3-CHANNEL DS3/E3/STS-1 TO STS-3/STM-1 MAPPER IC DATA SHEET

N0VEMBER 2006 *REV.1.2.0.*

GENERAL DESCRIPTION

The XRT94L33 is a highly integrated SONET/SDH terminator designed for E3/DS3/STS-1 mapping/de-mapping functions from either the STS-3 or STM-1 data stream. The XRT94L33 interfaces directly to the optical transceiver

The XRT94L33 processes the section, line and path overhead in the SONET/SDH data stream and also performs ATM and PPP PHY-layer processing. The processing of path overhead bytes within the STS-1s or TUG-3s includes 64 bytes for storing the J1 bytes. Path overhead bytes can be accessed through the microprocessor interface or via serial interface.

The XRT94L33 uses the internal E3/DS3 De-Synchronizer circuit with an internal pointer leak algorithm for clock smoothing as well as to remove the jitter due to mapping and pointer movements. These De-Synchronizer circuits do not need any external clock reference for its operation.

The SONET/SDH transmit blocks allow flexible insertion of TOH and POH bytes through both Hardware and Software. Individual POH bytes for the transmitted SONET/SDH signal are mapped either from the XRT94L33 memory map or from external interface. A1, A2 framing pattern, C1 byte and H1, H2 pointer byte are generated.

The SONET/SDH receive blocks receive SONET STS-3 signal or SDH STM-1 signal and perform the necessary transport and path overhead processing.

The XRT94L33 provides a line side APS (Automatic Protection Switching) interface by offering redundant receive serial interface to be switched at the frame boundary.

The XRT94L33 provides 3 mappers for performing STS-1/VC-3 to STS-1/DS3/E3 mapping function, one for each STS-1/DS3/E3 framers.

A PRBS test pattern generation and detection is implemented to measure the bit-error performance.

A general-purpose microprocessor interface is included for control, configuration and monitoring.

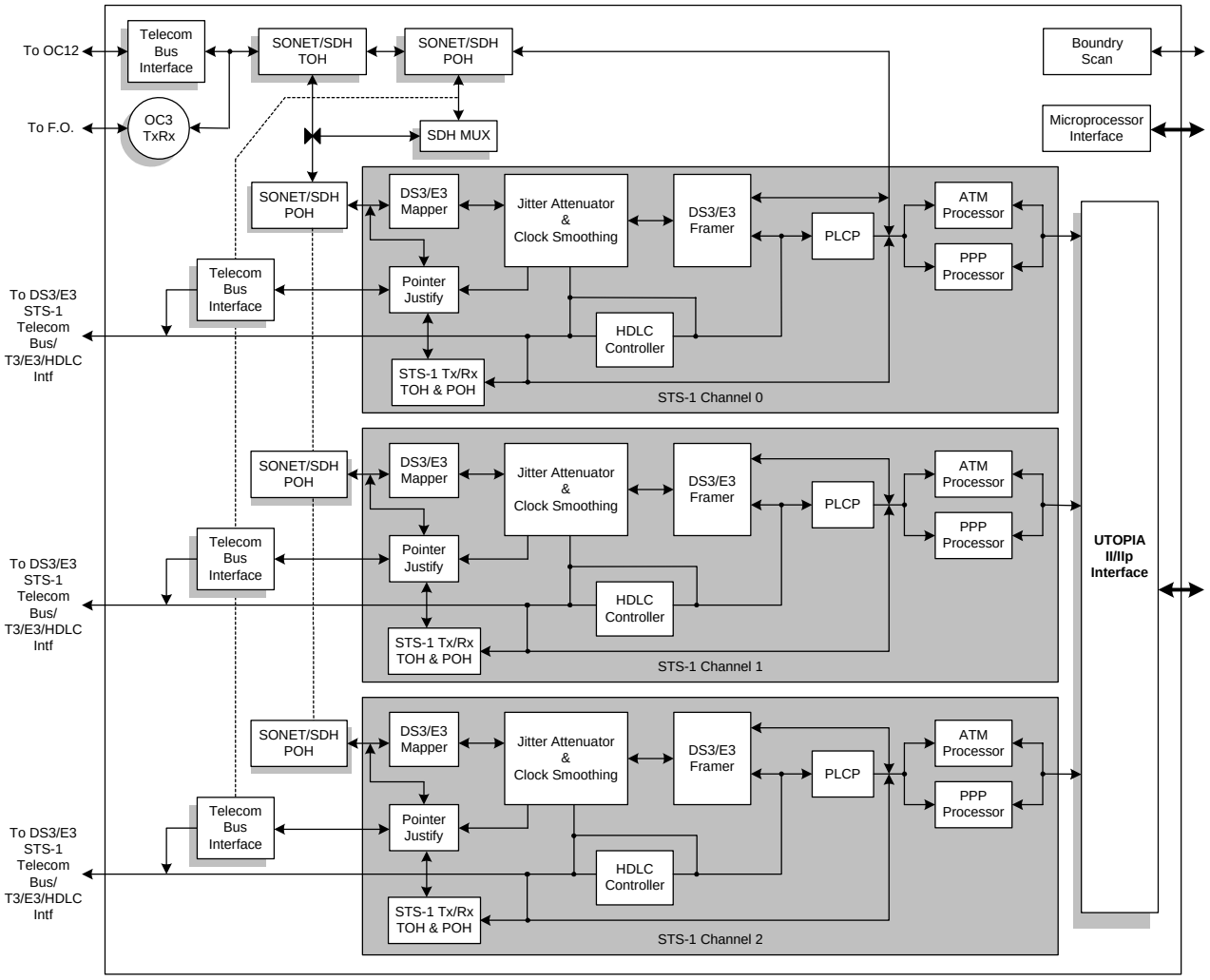
APPLICATIONS

- Network switches
- Add/Drop Multiplexer
- W-DCS Digital Cross Connect Systems

FEATURES

- Provides DS3/ E3 mapping/de-mapping for up to 3 tributaries through SONET STS-1 or SDH AU-3 and/or TUG-3/AU-4 containers
- Generates and terminates SONET/SDH section, line and path layers
- Integrated SERDES with Clock Recovery Circuit
- Provides SONET frame scrambling and descrambling
- Integrated Clock Synthesizer that generates 155 MHz and 77.76 MHz clock from an external 12.96/19.44/77.76 MHz reference clock
- Integrated 3 E3/DS3/STS-1 De-Synchronizer circuit that de-jitter gapped clock to meet 0.05UIpp jitter requirements
- Access to Line or Section DCC
- Level 2 Performance Monitoring for E3 and DS3
- Supports mixing of STS-1E and DS3 or E3 and DS3 tributaries
- UTOPIA Level 2 interface for ATM or level 2P for Packets
- E3 and DS3 framers for both Transmit and Receive directions
- Complete Transport/Section Overhead Processing and generation per Telcordia and ITU standards
- Single PHY and Multi-PHY operations supported
- Full line APS support for redundancy applications
- Loopback support for both SONET/SDH as well as E3/DS3/STS-1
- Boundary scan capability with JTAG IEEE 1149
- 8-bit microprocessor interface
- 3.3 V ± 5% Power Supply; 5 V input signal tolerance
- -40°C to +85°C Operating Temperature Range
- Available in a 504 Ball TBGA package

Block Diagram of the XRT94L33



ORDERING INFORMATION

PART NUMBER	PACKAGE TYPE	OPERATING TEMPERATURE RANGE
XRT94L33IB	27 x 27 504 Lead TBGA	-40°C to +85°C

PIN DESCRIPTION of the XRT94L33 (Rev. B)

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
				MICROPROCESSOR INTERFACE
Y22	PCLK	I	TTL	**Microprocessor Interface Clock Input:** This clock input signal is only used if the Microprocessor Interface has been configured to operate in one of the Synchronous Mode (e.g., Power PC 403 Mode). If the Microprocessor Interface is configured to operate in one of these modes, then it will use this clock signal to do the following. • To sample the CS*, WR*/R/W*, A[14:0], D[7:0], RD*/DS* and DBEN input pins, and • To update the state of D[7:0] and the RDY/DTACK output signals. ***NOTES:*** 1. The Microprocessor Interface can work with μPCLK frequencies ranging up to 33MHz. 2. This pin is inactive if the user has configured the Microprocessor Interface to operate in either the Intel-Asynchronous or the Motorola-Asynchronousl Modes. In this case, the user should tie this pin to GND.
AD25 AD23 AC21	PTYPE_0 PTYPE_1 PTYPE_2	I	TTL	**Microprocessor Type Select input:** These three input pins permit the user to configure the Microprocessor Interface block to readily support a wide variety of Microprocessor Interfaces. The relationship between the settings of these input pins and the corresponding Microprocessor Interface configuration is presented below. **PTYPE[2:0] Microprocessor Interface Mode** 000 Intel-Asynchronous Mode 001 Motorola – Asynchronous Mode 010 Intel X86 011 Intel I960 100 IDT3051/52 (MIPS) 101 Power PC 403 Mode 111 Motorola 860

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AD27 AB25 W23 Y24 AD26 AC25 AA24 Y23 AE24 AB20 AD22 AC20 AD21 AE23 AF24	PADDR_0 PADDR_1 PADDR_2 PADDR_3 PADDR_4 PADDR_5 PADDR_6 PADDR_7 PADDR_8 PADDR_9 PADDR_10 PADDR_11 PADDR_12 PADDR_13 PADDR_14	I	TTL	**Address Bus Input pins (Microprocessor Interface):** These pins permit the Microprocessor to identify on-chip registers and Buffer/Memory locations (within the XRT94L33) whenever it performs READ and WRITE operations with the XRT94L33.
AD20 AC19 AE22 AG24 AE21 AD19 AF23 AE20	PDATA_0 PDATA_1 PDATA_2 PDATA_3 PDATA_4 PDATA_5 PDATA_6 PDATA_7	I/O	TTL	**Bi-Directional Data Bus pins (Microprocessor Interface):** These pins are used to drive and receive data over the bi-directional data bus, whenever the Microprocessor performs READ and WRITE operations with the Microprocessor Interface of the XRT94L33.

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AF22	PWR_L/ R/W*	I	TTL	**Write Strobe/Read-Write Operation Identifier:** The exact function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in, as described below. **Intel-Asynchronous Mode – WR* - Write Strobe Input:** If the Microprocessor Interface is configured to operate in the Intel-Asynchronous Mode, then this input pin functions as the WR* (Active-Low WRITE Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the input buffers (associated with the Bi-Directional Data Bus pins, D[7:0]) will be enabled. The Microprocessor Interface will latch the contents on the Bi-Directional Data Bus (into the "target" register or address location, within the XRT94L33) upon the rising of this input. **Motorola-Asynchronous Mode - R/W* - Read/Write Operation Identification Input Pin:** If the Microprocessor Interface is operating in the "Motorola-Asynchronous" Mode, then this pin is functionally equivalent to the "R/W*" input pin. In the Motorola Mode, a "READ" operation occurs if this pin is held at a logic "1", coincident to a falling edge of the RD/DS* (Data Strobe) input pin. **PowerPC 403 Mode – R/W* - Read/Write Operation Identification Input:** If the Microprocessor Interface is configured to operate in the PowerPC 403 Mode, then this input pin will function as the "Read/Write Operation Identification" input pin. Anytime the Microprocessor Interface samples this input signal at a logic "low" (while also sampling the CS* input pin "low") upon the rising edge of PCLK, then the Microprocessor Interface will (upon the very same rising edge of PCLK) latch the contents of the Address Bus (A]14:0]) into the Microprocessor Interface circuitry, in preparation for this forthcoming READ operation. At some point (later in this READ operation) the Microprocessor will also assert the DBEN*/OE* input pin, and the Microprocessor Interface will then place the contents of the "target" register (or address location within the XRT94L33) upon the Bi-Directional Dat Bus pins (D[7:0]), where it can be read by the Microprocessor. Anytime the Microprocessor Interface samples this input signal at a logic high (while also sampling the CS* input pin at a logic "low") upon the rising edge of PCLK, then the Microprocessor Interface will (upon the very same rising edge of PCLK) latch the contents of the Address Bus (A[14:0]) into the Microprocessor Interface circuitry, in preparation for the forthcoming WRITE operation. At some point (later in this WRITE operation) the Microprocessor will also assert the RD*/DS*/WE* input pin, and the Microprocessor Interface will then latch the contents of the Bi-Directional Data Bus (D[7:0]) into the contents of the "target" register or buffer location (within the XRT94L33).

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AC18	PRD_L/ DS*/ WE*	I	TTL	**READ Strobe /Data Strobe:** The exact function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in, as described below. **Intel-Asynchronous Mode – RD* - READ Strobe Input:** If the Microprocessor Interface is operating in the Intel-Asynchronous Mode, then this input pin will function as the RD* (Active Low READ Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the XRT94L33 will place the contents of the addressed register (or buffer location) on the Microprocessor Bi-directional Data Bus (D[7:0]). When this signal is negated, the Data Bus will be tri-stated. **Motorola-Asynchronous (68K) Mode – DS* - Data Strobe Input:** If the Microprocessor Interface is operating in the Motorola Asynchronous Mode, then this input will function as the DS* (Data Strobe) input signal. **PowerPC 403 Mode – WE* - Write Enable Input:** If the Microprocessor Interface is operating in the PowerPC 403 Mode, then this input pin will function as the WE* (Write Enable) input pin. Anytime the Microprocessor Interface samples this active-low input signal (along with CS* and WR*/R/W*) also being asserted (at a logic level) upon the rising edge of PCLK, then the Microprocessor Interface will (upon the very same rising edge of PCLK) latch the contents on the Bi-Directional Data Bus (D[7:0]) into the “target” on-chip register or buffer location within the XRT94L33.

