY_ENB and TPHY_SOP are optional and both could be tied low indefinitely.

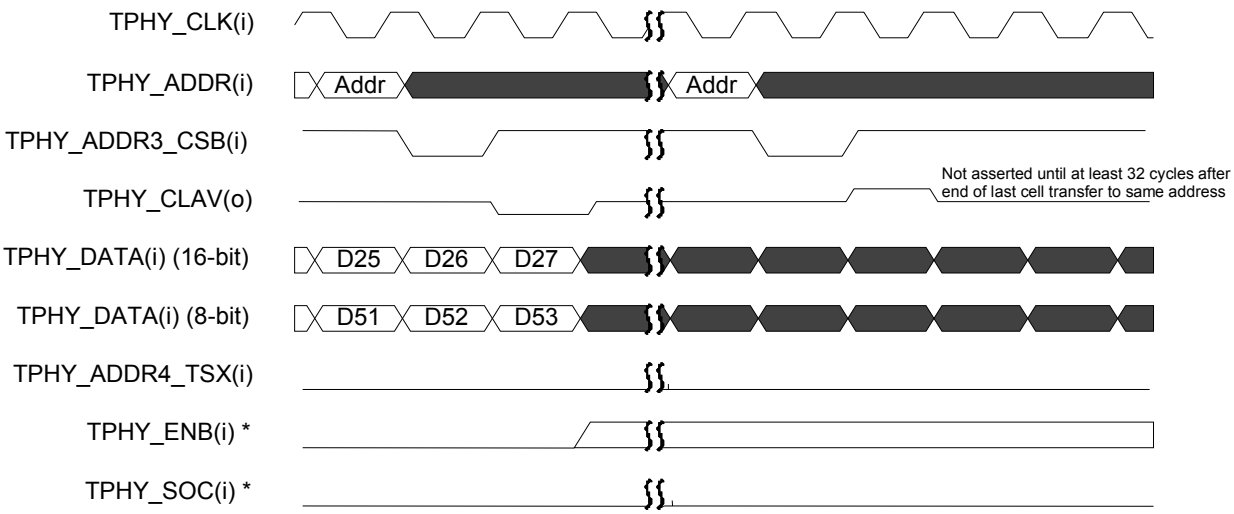
Figure 109 SNK_INTF Start-of-Transfer (Any-PHY PHY Mode)



* TPHY_ENB and TPHY_SOC could be tied low for Any-PHY mode

Figure 110 shows the end of transfer for Any-PHY mode. In an identical fashion as Utopia 2 multi-address mode, Any-PHY mode also reserves one space in the MCFF FIFO for each of the four slave addresses and relies on cells being read out by the UMUX before the associated CLAV signal for that address can be asserted. As with Utopia 2 multi-address mode, it takes a minimum of 32 cycles after a cell transfer to a particular address has ended for the associated CLAV to be asserted.

Figure 110 SNK_INTF End-of-Transfer (Any-PHY PHY Mode)



* TPHY_ENB and TPHY_SOC could be tied low for Any-PHY mode

15.3 Processor I/F

The microprocessor accesses the device by means of the CSB, RDB, and WRB lines. To perform a write cycle, the microprocessor asserts the CSB and WRB lines. The AAL1gator-32 then passes the address and data to the RAM interface or internal registers and, when complete, signals the microprocessor with the ACKB line. See Figure 111 below. To perform a read, the microprocessor asserts CSB and RDB and waits for ACKB before reading the data. See Figure 112 below.

If CSB, WRB and RDB are all active at the same time, all chip outputs go tri-state. Therefore, WRB and RDB should never be active at the same time.

The amount of time it takes for ACKB to go active is dependant on what other operations are happening at the same time and where the access is targeted. Register or internal memory operations will be responded to within a few cycles. Accesses to RAM usually take less than 10 clock cycles but sometimes can take much longer if there is a lot of contention for the RAM.

If interfacing to a multiplexed address data bus, an optional ALE signal is provided. If ALE is not required it should be tied high. See Figure 113 and Figure 114.

Figure 111 Microprocessor Write Access

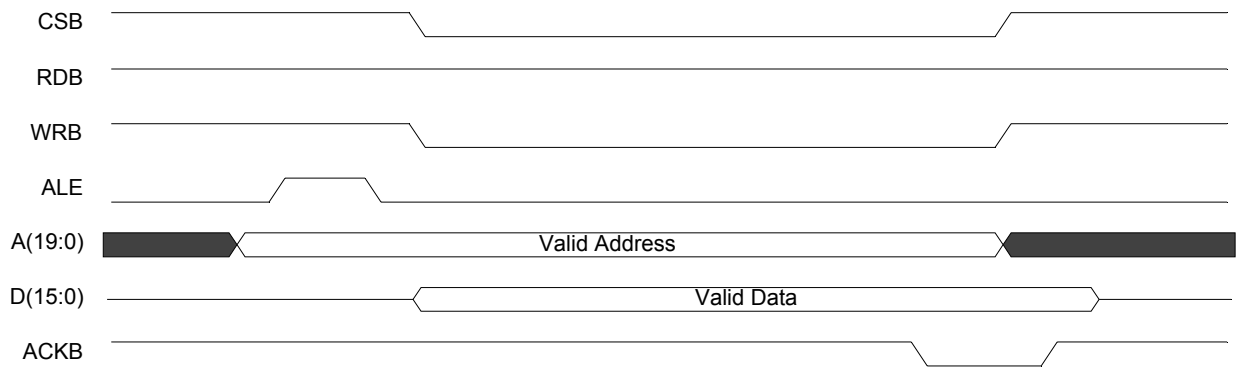


Figure 112 Microprocessor Read Access

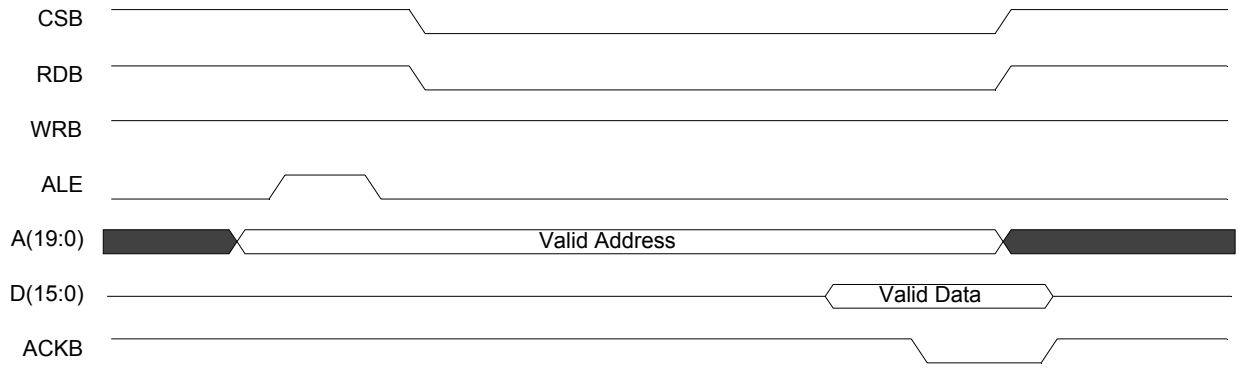


Figure 113 Microprocessor Write Access with ALE

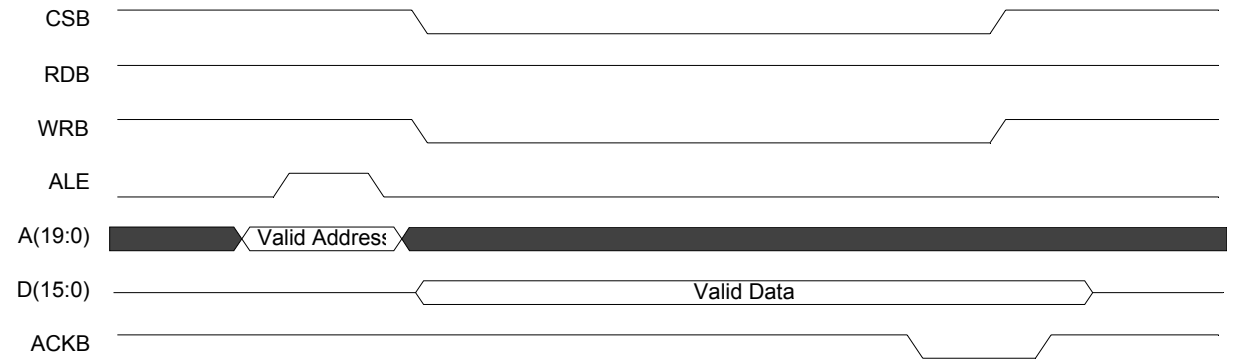


Figure 114 Microprocessor Read Access with ALE

15.4 External Clock Generation Control I/F (CGC)

Status information is played out on the External Clock Control (CGC) Interface which includes the external output signals: CGC_DOUT(3:0), CGC_LINE(4:0), SRTS_STBH, and ADAP_STBH.

Information is played out for all chip modes (Direct Low Speed, SBI, H-MVIP and High Speed).

The interface clocking operates according to the following pseudocode:

- If a channel pair is being serviced, start a state machine to play out the channel status, as shown in Table 38 asserting ADAP_STBH as each state is played out.
- Else, if an SRTS difference value is ready, it is played out with SRTS_STBH asserted and ADAP_STBH deasserted.
- Else if a cell is received and a valid averaged relative buffer depth can be computed, start a state machine to play out the averaged relative buffer depth and queue number, as shown in Table 39, asserting ADAP_STBH as each state is played out.
- Once playout of a certain data type has begun if will not be interrupted.

15.4.1 SRTS Data Output

In SRTS mode the CGC block puts out a 4 bit value which represents the difference between the local SRTS value and the received SRTS value. Figure 115 below shows a typical CGC output in SRTS mode.

Figure 115 SRTS Data

15.4.2 Channel Underrun Status Output

Figure 116 below shows an example of the channel underrun status being reported on the CGC Interface. In the example the line number is 19 (A1SP = 2, local line number = 3) the channel pair is 7 (channels 14 and 15), and the high channel (channel 15) is in underrun while the low channel (channel 14) is in normal mode. These values are encoded into the 4 data words following the format shown in Table 38.

Status is only reported when a channel enters or exits underrun.

Figure 116 Channel Status Functional Timing

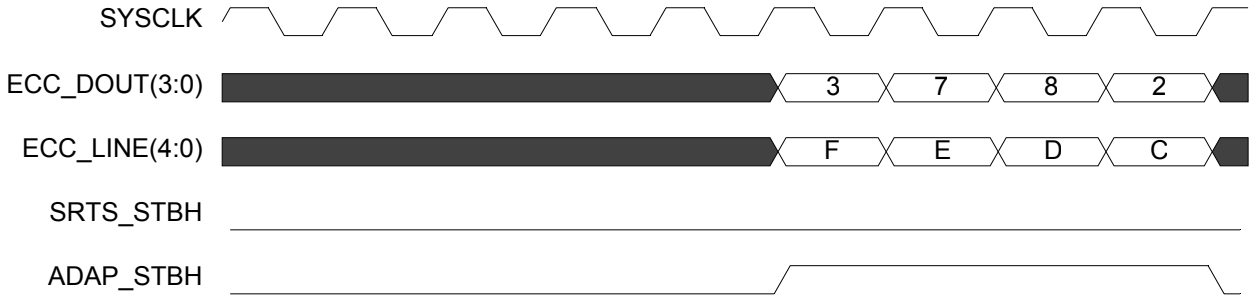


Table 38 Channel Status

CGC_LINE(4:0) Value	CGC_DOUT(3:0) Value			
	3	**2**	**1**	**0**
15	0	Line(3)	Line(2)	Line(1)
14	Channel(4)	Channel(3)	Channel(2)	Channel(1)
13	underrun_h	0	underrun_l	0
12	0	0	A1SP(1)	A1SP(0)
	0	0	0	0

15.4.3 Adaptive Status Output

The AAL1gator provides the average buffer depth in units of bytes for an external circuit to generate an adaptive clock. (If adap_filt_size is set to zero it will provide the current buffer depth with no averaging). The general mechanism is often termed “buffer centering”. A clock delta value is determined externally by subtracting the nominal buffer depth(value of CDVT) from the actual buffer depth. This delta value is then transformed into the frequency selection for an external frequency synthesizer. The closed-loop action of this circuit causes the delta value to find a center point. When the delta is above the center point, there is too much data buffered and the frequency must be increased. When the delta is below the center point, the frequency must be reduced.

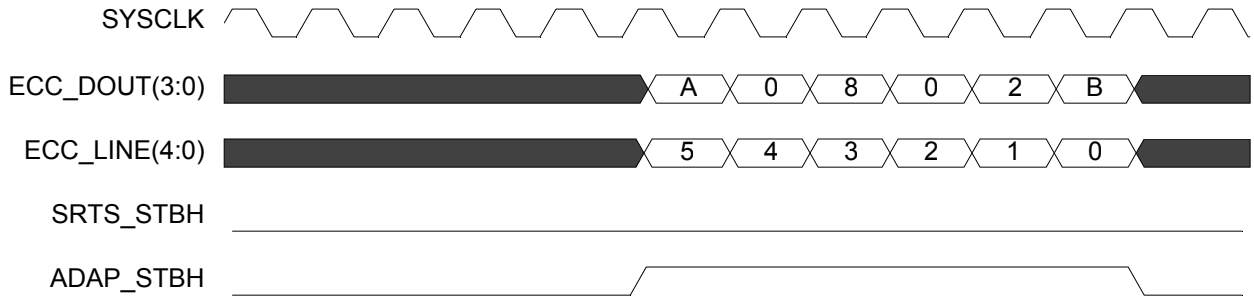
The AAL1gator implements a programmable weighted moving average internally. However if an alternative adaptive algorithm is desired then this information can be processed externally. Also in High Speed mode, since an E3 or DS3 clock cannot be internally synthesized, this information can be used for external synthesis of an E3 or DS3 clock.

Figure 117 below shows an example of the CGC Interface for adaptive data. In this example the line number is 21, and the average buffer depth in units of bytes is 43. These values are encoded into the 6 data words following the format shown in Table 39.

Table 39 Frame Difference

CGC_LINE_OUT(4:0) Value	CGC_DOUT(3:0) Value			
	3	2	1	0
5	queue(7)	queue(6)	queue(5)	queue(4)
4	queue(3)	queue(2)	queue(1)	queue(0)
3	A1SP(1)	A1SP(0)	buff_depth(13)	buff_depth(12)
2	buff_depth(11)	buff_depth(10)	buff_depth(9)	buff_depth(8)
1	buff_depth(7)	buff_depth(6)	buff_depth(5)	buff_depth(4)
0	buff_depth(3)	buff_depth(2)	buff_depth(1)	buff_depth(0)

Figure 117 Adaptive Data Functional Timing



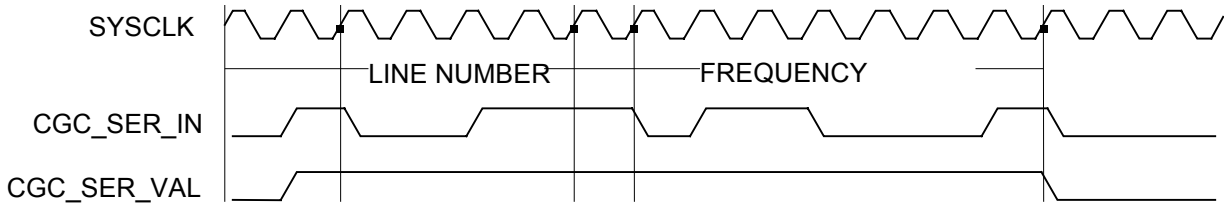
15.5 Ext Freq Select Interface

The Ext Freq Select Interface allows an external source to directly select the line clock frequency from any one of 171 T1 or 240 E1 frequencies centered around the nominal clock rate. For T1 the legal input values are –83 to 88. For E1 the legal input values are –128 to 111. Any values outside of this range will be clamped to these levels. These levels correspond to a +/-200 ppm T1 clock and a +/- 100 ppm E1 clock.

The External Interface block has two input ports that allow an external source to control the frequency synthesizers internal to the CGC. These two ports are CGC_SER_D and CGC_VALID. CGC_SER_D contains the data that selects one of the 171/240 frequencies and CGC_VALID indicates when this data is valid. There should be a rising edge of CGC_VALID at the start of each timing message. And CGC_VALID should go low at the end of each timing message. CGC_VALID only needs to be deactivated for one cycle between timing messages. The CGC block decodes the incoming line number and if the address matches one of its 8 line numbers passes the data on to the appropriate frequency synthesizer.

Figure 118 below shows an example of where Line 19 is being programmed to run with the frequency setting of –79 (Two's complement). See the section on the Frequency Synthesizer block for a discussion of the different frequency settings which range from -83 to 88 in T1 mode and –128 to 111 in E1 mode.

Figure 118 Ext Freq Select Functional Timing



15.6 Line Interface Timing

15.6.1 16 Line Mode

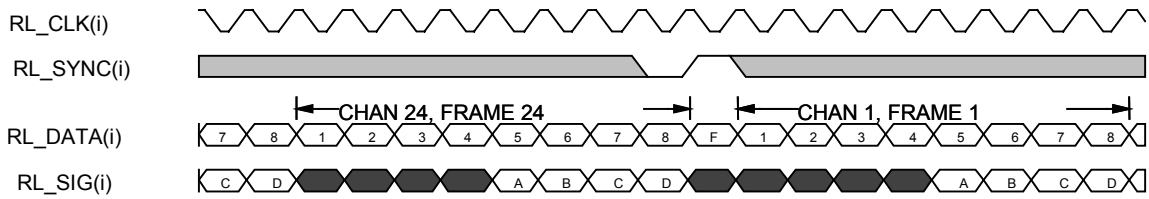
15.6.1.1 Receive Line Timing

In T1 mode, the rising edge of RL_SYNC must coincide with the leading edge of the F bit. If RL_SYNC is programmed as an MF pulse, then that edge should also be the leading edge of a multiframe. AI input signals are sampled on the falling edge of RL_CLK, as shown in the following figure.

Figure 119 and Figure 120 show how the transmitter receives data from the line interface. These lines typically interface with the receive output portion of the corresponding framer. The timing parameters are given in the AC timing section. RL_SYNC is used in structured modes to align the frame and/or multiframe of the incoming data. RL_SYNC can be configured to be a frame sync or a multi-frame sync by the value of MF_SYNC_MODE. This input is ignored in unstructured modes.

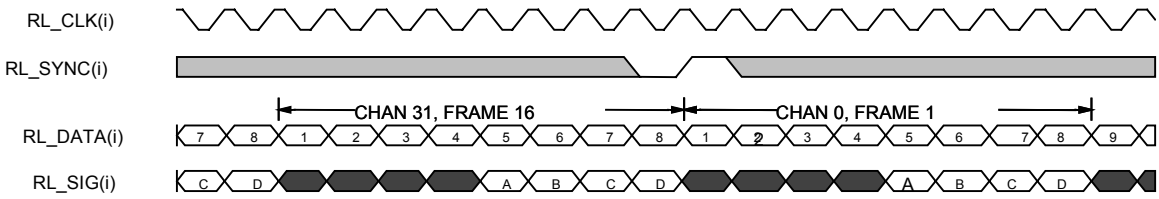
In T1 mode the rising edge of RL_SYNC must coincide with leading edge of the F bit. If RL_SYNC is programmed as a MF pulse then that edge should also be the leading edge of a multiframe. All input signals are sampled on the falling edge of RL_CLK. as shown in the following figure.

Figure 119 Receive Line Side T1 Timing(RL_CLK = 1.544 MHz)



In E1 mode the rising edge of RL_SYNC should coincide with the leading edge of the bit 1 of the first channel within a frame. If RL_SYNC is programmed as a MF pulse then that edge should also be the leading edge of a multiframe. All input signals are sampled on the falling edge of RL_CLK. as shown in the following figure.

Figure 120 Receive Line Side E1 Timing(RL_CLK = 2.048 MHz)

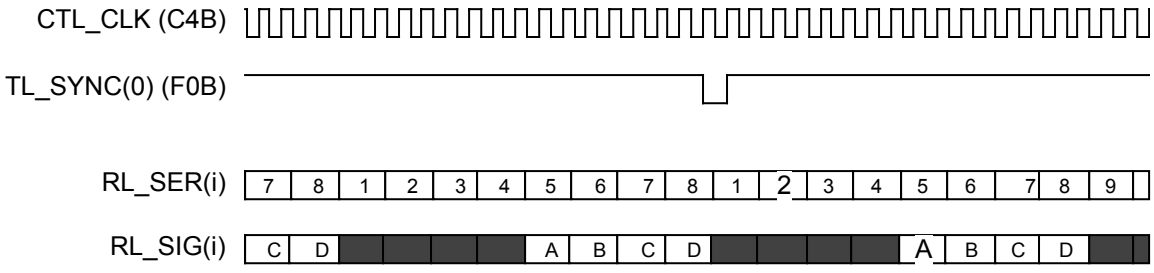


When the line is in MVIP-90 mode, as controlled by the MVIP_EN mode bit in the Low Speed Line Configuration Register, a common active low frame pulse is used; F0B. This frame pulse is sampled using the falling edge of the 4 MHz C4Binput clock signal. C4B is also used to clock the data. The AAL1gator-32 samples the data provided on RL_SER[n] at the ¾ point of the data bit using the rising edge of C4B. 1 is the most significant bit and 8 is the least significant bit of each octet.

Note that GEN_SYNC is ignored in this mode. F0B must be externally generated and must be only one 4 MHz clock cycle wide.

CAS signaling can be transported by passing it during the last nibble of each time slot.

Figure 121 MVIP-90 Receive Functional Timing

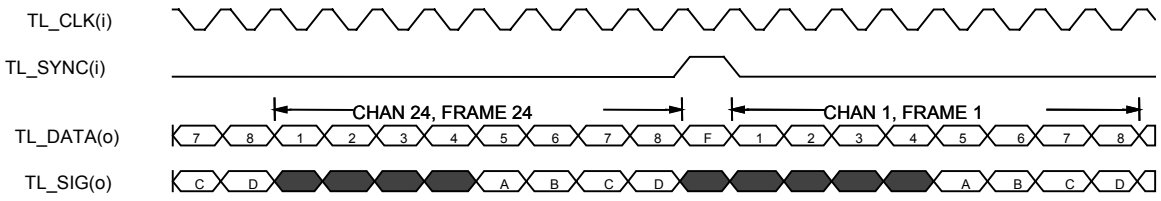


15.6.1.2 Transmit Line Timing

Figure 122 and Figure 123, show how the receiver sends data to the line interface. These lines typically interface with the transmit input portion of the corresponding framer. The timing parameters are given in the AC timing section. TL_SYNC is used in structured modes to align the frame and/or multiframe of the incoming data. TL_SYNC can be configured to be a frame sync or a multi-frame sync by the value of MF_SYNC_MODE. This input is ignored in unstructured modes.

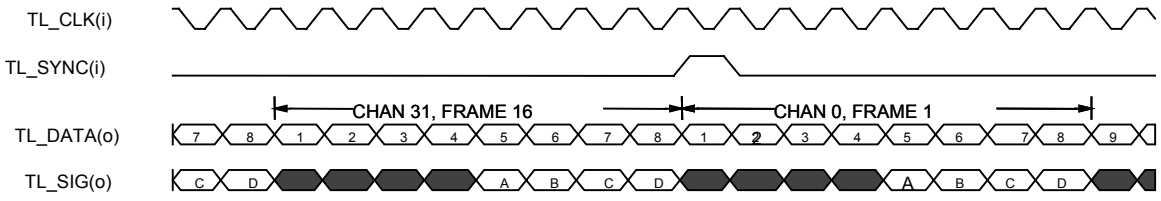
In T1 mode the rising edge of TL_SYNC coincides with leading edge of the F bit. If TL_SYNC is programmed as a MF pulse then that edge will also be the leading edge of a multiframe. All output signals are driven on the rising edge of TL_CLK. as shown in the following figure.

Figure 122 Transmit Line Side T1 Timing(TL_CLK = 1.544 MHz)



In E1 mode the rising edge of TL_SYNC coincides with the leading edge of the bit 1 of the first channel within a frame. If TL_SYNC is programmed as a MF pulse then that edge should also be the leading edge of a multiframe. All input signals are sampled on the falling edge of TL_CLK. as shown in the following figure.

Figure 123 Transmit Line Side E1 Timing(TL_CLK = 2.048 MHz)



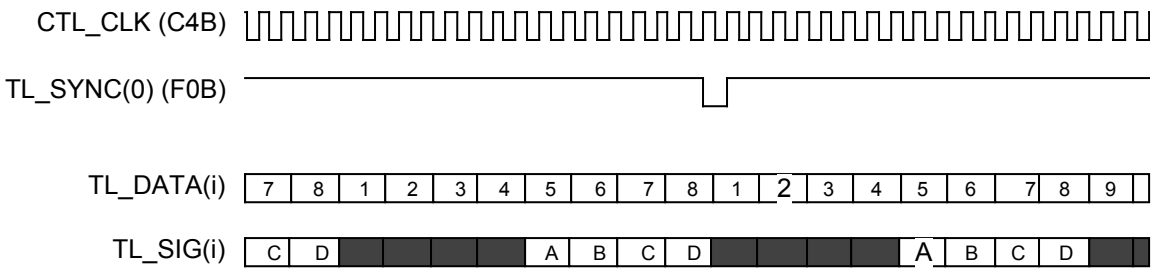
When the line is in MVIP-90 mode, as controlled by the MVIP_EN mode bit in the Low Speed Line Configuration Register, a common active low frame pulse is used; F0B. This frame pulse is sampled using the falling edge of the 4 MHz C4B input clock signal.

Note that GEN_SYNC is ignored in this mode. F0B must be externally generated and must be only one 4 MHz clock cycle wide.

The AAL1gator-32 updates the data provided on TL_SER[n] on every second falling edge of C4B. The first bit of the next frame is updated on TL_SER on the falling C4B clock edge for which F0B is also sampled low. 1 is the most significant bit and 8 is the least significant bit of each octet.

CAS signaling can be transported by passing it during the last nibble of each time slot.

Figure 124 MVIP-90 Transmit Functional Timing



15.6.2 H-MVIP Timing

In H-MVIP mode eight 8 Mbps incoming external links are broken into thirty-two 2 Mbps internal links. Also thirty-two 2Mbps internal links are combined into eight 8 Mbps outgoing external lines.

A common active low frame pulse; F0B; is used to indicate the start of a 128 time slot frame and is shared by all incoming and outgoing lines. The F0B signal is expected to be driven off the rising edge of C4B and is sampled using the falling edge of C4B. The sync signal is used to generate internal RLI_FSYNC and TLI_FSYNC signals for each 2 Mbps link. In addition GEN_SYNC is ignored as the frame pulse is always externally generated from F0B. Due to pipelining delays, the sync signals are offset from the start of frame on the internal links.

15.6.2.1 Receive Side Timing.

In H-MVIP mode there is a 16 MHz (C16B) clock that is used to clock in data and a 4 MHz clock (C4B) which is used to clock in the frame pulse. The falling edge of C4B is used to clock in F0B. This falling edge also coincides with the start of reception of the first data bit of the frame. C16B is used by the internal clock mux logic to generate a 2.048 MHz TLI_CLK signal for each internal local link. Data is captured on the second rising edge of C16B that occurs for each data bit.

CAS signaling can be transported by passing it during the last nibble of each time slot.

Figure 125 shows the timing around F0B pulse and the lower two local links within a group of four. Figure 125 shows an expanded view, including all four local links within each group of four. All signals prefixed with “RLI_” are internal signals and are not visible.