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AG23	ALE/ AS_L	I	TTL	**Address Latch Enable/Address Strobe:** The exact function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in, as described below. **Intel-Asynchronous Mode - ALE** If the Microprocessor Interface (of the XRT94L33) has been configured to operate in the Intel-Asynchronous Mode, then this active-high input pin is used to latch the address (present at the Microprocessor Interface Address Bus input pins (A[14:0]) into the XRT94L33 Microprocessor Interface block and to indicate the start of a READ or WRITE cycle. Pulling this input pin "high" enables the input bus drivers for the Address Bus input pins (A[14:0]). The contents of the Address Bus will be latched into the XRT94L33 Microprocessor Interface circuitry, upon the falling edge of this input signal. **Motorola-Asynchronous (68K) Mode – AS*** If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this active-low input pin is used to latch the data (residing on the Address Bus, A[14:0]) into the Microprocessor Interface circuitry of the XRT94L33. Pulling this input pin "low" enables the input bus drivers for the Address Bus input pins. The contents of the Address Bus will be latched into the Microprocessor Interface circuitry, upon the rising edge of this signal. **PowerPC 403 Mode – No Function – Tie to GND**: If the MIcroprocessor Interface has been configured to operate in the PowerPC 403 Mode, then this input pin has no role nor function and should be tied to GND.
AE19	PCS_L	I	TTL	**Chip Select Input:** The user must assert this active low signal in order to select the Microprocessor Interface for READ and WRITE operations between the Microprocessor and the XRT94L33 on-chip registers, LAPD and Trace Buffer locations.

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION

AD18	PRDY_L/
	DTACK*
	RDY
		O	CMOS	**READY or DTACK Output:**
				
				The exact function of this input pin depends upon wich mode the Microprocessor Interface has been configured to operate in, as described below.
				
				Intel Asynchronous Mode – RDY* - READY output:
				
				If the Microprocessor Interface has been configured to operate in the Intel-Asyncrhronous Mode, then this output pin will function as the “active-low” READY output.
				
				During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic “low” level ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic “low” level, then it is now safe for it to move on and execute the next READ or WRITE cycle.
				
				If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic “high” level, then the MIcroprocessor is expected to extend this READ or WRITE cycle, until it detect this output pin being toggled to the logic low level.
				
				Motorola Mode – DTACK* - Data Transfer Acknowledge Output:
				
				If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this output pin will function as the “active-low” DTACK* ouytput.
				
				During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic low level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic “low” leve, then it is now safe for it to move on and execute the next READ or WRITE cycle.
				
				If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic “high” level, then the MIcroprocessor is expected to extend this READ or WRITE cycle, until it detects this output pin being toggled to the logic low level.
				
				PowerPC 403 Mode – RDY – Ready Output:
				
				If the Microprocessor Interface has been configured to operate in the PowerPC 403 Mode, then this output pin will function as the “active-high” READY output.
				
				During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic high level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has sampled this signal being at a logic “high” level (upon the rising edge of PCLK) then it is now safe for it to move on and execute the next READ or WRITE cycle.
				
				The Microprocessor Interface will update the state of this output pin upon the rising edge of PCLK.

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AF21	PDBEN_L	I	TTL	**Bi-directional Data Bus Enable Input pin:** This input pin permits the user to either enable or tri-state the Bi-Directional Data Bus pins (D[7:0]), as described below. Setting this input pin "low" enables the Bi-directional Data bus. Setting this input "high" tri-states the Bi-directional Data Bus.
AF20	PBLAST_L	I	TTL	**Last Burst Transfer Indicator input pin:** If the Microprocessor Interface is operating in the Intel-I960 Mode, then this input pin is used to indicate (to the Microprocessor Interface block) that the current data transfer is the last data transfer within the current burst operation. The Microprocessor should assert this input pin (by toggling it "Low") in order to denote that the current READ or WRITE operation (within a BURST operation) is the last operation of this BURST operation. ***Note:*** *The user should connect this input pin to GND whenever the Microprocessor Interface has been configured to operate in the Intel-Async, Motorola 68K and IBM PowerPC 403 modes.*
AG22	PINT_L	O	CMOS	**Interrupt Request Output:** This open-drain, active-low output signal will be asserted when the Mapper/Framer device is requesting interrupt service from the Microprocessor. This output pin should typically be connected to the "Interrupt Request" input of the Microprocessor.
AB24	RESET_L	I	TTL	**Reset Input:** When this "active-low" signal is asserted, the XRT94L33 will be asynchronously reset. When this occurs, all outputs will be "tri-stated" and all on-chip registers will be reset to their "default" values.
AE18	DIRECT_ADD_SEL	I	TTL	**Address Location Select input pin:** This input pin must be pulled "HIGH" in order to permit normal operation of the Microprocessor Interface.
SONET/SDH SERIAL LINE INTERFACE PINS				
T3	RXLDAT_P	I	LVPECL	**Receive STS-3/STM-1 Data – Positive Polarity PECL Input:** This input pin, along with RXLDAT_N functions as the Recovered Data Input, from the Optical Transceiver or as the Receive Data Input from the system back-plane ***Note:*** *For APS (Automatic Protection Switching) purposes, this input pin, along with "RXLDAT_N" functions as the "Primary" STS-3/STM-1 Receive Data Input Port.*

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
T2	RXLDAT_N	I	LVPECL	**Receive STS-3/STM-1 Data – Negative Polarity PECL Input:** This input pin, along with RXLDAT_P functions as the Recovered Data Input, from the Optical Transceiver or as the Receive Data Input from the system back-plane. ***Note:*** *For APS (Automatic Protection Switching) purposes, this input pin, along with "RXLDAT_P" functions as the "Primary Receive STS-3/STM-1 Data Input Port"*
U2	RXLDAT_R_P	I	LVPECL	**Receive STS-3/STM-1 Data – Positive Polarity PECL Input – Redundant Port:** This input pin, along with "RXLDAT_R_N" functions as the Recovered Data Input, from the Optical Transceiver or as the Receive Data Input from the system back-plane. ***Note:*** *For APS (Automatic Protection Switching) purposes, this input pin, along with "RXLDAT_R_N" functions as the "Redundant Receive STS-3/STM-1 Data Input Port".*
U1	RXLDAT_R_N	I	LVPECL	**Receive STS-3/STM-1 Data – Negative Polarity PECL Input – Redundant Port:** This input pin, along with "RXLDAT_R_P" functions as the Recovered Data Input, from the Optical Transceiver or as the Receive Data Input from the system back-plane. ***Note:*** *For APS (Automatic Protection Switching) purposes, this input pin, along with "RXLDAT_R_N" functions as the "Redundant Receive STS-3/STM-1 Data Input Port".*
AE27	RXCLK_19MHZ	O	CMOS	**19.44MHz Recovered Output Clock:** This pin outputs a 19.44MHz clock signal that has been derived from the incoming STS-3/STM-1 line signal (via the Receive STS-3/STM-1 Clock and Data Recovery PLL). If the user wishes to operate the STS-3/STM-1 Interface in the "loop-timing" mode, then the user should route this particular signal through a "narrow-band" PLL (in order to attenuate any jitter within this signal) prior to routing it to the REFTTL input pin.
P3	REFCLK_P	I	LVPECL	**Transmit Reference Clock – Positive Polarity PECL Input:** This input pin, along with "REFCLK_N" and "REFTTL" can be configured to function as the timing source for the STS-3/STM-1 Transmit Interface Block. If the user configures these two input pins to function as the timing source, then the user must apply a 155.52MHz clock signal, in the form of a PECL signal to these input pins. The user can configure these two inputs to function as the timing source by writing the appropriate data into the "Transmit Line Interface Control Register " (Address Location = 0x0383) ***Note:*** *Users should set this pin to "1" if "REFTTL" clock input is used*

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
P2	REFCLK_N	I	LVPECL	**Transmit Reference Clock – Negative Polarity PECL Input:** This input pin, along with "REFCLK_P" and "REFTTL" can be configured to function as the timing source for the STS-3/STM-1 Transmit Interface Block. If the user configures these two input pins to function as the timing source, then the user must apply a 155.52MHz clock signal, in the form of a PECL signal to these input pins. The user can configure these two inputs to function as the timing source by writing the appropriate data into the "Transmit Line Interface Control Register " (Address Location = 0x0383) ***Note:*** *Users should set this pin to "0" if "REFTTL" clock input is used*
P5	TXLDATO_P	O	LVPECL	**Transmit STS-3/STM-1 Data - Positive Polarity PECL Output:** This output pin, along with TXLDATO_N functions as the Transmit Data Output, to the Optical Transceiver or to the system back-plane. For "High-Speed" Back-Plane Applications, the user should note that data is output from these output pins upon the rising/falling edge of "TXLCLKO_P/TXLCLKO_N". ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLDATO_N" functions as the "Primary" Transmit STS-3/STM-1 Data Output Port.*
P6	TXLDATO_N	O	LVPECL	**Transmit STS-3/STM-1 Data – Negative Polarity PECL Output:** This output pin, along with TXLDATO_P functions as the Transmit Data Output, to the Optical Transceiver or to the system back-plane. For "High-Speed" Back-Plane Applications, the user should note that data is output from these output pins upon the rising/falling edge of TXLCLKO_P/TXLCLKO_N. ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLDATO_P" functions as the "Primary" Transmit STS-3/STM-1 Data Output Port.*
M4	TXLDATO_R_P	O	LVPECL	**Transmit STS-3/STM-1 Data - Positive Polarity PECL Output - Redundant Port:** This output pin, along with TXLDATO_R_N functions as the Transmit Data Output, to the Optical Transceiver or to the system back-plane. For "High-Speed" Back-Plane Applications, the user should note that data is output from these output pins upon the rising/falling edge of "TXLCLKO_R_P/TXLCLKO_R_N"). ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLDATO_N" functions as the "Redundant" Transmit STS-3/STM-4 Data Output Port.*

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
M3	TXLDATO_R_N	O	LVPECL	**Transmit STS-3/STM-1 Data - Negative Polarity PECL Output - Redundant Port:** This output pin, along with TXLDATO_R_P functions as the Transmit Data Output, to the Optical Transceiver (for transmission to remote terminal equipment) or to the system back-plane (for transmission to some other System board) For "High-Speed" Back-Plane Applications, the user should note that data is output from these output pins upon the rising/falling edge of "TXLCLKO_R_P/TXLCLKO_R_N"). ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLDATO_R_P" functions as the "Redundant" Transmit STS-3/STM-1 Data Output Port.*
N6	TXLCLKO_P	O	LVPECL	**Transmit STS-3/STM-1 Clock – Positive Polarity PECL Output:** This output pin, along with TXLCLKO_N functions as the Transmit Clock Output signal. These output pins are typically used in "High-Speed" Back-Plane Applications. In this case, outbound STS-3/STM-1 data is output via the "TXLDATO_P/TXLDATO_N" output pins upon the rising edge of this clock signal. ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLCLKO_N" functions as the "Primary Transmit Output Clock" signal.*
N5	TXLCLKO_N	O	LVPECL	**Transmit STS-3/STM-1 Clock – Negative Polarity PECL Output:** This output pin, along with TXLCLKO_P functions as the Transmit Clock Output signal. These output pins are typically used in "High-Speed" Back-Plane Applications. In this case, outbound STS-3/STM-1 data is output via the "TXLDATO_P/TXLDATO_N" output pins upon the falling edge of this clock signal. ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLCLKO_N" functions as the "Primary Transmit Output Clock" signal.*
M1	TXLCLKO_R_P	O	LVPECL	**Transmit STS-3/STM-1 Clock – Positive Polarity PECL Output – Redundant Port:** This output pin, along with TXLCLKO_R_N functions as the Transmit Clock Output signal. These output pins are typically used in "High-Speed" Back-Plane Applications. In this case, outbound STS-3/STM-1 data is output via the "TXLDATO_R_P/TXLDATO_R_N" output pins upon the rising edge of this clock signal. ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLCLKO_R_N" functions as the "Redundant Transmit Output Clock" signal.*

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
M2	TXLCLKO_R_N	O	LVPECL	**Transmit STS-3/STM-1 Clock – Negative Polarity PECL Output – Redundant Port:** This output pin, along with TXLCLKO_R_P functions as the Transmit Clock Output signal. These output pins are typically used in "High-Speed" Back-Plane Applications. In this case, outbound STS-3/STM-1 data is output via the "TXLDATO_R_P/TXLDATO_R_N" output pins upon the rising edge of this clock signal. ***Note:*** *For APS (Automatic Protection Switching) purposes, this output pin, along with "TXLCLKO_R_P" functions as the "Redundant Transmit Output Clock" signal.*
P1	REFTTL	I	TTL	**19.44MHz or 77.76MHz Clock Synthesizer Reference Clock Input Pin:** The exact function of this input pin depends upon whether the user enables the "Clock Synthesizer" block or not. **If Clock Synthesizer is Enabled.** If the "Clock Synthesizer" block is enabled, then it will be used to generate the 155.52MHz, 19.44MHz and/or 77.76MHz clock signal for the STS-3/STM-1 block. In this mode, the user should apply a clock signal of either of the following frequencies to this input pin. • 19.44 MHz • 38.88 MHz • 51.84 MHz • 77.76 MHz Afterwards, the user needs to write the appropriate data into the "Transmit Line Interface Control Register" (Address Location = 0x0383) in order to (1) configure the Clock Synthesizer Block to accept any of the above-mentioned signals and generate a 155.52MHz, 19.44MHz or 77.76MHz clock signal, (2) to configure the Clock Synthesizer to function as the Clock Source for the STS-3/STM-1 block. **If Clock Synthesizer is NOT Enabled:** If the "Clock Synthesizer" block is NOT enabled, then it will NOT be used to generate the 19.44MHz and/or 77.76MHz clock signal, for the STS-3/STM-1 block. In this configuration seting, the user MUST apply a 19.44MHz clock signal to this input pin.

PIN #	SIGNAL NAME	I/O	SIGNAL TYPE	DESCRIPTION
AG3	LOSTTL	I	TTL	**Loss of Optical Carrier Input – Primary:** The user is expected to connect the "Loss of Carrier" output (from the Optical Transceiver) to this input pin. If this input pin and the LOSPECL_P pin are pulled "high", or if both of these input pins are pulled "low", tthen the Receive STS-3 TOH Processor block will declare a "Loss of Optical Carrier" condition. ***Note:*** *This input pin is only active if the "Primary Port" is active. This input pin is inactive if the "Redundant Port" is active.*
AG25	LOSTTL_R	I	TTL	**Loss of Optical Carrier Input – Redundant:** The user is expected to connect the "Loss of Carrier" output (from the Optical Transceiver) to this input pin. If this input pin and the LOSPECL_R are pulled "high", or if both of these input pins are pulled "low", then the Receive STS-3 TOH Processor block will declare a "Loss of Optical Carrier" condition. ***Note:*** *This input pin is only active if the "Redundant Port" is active. This input pin is inactive if the "Primary Port" is active.*
L4	LOSPECL_P	I	LVPECL	**Loss of PECL Interface Input – Primary:** If this input pin is pulled "high", then the Receive STS-3 TOH Processor block will declare a "Loss of PECL Interface" condition. ***Note:*** *This input pin is only active if the "Primary Port" is active. This input pin is inactive if the "Redundant Port" is active.*
L3	LOSPECL_R	I	LVPECL	**Loss of PECL Interface Input – Redundant:** If this input pin is pulled "high", then the Receive STS-3 TOH Processor block will declare a "Loss of PECL Interface" condition. ***Note:*** *This input pin is only active if the "Redundant Port" is active. This input pin is inactive if the "Primary Port" is active.*
V1	LOCKDET	O	CMOS	**Lock Detect Output Pin – Clock and Data Recovery PLL Block** This output pin indicates whether the Clock and data recovery PLL block has obtained lock to incoming STS-3/STM-1 signal or not. This pin pulses high if internal VCO frequency is within 0.05% of external reference clock This pin pulses low if internal VCO frequency is beyond 0.05% of external reference clock, and Loss of lock is declared.

STS-3/STM-1 TELECOM BUS INTERFACE – TRANSMIT DIRECTION				
E1	TXA_CLK/ TxAPSCLK	O I/O	CMOS	**Transmit STS-3/STM-1 Telecom Bus Interface - Clock Output Signal:** This output clock signal functions as the clock source for the Transmit STS-3/STM-1 Telecom Bus. All signals, that are output via the Transmit STS-3/STM-1 Telecom Bus Interface (e.g., TXA_C1J1, TXA_ALARM, TXA_DP, TXA_PL and TXA_D[7:0]) are updated upon the rising edge of this clock signal. This clock signal operates at 19.44MHz and is derived from the Clock Synthesizer block. **Transmit Payload APS Bus Interface Clock Input/Output signal – TxAPSCLK:** This pin can only be configured to operate in this mode if the XRT94L33 has been configured to operate in either the "ATM UNI" over "PPP over STS-3c" Mode.
F2	TXA_C1J1	O	CMOS	**Transmit STS-3/STM-1 Telecom Bus Interface - C1/J1 Byte Phase Indicator Output Signal:** This output pin pulses "high" under the following two conditions. • Whenever the C1 byte is being output via the "TxA_D[7:0]" output, and • Whenever the J1 byte is being output via the "TxA_D[7:0]" output. ***Notes:*** 1. *The Transmit STS-3/STM-1 Telecom Bus Interface will indicate that it is currently transmitting the C1 byte (via the TXA_D[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "TXA_CLKTXA_CK") and keeping the "TXA_PL" output pin pulled "LOW".* 2. *The Transmit STS-3/STM-1 Telecom Bus will indicate that it is currently transmitting the J1 byte (via the TXA_D[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "TXA_CLKTXA_CK") while the "TXA_PL" output pin is pulled "HIGH".* 3. *This output pin is only active if the Transmit STS-3/STM-1 Telecom Bus Interface block is enabled and is configured to operate in the "Re-Phase OFF" Mode.*

E2	TXA_ALARM/ TxAPSPAR	O I/O	CMOS TTL/ CMOS	**Transmit STS-3/STM-1 Telecom Bus Interface – Alarm Indicator Output signal:** This output pin pulses "high", coincident to the instant that the Transmit STS-3/STM-1 Telecom Bus Interface outputs an byte of any STS-1 or STS-3c signal (via the "TXD_D[7:0]" output pins) that is carrying an AIS-P indicator. This output pin is "low" for all other conditions. **NOTE:** This output pin is only active if the Transmit STS-3/STM-1 Telecom Bus Interface is enabled and has been configured to operate in the "Re-Phase OFF" Mode. **Transmit Payload APS Bus Interface – Parity Input/Output pin:** This pin can only be configured to operate in this role/function if the XRT94L33 has been configured to operate in the "ATM UNI" or the "PPP over STS-3c" Mode. Please see the "XRT94L33_Pin_Description_ATM_PPP.pdf" document for more information.
H3	TXA_DP	O	CMOS	**Transmit STS-3/STM-1 Telecom Bus Interface – Parity Output pin:** This output pin can be configured to function as either one of the following. The EVEN or ODD parity value of the bits which are output via the "TXA_D[7:0]" output pins. The EVEN or ODD parity value of the bits which are being output via the "TXA_D[7:0]" output pins and the states of the "TXA_PL" and "TXA_C1J1" output pins. *NOTES:* *a. The user can make any one of these configuration selections by writing the appropriate value into the "Telecom Bus Control" Register (Address Location = 0x0137).* b. *This output pin is only active if the XRT94L33 has been configured to output its STS-3/STM-1 or STS-3c data via the Transmit STS-3/STM-1 Telecom Bus Interface block.*

G4	TxSBFP	I	TTL	**Transmit STS-3/STM-1 Frame Alignment Sync Input:** The Transmit STS-3 TOH Processor Block can be configured to initiate its generation of a new "outbound" STS-3/STM-1 frame based upon an externally supplied 8kHz clock signal to this input pin. If the user opts to use this feature, then the Transmit STS-3/STM-1 Telecom Bus Interface will begin transmitting the very first byte of given STS-3 or STM-1 frame, upon sensing a rising edge (of the 8kHz signal) at this input pin. ***Notes:*** 1. *If the user connects this input pin to GND, then the Transmit STS-3 TOH Processor block will generate its "outbound" STS-3/STM-1 frames asynchronously, with respect to any input signal.* 2. *This input signal must be synchronized with the signal that is supplied to the REFTTL input pin. Failure to insure this will result in bit errors being generated within the outbound STS-3/STM-1 signal.* 3. *The user must supply an 8kHz pulse (to this input pin) that has a width of approximately 51.4412.8ns (one 19.44MHz clock period). The user must not apply a 50% duty cycle 8kHz signal to this input pin.* 4. *Register "HRSYNC_DLY" (Address Location: 0x0135) defines the timing for TxSBFP input pin.*
K5	TxA_PL/ TxAPSReq	O I/O	CMOS TTL/ CMOS	**Transmit STS-3/STM-1 Telecom Bus Interface – Payload Data Indicator Output Signal:** This output pin indicates whether the Transmit STS-3/STM-1 Telecom Bus Interface is currently placing a Transport Overhead byte or a "non-Transport Overhead Byte (e.g., STS-1 SPE, STS-3c SPE, VC-3 or VC-4 data) via the "TXA_D[7:0]" output pins. This output pin is pulled "low" for the duration that the Transmit STS-3/STM-1 Telecom Bus Interface is transmitting a Transport Overhead byte via the "TXA_D[7:0]" output pins. Conversely, this output pin is pulled "high" for the duration that the STS-3/STM-1 Transmit Telecom Bus is transmitting something other than a Transport Overhead byte via the "TXA_D[7:0]" output pins. **Transmit Payload APS Bus Interface – Request Input/Output pin:** This pin can only be configured to operate in this role if the XRT94L33 has been configured to operate in either the "ATM UNI" or "PPP over STS-3c" Mode.

J4 G3 D1 F3 J5 H4 D2 E3	TxA_D0/ TxAPSDat0 TxA_D1/ TxAPSDat1 TxA_D2/ TxAPSDat2 TxA_D3/ TxAPSDat3 TxA_D4/ TxAPSDat4 TxA_D5/ TxAPSDat5 TxA_D6/ TxAPSDat6 TxA_D7/ TxAPSDat7	O O I/O	CMOS CMOS CMOS/ TTL	**Transmit STS-3/STM-1 Telecom Bus Interface – Transmit Output Data Bus pins:** These 8 output pins function as the "Transmit STS-3/STM-1 Telecom Bus Interface" – Data bus output pins. If the STS-3/STM-1 Telecom Bus Interface is enabled, then all "outbound" STS-3/STM-1 data is output via these pins (in a byte-wide manner), upon the rising edge of the "TXA_CLK" output clock signal. **Transmit Payload APS Bus Interface – Data Input/Output pins:** These pins can only be configured to operate in this function/role if the XRT94L33 has been configured to operate in the "ATM UNI" or "PPP over STS-3c" Mode.

STS-3/STM-1 TELECOM BUS INTERFACE – RECEIVE DIRECTION				
W2	RxD_CLK/ RxAPSClk	I I I/O	TTL TTL TTL/ CMOS	**Receive STS-3/STM-1 Telecom Bus Interface - Clock Input Signal:** This input clock signal functions as the clock source for the Receive STS-3/STM-1 Telecom Bus Interface block. All input signals are sampled upon the falling edge of this input clock signal. This clock signal should operate at 19.44MHz. ***Note:*** *This input pin is only used if the "STS-3/STM-1 Telecom Bus" has been enabled. It should be connected to GND otherwise.* **Receive Payload APS Bus Interface - Clock input/output signal:** This input can only be configured to operate in this role/function if the XRT94L33 has been configured to operate in either the "ATM UNI" or "PPP over STS-3c" Mode.

AA3	RxD_PL	I	TTL	**Receive STS-3/STM-1 Telecom Bus Interface – Payload Data Indicator Output Signal:** This input pin indicates whether or not the Receive STS-3/STM-1 Telecom Bus Interface is currently receiving Transport Overhead bytes or "non-Transport Overhead bytes (e.g., STS-1 SPE, STS-3c SPE,VC-3 or VC-4 data) via the "RXD_D[7:0]" input pins. This input pin should be pulled "low" for the duration that "STS-3/STM-1 Receive STS-3/STM-1 Telecom Bus Interface is receiving a Transport Overhead byte via the "RXD_D[7:0]" input pins. Conversely, this input pin should be pulled "high" for the duration that the Receive STS-3/STM-1 Telecom Interface Bus is receiving something other than a Transport Overhead byte via the "RXD_D[7:0]" input pins. ***Note:*** *The user should connect this pin to GND if the STS-3/STM-1 Telecom Bus is NOT enabled.*
AD1	RxD_C1J1/ RxAPSVal	I I/O	TTL TTL/ CMOS	**Receive STS-3/STM-1 Telecom Bus Interface - C1/J1 Byte Phase Indicator Input Signal/Receive APS Valid Indicator Input/Output signal:** The exact function of this input pin depends upon (1) whether the STS-3/STM-1 Telecom Bus Interface has been enabled or not, and (2) whether the Payload APS Bus has been enabled or not. **If the STS-3/STM-1 Telecom Bus Interface has been enabled – RxD_C1J1:** This input pin should be pulsed "high" during both of the following conditions. a. Coincident to whenever the C1/J0 byte is being applied to the Receive STS-3/STM-1 Telecom Bus – Data Input pins (RXD_D[7:0]). b. b. Coincident to whenever the J1 byte is being applied to the Receive STS-3/STM-1 Telecom Bus – Data Input pins (RXD_D[7:0]) input. *NOTE: This input pin should be pulled "low" during all other times.* **Receive Payload APS Bus Interface – Data Valid Input/Output Signal:** This pin can only be configured to operate in this role if the XRT94L33 has been configured to operate in either the "ATM UNI" or "PPP over STS-3c" Mode.

AB3	RxD_DP	I	TTL	**Receive STS-3/STM-1 Telecom Bus Interface – Parity Input pin:** This input pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits which are input via the "RXD_D[7:0]" input pins. The EVEN or ODD parity value of the bits which are being input via the "RXD_D[7:0]" input and the states of the "RXD_PL" and "RXD_C1J1" input pins. ***Notes:*** 1. *The user can make any one of these configuration selections by writing the appropriate value into the "Telecom Bus Control" register (Address Location = 0x0137).* 2. *The user should connect this pin to GND if the STS-3/STM-1 Telecom Bus Interface is disabled.*
W1	RxD_ALARM/ RxAPSPAR	I I/O	TTL TTL/ CMOS	**Receive STS-3/STM-1 Telecom Bus Interface – Alarm Indicator Input:** This input pin pulses "high" coincident to whether the Receive STS-3/STM-1 Telecom Bus Interface block is receiving a byte (via the "RxD_D[7:0] input pins) that is a part of any STS-1 or STS-3c signal that is carrying the AIS-P indicator. ***Note:*** *If the RxD_ALARM input signal pulses "HIGH" for any given STS-1 signal (within the "incoming" STS-3), then the corresponding Receive SONET POH Processor block will automatically declare the AIS-P defect condition.* **RxAPSParity – Receive Payload APS Bus Interface – Parity Input/Output Pin:** This pin can only be configured to operate in this role if the XRT94L33 has been configured to operate in either the "ATM UNI" or "PPP over STS-3c" Mode.