Figure 125 Receive H-MVIP Timing, Close-Up View

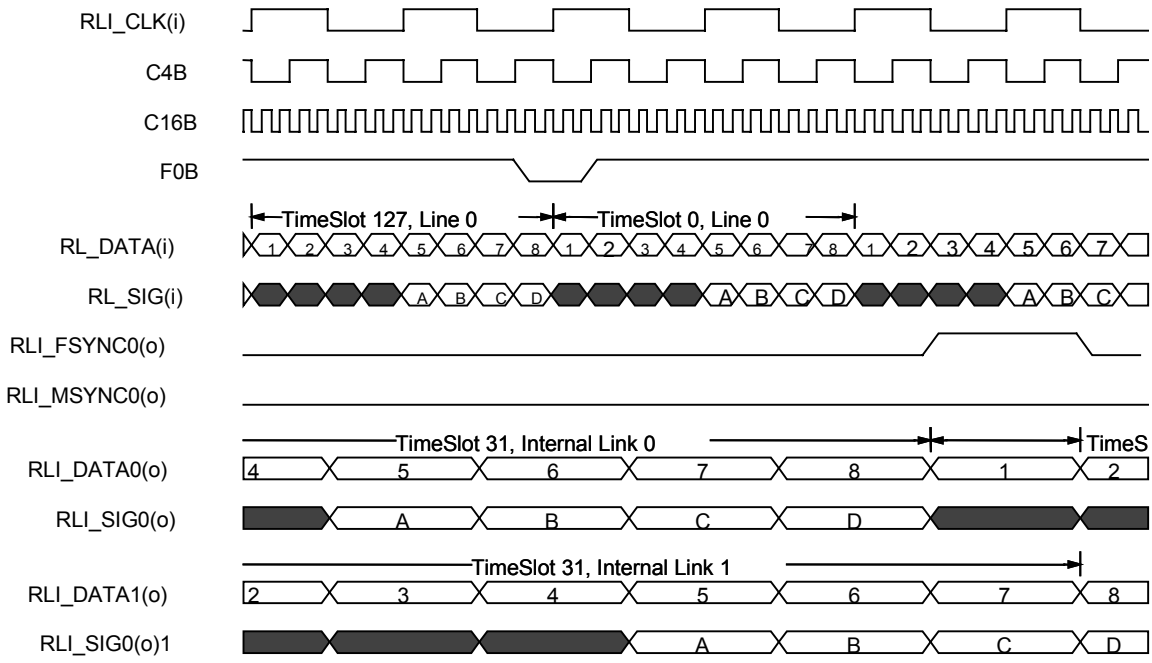
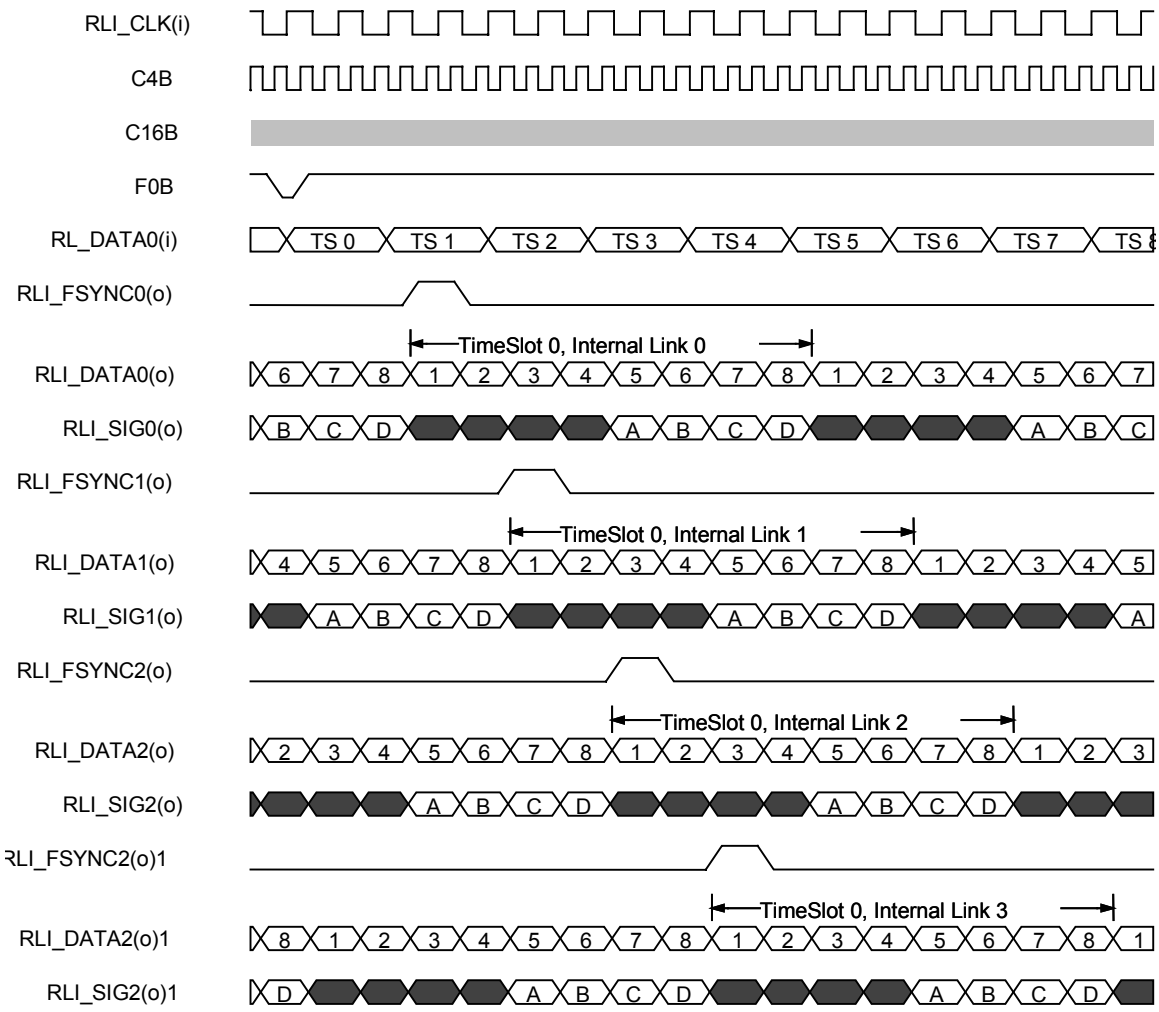


Figure 126 Receive H-MVIP Timing, Expanded View



15.6.2.2 Transmit Side Timing

In H-MVIP mode, on the transmit side, there is a 16 MHz clock (C16B) that is used to clock out data and a 4 MHz clock (C4B) which is used to clock in the frame pulse. The falling edge of C4B is used to clock in F0B. This falling edge also coincides with the start of transmission of the first data bit of the frame. C16B is used by the internal clock mux logic to generate a 2.048 MHz TLI_CLK signal for each internal local link.

CAS signaling can be transported by passing it during the last nibble of each time slot.

Figure 127 shows the timing around F0B pulse and the lower two local links within a group of four. Figure 127 shows an expanded view, including all four local links within each group of four. All signals prefixed with "TLI_" are internal signals and are not visible.

Figure 127 Transmit H-MVIP Timing, Close-up View

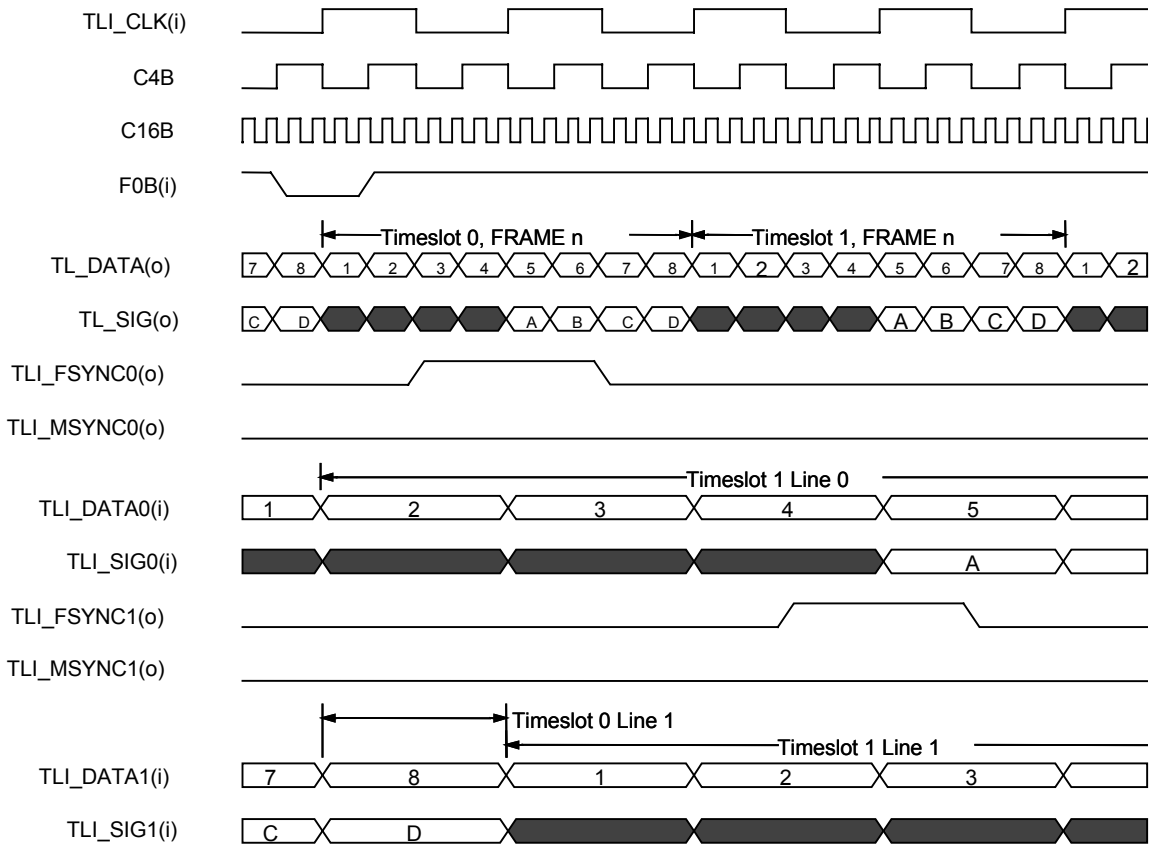
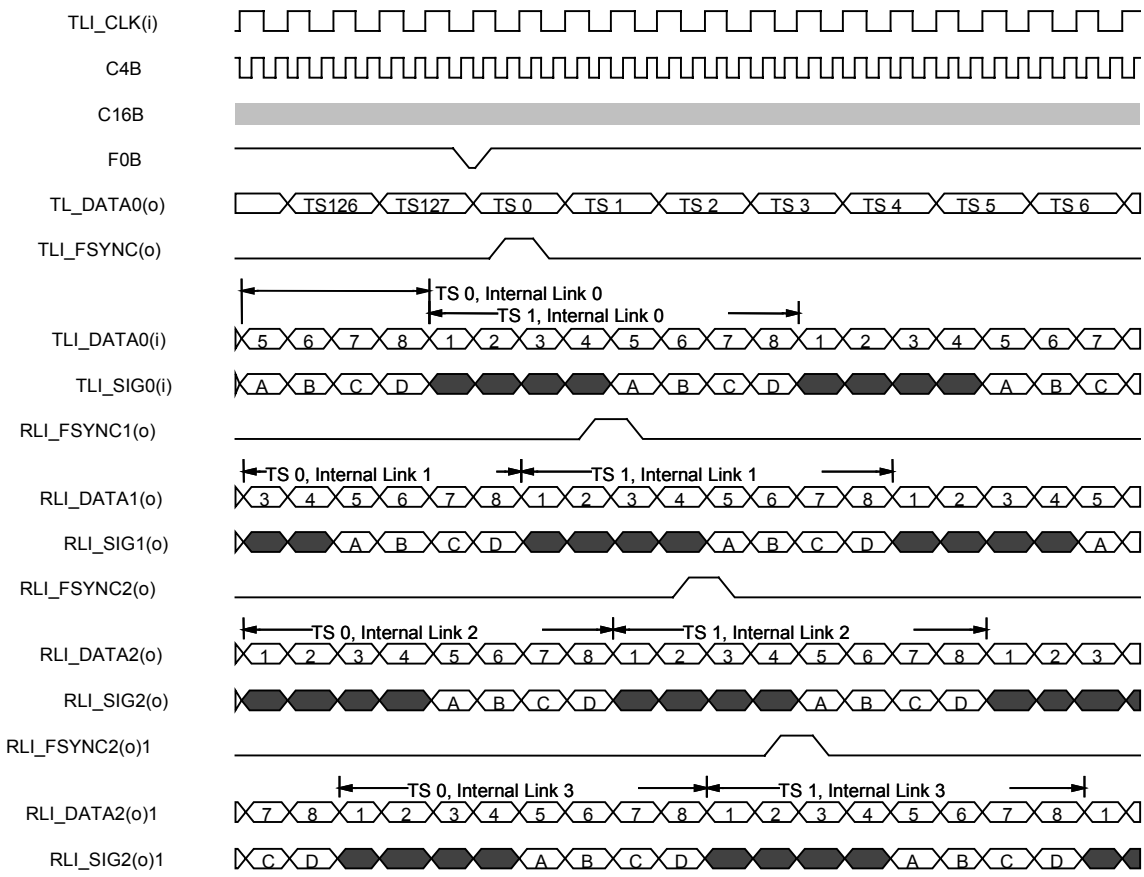


Figure 128 Transmit H-MVIP Timing, Expanded View



15.6.3 SBI Interface

15.6.3.1 Drop Bus Timing

Figure 129 illustrates the operation of the SBI DROP Bus, using a negative justification on the second to last V3 octet as an example. The justification is indicated by asserting DPL high during the V3 octet. The timing diagram also shows the location of one of the tributaries by asserting DV5 high during the V5 octet.

Figure 129 SBI DROP Bus T1/E1 Functional Timing

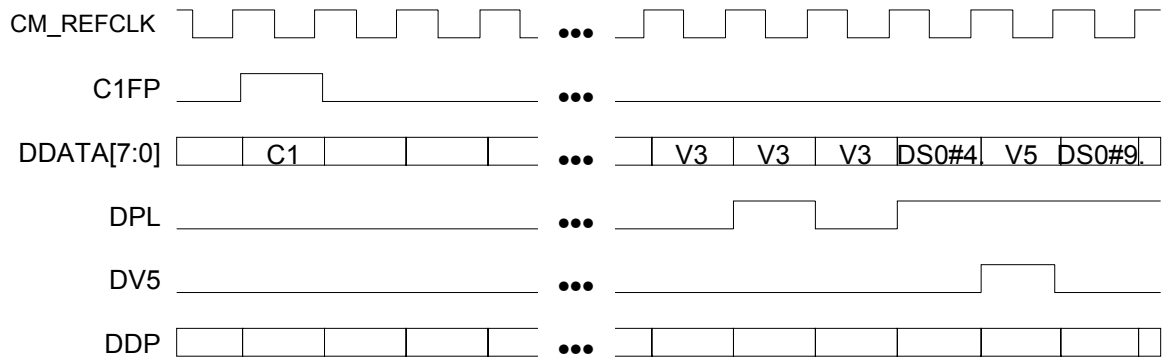
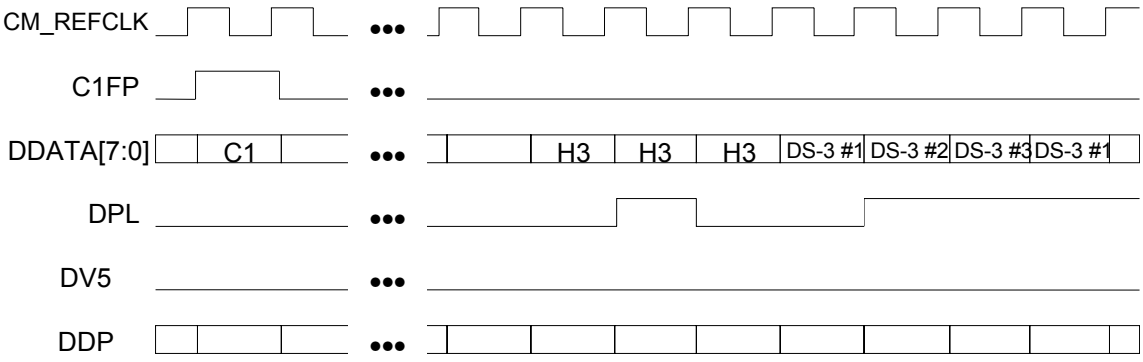


Figure 130 shows three DS-3 tributaries mapped onto the SBI bus. A negative justification is shown for DS-3 #2 during the H3 octet with DPL asserted high. A positive justification is shown for DS-3#1 during the first DS-3#1 octet after H3 which has DPL asserted low.