Y2	RxD_D0/ RxHRDat0/ RxAPSDat0	I I I/O	TTL TTL TTL/	**Receive STS-3/STM-1 Telecom Bus Interface – Receive Input Data Bus pins - RxD_D[7:0]:** These 8 input pins function as the Receive STS-3/STM-1 Telecom Bus Interface - Input data bus. All incoming STS-3/STM-1 data is sampled and latched (into the XRT94L33, via these input pins) upon the falling edge of the "RXD_CLK" input clock signal. **RxHRDat[7:0]: Receive data inputs for high-rate device** **Receive Payload APS Bus Interface – Data Bus Input/Output Pins:** These pins can only be configured to function in this role if the XRT94L33 has been configured to operate in the "ATM UNI" or "PPP over STS-3c" Mode. These pins cannot be configured to support "Payload APS" operation if the XRT94L33 has been configured to operate in an "Aggregation" role.
AD2	RxD_D1 RxHRDat1/ RxAPSDat1		CMOS	
AC3	RxD_D2 RxHRDat2/ RxAPSDat2			
AA4	RxD_D3 RxHRDat3/ RxAPSDat3			
AB4	RxD_D4 RxHRDat4/ RxAPSDat4			
Y1	RxD_D5 RxHRDat5/ RxAPSDat5			
AD3	RxD_D6 RxHRDat6/ RxAPSDat6			
AA5	RxD_D7 RxHRDat7/ RxAPSDat7			

SONET/SDH OVERHEAD INTERFACE – TRANSMIT DIRECTION				
H6	TxTOHClk	O	CMOS	**Transmit TOH Input Port – Clock Output:** This output pin, along with the "TxTOHEnable", "TxTOHFrame" output pins and the "TxTOH" and "TxTOHIns" input pins function as the "Transmit TOH Input Port". The Transmit TOH Input Port permits the user to externally insert his/her own value(s) for the TOH bytes (within the outbound STS-3/STM-1 signal). This output pin provides the user with a clock signal. If the "TxTOHEnable" output pin is "HIGH" and if the "TxTOHIns" input pin is pulled "HIGH", then the user is expected to provide a given bit (within the "TOH") to the "TxTOH" input pin, upon the falling edge of this clock signal. The data, residing on the "TxTOH" input pin will be latched into the XRT94L33 upon the rising edge of this clock signal. ***Note:*** *The Transmit TOH Input Port only supports the insertion of the TOH within the very first STS-1 of the outbound STS-3 signal.*
G5	TxTOHEnable	O	CMOS	**Transmit TOH Input Port – TOH Enable (or READY) indicator:** This output pin, along with the "TxTOHClk", "TxTOHFrame" output pins and the "TxTOH" and "TxTOHIns" input pins function as the "Transmit TOH Input Port". This output pin will toggle and remain "HIGH" anytime the "Transmit TOH Input Port" is ready to externally accept TOH data via the "TxOH" input pin. To externally insert user values of TOH into the "outbound" STS-3 data stream via the "Transmit TOH Input Port", do the following. • Continuously sample the state of "TxTOHFrame" and this output pin upon the rising edge of "TxTOHClk". • Whenever this output pin pulses "HIGH", then the user's external circuitry should drive the "TxTOHIns" input pin "HIGH". • Next, the user should output the next TOH bit, onto the "TxTOH" input pin, upon the rising edge of "TxTOHClk"

F8	TxTOH	I	TTL	**Transmit TOH Input Port – Input pin:** This input pin, along with the "TxTOHIns" input pin, the "TxTOHEnable" and "TxTOHFrame" and "TxTOHClk" output pins function as the "Transmit TOH Input Port". To externally insert user values of TOH into the outbound STS-3 data stream via the "Transmit TOH Input Port", do the following. • Continuously sample the state of "TxTOHFrame" and "TxTOHEnable" upon the rising edge of "TxTOHClk. • Whenever "TxTOHEnable" pulses "HIGH", then the user's external circuitry should drive the "TxTOHIns" input pin "HIGH". • Next, the user should output the next TOH bit, onto this input pin, upon the rising edge of "TxTOHClk". The "Transmit TOH Input Port" will sample the data (on this input pin) upon the falling edge of "TxTOHClk". ***Note:*** *Data at this input pin will be ignored (e.g., not sampled) unless the "TxTOHEnable" output pin is "HIGH" and the "TxTOHIns" input pin is pulled "HIGH".*
E8	TxTOHFrame	O	CMOS	**Transmit TOH Input Port – STS-3/STM-1 Frame Indicator:** This output pin, along with "TxTOHClk", "TxTOHEnable output pins, and the "TxTOH" and "TxTOHIns" input pins function as the "Transmit TOH Input Port". This output pin will pulse high (for one period of TxTOHClk), one "TxTOHClk" clock period prior to the first "TOH bit" of a given STS-3 frame, being expected via the "TxTOH" input pin. To externally insert user values of TOH into the "outbound" STS-3 data stream via the "Transmit TOH Input Port", do the following. • Continuously sample the state of "TxTOHEnable" and this output pin upon the rising edge of "TxTOHClk". • Whenever the "TxTOHEnable" output pin pulse "HIGH", then the user's external circuitry should drive the "TxTOHIns" input pin "HIGH". • Next, the user should output the next TOH bit, onto the "TxTOH" input pin, upon the rising edge of "TxTOHClk". ***Note:*** *The external circuitry (which is being interfaced to the "Transmit TOH Input Port" can use this particular output pin to denote the boundary of STS-3 frames.*

D6	TxTOHIns	I	TTL	**Transmit TOH Input Port – Insert Enable Input pin:** This input pin, along with the "TxTOH" input pin, and the "TxTOHEnable", "TxTOHFrame" and "TxTOHClk" output pins function as the "Transmit TOH Input Port". This input pin permits the user to either enable or disable the "Transmit TOH Input Port". If this input pin is "LOW", then the "Transmit TOH Input Port" will be disabled and will not sample and insert (into the "outbound" STS-3 data stream) any data residing on the "TxTOH" input, upon the rising edge of "TxTOHClk" If this input pin is "HIGH", then the "Transmit TOH Input Port" will be enabled. In this mode, whenever the "TxTOHEnable" output pin is also "HIGH", the "Transmit TOH Input Port" will sample and latch any data that is presented on the "TxTOH" input pin, upon the rising edge of "TxTOHClk". To externally insert user values of TOH into the "outbound" STS-3 data stream via the "Transmit TOH Input Port", do the following. • Continuously sample the state of "TxTOHFrame" and "TxTOHEnable" upon the rising edge of "TxTOHClk". • Whenever the "TxTOHEnable" output pin is sampled "high" then the user's external circuitry should drive this input pin "HIGH". • Next, the user should output the next TOH bit, onto the "TxTOH" input pin, upon the falling edge of "TxTOHClk". The "Transmit TOH Input Port" will sample the data (on this input pin) upon the falling edge of "TxTOHClk".] ***Notes:*** *1. Data applied to the "TxTOH" input pin will be sampled according to the following insertion priority scheme:* *2. For DCC, E1, F1, E2 bytes, "TxTOH" input pin will be sampled if both "TxTOHEnable" and "TxTOHIns" are high.* *3. For other TOH bytes, "TxTOH" input pin will be sampled if both "TxTOHEnable" and "TxTOHIns" are high or if both "TxTOHIns" and "Software Insertion Enabled" are "low".*

B4	TxLDCCEnable	O	CMOS	**Transmit – Line DCC Input Port – Enable Output pin:** This output pin, along with the "TxTOHClk" output pin and the "TxLDCC" input pin permit the user to insert their value for the D4, D5, D6, D7, D8, D9, D10, D11 and D12 bytes into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and will insert into the D4, D5, D6, D7, D8, D9, D10, D11 and D12 byte-fields, within the "outbound" STS-3 data-stream. The Line DCC HDLC Controller circuitry (which is connected to the "TxTOHClk", the "TxSDCC" and this output pin, is suppose to do the following. It should continuously monitor the state of this output pin. Whenever this output pin pulses "HIGH", then the Line DCC HDLC Controller circuitry should place the next Line DCC bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto the "TxLDCC" input pin, upon the rising edge of "TxTOHClk". Any data that is placed on the "TxLDCC" input pin, will be sampled upon the falling edge of "TxOHClk".
D7	TxSDCCEnable	O	CMOS	**Transmit – Section DCC Input Port – Enable Output pin:** This output pin, along with the "TxTOHClk" output pin and the "TxSDCC" input pin permit the user to insert their value for the D1, D2 and D3 bytes, into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and will insert into the D1, D2 and D3 byte-fields, within the "outbound" STS-3 data-stream. The Section DCC HDLC Controller circuitry (which is connected to the "TxTOHClk", the "TxSDCC" and this output pin, is suppose to do the following. It should continuously monitor the state of this output pin. Whenever this output pin pulses "HIGH", then the Section DCC HDLC Controller circuitry should place the next Section DCC bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto the "TxSDCC" input pin, upon the rising edge of "TxTOHClk". Any data that is placed on the "TxSDCC" input pin, will be sampled upon the falling edge of "TxOHClk".

C5	TxSDCC	I	TTL	**Transmit - Section DCC Input Port – Input pin:** This input pin, along with the "TxSDCCEnable" and the "TxTOHClk" output pins permit the user to insert their value for the D1, D2 and D3 bytes, into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and insert it into the D1, D2 and D3 byte fields, within the "outbound" STS-3 data-stream. The Section DCC HDLC Circuitry that is interfaced to this input pin, the "TxSDCCEnable" and the "TxTOHClk" pins is suppose to do the following. It should continuously monitor the state of the "TxSDCCEnable" input pin. Whenever the "TxSDCCEnable" input pin pulses "HIGH", then the Section DCC HDLC Controller circuitry should place the next Section DCC bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto this input pin upon the rising edge of "TxTOHClk". Any data that is placed on the "TxSDCC" input pin, will be sampled upon the falling edge of "TxTOHClk". ***Note:*** *This pin should be connected to GND if it is not used.*
D8	TxLDCC	I	TTL	**Transmit - Line DCC Input Port:** This input pin, along with the "TxLDCCEnable" and the "TxTOHClk" pins permit the user to insert their value for the D4, D5, D6, D7, D8, D9, D10, D11 and D12 bytes, into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and insert it into the D4, D5, D6, D7, D8, D9, D10, D11 and D12 byte-fields, within the "outbound" STS-3 data-stream. Whatever Line DCC HDLC Controller Circuitry is interface to the this input pin, the "TxLDCCEnable" and the "TxTOHClk" is suppose to do the following. It should continuously monitor the state of the "TxLDCCEnable" input pin. Whenever the "TxLDCCEnable" input pin pulses "HIGH", then the Section DCC Interface circuitry should place the next Line DCC bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto the "TxLDCC" input pin, upon the rising edge of "TxTOHClk". Any data that is placed on the "TxLDCC" input pin, will be sampled upon the falling edge of "TxTOHClk". ***Note:*** *This pin should be connected to GND if it is not used.*

E9	TxE1F1E2Enable	O	CMOS	**Transmit E1-F1-E2 Byte Input Port – Enable (or Ready) Indicator Output pin:** This output pin, along with the "TxTOHClk" output pin and the "TxE1F1E2" input pin permit the user to insert their value for the E1, F1 and E2 bytes, into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and will insert into the E1, F1 and E2 byte-fields, within the "outbound" STS-3 data-stream. Whatever external circuitry (which is connected to the "TxTOHClk", the "TxE1F1E2" and this output pin), is suppose to do the following. It should continuously monitor the state of this output pin. Whenever this output pin pulses "HIGH", then the external circuitry should place the next "orderwire" bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto the "TxE1F1E2" input pin, upon the rising edge of "TxTOHClk". Any data that is placed on the "TxE1F1E2" input pin, will be sampled upon the falling edge of "TxOHClk".
C6	TxE1F1E2Frame	O	CMOS	**Transmit E1-F1-E2 Byte Input Port – Framing Output Pin.** This output pin pulses "HIGH" for one period of "TxTOHClk", one "TxTOHClk" bit-period prior to the "Transmit E1-F1-E2 Byte Input Port" expecting the very first byte of the E1 byte, within a given "outbound" STS-3 frame.
A4	TxE1F1E2	I	TTL	**Transmit E1-F1-E2 Byte Input Port – Input Pin:** This input pin, along with the "TxE1F1E2Enable" and the "TxTOHClk" output pins permit the user to insert their value for the E1, F1 and E2 bytes, into the Transmit STS-3 TOH Processor Block. The Transmit STS-3 TOH Processor block will accept this data and insert it into the E1, F1 and E2 byte fields, within the "outbound" STS-3 data-stream. Whatever external circuitry that is interfaced to this input pin, the "TxE1F1E2Enable" and the "TxTOHClk" pins is suppose to do the following. It should continuously monitor the state of the "TxE1F1E2Enable" input pin. Whenever the "TxE1F1E2Enable" input pin pulses "HIGH", then the external circuitry should place the next "orderwire" bit (to be inserted into the "Transmit STS-3 TOH Processor" block) onto this input pin upon the rising edge of "TxTOHClk". Any data that is placed on the "TxE1F1E2" input pin, will be sampled upon the falling edge of "TxTOHClk". ***Note:*** *This pin should be connected to GND if it is not used.*

C7	TXPOH	I	TTL	**Transmit Path Overhead Input Port – Input pin.** This pin is used for the Transmit AU-4/VC-4 Mapper POH Processor Block when TUG-3 mapping is used. This input pin permits the user to insert the POH data into the Transmit AU-4/VC-4 Mapper POH Processor blocks for insertion and transmission via the "outbound" STS-3 signal. In this mode, the external circuitry (which is being interfaced to the "Transmit Path Overhead Input Port" is suppose to monitor the following output pins; • TxPOHFrame_n • TxPOHEnable_n • TxPOHClk_n The "TxPOHFrame_n" output pin will toggle "high" upon the rising edge of "TxPOHClk_n" approximately one "TxPOHClk_n" period prior to the "TxPOH" port being ready to accept and process the first bit within J1 byte (e.g., the first POH byte). The "TxPOHFrame_n" output pin will remain "high" for eight consecutive "TxPOHClk_n" periods. The external circuitry should use this pin to note STS-1 SPE frame boundaries. The "TxPOHEnable_n" output pin will toggle "high" upon the rising edge of "TxPOHClk_n" approximately one "TxPOHClk_n" period prior to the "TxPOH" port being ready to accept and process the first bit within a given POH byte. To externally insert a given POH byte: (1) assert the "TxPOHIns_n" input pin by toggling it "high", and (2) place the value of the first bit (within this particular POH byte) on this input pin upon the very next rising edge of "TxPOHClk_n". This data bit will be sampled upon the very next falling edge of "TxPOHClk_n". The external circuitry should continue to keep the "TxPOHIns_n" input pin "high" and advancing the next bits (within the POH bytes) upon each rising edge of "TxPOHClk_n".
D9	TXPOHCLK	O	TTL	**Transmit Path Overhead Input Port – Clock Output pin:** This pin is used for the Transmit AU-4/VC-4 Mapper POH Processor Block when TUG-3 mapping is used. This output pin, along with "TxPOH", "TxPOHEnable", "TxPOHIns" and "TxPOHFrame" function as the "Transmit Path Overhead (TxPOH) Input Port". The "TxPOHFrame" and "TxPOHEnable" output pins are updated upon the falling edge this clock output signal. The "TxPOHIns" input pins and the data residing on the "TxPOH" input pins are sampled upon the next falling edge of this clock signal.

B5	TXPOHFRAME	O	TTL	**Transmit Path Overhead Input Port – Frame Output pin:** This pin is used for the Transmit AU-4/VC-4 Mapper POH Processor Block when TUG-3 mapping is used. This output pin, along with the "TxPOH", "TxPOHEnable", "TxPOHIns" and "TxPOHClk" function as the "Transmit Path Overhead Input Port". If the user is only inserting POH data via these input pins: ***Note:*** *In this mode, the "TxPOH" port will pulse these output pins "high" whenever it is ready to accept and process the J1 byte (e.g., the very first POH byte) via this port.*
C8	TXPOHINS	I	TTL	**Transmit Path Overhead Input Port – Insert Enable Input pin:** This pin is used for the Transmit AU-4/VC-4 Mapper POH Processor Block when TUG-3 mapping is used. These input pins, along with "TxPOH", "TxPOHEnable", "TxPOHFrame" and "TxPOHClk" function as the "Transmit Path Overhead (TxPOH) Input Port. These input pins permit the user to enable or disable the "TxPOH" input port. If these input pins are pulled "high", then the "TxPOH" port will sample and latch data via the corresponding "TxPOH" input pins, upon the falling edge of "TxPOHClk". ***Note:*** *Conversely, if these input pins are pulled "low", then the "TxPOH" port will NOT sample and latch data via the corresponding "TxPOH" input pins.*
B6	TXPOHENABLE	O	TTL	**Transmit Path Overhead Input Port – POH Indicator Output pin:** This pin is used for the Transmit AU-4/VC-4 Mapper POH Processor Block when TUG-3 mapping is used. These output pins, along with "TxPOH", "TxPOHIns", "TxPOHFrame" and "TxPOHClk" function as the "Transmit Path Overhead (TxPOH) Input Port". These output pins will pulse "high" anytime the "TxPOH" port is ready to accept and process POH bytes. These output pins will be "low" at all other times.

E10 B8 D11	TxPOH_0 TxPOH_1 TxPOH_2	I	TTL	**Transmit Path Overhead Input Port – Input pin.** These input pins permit the user to insert the POH data into each of the 3 Transmit SONET POH Processor blocks (for insertion and transmission via the "outbound" STS-3 signal. If the user is only inserting POH data via these input pins: In this mode, the external circuitry (which is being interfaced to the "Transmit Path Overhead Input Port" is suppose to monitor the following output pins; • TxPOHFrame_n • TxPOHEnable_n • TxPOHClk_n The "TxPOHFrame_n" output pin will toggle "high" upon the rising edge of "TxPOHClk_n" approximately one "TxPOHClk_n" period prior to the "TxPOH" port being ready to accept and process the first bit within J1 byte (e.g., the first POH byte). The "TxPOHFrame_n" output pin will remain "high" for eight consecutive "TxPOHClk_n" periods. The external circuitry should use this pin to note STS-1 SPE frame boundaries. The "TxPOHEnable_n" output pin will toggle "high" upon the rising edge of "TxPOHClk_n" approximately one "TxPOHClk_n" period prior to the "TxPOH" port being ready to accept and process the first bit within a given POH byte. To externally insert a given POH byte: (1) assert the "TxPOHIns_n" input pin by toggling it "high", and (2) place the value of the first bit (within this particular POH byte) on this input pin upon the very next rising edge of "TxPOHClk_n". This data bit will be sampled upon the very next falling edge of "TxPOHClk_n". The external circuitry should continue to keep the "TxPOHIns_n" input pin "high" and advancing the next bits (within the POH bytes) upon each rising edge of "TxPOHClk_n".
A5 A6 A7	TxPOHClk_0 TxPOHClk_1 TxPOHClk_2	O	CMOS	**Transmit Path Overhead Input Port – Clock Output pin:** These output pins, along with "TxPOH_n", "TxPOHEnable_n", "TxPOHIns_n" and "TxPOHFrame" function as the "Transmit Path Overhead (TxPOH) Input Port". The "TxPOHFrame" and "TxPOHEnable" output pins are updated upon the falling edge this clock output signal. The "TxPOHIns_n" input pins and the data residing on the "TxPOH_n" input pins are sampled upon the next falling edge of this clock signal.

C9 C10 A8	TxPOHFrame_0 TxPOHFrame_1 TxPOHFrame_2	O	CMOS	**Transmit Path Overhead Input Port – Frame Output pin:** These output pins, along with the "TxPOH_n", "TxPOHEnable_n", "TxPOHIns_n" and "TxPOHClk_n" function as the "Transmit Path Overhead Input Port". The exact function of these output pins depends upon whether the user inserting POH or TOH data via the "TxPOH_n" input pins. If the user is only inserting POH data via these input pins: The "TxPOH" port will pulse these output pins "high" whenever it is ready to accept and process the J1 byte (e.g., the very first POH byte) via this port. ***Notes:*** *1. The externally circuitry can determine whether the "TxPOH" port is expecting the A1 byte or the J1 byte, by checking the state of the corresponding "TxPOHEnable" output pin. If the "TxPOHEnable_n" output pin is "LOW" while the "TxPOHFrame_n" output pin is "HIGH", then the "TxPOH" port is ready to process the A1 (TOH) bytes.* *2. If the "TxPOHEnable_n" output pin is "HIGH" while the "TxPOHFrame_n" output pin is "HIGH", then the "TxPOH" port is ready to process the J1 (POH) bytes.*
D10 E11 C11	TxPOHIns_0 TxPOHIns_1 TxPOHIns_2	I	TTL	**Transmit Path Overhead Input Port – Insert Enable Input pin:** These input pins, along with "TxPOH_n", "TxPOHEnable_n", "TxPOHFrame_n" and "TxPOHClk_n" function as the Transmit Path Overhead (TxPOH) Input Port. These input pins permit the user to enable or disable the "TxPOH" input port. If these input pins are pulled "high", then the "TxPOH" port will sample and latch data via the corresponding "TxPOH" input pins, upon the falling edge of "TxPOHClk_n". Conversely, if these input pins are pulled "low", then the "TxPOH" port will NOT sample and latch data via the corresponding "TxPOH" input pins. ***Note:*** *If the "TxPOHIns_n" input pin is pulled "LOW", this setting will be overridden if the user has configured the "Transmit SONET/STS-1 POH Processor" or "Transmit STS-1 TOH Processor" blocks to accept certain POH or TOH overhead bytes via the external port.*
B7 B9 B10	TxPOHEnable_0 TxPOHEnable_1 TxPOHEnable_2	O	CMOS	**Transmit Path Overhead Input Port – POH Indicator Output pin:** These output pins, along with "TxPOH_n", "TxPOHIns_n", "TxPOHFrame_n" and "TxPOHClk_n" function as the "Transmit Path Overhead (TxPOH) Input Port". These output pins will pulse "high" anytime the "TxPOH" port is ready to accept and process POH bytes. These output pins will be "low" at all other times.
TRANSMIT LINE/ SYSTEM SIDE INTERFACE PINS				

C12	TXDS3CLK_0 TXE3CLK_0	I	TTL	**Transmit DS3/E3 Reference Clock Input – Channel 0 (Not used for Mapper Applications):** The exact manner in which the user should handle this input pin depends upon whether Channel 0 has been configured to operate in the Mapper Mode or in the ATM UNI/PPP Mode. **If Channel 0 is configured to operate in the Mapper Mode**: If Channel 0 has been configured to operate in the Mapper Mode, then this input pin supports no function, and should, therefore, be connected to GND. **If Channel 0 is configured to operate in the ATM UNI/PPP/Clear Channel Mode:** If Channel 0 (within the XRT94L33) has been configured to operate in the ATM UNI/PPP Mode, then this input pin will function as the timing reference clock signal for the Transmit STS-1/DS3/E3 Framer block circuitry, provided that Channel 0 has been configured to operate in the Local Timing Mode. If Channel 0 has been configured to operate in the DS3 Mode, then the user is expected to apply a 44.736MHz clock signal to this input pin. Likewise, if Channel 0 has been configured to operate in the E3 Mode, then the user is expected to apply a 34.368MHz clock signal to this input pin. ***Note:*** *For more information on using the XRT94L33 for ATM UNI/PPP applications, the user should consult the XRT94L33 1-Channel STS-3c/3-Channel DS3/E3/STS-1 ATM UNI/PPP Data Sheet.*
B20	TXDS3CLK_1 TXE3CLK_1	I	TTL	**Transmit DS3/E3 Reference Clock Input – Channel 1 (Not used for Mapper Applications):** The exact manner in which the user should handle this input pin depends upon whether Channel 1 has been configured to operate in the Mapper Mode or in the ATM UNI/PPP Mode. **If Channel 1 is configured to operate in the Mapper Mode**: If Channel 1 has been configured to operate in the Mapper Mode, then this input pin supports no function, and should, therefore, be connected to GND. **If Channel 1 is configured to operate in the ATM UNI/PPP Mode:** If Channel 1 (within the XRT94L33) has been configured to operate in the ATM UNI/PPP Mode, then this input pin will function as the timing reference clock signal for the Transmit STS-1/DS3/E3 Framer block circuitry, provided that Channel 1 has been configured to operate in the Local Timing Mode. If Channel 1 has been configured to operate in the DS3 Mode, then the user is expected to apply a 44.736MHz clock signal to this input pin. Likewise, if Channel 1 has been configured to operate in the E3 Mode, then the user is expected to apply a 34.368MHz clock signal to this input pin. ***Note:*** *For more information on using the XRT94L33 for ATM UNI/PPP applications, the user should consult the XRT94L33 1-Channel STS-3c/3-Channel DS3/E3/STS-1 ATM UNI/PPP Data Sheet.*

AF17	TXDS3CLK_2 TXE3CLK_2	I	TTL	**Transmit DS3/E3 Reference Clock Input – Channel 2 (Not used for Mapper Applications):** The exact manner in which the user should handle this input pin depends upon whether Channel 2 has been configured to operate in the Mapper Mode or in the ATM UNI/PPP Mode. **If Channel 2 is configured to operate in the Mapper Mode**: If Channel 2 has been configured to operate in the Mapper Mode, then this input pin supports no function, and should, therefore, be connected to GND. **If Channel 2 is configured to operate in the ATM UNI/PPP Mode:** If Channel 2 (within the XRT94L33) has been configured to operate in the ATM UNI/PPP Mode, then this input pin will function as the timing reference clock signal for the Transmit STS-1/DS3/E3 Framer block circuitry, provided that Channel 2 has been configured to operate in the Local Timing Mode. If Channel 2 has been configured to operate in the DS3 Mode, then the user is expected to apply a 44.736MHz clock signal to this input pin. Likewise, if Channel 2 has been configured to operate in the E3 Mode, then the user is expected to apply a 34.368MHz clock signal to this input pin. ***Note:*** *For more information on using the XRT94L33 for ATM UNI/PPP applications, the user should consult the XRT94L33 1-Channel STS-3c/3-Channel DS3/E3/STS-1 ATM UNI/PPP Data Sheet.*
B11 A22 AD16	TxOHClk_0 TxOHClk_1 TxOHClk_2	O	CMOS	**Transmit Overhead Clock Output:** This output pin functions as the “Transmit Overhead Clock” output for the transmit system side interface when the XRT94L33 is configured to operate in STS-1/DS3/E3 mode, however, it functions as the “Transmit STS-1 Overhead” clock output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** This output pin functions as the “Transmit Overhead Data Input Interface clock signal. If the user enables the “Transmit Overhead Data Input Interface” block by asserting the “TxOHIns” input pin, then the Transmit Overhead Data Input Interface block will sample and latch the data (residing on the “TxOH_n” input pin) upon the falling edge of this signal. **When configured to operate in STS-1 mode:** These output pins, along with “TxOH_n”, “TxOHEnable_n”, “TxOHIns_n” and “TxOHFrame” function as the “Transmit Path Overhead (TxOH) Input Port”. The “TxOHFrame” and “TxOHEnable” output pins are updated upon the falling edge this clock output signal. The “TxOHIns_n” input pins and the data residing on the “TxOH_n” input pins are sampled upon the falling edge of this clock signal.

D12 C18 AC16	TxOHENABLE_0 TxOHENABLE_1 TxOHENABLE_2	O	CMOS	**Transmit Overhead Enable Output indicator** This output pin functions as the "Transmit Overhead Enable" output indicator for the transmit system side interface when the XRT94L33 is configured to operate in STS-1/DS3/E3 mode, however, it functions as the "Transmit STS-1 Overhead Enable" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** The Channel will assert this output pin, for one "TxInClk" period, just prior to the instant that the Transmit Overhead Data Input Interface will be sampling and processing an overhead bit. If the local terminal equipment intends to insert its own value for an overhead bit, into the outbound DS3 or E3 data stream, then it is expected to sample the state of this signal, upon the falling edge of "TxInClk". Upon sampling the "TxOHEnable_n" signal high, the local terminal equipment should (1) place the desired value of the overhead bit, onto the "TxOH_n" input pin and (2) assert the "TxOHIns_n" input pin. The Transmit Overhead Data Input Interface block will sample and latch the data on the "TxOH_n" signal, upon the rising edge of the very next "TxInClk_n" input signal. **When configured to operate in STS-1 mode:** These output pins, along with "TxOH_n", "TxOHIns_n", "TxOHFrame_n" and "TxOHClk_n" function as the "Transmit Path Overhead (TxOH) Input Port". These output pins will pulse "high" anytime the "TxOH" port is ready to accept and process POH bytes. These output pins will be "low" at all other times.

E12 E17 AB16	TxOH_0 TxOH_1 TxOH_2	I	TTL	**Transmit Overhead Data Input:** This input pin functions as the "Transmit Overhead Data" output indicator for the transmit system side interface when the XRT94L33 is configured to operate in STS-1/DS3/E3 mode, however, it functions as the "Transmit STS-1 Overhead Enable" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** The Transmit Overhead Data Input Interface accepts overhead via these input pins, and insert this data into the "overhead" bit positions within the outbound DS3 or E3 frames. If the "TxOHIns_n" input pin is pulled "high", then the Transmit Overhead Data Input Interface will sample the overhead data, via this input pin, upon the falling edge of the TxOHClk_n output signal. Conversely, if the TxOHIns_n input pin is NOT pulled "high", then the Transmit Overhead Data Input Interface block will be inactive and will not accept any overhead data via the TxOH_n input pin. **When configured to operate in STS-1 mode:** These input pins permit the user to do the following. 1. To insert the POH data into each of the 3 Transmit STS-1 POH Processor blocks (for insertion and transmission via each of the "outbound" STS-1 signals). 2. To insert the TOH data into each of the 3 Transmit STS-1 TOH Processor blocks (for insertion and transmission via each of the "outbound" STS-1 signals). The exact function of these input pins, depend upon whether the user have opted to insert the TOH data into the 3 Transmit STS-1 TOH Processor blocks, or not.