Figure 130 SBI DROP Bus DS3 Functional Timing

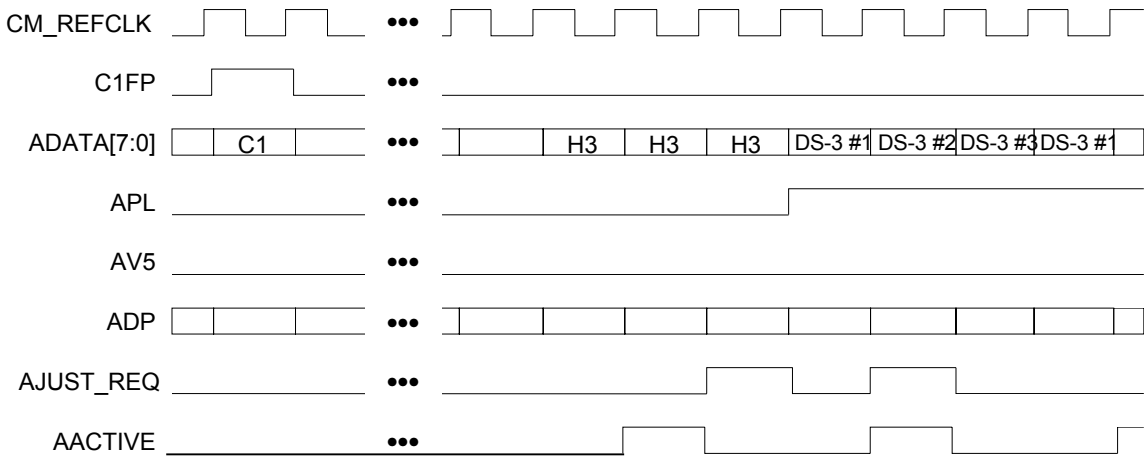


15.6.3.2 SBI ADD Bus Interface Timing

The SBI ADD bus functional timing for the transfer of tributaries whether T1/E1 or DS3 is the same as for the SBI DROP bus. The only difference is that the SBI ADD bus has a few additional signals. The AJUST_REQ signal is used to by the AAL1_LI in SBI master timing mode to provide transmit timing to SBI link layer devices. The AACTIVE signal is asserted whenever the AAL1_LI is driving the SBI ADD bus and is used with similar signals of other devices to detect and protect against SBI ADD bus conflicts.

Figure 131 illustrates the operation of the SBI ADD Bus, using positive and negative justification requests as an example. (The responses to the justification requests would take effect during the next multi-frame.) The negative justification request occurs on the DS-3#3 tributary when AJUST_REQ is asserted high during the H3 octet. The positive justification occurs on the DS-3#2 tributary when AJUST_REQ is asserted high during the first DS-3#2 octet after the H3 octet. The AACTIVE signal is shown for the case in which the AAL1_LI is driving DS-3#2 onto the SBI ADD bus.

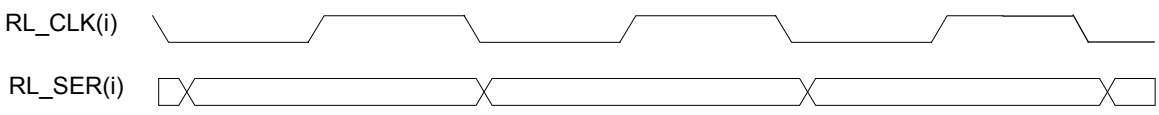
Figure 131 SBI ADD Bus Adjustment Request Functional Timing



15.6.4 DS3/E3 Timing

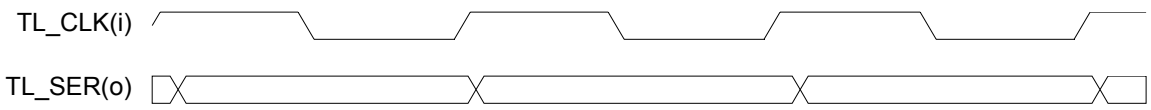
In UDF-HS mode there is no structure and the data is sampled using the falling edge of RL_CLK as shown in Figure 132. Because of the higher frequency, data should be updated using the falling edge of RL_CLK instead of the rising edge, which is used in low speed mode. This allows the full clock cycle to be used instead of half a clock cycle.

Figure 132 Receive High-Speed Functional Timing



In UDF-HS mode there is no structure and the data is driven using the rising edge of TL_CLK as shown in Figure 133. Data should be latched using the rising edge of TL_CLK so that the full cycle can be used.

Figure 133 Transmit High-Speed Functional Timing



16 ABSOLUTE MAXIMUM RATINGS

Maximum rating are the worst case limits that the device can withstand without sustaining permanent damage. They are not indicative of normal mode operation conditions.

Table 40 Absolute Maximum Ratings

Ambient Temperature under Bias	-40°C to +85°C
Storage Temperature	-40°C to +125°C
Supply Voltage (+3.3 Volt $V_{DD3.3}$)	-0.3V to +4.6V
Supply Voltage (+2.5 Volt $V_{DD2.5}$)	-0.3V to +3.5V
Voltage on Any Pin	-0.3V to 5.5V
Static Discharge Voltage	±1000 V
Latch-Up Current	±100 mA
DC Input Current	±20 mA
Lead Temperature	+230°C
Absolute Maximum Junction Temperature	+150°C

17 D.C. CHARACTERISTICS

(T_A = -40°C to +85°C, $V_{DD3.3}$ = 3.0 to 3.6 V, $V_{DD2.5}$ = 2.3 to 2.7 V)

Table 41 AAL1GATOR-32 D.C. Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Conditions
$V_{DD3.3}$	3.3V Power Supply	3.0	3.3	3.6	Volts	Note 5.
$V_{DD2.5}$	2.5V Power Supply	2.3	2.5	2.7	Volts	Note 5.
V_{IL}	Input Low Voltage	-0.5		0.8	Volts	All non-clock inputs, except TL_CLK. Also not rl_sync[3,2,1] and RSTB and TRSTB
V_{IH}	Input High Voltage	2.0		5.5	Volts	All non-clock inputs, , except TL_CLK. Also not rl_sync[3,2,1] and RSTB and TRSTB
V_{OL}	Output or Bi-directional Low Voltage			0.4	Volts	I_{OL} = -8 mA for the SBI and UTOPIA outputs. –-4mA for TDO. –6 mA for everything else. Notes 3, 5.
V_{OH}	Output or Bi-directional High Voltage	2.4			Volts	I_{OH} = 8 mA for the SBI and UTOPIA outputs. 4mA for TDO. 6 mA for everything else. Notes 3, 5.
V_{T+}	Schmitt Triggered Input High Voltage	2.0		5.5	Volts	All clock inputs, except TL_CLK. Also rl_sync[3,2,1] and RSTB and TRSTB
V_{T-}	Schmitt Triggered Input Low Voltage	-0.2		0.6	Volts	All clock inputs, except TL_CLK. Also rl_sync[3,2,1] and RSTB and TRSTB
I_{ILPU}	Input Low Current	+10	45	+150	µA	V_{IL} = GND, Notes 1, 3, 5.
I_{IHPU}	Input High Current	-10	0	+10	µA	$V_{IH} = V_{DD}$, Notes 1, 3

Symbol	Parameter	Min	Typ	Max	Units	Conditions
I_{IL}	Input Low Current	-10	0	+10	µA	V_{IL} = GND, Notes 2, 3
I_{IH}	Input High Current	-10	0	+10	µA	$V_{IH} = V_{DD}$, Notes 2, 3
C_{IN}	Input Capacitance		5		pF	Excludes package. Package typically 2 pF. Note 5.
C_{OUT}	Output Capacitance		5		pF	All pins. Excludes package. Package typically 2 pF. Note 5.
C_{IO}	Bi-directional Capacitance		5		pF	All pins. Excludes package. Package typically 2 pF. Note 5.
L_{PIN}	Pin Inductance		2		nH	All pins. Note 5.
I_{DDOP} (SBI2.7v)	Core Operating Current (SBI mode 2.7v).			500	mA	SBI Mode, Outputs typically loaded. All 32 links in E1 mode.
I_{DDOP} ($_{SBI3.3v}$)	I/O Operating Current (SBI mode 3.3v)		65		mA	SBI Mode, Outputs typically loaded. All 32 links in E1 mode.
I_{DDOP} (HS2.5v)	Core Operating Current (HS mode 2.5v).			300	mA	HS Mode, Outputs typically loaded. Two HS lines at 52 MHz.
I_{DDOP} (HS3.3v)	I/O Operating Current (HS mode 3.3v).		80		mA	HS Mode, Outputs typically loaded. Two HS lines at 52 MHz.

Notes on D.C. Characteristics:

1. Input pin or bi-directional pin with internal pull-up resistor.
2. Input pin or bi-directional pin without internal pull-up resistor.
3. Negative currents flow into the device (sinking), positive currents flow out of the device (sourcing).
4. Input pin or bi-directional pin with internal pull-down resistor.
5. Typical values are given as a design aid. The product is not tested to the typical values given in the data sheet.

18 A.C. TIMING CHARACTERISTICS

(T_A = -40°C to +85°C, $V_{DD3.3}$ = 3.0 to 3.6 V, $V_{DD2.5}$ = 2.3 to 2.7 V)

Notes on Input Timing:

1. When a set-up time is specified between an input and a clock, the set-up time is the time in nanoseconds from the 1.4 Volt point of the input to the 1.4 Volt point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4 Volt point of the clock to the 1.4 Volt point of the input.
3. It is recommended that the transition time on all clock inputs is less than 15 ns.

Notes on Output Timing:

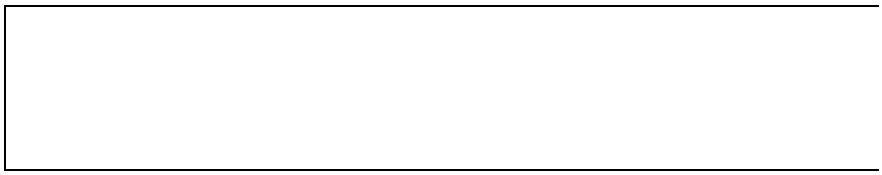
1. Output propagation delay time is the time in nanoseconds from the 1.4 Volt point of the reference signal to the 1.4 Volt point of the output.
2. Maximum and minimum output propagation delays are measured with a 100 pF load on all the outputs for the UTOPIA and SBI interface, and 50 pF load on microprocessor and TL_CLK outputs, and 25 pF on every other output.
3. Output tristate delay is the time in nanoseconds from the 1.4 Volt point of the reference signal to the point where the total current delivered through the output is less than or equal to the leakage current.

18.1 Reset Timing

Table 42 RTSB Timing

Symbol	Description	Min	Max	Units
tVRSTB	RSTB Pulse Width	64		SYS_CLK cycles*
*SYS_CLK must cycle through at least 64 rising edges while RSTB=0.				

Figure 134 RSTB Timing



18.2 SYS_CLK Timing

Table 43 SYS_CLK Timing

Symbol	Description	Min	Max	Units
fSYS	SYS_CLK Frequency (See notes below)	25	45	MHz
dSYS	SYS_CLK Duty Cycle	40	60	

Notes on SYS_CLK Timing:

1. Inputs are latched on rising edge of SYS_CLK. Outputs are driven off rising edge of SYS_CLK.

2. If any internal TL_CLK synthesizer is used, SYS_CLK must be 38.88 MHz +/- 50 ppm.

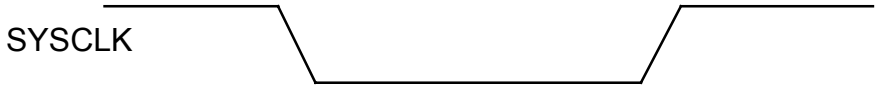
3. If no internal TL_CLK synthesizer is used, the SYS_CLK tolerance can be relaxed to +/- 200 ppm.

4. If it is desired that the internally synthesized TL_CLK is locked to a network clock, then SYS_CLK needs to be locked to that network clock.

5. To maintain sufficient bandwidth on all lines, SYS_CLK must be at least 38.88 MHz. For each line that is not used on the most loaded A1SP, the minimum frequency can be decreased by 4.5 MHz, if the internal clock synthesizers are not used.

6. The SYS_CLK minimum frequency is due to the internal DLL.

Figure 135 SYS_CLK Timing



18.3 NCLK Timing

Table 44 NCLK Timing

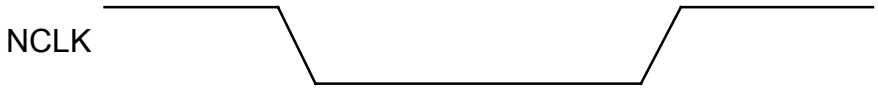
Symbol	Description	Min	Max	Units
fNCLK	NCLK Frequency (See notes below)		77.76 +50ppm	MHz
dNCLK	NCLK Duty Cycle	40	60	

Notes on NCLK Timing:

1. If DS3 is used NCLK should be 77.76 +/- 50 ppm.

2. An internal divider is available per A1SP, which can divide down the NCLK frequency. The internal NCLK frequency needs to be 2.43 MHz +/- 50 ppm for E1 and T1, 38.88 MHz +/- 50 ppm for E3, and 77.76 MHz +/-50 ppm for DS3.