E12 E17 AB16	TxOH_0 TxOH_1 TxOH_2	I	TTL	**Continued** **If the user is only inserting POH data via these input pins:** In this mode, the external circuitry (which is being interfaced to the "Transmit Path Overhead Input Port" is suppose to monitor the following output pins. • TxOHFrame_n • TxOHEnable_n • TxOHClk_n The "TxOHFrame_n" output pin will toggle "high" upon the falling edge of "TxOHClk_n" approximately one "TxOHClk_n" period prior to the "TxOH" port being ready to accept and process the first bit within J1 byte (e.g., the first POH byte). The "TxOHFrame_n" output pin will remain "high" for eight consecutive "TxOHClk_n" periods. The external circuitry should use this pin to note STS-1 SPE frame boundaries. The "TxOHEnable_n" output pin will toggle "high" upon the falling edge of "TxOHClk_n" approximately one "TxOHClk_n" period prior to the "TxOH" port being ready to accept and process the first bit within a given POH byte. If the user wishes to externally insert a given POH byte; (1) assert the "TxOHIns_n" input pin by toggling it "high", and **(2)** place the value of the first bit (within this particular POH byte) on this input pin upon the very next falling edge of "TxOHClk_n". This data bit will be sampled upon the very next falling edge of "TxOHClk_n". The external circuitry should continue to keep the "TxOHIns_n" input pin "high" and advancing the next bits (within the POH bytes) upon each rising edge of "TxOHClk_n".

E12 E17 AB16	TxOH_0 TxOH_1 TxOH_2	I	TTL	**Continued** **If the user is inserting both POH and TOH data via these input pins:** In this mode, the external circuitry (which is being interfaced to the "Transmit Path Overhead Input Port" is suppose to monitor the following output pins. • TxOHFrame_n • TxOHEnable_n • TxOHClk_n The "TxOHFrame_n" output pin will toggle "high" twice during a given STS-1 frame period. First, this output pin will toggle high coincident with the "TxOH" port being ready to accept and process the A1 byte (e.g., the very first TOH byte). Second, this output pin will toggle "high" coincident with the "TxOH" port being ready to accept and process the J1 byte (e.g., the very first POH byte). If the externally circuitry samples the "TxOHFrame_n" output pin "high", and the "TxOHEnable_n" output pin "low", then the "TxOH" port is now ready to accept and process the very first TOH byte. If the externally circuitry samples the "TxOHFrame_n" output pin "high" and the "TxOHEnable_n" output pin "high", then the "TxOH" port is now ready to accept and process the very first POH byte. To externally insert a given POH or TOH byte; (1) assert the "TxOHIns_n" input pin by toggling it "high", and (2) place the value of the first bit (within this particular POH or TOH byte) on this input upon the very next falling edge of "TxOHClk_n" This data bit will be sampled upon the very next falling edge of "TxOHClk_n". The external circuitry should continue to keep the "TxOHIns_n" input pin "high" and advancing the next bits (within the POH bytes) upon each rising edge of "TxOHClk_n".

F12 B19 AG19	TxOHINS_0 TxOHINS_1 TxOHINS_2	I	TTL	**Transmit Overhead Data Insert Input:** This input pin functions as the "Transmit Overhead Data Insert" input indicator for the transmit system side interface when the XRT94L33 is configured to operate in STS-1/DS3/E3 mode, however, it functions as the "Transmit STS-1 Overhead Enable" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** This input pin permits the user to either enable or disable the "Transmit Overhead Data Input Interface" block within the DS3/E3 Frame Generator block. If the Transmit Overhead Data Input Interface block is enabled, then the DS3/E3 Frame Generator block will accept overhead data (from the local terminal equipment) via the "TxOH_n" input pin; and insert this data into the overhead bit positions within the outbound DS3 or E3 data stream. Conversely, if the Transmit Overhead Data Input Interface block is disabled, then the DS3/E3 Frame Generator block it will NOT accept overhead data from the local terminal equipment. Pulling this input pin "high" enables the "Transmit Overhead Data Input Interface" block. Pulling this input pin "low" disables the "Transmit Overhead Data Input Interface" block **When configured to operate in STS-1 mode:** These input pins, along with "TxOH_n", "TxOHEnable_n", "TxOHFrame_n" and "TxOHClk_n" function as the "Transmit Overhead (TxOH) Input Port. These input pins permit the user to enable or disable the "TxOH" input port. If these input pins are pulled "high", then the "TxOH" port will sample and latch data via the corresponding "TxOH" input pins, upon the falling edge of "TxOHClk_n". Conversely, if these input pins are pulled "low", then the "TxOH" port will NOT sample and latch data via the corresponding "TxOH" input pins. ***Note:*** *If the "TxOHIns_n" input pin is pulled "LOW", this setting will be overridden if the user has configured the "Transmit SONET/STS-1 POH Processor" or "Transmit STS-1 TOH Processor" blocks to accept certain POH or TOH overhead bytes via the external port.*

A9 D17 AF18	TxOHFRAME_0 TxOHFRAME_1 TxOHFRAME_2	O	CMOS	**Transmit Overhead Framing Pulse:** This input pin functions as the "Transmit Overhead Framing" Pulse for the transmit system side interface when the XRT94L33 is configured to operate in DS3/E3 mode, however, it functions as the "Transmit STS-1 Overhead Enable" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** This output pin pulses high (for one TxOHClk_n" period) coincident with the instant that the DS3/E3 Frame Generator block will be accepting the very first overhead bit within an outbound DS3 or E3 frame (via Transmit Overhead Data Input Interface). **When configured to operate in STS-1 mode:** These output pins, along with the "TxOH_n", "TxOHEnable_n", "TxOHIns_n" and "TxOHClk_n" function as the "Transmit Overhead Input Port". The exact function of these output pins depends upon whether the user inserting POH or TOH data via the "TxOH_n" input pins. **If the user is only inserting POH data via these input pins:** In this mode, the "TxOH" port will pulse these output pins "high" whenever it is ready to accept and process the J1 byte (e.g., the very first POH byte) via this port. **If the user is inserting both POH and TOH data via these input pins:** In this mode, the "TxOH" port will pulse these output pins "high" coincident with the following. Whenever the "TxOH" port is ready to accept and process the A1 byte (e.g., the very first TOH byte) via this port. Whenever the "TxOH" port is ready to accept and process the J1 byte (e.g., the very first POH byte) via this port. ***Notes:*** *1. The externally circuitry can determine whether the "TxOH" port is expecting the A1 byte or the J1 byte, by checking the state of the corresponding "TxOHEnable" output pin. If the "TxOHEnable_n" output pin is "LOW" while the "TxOHFrame_n" output pin is "HIGH", then the "TxOH" port is ready to process the A1 (TOH) bytes.* *2. If the "TxOHEnable_n" output pin is "HIGH" while the "TxOHFrame_n" output pin is "HIGH", then the "TxOH" port is ready to process the J1 (POH) bytes.*

AF19 AG21 AE17	STUFFCNTL_0/ TXHDLC_CLK_0/ STUFFCNTL_1/ TXHDLC_CLK_1/ STUFFCNTL_2/ TXHDLC_CLK_2/	I/O	TTL/CMOS	**Transmit PLCP Processor Block – Nibble Trailer Stuff Control Input pin/Transmit High-Speed HDLC Controller Input Interface – Clock Output pin – Channel n:** The exact function of this input pin depends upon (1) whether the XRT94L33 has been configured to operate in the ATM UNI/PLCP Mode and (2) whether a given DS3/E3 Framer block/Channel has been configured to operate in the "High-Speed HDLC Controller" Mode, as described below. **ATM UNI Mode - STUFFCNT_n: Transmit PLCP Processor block Nibble-Trailer Stuff Control Input pin – Channel n - STUFFCNT_n:** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode. For more information on this pin operating in this mode, please see the XRT94L33 Pin Description for ATM UNI/PPP Applications. **High-Speed HDLC Controller Mode – Transmit HDLC Controller Input Interace Block - Clock output signal – Channel n – TxHDLCClk_n:** This output signal functions as the "demand" clock for the Transmit High-Speed HDLC Controller Input Interface block, associated with the DS3/E3 Framer blocks. Whenever the user pulls the "Snd_Msg_n" input pin "high" then the Transmit High-Speed HDLC Controller block will begin to sample and latch the contents of the "TxHDLCDat[7:0] input pins upon the falling edge of this clock signal. The user is advised to configure their terminal equipment circuitry to output (or place) data onto the "TxHDLCDat[7:0] bus upon the rising edge of this clock signal. Since the Transmit HDLC Controller block is sampling and latching 8-bits of data at a given time, it may be assumed that the frequency of the TxHDLC_CLK_n output signal is either 34.368MHz/8 or 44.736MHz/8. In general, this presumption is true. However, because the Transmit HDLC Controller block is also performing "Zero-Stuffing" of the user data that it accepts from the Terminal Equipment, the frequency of this signal may be slower. ***Note:*** *The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

AC17 AD17 AG20	EIGHTKHZSYNC_0/ RXHDLC_CLK_0/ EIGHTKHZSYNC_1/ RXHDLC_CLK_1/ EIGHTKHZSYNC_2/ RXHDLC_CLK_2/	I/O	TTL/CMOS	**Transmit PLCP Processor Block – 8kHz Framing Alignment Input/Receive High-Speed HDLC Controller Output Interface Block – Clock Output – Channel n:** The exact function of this input pin depends upon (1) whether the XRT94L33 has been configured to operate in the ATM UNI/PLCP Mode and (2) whether Channel n has been configured to operate in the "High-Speed HDLC Controller" Mode, as described below. **ATM UNI Mode - EIGHTKHZSYNC_n: Transmit PLCP Processor Block 8kHz Framing Alignment Input:** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode. For more information on this pin operating in this mode, please see the XRT94L33 Pin Description for ATM UNI/PPP Applications. **High-Speed HDLC Controller Mode - Receive High-Speed HDLC Controller Output Interface Block - Clock output signal – Channel n – RxHDLCClk_n:** This output pin functions as the "Receive High-Speed HDLC Controller Output Interface block – clock output signal for Channel n. The Receive High-Speed HDLC Controller Output Interface block outputs the contents of all received HDLC frames and flag sequence octets via the Receive High-Speed HDLC Controller Output Interface block – Data Bus output pins (RxHDLCDat_n[7:0]) upon the rising edge of this clock signal. The user is advised to configure the terminal equipment to sample the contents of the RxHDLCDat_n[7:0] output pins upon the falling edge of this clock signal. ***Note:*** *The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*
D27	TXPERR	I	TTL	**For Mapper applications, please connect this pin to GND.**
G25	TxPEOP	I	TTL	**For Mapper applications, please connect this pin to GND.**
F25	TxMOD_0	I	TTL	**For Mapper applications, please connect this pin to GND.**
J24	TxUPRTY/ TxPPRTY	I	TTL	**For Mapper applications, please connect this pin to GND.**

H27 G27 L24 J26 L23 K25 F27 H26 G26 K24 J25 E27 K23 F26 H25 E26	TxUDATA_0/ TxPDATA_0 TxUDATA_1/ TxPDATA_1 TxUDATA_2/ TxPDATA_2 TxUDATA_3/ TxPDATA_3 TxUDATA_4/ TxPDATA_4 TxUDATA_5/ TxPDATA_5 TxUDATA_6/ TxPDATA_6 TxUDATA_7/ TxPDATA_7 TxUDATA_8/ TxPDATA_8 TxUDATA_9/ TxPDATA_9 TxUDATA_10/ TxPDATA_10 TxUDATA_11/ TxPDATA_11 TxUDATA_12/ TxPDATA_12 TxUDATA_13/ TxPDATA_13 TxUDATA_14/ TxPDATA_14 TxUDATA_15/ TxPDATA_15	I	TTL	**For Mapper applications, please connect these input pins to GND.**
M24 M23 J27 K26 L25	TxUADDR_0 TxUADDR_1 TxUADDR_2 TxUADDR_3 TxUADDR_4	I	TTL	**For Mapper applications, please connect these input pins to GND.**
L26	TxUClav/TxPPA	O	CMOS	**For Mapper applications, please leave this pin open.**
M25	TxUSOC/ TXPSOP/ TXPSOC	I	TTL	**For Mapper applications, please connect this pin to GND.**
K27	TxTSX / TXPSOF	I	TTL	**For Mapper applications, please connect this pin to GND.**
M26	TXUENB_L/ TXPENB_L	I	TTL	**For Mapper applications, please connect this pin to VDD.**
L27	TXUCLKO/ TXPCLKO	O	CMOS	**For Mapper applications, please leave this pin open.**
M27	TXUCLK/ TXPCLK	I	TTL	**For Mapper applications, please connect this pin to GND.**

STS-1 TELECOM BUS INTERFACE – TRANSMIT DIRECTION

C14	STS1TXA_CK_0 TXSENDFCS_0 TXGFCCLK_0	I I O	TTL TTL CMOS	**STS-1 Transmit Telecom Bus Clock Input pin/Transmit HDLC Control Block Send FCS Command Input pin – Channel 0:** The exact function of this input pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled – STS1TXA_CLK_0 - "STS-1 Transmit Telecom Bus" Transmit Clock Input – Channel 0:** This input clock signal functions as the clock source for the STS-1 Transmit Telecom Bus, associated with Channel 0. All input signals (e.g., STS1TXA_ALARM_0, STS1TXA_D_0[7:0], STS1TXA_DP_0, STS1TXA_PL_0, STS1TXA_C1J1_0) are sampled upon the falling edge of this input clock signal. This clock signal should operate at 19.44MHz. (For STS-3 mode) or 6.48MHz (for STS-1 mode) **If STS-1 Telecom Bus (Channel 0) has NOT been enabled:** If STS-1 Telecom Bus (Channel 0) has not been enabled, then this particular pin can be configured to function in either of the following roles. **TXSENDFCS_0 (Transmit HDLC Controller block Send FCS Command Input – High Speed HDLC Controller Mode Only)** The user's terminal equipment is expected to control both this input pin and the "TXSENDMSG_0" input pin during the construction and transmission of each outbound HDLC frame. This input pin permits the user to command the Transmit HDLC Controller block to compute and insert the computed FCS value into the back-end of the "outbound" HDLC frame as a trailer. If the user has configured the Transmit HDLC Controller to compute and insert a CRC-16 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for two periods of TxHDLCClk_0. Likewise, if the user has configured the Transmit HDLC Controller to compute and insert a CRC-32 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for four periods of TxHDLCClk_0. **TXGFCCLK_0 (Transmit GFC Nibble-Field Input Port clock signal Input) – ATM Applications ONLY.** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode. ***Note:*** *The user should tie this pin to GND the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

E19	STS1TXA_CK_1 TXSENDFCS_1 TXGFCCLK_1	I I O	TTL TTL CMOS	**STS-1 Transmit Telecom Bus Clock Input pin/Transmit HDLC Control Block Send FCS Command Input pin – Channel 1:** The exact function of this input pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – **STS1TXA_CLK_1 -** "STS-1 Transmit Telecom Bus" Clock Input – Channel 1: This input clock signal functions as the clock source for the STS-1 Transmit Telecom Bus, associated with Channel 1. All input signals, (e.g., STS1TXA_ALARM_1, STS1TXA_D_1[7:0], STS1TXA_DP_1, STS1TXA_PL_1, STS1TXA_C1J1_1) are sampled upon the falling edge of this input clock signal. This clock signal should operate at 19.44MHz. (For STS-3 mode) or 6.48MHz (for STS-1 mode) **If STS-1 Telecom Bus (Channel 1) has NOT been enabled:** If STS-1 Telecom Bus (Channel 1) has not been enabled, then this particular pin can be configured to function in either of the following roles. **TXSENDFCS_1 (Transmit HDLC Controller block Send FCS Command Input – High Speed HDLC Controller Mode Only)** The user's terminal equipment is expected to control both this input pin and the "TXSENDMSG_1" input pin during the construction and transmission of each outbound HDLC frame. This input pin permits the user to command the Transmit HDLC Controller block to compute and insert the computed FCS value into the back-end of the "outbound" HDLC frame as a trailer. If the user has configured the Transmit HDLC Controller to compute and insert a CRC-16 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for two periods of TxHDLCClk_1. Likewise, if the user has configured the Transmit HDLC Controller to compute and insert a CRC-32 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for four periods of TxHDLCClk_1. **TXGFCCLK_1 (Transmit GFC Nibble-Field Input Port clock signal Input) – ATM Applications ONLY.** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode. *NOTE: The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

AC14	STS1TXA_CK_2 TXSENDFCS_2 TXGFCCLK_2	IO	TTL CMOS CMOS	**STS-1 Transmit Telecom Bus Clock Input pin/Transmit HDLC Control Block Send FCS Command Input pin – Channel 2:** The exact function of this input pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – **STS1TXA_CLK_2 –** "STS-1 Transmit Telecom Bus" Transmit Clock Input – Channel 2: This input clock signal functions as the clock source for the STS-1 Transmit Telecom Bus, associated with Channel 2. All input signals, (e.g., STS1TXA_ALARM_2, STS1TXA_D_2[7:0]", STS1TXA_DP_2, STS1TXA_PL_2, STS1TXA_C1J1_2) are sampled upon the falling edge of this input clock signal. This clock signal should operate at 19.44MHz. (For STS-3 mode) or 6.48MHz (for STS-1 mode) **If STS-1 Telecom Bus (Channel 1) has NOT been enabled:** If STS-1 Telecom Bus (Channel 1) has not been enabled, then this particular pin can be configured to function in either of the following roles. **TXSENDFCS_2 (Transmit HDLC Controller block Send FCS Command Input – High Speed HDLC Controller Mode Only)** The user's terminal equipment is expected to control both this input pin and the "TXSENDMSG_2" input pin during the construction and transmission of each outbound HDLC frame. This input pin permits the user to command the Transmit HDLC Controller block to compute and insert the computed FCS value into the back-end of the "outbound" HDLC frame as a trailer. If the user has configured the Transmit HDLC Controller to compute and insert a CRC-16 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for two periods of TxHDLCClk_2. Likewise, if the user has configured the Transmit HDLC Controller to compute and insert a CRC-32 value into the "outbound" HDLC frame, then the terminal equipment is expected to pull this input pin "high" for four periods of TxHDLCClk_2. **TXGFCCLK_2 (Transmit GFC Nibble-Field Input Port clock signal Input) – ATM Applications ONLY.** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode. *NOTE: The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

E14	STS1TXA_PL_0 TXSENDMSG_0	I	TTL	**STS-1 Transmit Telecom Bus – Payload Indicator Signal input/Transmit HDLC Controller block Send Message Command Input pin – Channel 0:** The exact function of this input depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled – STS1TXA_PL_0 - STS-1 Transmit Telecom Bus – Payload Indicator Signal – Channel 0:** This input pin indicates whether or not "Transport Overhead" (TOH) bytes are being input via the "TXA_D_0[7:0]" input pins. This input pin should be pulled "low" for the duration that the "STS-1 Transmit Telecom Bus is receiving a TOH byte, via the "TXA_D_0[7:0]" input pins. Conversely, this input pin should be pulled "high" at all other times. ***Note:*** *This input signal is sampled upon the falling edge of "STS1TXA_CK_0".* **If STS-1 Telecom Bus (Channel 0) has NOT been enabled:** If STS-1 Telecom Bus (Channel 0) has not been enabled, then this particular pin can either be configured to function as the "TxSENDMSG_0" input pin (if the DS3/E3 Framer block within Channel 0 has been configured to operate in the "High-Speed HDLC Controller Mode), or the user should simply tie this input pin to GND. The details of this pin's role as the "TxSENDMSG_0" input pin is described below. **TXSENDMSG_0 (Transmit HDLC Controller block Send Message Command Input – High Speed HDLC Controller Mode Only)** This input pin permits the user to command the Transmit HDLC Controller block (associated with Channel 0) to begin sampling and latching the data which is being applied to the "TxHDLCDat_0[7:0]" input pins. If the user pulls this input pin "high", then the Transmit HDLC Controller block samples and latches the data which is applied to the "TxHDLCDat_0[7:0]" input pins upon the rising edge of "TxHDLCClk_0". Each byte of this sampled data will ultimately be encapsulated into an outbound HDLC frame and will be mapped into the payload bits within the outbound DS3/E3 frames via the DS3/E3 Frame Generator block. If the user pulls this input pin "low" then the Transmit HDLC Controller block will NOT sample and latch the contents on the "TxHDLCDat_0[7:0]" input pins, and the Transmit HDLC Controller block will simply generate a continuous stream of flag sequence octets (0x7E). ***Note:*** *The user should tie this pin to GND if the DS3/E3 Framer block* has NOT been configured *to operate in the "High-Speed HDLC Controller" Mode.*

C22	STS1TXA_PL_1 TXSENDMSG_1:	I	TTL	***STS-1 Transmit Telecom Bus – Payload Indicator Signal input/Transmit HDLC Controller block Send Message Command Input pin – Channel 1:*** *The exact function of this input pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not.* ***If STS-1 Telecom Bus (Channel 1) has been enabled – STS1TXA_PL_1 - STS-1 Transmit Telecom Bus – Payload Indicator Signal – Channel 1:*** *This input pin indicates whether or not "Transport Overhead" (TOH) bytes are being input via the "TXA_D_1[7:0]" input pins.* *This input pin should be pulled "low" for the duration that the STS-1 Transmit Telecom Bus is receiving a TOH byte, via the "TXA_D_1[7:0]" input pins. Conversely, this input pin should be pulled "high" at all other times.* ***Note:*** *This input signal is sampled upon the falling edge of "STS1TXA_CK_1".* ***If STS-1 Telecom Bus (Channel 1) has NOT been enabled:*** *If STS-1 Telecom Bus (Channel 1) has not been enabled, then this particular pin can either be configured to function as the "TxSENDMSG_1" input pin (if the DS3/E3 Framer block within Channel 1 has been configured to operate in the "High-Speed HDLC Controller Mode), or the user should simply tie this input pin to GND. The details of this pin's role as the "TxSENDMSG_1" input pin is described below.* ***TXSENDMSG_1 (Transmit HDLC Controller block Send Message Command Input – High Speed HDLC Controller Mode ONLY)*** *This input pin permits the user to command the Transmit HDLC Controller block (associated with Channel 1) to begin sampling and latching the data which is being applied to the "TxHDLCDat_1[7:0]" input pins.* *If the user pulls this input pin "high", then the Transmit HDLC Controller block samples and latches the data which is applied to the "TxHDLCDat_1[7:0]" input pins upon the rising edge of "TxHDLCClk_1". Each byte of this sampled data will ultimately be encapsulated into an outbound HDLC frame and will be mapped into the payload bits within the outbound DS3/E3 frames via the DS3/E3 Frame Generator block.* *If the user pulls this input pin "low" then the Transmit HDLC Controller block will NOT sample and latch the contents on the "TxHDLCDat_1[7:0]" input pins, and the Transmit HDLC Controller block will simply generate a continuous stream of flag sequence octets (0x7E).* ***Note:*** *The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

AD14	STS1TXA_PL_2 TXSENDMSG_2:	I	TTL	***STS-1 Transmit Telecom Bus – Payload Indicator Signal input/Transmit HDLC Controller block Send Message Command Input pin – Channel 2:*** *The exact function of this input pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not.* ***If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Payload Indicator Signal – Channel 2:* *This input pin indicates whether or not "Transport Overhead" (TOH) bytes are being input via the "TXA_D_2[7:0]" input pins.* *This input pin should be pulled "low" for the duration that the STS-1 Transmit Telecom Bus is receiving a TOH byte, via the "TXA_D_2[7:0]" input pins. Conversely, this input pin should be pulled "high" at all other times.* ***Note:** This input signal is sampled upon the falling edge of "STS1TXA_CK_2".* ***If STS-1 Telecom Bus (Channel 2) has NOT been enabled:*** *If STS-1 Telecom Bus (Channel 2) has not been enabled, then this particular pin can either be configured to function as the "TxSENDMSG_2" input pin (if the DS3/E3 Framer block within Channel 2 has been configured to operate in the "High-Speed HDLC Controller Mode), or the user should simply tie this input pin to GND. The details of this pin's role as the "TxSENDMSG_2" input pin is described below.* ***TXSENDMSG_2 (Transmit HDLC Controller block Send Message Command Input – High Speed HDLC Controller Mode ONLY)*** *This input pin permits the user to command the Transmit HDLC Controller block (associated with Channel 2) to begin sampling and latching the data which is being applied to the "TxHDLCDat_2[7:0]" input pins.* *If the user pulls this input pin "high", then the Transmit HDLC Controller block samples and latches the data which is applied to the "TxHDLCDat_2[7:0]" input pins upon the rising edge of "TxHDLCClk_2". Each byte of this sampled data will ultimately be encapsulated into an outbound HDLC frame and will be mapped into the payload bits within the outbound DS3/E3 frames via the DS3/E3 Frame Generator block.* *If the user pulls this input pin "low" then the Transmit HDLC Controller block will NOT sample and latch the contents on the "TxHDLCDat_2[7:0]" input pins, and the Transmit HDLC Controller block will simply generate a continuous stream of flag sequence octets (0x7E).* ***Note:** The user should tie this pin to GND if the DS3/E3 Framer block has NOT been configured to operate in the "High-Speed HDLC Controller" Mode.*

D14	STS1TXA_C1J1_0 RXDS3LINECLK_0	I	TTL	**STS-1 Transmit Telecom Bus C1/J1 Byte Phase Indicator Input Signal/Receive DS3/E3/STS-1 Clock Input from LIU (Channel 0):** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled - STS-1 Transmit Telecom Bus C1/J1 Byte Phase Indicator Input Signal (Channel 0):** This input pin should be pulsed "high" during both of the following conditions. Whenever the C1 byte is being input to the STS-1 Transmit Telecom Bus (TXA_D_0[7:0]) input pins. Whenever the J1 byte is being input to the STS-1 Transmit Telecom Bus (TXA_D_0[7:0]) input pins. This input pin should be pulled "low" at all other times. **If STS-1 Telecom Bus (Channel 0) has NOT been enabled - RXDS3LINECLK_0 (Receive DS3/E3/STS-1 clock input from LIU)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 0) uses this input pin to sample and latch the data that is present on the RxDS3POS_0 and RxDS3NEG_0 (for Dual-Rail Operation only) inputs. This input clock signal also functions as the timing source for the Ingress Direction signal and circuitry within the DS3/E3 Framer block of Channel 0. The user is expected to connect this input to the Recovered Clock Output of a DS3/E3/STS-1 LIU IC.

A24	STS1TXA_C1J1_1 RXDS3LINECLK_1/ RxSTS1LineClk_1	I	TTL	**Transmit STS-1 Telecom Bus Interface - C1/J1 Byte Phase Indicator Input Signal – Channel 1/Receive DS3/E3/STS-1 Clock Input from LIU – Channel 1:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If the STS-1 Telecom Bus (Channel 1) has been enabled - Transmit STS-1 Telecom Bus Interface - C1/J1 Byte Phase Indicator Input Signal (Channel 1):** This input pin should be pulsed “high” during both of the following conditions. Whenever the C1 byte is being input to the Transmit STS-1 Telecom Bus Interface input pins (TXA_D_1[7:0]). Whenever the J1 byte is being input to the Transmit STS-1 Telecom Bus Interface input pins (TXA_D_1[7:0]). This input pin should be pulled “low” at all other times. **If the STS-1 Telecom Bus (Channel 1) has NOT been enabled - RXDS3LINECLK_1 (Receive DS3/E3/STS-1 clock input from LIU)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 1) uses this input pin to sample and latch the data that is present on the RxDS3POS_1 and RxDS3NEG_1 (for Dual-Rail Operation only) inputs. This input clock signal also functions as the timing source for the Ingress Direction signal and circuitry within the DS3/E3 Framer block of Channel 1. The user is expected to connect this input to the Recovered Clock Output pin of an off chip DS3/E3/STS-1 LIU IC.

AF14	STS1TXA_C1J1_2 RXDS3LINECLK_2	I	TTL	**STS-1 Transmit Telecom Bus C1/J1 Byte Phase Indicator Input Signal/Receive DS3/E3/STS-1 Clock Input from LIU (Channel 2):** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** - STS-1 Transmit Telecom Bus C1/J1 Byte Phase Indicator Input Signal (Channel 2): This input pin should be pulsed “high” during both of the following conditions. Whenever the C1 byte is being input to the STS-1 Transmit Telecom Bus (TXA_D_2[7:0]) input pins. Whenever the J1 byte is being input to the STS-1 Transmit Telecom Bus (TXA_D_2[7:0]) input pins. This input pin should be pulled “low” at all other times. **Is STS-1 Telecom Bus (Channel 2) has NOT been enabled - RXDS3LINECLK_2 (Receive DS3/E3/STS-1 clock input from LIU)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 2) uses this input pin to sample and latch the data that is present on the RxDS3POS_2 and RxDS3NEG_2 (for Dual-Rail Operation only) inputs. This input clock signal also functions as the timing source for the Ingress Direction signal and circuitry within the DS3/E3 Framer block of Channel 2. The user is expected to connect this input to the Recovered Clock Output of a DS3/E3/STS-1 LIU IC.