Figure 136 NCLK Timing

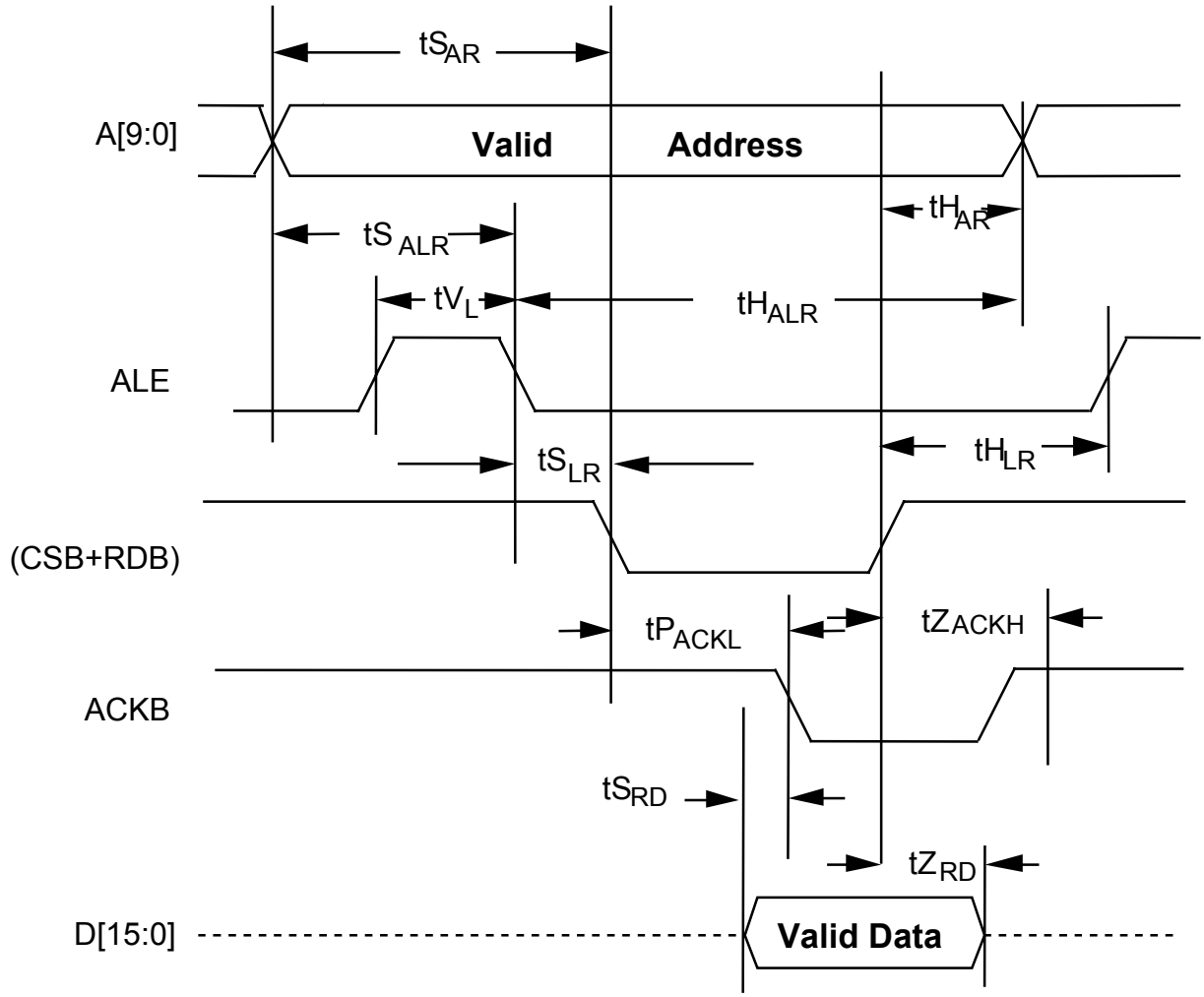


18.4 MICROPROCESSOR INTERFACE TIMING CHARACTERISTICS

Table 45 Microprocessor Interface Read Access

Symbol	Parameter	Min	Max	Units
tSAR	Address to Valid Read Set-up Time	10		ns
tHAR	Address to Valid Read Hold Time	5		ns
tSALR	Address to Latch Set-up Time	10		ns
tHALR	Address to Latch Hold Time	10		ns
tVL	Valid Latch Pulse Width	5		ns
tSLR	Latch to Read Set-up	0		ns
tHLR	Latch to Read Hold	5		ns
tPACKL	Valid Read to Valid ACKB Propagation Delay	3	400*	SYS CLK cycles
tZRD	Valid Read Negated to Output Tri-state		20	ns
tZACKH	Valid Read Negated to ACK Tri-state		20	ns
tSRD	Data to Valid ACKB Setup time	5		ns
* Microprocessor may momentarily experience excessive delays when accessing the external SSRAM, but will average 10-15 SYSCLK cycles. Microprocessor accesses to internal registers will not experience these excessive delays.				

Figure 137 Microprocessor Interface Read Timing



Notes on Microprocessor Interface Read Timing:

1. Output propagation delay time is the time in nanoseconds from the 1.4 Volt point of the reference signal to the 1.4 Volt point of the output.
2. Maximum output propagation delays are measured with a 50 pF load on the Microprocessor Interface data bus, (D[15:0]).
3. A valid read cycle is defined as a logical OR of the CSB and the RDB signals.
4. In non-multiplexed address/data bus architectures, ALE should be held high so parameters tSALR, tHALR, tVL, and tSLR are not applicable.

5. Parameter tHAR is not applicable if address latching is used.

6. Read Timing is dependent upon the region read and the SYS_CLK frequency. With a 40 MHZ SYS_CLK, Registers reads typically complete within 70 ns. Internal memory table reads typically complete within 110 ns, and external memory reads typically complete within 180ns to 200 ns. However memory accesses can take up to 10 us if there is a lot of contention.

7. WRB must be high during reads.

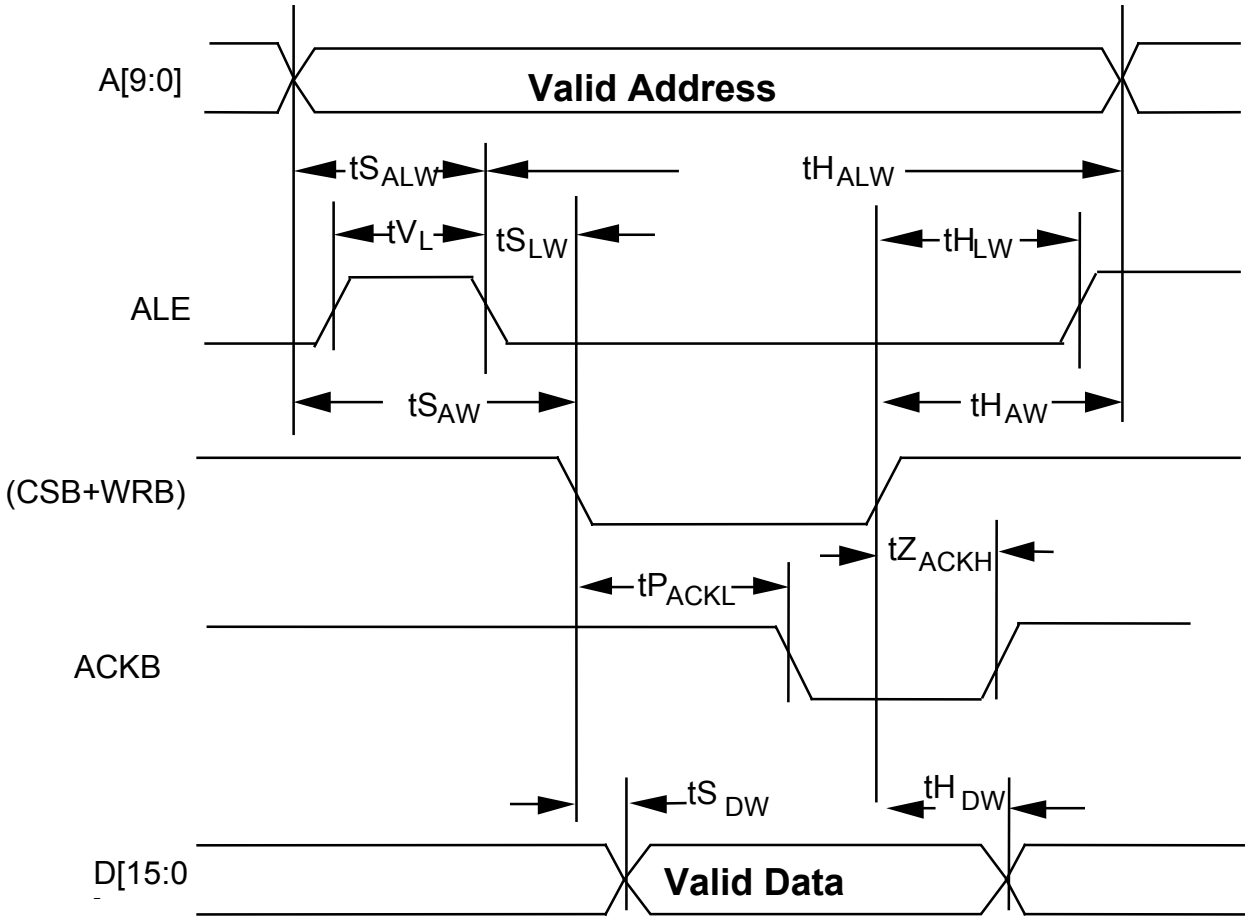
8. When reading the EXT_TRIAC or INS_TRIAC register (SBI control registers) it is possible for the BUSY bit to clear in the middle of the read.

9. Back to back reads of the same SBI interrupt register must be at least 250 ns apart

Table 46 Microprocessor Interface Write Access

Symbol	Parameter	Min	Max	Units
TSAW	Address to Valid Write Set-up Time	10		ns
TSDW	Write active to Data Valid Set-up Time		10	ns
TSALW	Address to Latch Set-up Time	10		ns
THALW	Address to Latch Hold Time	10		ns
TVL	Valid Latch Pulse Width	5		ns
TSLW	Latch to Write Set-up	0		ns
THLW	Latch to Write Hold	5		ns
THDW	Data to Valid Write Hold Time	5		ns
THAW	Address to Valid Write Hold Time	5		ns
TPACKL	Valid Write to Valid ACKB Propagation Delay	3	400	SYS_CLK cycles
TZACKH	Valid Write Negated to Output Tri-state		20	ns
* Microprocessor may momentarily experience excessive delays when accessing the external SSRAM, but will average 10-15 SYSCLK cycles. Microprocessor accesses to internal registers will not experience these excessive delays.				

Figure 138 Microprocessor Interface Write Timing



Notes on Microprocessor Interface Write Timing:

1. A valid write cycle is defined as a logical OR of the CSB and the WRB signals. These signals must be held active until ACKB goes low.

2. In non-multiplexed address/data bus architectures, ALE should be held high so parameters tSALW, tHALW, tVL, tSLW and tHLW are not applicable.

3. Parameter tHAW is not applicable if address latching is used.

4. Write Timing is dependent upon the region read and the SYS_CLK frequency. With a 40 MHZ SYS_CLK, Registers writes typically complete within 70 ns. Internal memory table writes typically complete within 110 ns, and external memory writes typically complete within 180ns to 200 ns. However memory accesses can take up to 10 us if there is a lot of contention.

5. Data is sampled internally one SYS_CLK cycle after CSB and WRB are detected low.

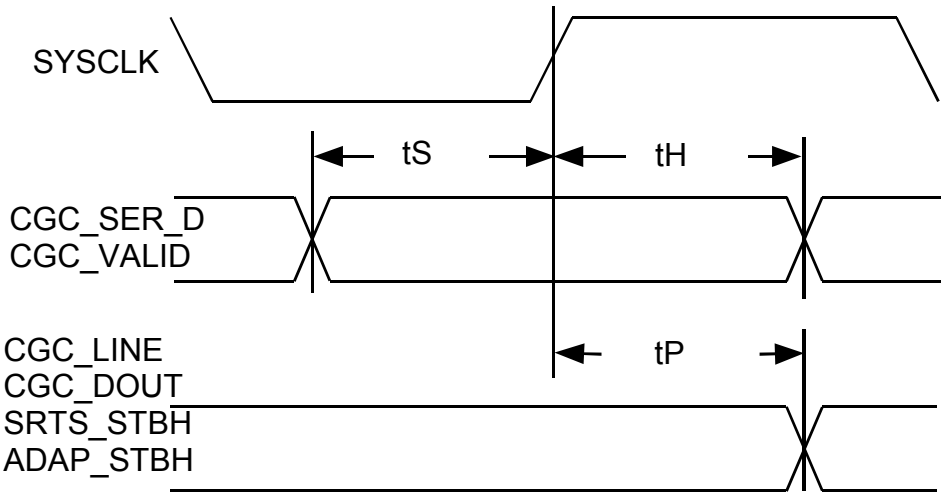
6. RDB must be high during writes.

18.5 External Clock Generation Control Interface

Table 47 External Clock Generation Control Interface

Symbol	Description	Min	Max	Units
f_{SCLK}	SYS_CLK Frequency (See notes below)	25	45	MHz
D_{SCLK}	SYS_CLK Duty Cycle	40	60	%
tS	Input Set-up time to SYS_CLK	4		ns
tH	Input Hold time to SYS_CLK	1		ns
tP	SYS_CLK High to Output Valid	1	12	ns

Figure 139 External Clock Generation Control Interface Timing

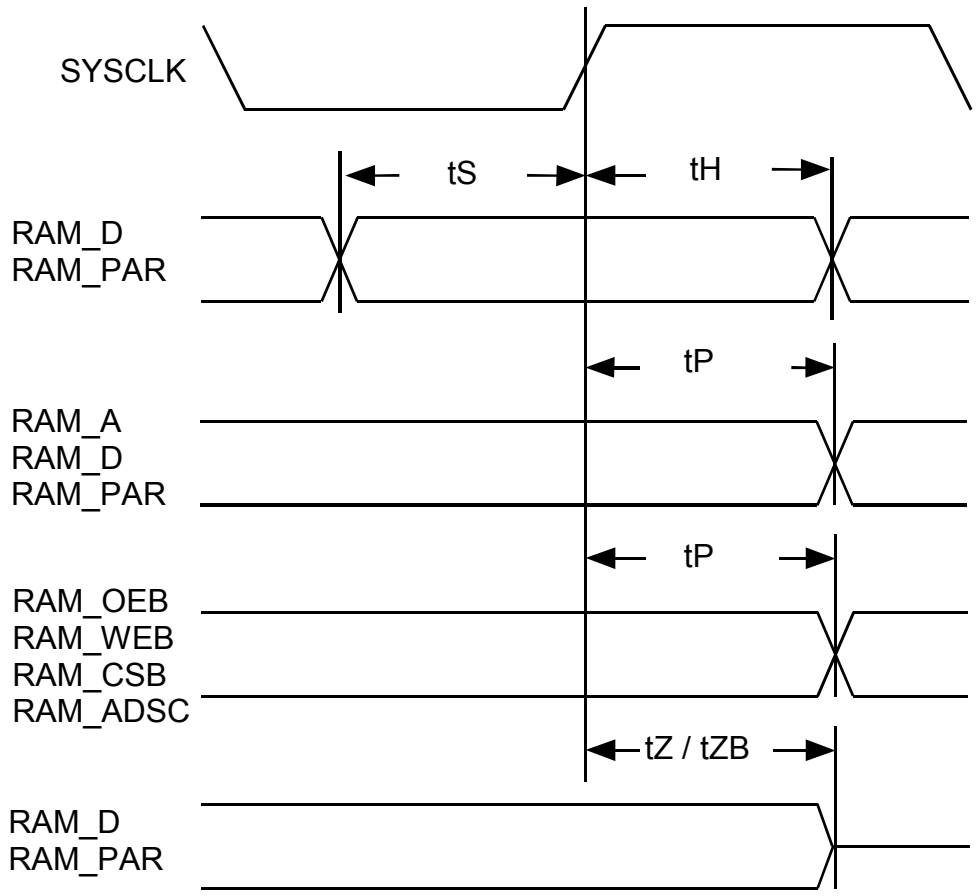


18.6 RAM Interface

Table 48 RAM Interface

Symbol	Description	Min	Max	Units
F_{SCLK}	SYS_CLK, Frequency	25	45	MHz
D_{SCLK}	SYS_CLK Duty Cycle	40	60	%
tS	Input Set-up time to SYS_CLK	4		ns
tH	Input Hold time to SYS_CLK	1		ns
tP	SYS_CLK High to Output Valid	1.5	12	ns
tZ	SYS_CLK High to Output High-Impedance	1.5	12	ns
tZB	SYS_CLK High to Output Driven	1.5	12	ns

Figure 140 RAM Interface Timing



18.7 UTOPIA INTERFACE

Table 49 UTOPIA Source and Sink Interface

Symbol	Description	Min	Max	Units
F	TATM_CLK/RATM_CLK Frequency	10	52	MHz
tD	TATM_CLK/RATM_CLK Duty Cycle	40	60	%
tS	Input Set-up time to TATM_CLK/RATM_CLK	4		ns
tH	Input Hold time to TATM_CLK/RATM_CLK	1		ns
tP	TATM_CLK/RATM_CLK High to Output Valid	2	14	ns
tZ	TATM_CLK/RATM_CLK High to Output High-Impedance	2	14	ns
tZB	TATM_CLK/RATM_CLK High to Output Driven	2	14	ns

Figure 141 Sink UTOPIA Interface Timing

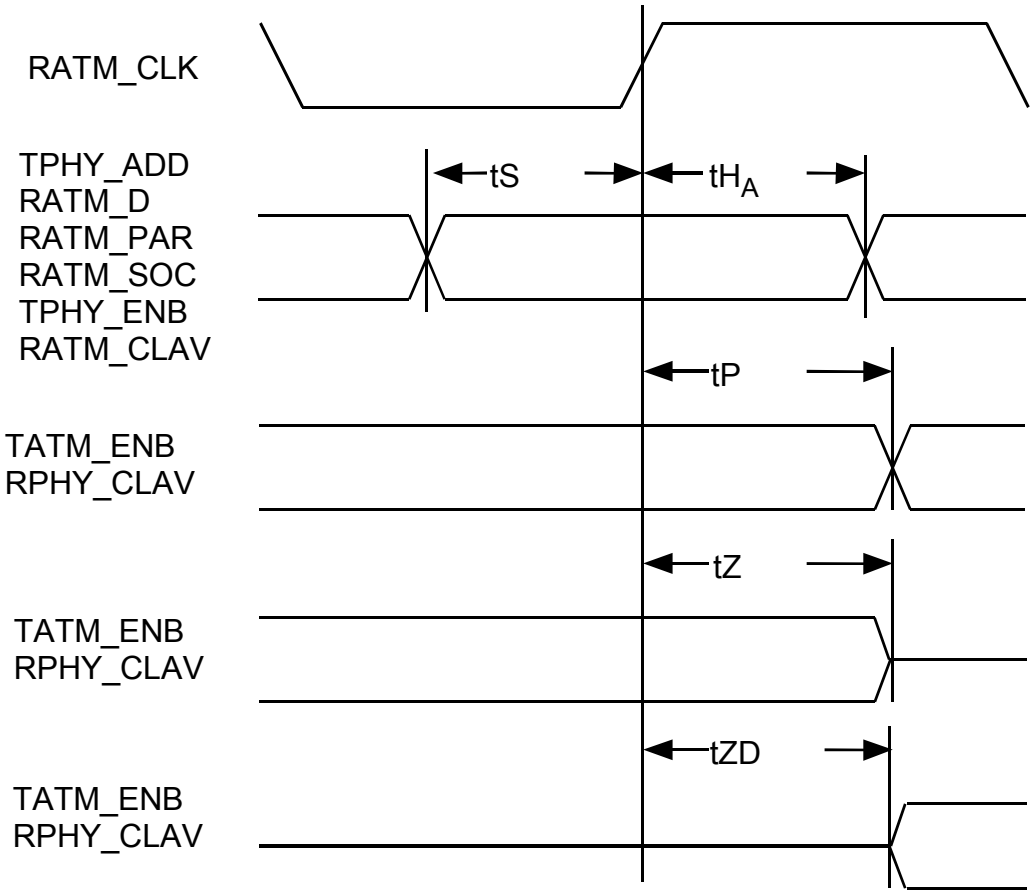
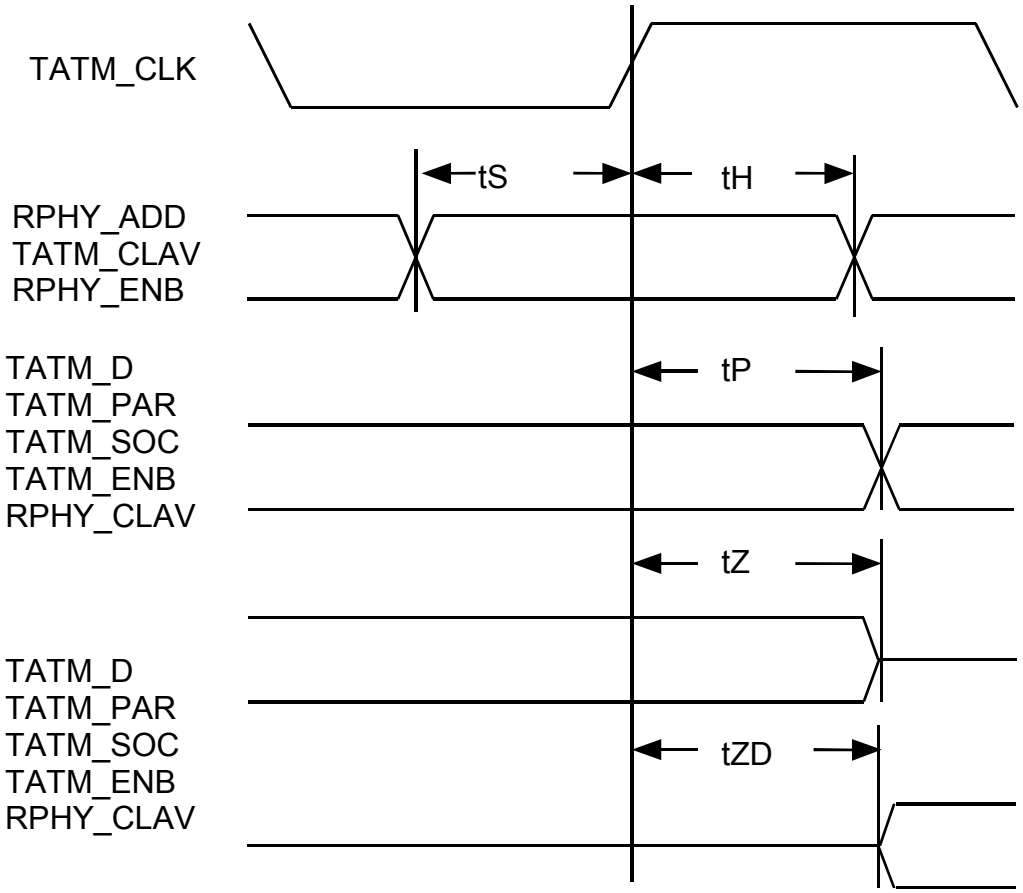


Figure 142 Source UTOPIA Interface Timing



18.8 LINE I/F Timing

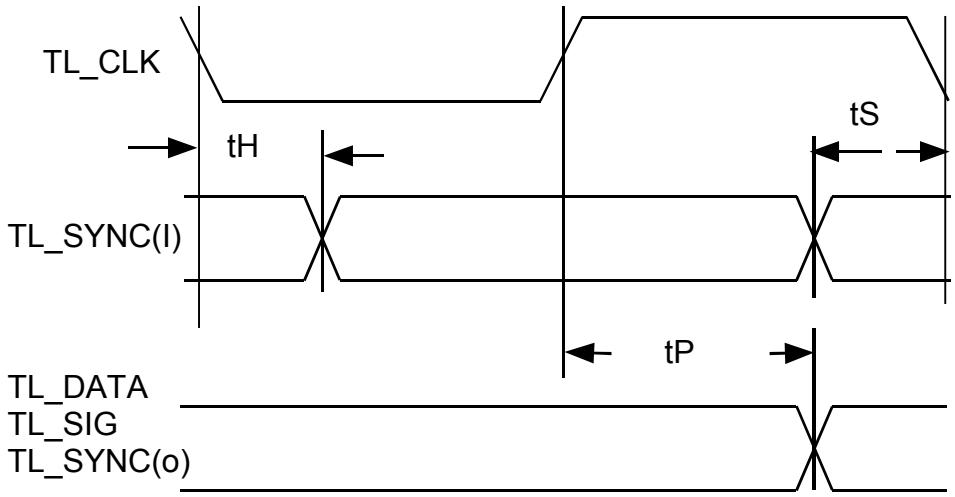
18.8.1 Direct Low Speed Timing

18.8.1.1 Transmit

Table 50 Transmit Low Speed Interface Timing

Symbol	Description	Min	Max	Units
f_T	TL_CLK Frequency	0	15	MHz
d_T	TL_CLK Duty Cycle	40	60	%
tS	Input Set-up time to falling edge of TL_CLK	10		ns
tH	Input Hold time from falling edge TL_CLK	10		ns
tP	TL_CLK High to Output Valid	3	15	ns

Figure 143 Transmit Low Speed Interface Timing



Notes on Transmit Low Speed Timing:

1. Outputs are driven using the rising edge of TL_CLK and inputs are expected to be driven using the rising edge of TL_CLK.

2. Inputs are latched using the falling edge of TL_CLK.

3. The maximum frequency per line in low speed mode is 15 MHz. However aggregate speed for all lines within an A1SP must be 20 MHz or less with a SYS_CLK of 38.88 MHz.

18.8.1.2 Receive

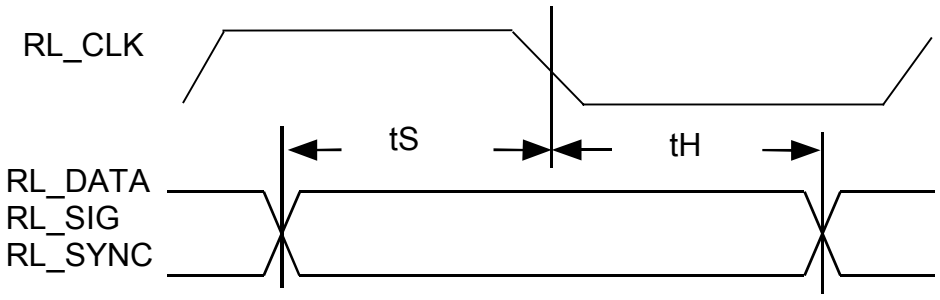
Table 51 Receive Low Speed Interface Timing

Symbol	Description	Min	Max	Units
f_R	RL_CLK Frequency (See note 3)	0	15	MHz
d_R	RL_CLK Duty Cycle	40	60	%
tS	Input Set-up time to falling edge of RL_CLK	10		ns
tH	Input Hold time from falling edge RL_CLK	10		ns

Notes on Receive Low Speed Timing:

1. Inputs are expected to be driven using the rising edge of RL_CLK.
2. Inputs are latched using the falling edge of RL_CLK.
3. The maximum frequency per line in low speed mode is 15 MHz. However aggregate speed for all lines within an A1SP must be 20 MHz or less with a SYS_CLK of 38.88 MHz.
4. For applications which use SRTS clock recovery, the RL_CLK must not be a gapped clock and jitter should be less than .3 UI.

Figure 144 Receive Low Speed Interface Timing



18.8.2 SBI Timing

Table 52 Clocks and SBI Frame Pulse (Figure 145)

Symbol	Description	Min	Max	Units
	REFCLK Frequency	19.44 – 50 ppm	19.44 +50 ppm	MHz
	REFCLK Duty Cycle	40	60	%
TS_{C1FP}	C1FP* Set-Up Time to REFCLK	4		ns
TH_{C1FP}	C1FP* Hold Time to REFCLK	0.5		ns

Note: *C1FP timing applies to both C1FP and C1FP_ADD

Figure 145 SBI Frame Pulse Timing

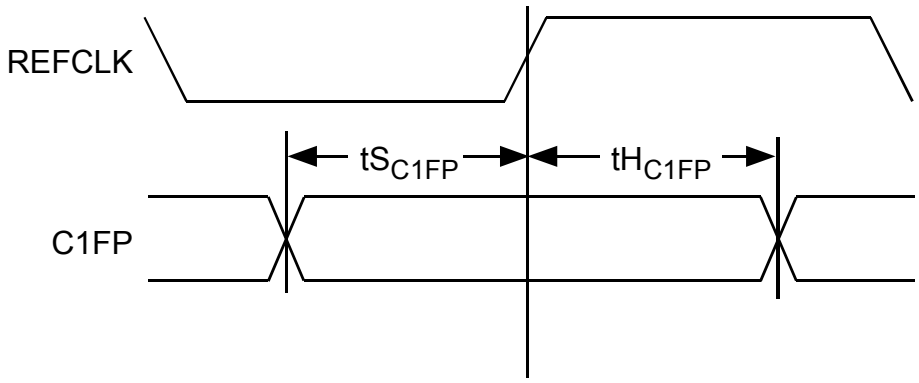


Table 53 SBI DROP BUS (Figure 146)

Symbol	Description	Min	Max	Units
$tS_{SBIDROP}$	All SBI DROP BUS Inputs Set-Up Time to REFCLK	4		ns
$tH_{SBIDROP}$	All SBI DROP BUS Inputs Hold Time to REFCLK	0		ns

Figure 146 SBI DROP BUS Timing

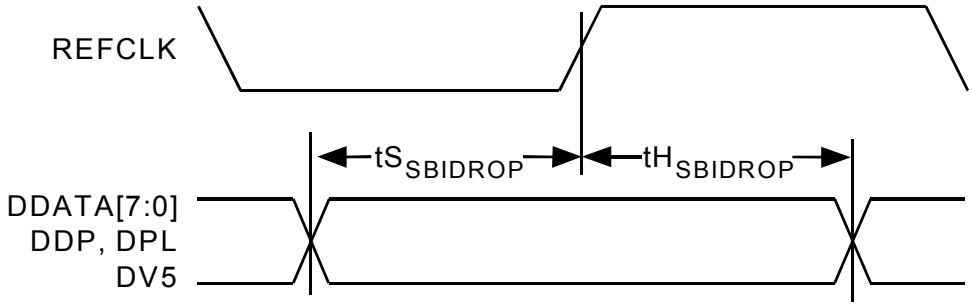


Table 54 SBI ADD BUS (Figure 147 to Figure 148)

Symbol	Description	Min	Max	Units
tS_{SBIADD}	AJUST_REQ Set-Up Time to REFCLK	4		ns
tH_{SBIADD}	AJUST_REQ Hold Time to REFCLK	0		ns
$tP_{AACTIVE}$	REFCLK to AACTIVE Valid	3	15	ns
tP_{SBIADD}	REFCLK to All SBI ADD BUS Outputs (except AACTIVE) Valid	3	20	ns
tZ_{SBIADD}	REFCLK to All SBI ADD BUS Outputs (except AACTIVE) Tristate	3	20	ns
tP_{OUTEN}	ADETECT low to All SBI ADD BUS Outputs (except AACTIVE) Valid	3	12.5	ns
tZ_{OUTEN}	ADETECT high to All SBI ADD BUS Outputs (except AACTIVE) Tristate	3	12.5	ns

Figure 147 SBI ADD BUS Timing

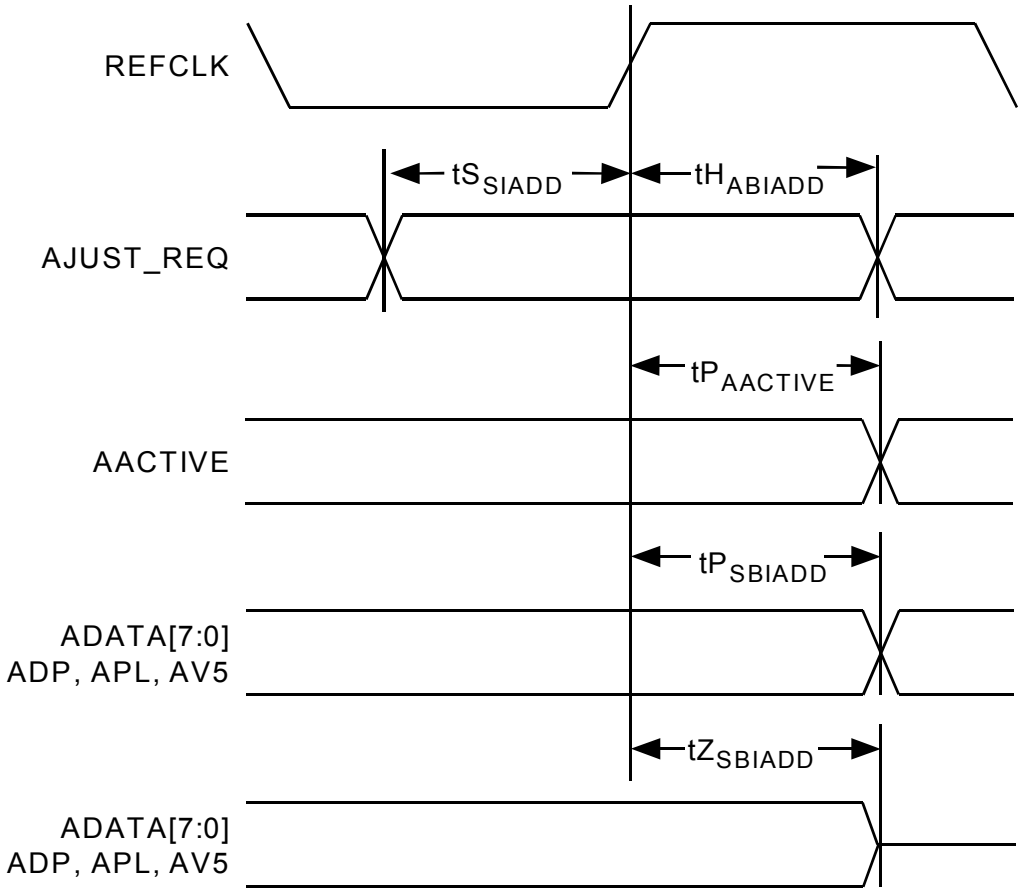
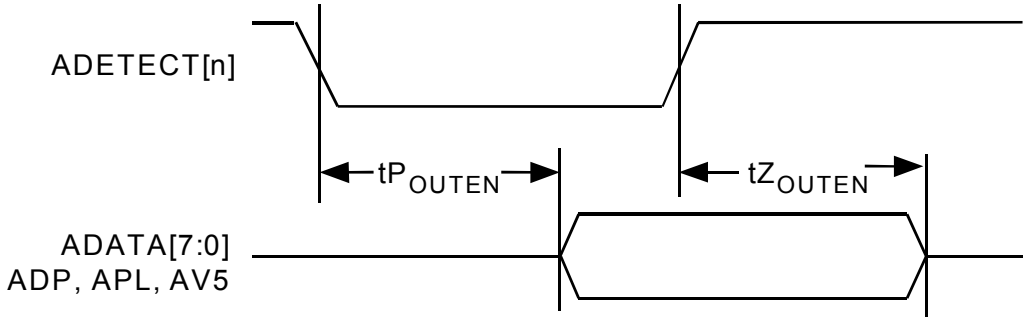


Figure 148 SBI ADD BUS Collision Avoidance Timing



18.8.3 H-MVIP Timing

Table 55 H-MVIP Sink Timing

Symbol	Description	Min	Max	Units
	C16B Frequency (See Note 1)	16.368	16.400	MHz
	C16B Duty Cycle	40	60	%
	C4B Frequency (See Note 2)	4.092	4.100	MHz
	C4B Duty Cycle	40	60	%
tP_{C16C4}	C16B to C4B skew	-10	10	ns
tS_{C16B}	RL_DATA, RL_SIG Set-Up Time to rising edge of C16B	5		ns
tH_{C16B}	RL_DATA, RL_SIG Hold Time from rising edge of C16B	5		ns
tS_{C4B}	F0B Set-Up Time to falling edge of C4B	5		ns
tH_{C4B}	F0B Hold Time from falling edge of C4B	5		ns

Notes on H-MVIP Sink Timing:

1. Measured between any two C16B falling edges.
2. Measured between any two C4B falling edges.

Figure 149 H-MVIP Sink Data & Frame Pulse Timing

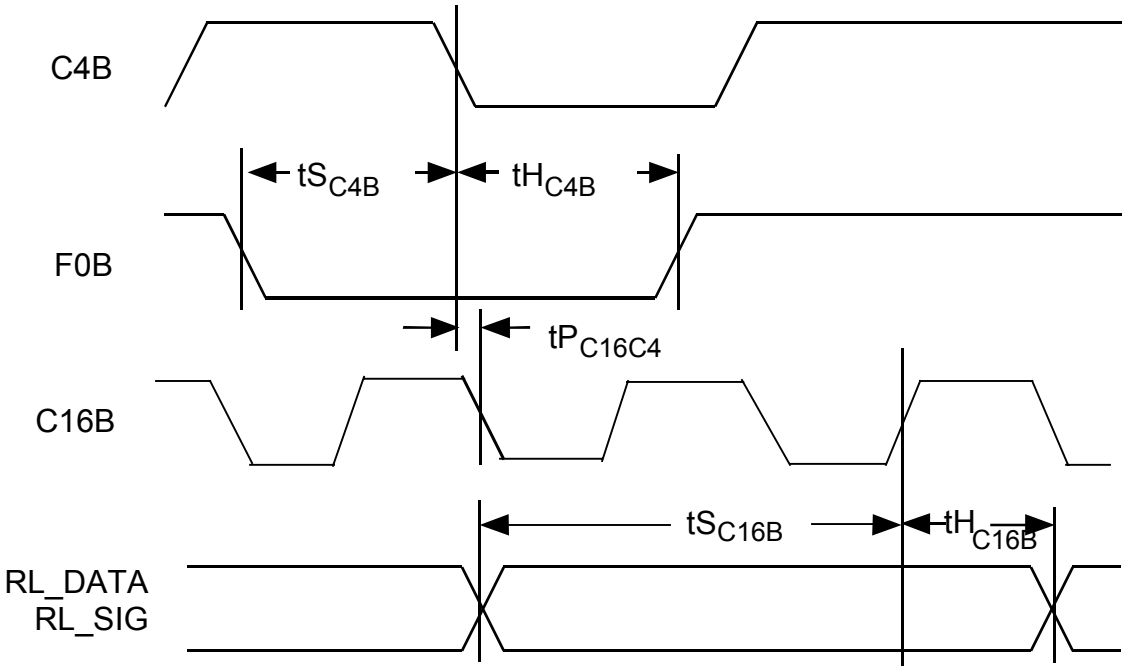


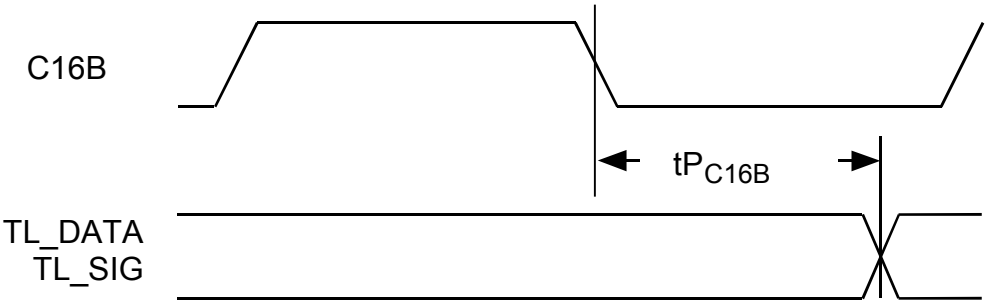
Table 56 H-MVIP Source Timing

Symbol	Description	Min	Max	Units
tP_{C16B}	C16B Low to TL_DATA, TL_SIG valid	3	15	ns

Notes on H-MVIP Source Timing:

1. Outputs are driven off the falling edge of C16B and are held for two C16B clock cycles

Figure 150 H-MVIP Ingress Data Timing



18.8.4 High Speed Timing

18.8.4.1 Transmit

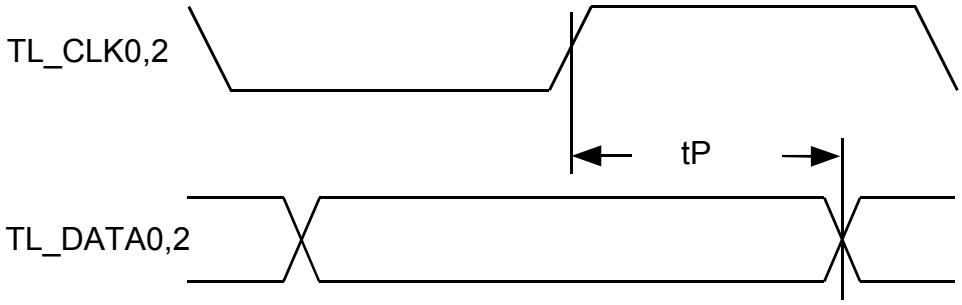
Table 57 Transmit High Speed Interface Timing

Symbol	Description	Min	Max	Units
f_T	TL_CLK0,2 Frequency (see note 2)	0	45	MHz
d_T	TL_CLK0,2 Duty Cycle	40	60	%
tP	TL_CLK0,2 High to Output Valid	3	13	ns

Notes on Transmit High Speed Timing:

1. Outputs are driven using the rising edge of TL_CLK.
2. The maximum frequency per line in High Speed mode is 45 MHz when SYS_CLK is 38.88 MHz. However if SYS_CLK is 45 MHz then the maximum frequency for TL_CLK0,2 is 52 MHz.

Figure 151 Transmit High Speed Timing



18.8.4.2 Receive

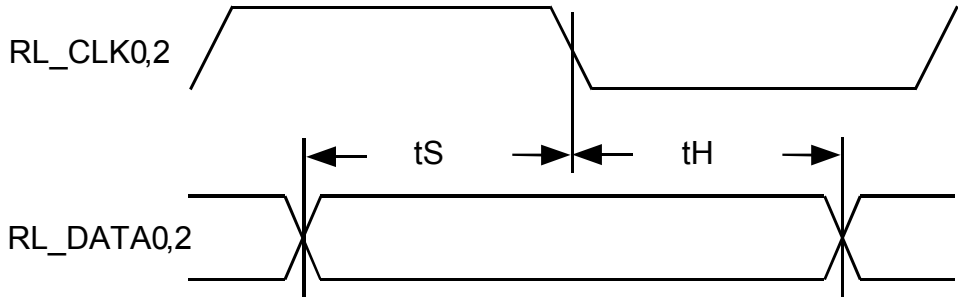
Table 58 Receive High Speed Interface Timing

Symbol	Description	Min	Max	Units
f_R	RL_CLK0,2 Frequency	0	45	MHz
d_R	RL_CLK0,2 Duty Cycle	40	60	%
tS	Input Set-up time to falling edge of RL_CLK0,2	5		ns
tH	Input Hold time from falling edge RL_CLK0,2	1		ns

Notes on Receive High Speed Timing:

1. Inputs are latched using the falling edge of RL_CLK.
2. The maximum frequency per line in High Speed mode is 45 MHz when SYS_CLK is 38.88 MHz. However if SYS_CLK is 45 MHz then the maximum frequency for RL_CLK0,2 is 52 MHz.
3. For applications which use SRTS clock recovery, the RL_CLK must not be a gapped clock and jitter should be less than .3 UI.

Figure 152 Receive High Speed Interface Timing

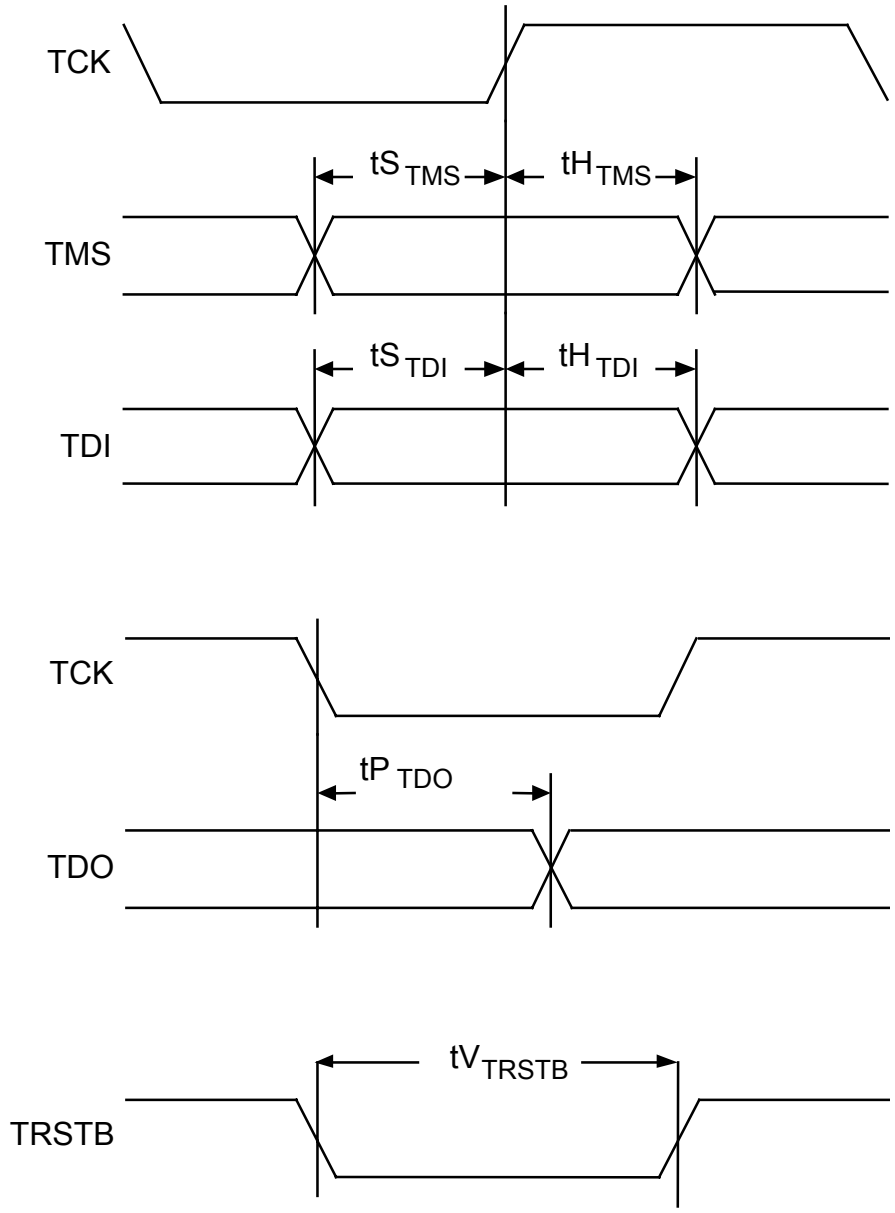


18.9 JTAG Timing

Table 59 JTAG Port Interface

Symbol	Description	Min	Max	Units
	TCK Frequency		1	MHz
	TCK Duty Cycle	40	60	%
tS_{TMS}	TMS Set-up time to TCLK	50		ns
tH_{TMS}	TMS Hold time to TCLK	50		ns
tS_{TDI}	TDI Set-up time to TCLK	50		ns
tH_{TDI}	TDI Hold time to TCLK	50		ns
tP_{TDO}	TCLK Low to TDO Valid	2	50	ns
tV_{TRSTB}	TRSTB Pulse Width	100		ns

Figure 153 JTAG Port Interface Timing



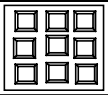
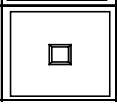
19 ORDERING AND THERMAL INFORMATION

Table 60 AAL1GATOR-32 (PM73122) Ordering Information

Part No.	Description
PM73122-BI	352 Pin Ball Grid Array (SBGA)

Table 61 AAL1GATOR-32 (PM73122) Thermal Information

PART NO.	AMBIENT TEMPERATURE	Theta Ja	Theta Jc
PM73122-BI	-40°C to +85°C	19.3 °C/W	1 °C/W

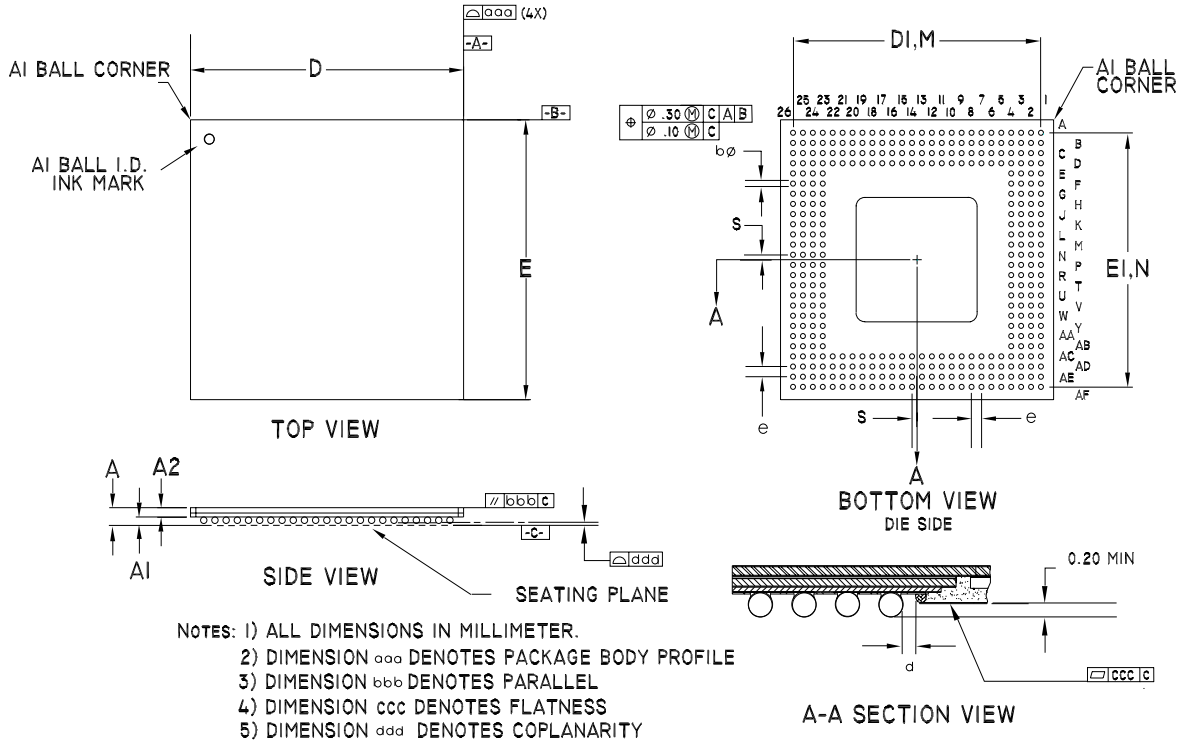
			Forced Air (Linear Feet per Minute)				
Theta JA @ 1.3 watts		Conv	100	200	300	400	500
Dense Board		19.3	17.3	15.9	15.0	14.6	14.6
JEDEC Board		13.6	12.6	11.9	11.4	10.9	10.3

1. DENSE Board is defined as a 3S3P board and consists of a 3x3 array of device PM73122-BI located as close to each other as board design rules allow. All PM73122-BI devices are assumed to be dissipating 1.4 Watts. Θ_{JA} listed is for the device in the middle of the array.

2. EDEC Board Θ_{JA} is the measured value for a single thermal device in the same package on a 2S2P board following EIA/JESD 51-3

20 MECHANICAL INFORMATION

Figure 154 352 Pin Enhanced Ball Grid Array (SBGA)



NOTES: 1) ALL DIMENSIONS IN MILLIMETER.
2) DIMENSION aaa DENOTES PACKAGE BODY PROFILE
3) DIMENSION bbb DENOTES PARALLEL
4) DIMENSION ccc DENOTES FLATNESS
5) DIMENSION ddd DENOTES COPLANARITY

PACKAGE TYPE: 352 PIN THERMAL BALL GRID ARRAY																
BODY SIZE: 35 X 35 X 1.51 MM																
DIM.	A	A1	A2	D	D1	E	E1	M,N	e	b	aaa	bbb	ccc	ddd	d	S
MIN.	1.30	0.50	0.80	34.90	31.65	34.90	31.65			0.60					0.50	
NOM.	1.51	0.60	0.91	35.00	31.75	35.00	31.75	26X26	1.27	0.75						0.635
MAX.	1.70	0.70	1.00	35.10	31.85	35.10	31.85			0.90	0.20	0.25	0.20	0.20		

21 DEFINITIONS

Transmit Signals	All signals related to transmitting ATM or TDM data from the chip.
Receive Signals	All signals related to receiving ATM or TDM data into the chip.
Transmit Blocks	All blocks which process data in the ingress direction; towards the ATM network.
Receive Blocks	All blocks which process data in the egress direction; away from the ATM network.
AAL1	ATM Adaptation Layer 1
ATM	Asynchronous Transfer Mode
CAS	Channel Associated Signaling
CBR	Constant Bit Rate
CCS	Common Channel Signaling
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CES	Circuit Emulation Services
CLP	Cell Loss Priority
CRC	Cyclic Redundancy Check
CRC-10	10-bit Cyclic Redundancy Check
CSD	Cell Service Decision
CSI	Convergence Sublayer Indication
DAC	Digital-to-Analog Converter
DACS	Digital Access Cross-connect System
DDS	Digital Data Service

DS0	Digital Signal Level 0
DS1	Digital Signal Level 1
DS3	Digital Signal Level 3
E1	European Digital Signal Level 1
E3	European Digital Signal Level 3
ESF	Extended Super Frame
FIFO	First-In, First-Out
FPGA	Field Programmable Gate Array
FXO	Foreign Exchange Office
FXS	Foreign Exchange Subscriber
HEC	Header Error Check
LIU	Line Interface Unit
LSB	Least Significant Bit
M13	Multiplexer Level 1 to Level 3
MIAC	Memory Interface and Arbitration Controller
MIB	Management Information Base
Mod	modulo
MPHY	Multi-PHY
MSB	Most Significant Bit
OAM	Operations, Administration, and Maintenance
PBX	Private Branch Exchange
PCR	Peak Cell Rate
PDH	Pleisochronous Digital Hierarchy
PHY	Physical Layer

PLL	Phase-Locked Loop
PCM	Pulse Code Modulation
PRBS	Pseudorandom Bit Sequence
PTI	Payload Type Indicator
RALP	Receive Adaptation Layer Processing
RATM	Receive UTOPIA ATM Layer
RFTC	Receive Frame Transfer Controller
RUTOPIA	Receive UTOPIA
SAR	Segmentation and Reassembly
SDF-FR	Structured Data Format, Frame-based
SDF-MF	Structured Data Format, Multiframe-based
SDU	Service Data Unit
SF	Super Frame
SN	Sequence Number
SNP	Sequence Number Protection
SOC	Start-Of-Cell
SP	Supervisory Processor
SPHY	Single PHY
SRAM	Static Random Access Memory
SRTS	Synchronous Residual Time Stamp
SSRAM	Synchronous Static Random Access Memory
TALP	Transmit Adaptation Layer Processor
TATM	Transmit UTOPIA ATM Layer
TDM	Time Division Multiplexing

TFTC	Transmit Frame Transfer Controller
TLIP	Transmit Line Interface Processor
TTL	Transistor-to-Transistor Logic
TUTOPIA	Transmit UTOPIA
UDF	Unstructured Data Format
UDF-HS	Unstructured Data Format, High Speed
UDF-ML	Unstructured Data Format, Multiple Line
UI	Unit Interval
UTOPIA	Universal Test and Operations Physical Interface for ATM
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VCO	Voltage Controlled Oscillator
VCXO	Voltage Controlled Crystal Oscillator
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VP	Virtual Path
VPI	Virtual Path Identifier
ZBT	Zero Bus Turnaround

PATENTS

The technology discussed is protected by the following Patent:

U.S. Patent No. 5,844,901

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
Web Site: http://www.pmc-sierra.com

Licensing SRTS Technology

Synchronous residual time stamp (SRTS) technology is patented by Telcordia Technologies (formerly Bellcore) under US patent 5,260,978. In a letter to the ATM Forum, Bellcore has stated that "Bellcore patents are available for licensing on a non-exclusive and nondiscriminatory basis and at reasonable royalties". It is our understanding that Telcordia prefers to make such licenses available to equipment vendors and does not make licenses available to integrated circuit component vendors at this time.

Accordingly, PMC-Sierra does not provide any patent licensing protection from infringement of US Patent No. 5,260,978, or any other third party patents, to any users of AAL1gator™ products. PMC-Sierra will not indemnify for or defend against patent infringement claims or suits brought by any third parties. In this regard, it is the customer's responsibility to obtain all necessary licenses. We recommend that any manufacturer that makes use of SRTS functionality (by using an AAL1gator product or via some other implementation) establishes its own license agreement with Telcordia.

PMC- 1981419 Issue date: June 2001