B14	STS1TXA_DP_0 RXDS3POS_0	I	TTL	**STS-1 Transmit Telecom Bus – Parity Input pin/Receive DS3/E3/STS-1 Positive-Polarity Data Input from LIU – Channel 0:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled – STS1TXA_DP_0 - STS-1 Transmit Telecom Bus Interface # 0 – Parity Input Pin:** This input pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits which are input via the "STS1TXA_D_0[7:0]" input pins. The EVEN or ODD parity value of the bits which are being input via the "STS1TXA_D_0[7:0]" input, and the states of the "STS1TXA_PL_0" and "STS1TXA_C1J1_0" input pins. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Interface Control Register – Byte 0" register (Address Location = 0x013B).* **If STS-1 Telecom Bus (Channel 0) has NOT been enabled - RXDS3POS_0 (Receive DS3/E3/STS-1 Positive-Polarity data input from LIU)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 0) will sample the data being applied to this input pin upon the user-selected edge of the "RXDS3LINECLK_0" input signal. If the user has configured Channel 0 to operate in the STS-1 Mode, or in the Single-Rail Mode (if also configured to operate in the DS3/E3 Mode), then all Recovered DS3, E3 or STS-1 data (from the DS3/E3/STS-1 LIU IC) should be applied to this input pin. If the user has configured Channel 0 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "positive-polarity" portion of the Recovered DS3/E3 data should be applied to this input pin.

C21	STS1TXA_DP_1 RXDS3POS_1	I	TTL	**STS-1 Transmit Telecom Bus – Parity Input pin/Receive DS3/E3/STS-1 Positive-Polarity Data Input from LIU – Channel 1:** The exact function of this input pin depends upon whether STS-1 Telecom Bus Interface # 1 has been enable or not. **If STS-1 Telecom Bus (Channel 1) has been enabled – STS1TXA_DP_1: STS-1 Transmit Telecom Bus Interface # 1 – Parity Input pin:** This input pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits which are input via the "STS1TXA_D_1[7:0]" input pins. The EVEN or ODD parity value of the bits which are being input via the "STS1TXA_D_1[7:0]" input and the states of the "STS1TXA_PL_1" and "STS1TXA_C1J1_1" input pins. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Interface Control Register – Byte 1" register (Address Location = 0x013A).* **If STS-1 Telecom Bus (Channel 1) has NOT been enabled - RXDS3POS_1 (Receive DS3/E3/STS-1 Positive-Polarity data input from LIU – Channel 1)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 1) will sample the data being applied to this input pin upon the user-selected edge of the "RXDS3LINECLK_1" input signal. If the user has configured Channel 1 to operate in the STS-1 Mode, or in the Single-Rail Mode (if also configured to operate in the DS3/E3 Mode), then all Recovered DS3, E3 or STS-1 data (from the DS3/E3/STS-1 LIU IC) should be applied to this input pin. If the user has configured Channel 1 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "positive-polarity" portion of the Recovered DS3/E3 data should be applied to this input pin.

AG15	STS1TXA_DP_2 RXDS3POS_2	I	TTL	**STS-1 Transmit Telecom Bus – Parity Input pin/Receive DS3/E3/STS-1 Positive-Polarity Data Input from LIU – Channel 2;** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled – STS1TXA_DP_2: STS-1 Transmit Telecom Bus Interface # 2 – Parity Input Pin:** This input pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits which are input via the "STS1TXA_D_2[7:0]" input pins. The EVEN or ODD parity value of the bits which are being input via the "STS1TXA_D_2[7:0]" input and the states of the "STS1TXA_PL_2" and "STS1TXA_C1J1_2" input pins. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Interface Control Register – Byte 2" register (Address Location = 0x0139).* **If STS-1 Telecom Bus (Channel 2) has NOT been enabled RXDS3POS_2 (Receive DS3/E3/STS-1 Positive-Polarity data input from LIU)** The DS3/E3 Framer block and the Receive STS-1 TOH Processor block (associated with Channel 2) will sample the data being applied to this input pin upon the user-selected edge of the "RXDS3LINECLK_2" input signal. If the user has configured Channel 2 to operate in the STS-1 Mode, or in the Single-Rail Mode (if also configured to operate in the DS3/E3 Mode), then all Recovered DS3, E3 or STS-1 data (from the DS3/E3/STS-1 LIU IC) should be applied to this input pin. If the user has configured Channel 2 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "positive-polarity" portion of the Recovered DS3/E3 data should be applied to this input pin.

A13	STS1TXA_ALARM_0 RXDS3NEG_0 RxLCV_0	I	TTL	**Transmit STS-1 Telecom Bus – Alarm Indicator Input/Receive DS3/E3 Negative-Polarity Data Input from LIU/Receive DS3/E3 Line Code Violation Input from LIU – Channel 0;** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Alarm Indicator Input: This input pin pulses "high" coincident to any STS-1 signal (which is carrying the AIS-P indicator) being applied to the STS1TXA_D_0[7:0] input data bus. ***Note:*** *If the STS1TXA_ALARM_0 input signal pulses "HIGH" for any given STS-1 signal (within the "incoming" STS-1), then the XRT94L33 will automatically declare the AIS-P defect condition for that particular STS-1 channel.* **If STS-1 Telecom Bus (Channel 0) has NOT been enabled:** If the STS-1 Telecom Bus (Channel 0) has NOT been enabled, then the role that this particular input pin plays depends upon whether Channel 0 is operating in the STS-1 Mode, the DS3/E3 Single-Rail Mode or in the DS3/E3 Dual-Rail Mode, as described below. **If Channel 0 is operating in the STS-1 Mode** If Channel 0 is operating in the STS-1 Mode, then the user should tie this pin to GND. **If Channel 0 is operating in the DS3/E3 Single-Rail Mode – Receive LCV Input from LIU** If Channel 0 is operating in both the DS3/E3 and Single-Rail Modes, then this input pin will function as the LCV (Line Code Violation) input. In this mode, the user is expected to connect the "LCV" output pin from the LIU IC to this input pin. The DS3/E3 Framer block will sample this input pin upon the "user-configured" edge of the "RXDS3LINECLK_0" clock signal, and the Primary Frame Synchronizer block (corresponding with Channel 0) will increment the PMON LCV or EXZ Event Count registers based upon the data sampled at this input pin. **If Channel 0 is operating in the DS3/E3 Dual-Rail Mode – Receive DS3/E3 Negative-Polarity Data Input from LIU** If the user has configured Channel 0 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "negative-polarity" portion of the Receive DS3/E3 data should be applied to this input pin.

D19	STS1TXA_ALARM_1 RXDS3NEG_1 RxLCV_1	I	TTL	**Transmit STS-1 Telecom Bus – Alarm Indicator Input/Receive DS3/E3 Negative-Polarity Data Input from LIU/Receive DS3/E3 Line Code Violation Input from LIU – Channel 1:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Alarm Indicator Input: This input pin pulses "high" coincident to any STS-1 signal (which is carrying the AIS-P indicator) being applied to the STS1TXA_D_1[7:0] input data bus. ***Note:*** *If the STS1TXA_ALARM_1 input signal pulses "HIGH" for any given STS-1 signal (within the "incoming" STS-1), then the XRT94L33 will automatically declare the AIS-P defect condition for that particular STS-1 channel.* **If STS-1 Telecom Bus (Channel 1) has NOT been enabled:** If the STS-1 Telecom Bus (Channel 1) has NOT been enabled, then the role that this particular input pin plays depends upon whether Channel 1 is operating in the STS-1 Mode, the DS3/E3 Single-Rail Mode or in the DS3/E3 Dual-Rail Mode, as described below. **If Channel 1 is operating in the STS-1 Mode** If Channel 1 is operating in the STS-1 Mode, then the user should tie this pin to GND. **If Channel 1 is operating in the DS3/E3 Single-Rail Mode – Receive LCV Input from LIU** If Channel 1 is operating in both the DS3/E3 and Single-Rail Modes, then this input pin will function as the LCV (Line Code Violation) input. In this mode, the user is expected to connect the "LCV" output pin from the LIU IC to this input pin. The DS3/E3 Framer block will sample this input pin upon the "user-configured" edge of the "RXDS3LINECLK_1" clock signal, and the Primary Frame Synchronizer block (corresponding with Channel 1) will increment the PMON LCV or EXZ Event Count registers based upon the data sampled at this input pin. **If Channel 1 is operating in the DS3/E3 Dual-Rail Mode – Receive DS3/E3 Negative-Polarity Data Input from LIU** If the user has configured Channel 1 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "negative-polarity" portion of the Receive DS3/E3 data should be applied to this input pin.

AF15	STS1TXA_ALARM_2 RXDS3NEG_2 RxLCV_2	I	TTL	**Transmit STS-1 Telecom Bus – Alarm Indicator Input/Receive DS3/E3 Negative-Polarity Data Input from LIU/Receive DS3/E3 Line Code Violation Input from LIU – Channel 2:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Alarm Indicator Input: This input pin pulses "high" coincident to any STS-1 signal (which is carrying the AIS-P indicator) being applied to the STS1TXA_D_2[7:0] input data bus. ***Note:*** *If the STS1TXA_ALARM_2 input signal pulses "HIGH" for any given STS-1 signal (within the "incoming" STS-1), then the XRT94L33 will automatically declare the AIS-P defect condition for that particular STS-1 channel.* **If STS-1 Telecom Bus (Channel 2) has NOT been enabled:** If the STS-1 Telecom Bus (Channel 2) has NOT been enabled, then the role that this particular input pin plays depends upon whether Channel 2 is operating in the STS-1 Mode, the DS3/E3 Single-Rail Mode or in the DS3/E3 Dual-Rail Mode, as described below. **If Channel 2 is operating in the STS-1 Mode** If Channel 2 is operating in the STS-1 Mode, then the user should tie this pin to GND. **If Channel 2 is operating in the DS3/E3 Single-Rail Mode – Receive LCV Input from LIU** If Channel 2 is operating in both the DS3/E3 and Single-Rail Modes, then this input pin will function as the LCV (Line Code Violation) input. In this mode, the user is expected to connect the "LCV" output pin from the LIU IC to this input pin. The DS3/E3 Framer block will sample this input pin upon the "user-configured" edge of the "RXDS3LINECLK_2" clock signal, and the Primary Frame Synchronizer block (corresponding with Channel 1) will increment the PMON LCV or EXZ Event Count registers based upon the data sampled at this input pin. **If Channel 2 is operating in the DS3/E3 Dual-Rail Mode – Receive DS3/E3 Negative-Polarity Data Input from LIU** If the user has configured Channel 2 to operate in both the DS3/E3 and the Dual-Rail Mode, then only the "negative-polarity" portion of the Receive DS3/E3 data should be applied to this input pin.

B13	STS1TXA_D0_0 TXHDLCDAT_0_0 TXGFCMSB_0	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 0** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 0: This input pin along with "STS1TXA_D_0[7:1]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. This particular input pin functions as the LSB (Least Significant Bit) input pin on the Transmit (Add) Telecom Bus – Input Data Bus. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". The LSB of any byte, which is being input into the "STS-1 Transmit Telecom Bus – Data Bus (for Channel 0) should be input via this pin. **TXHDLCDAT_0_0 (Transmit HDLC block data – Channel 0 – Input data pin 0)** If Channel 0 has been configured to operate in the "High-Speed HDLC Controller" Mode, then the System-Side Terminal Equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide **TXGFCMSB_0 (Transmit GFC MSB Indicator – Channel 0) – ATM Applications ONLY.** This pin only functions in this particular role if the XRT94L33 has been configured to operate in the ATM UNI Mode.
C13	STS1TXA_D1_0 TXHDLCDAT_1_0 TXGFC_0	I	TTL	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 1:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 1: This input pin along with "STS1TXA_D_0[7:2]" and "STS1TXA_D0_0 function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". **TXHDLCDAT_1_0 (Transmit HDLC block data – Channel 0 – Input data pin 1)** **TXGFC_0 (Transmit GFC data – Channel 0)**

D13	STS1TXA_D2_0 TXHDLCDAT_2_0 TXCELLTXED_0	I	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 2:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 2: STS1TXA_D2_0 This input pin along with "STS1TXA_D_0[7:3]" and "STS1TXA_D_0[1:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". **TXHDLCDAT_2_0 (Transmit HDLC block data – Channel 0 – Input data pin 2)** **TXCELLTXED_0 (Cell Transmitted – Channel 0)**
E13	STS1TXA_D3_0 TXHDLCDAT_3_0 SSI_CLK	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 3:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 2: STS1TXA_D3_0: This input pin along with "STS1TXA_D_0[7:4]" and "STS1TXA_D_0[2:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". **TXHDLCDAT_3_0 (Transmit HDLC block data – Channel 0 – Input data pin 3** **SSI_CLK (Slow Speed Interface for Ingress Path Clock)**
A12	STS1TXA_D4_0 TXHDLCDAT_4_0 TXDS3OHIND_0	IO	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 4:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 4: STS1TXA_D4_0: This input pin along with "STS1TXA_D_0[7:5]" and "STS1TXA_D_0[3:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". **TXHDLCDAT_4_0 (Transmit HDLC block data – Channel 0 – Input data pin 4)** **TXDS3OHIND_0 (Transmit DS3 Overhead Indicator – Channel 0)**

A11	STS1TXA_D5_0 TXHDLCDAT_5_0 TXDS3FP_0	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 5:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 5: STS1TXA_D5_0: This input pin along with "STS1TXA_D_0[7:6]" and "STS1TXA_D_0[4:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". **TXHDLCDAT_5_0** (Transmit HDLC block data – Channel 0 – Input data pin 5) TXDS3FP_0 (Transmit DS3 Frame Pulse – Channel 0) TXSBDATA_5_0
B12	STS1TXA_D6_0 TXHDLCDAT_6_0 TXDS3DATA_0 TXSBDATA_6_0	I	TTL	Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 6: The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. If STS-1 Telecom Bus (Channel 0) has been enabled – STS-1 Transmit Telecom Bus – Input Data Bus pin number 6: STS1TXA_D6_0: This input pin along with "STS1TXA_D7_0" and "STS1TXA_D_0[5:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". If STS-1 Telecom Bus (Channel 0) is disabled – TXHDLCDAT_6_0 (Transmit HDLC block data – Channel 0 – Input data pin 6) TXDS3DATA_0 (Transmit DS3 Data – Channel 0) TXSBDATA_6_0

A10	STS1TXA_D7_0 TXHDLCDAT_7_0 TXAISEN_0 TXSBDATA_7_0	I	TTL	Transmit STS-1 Telecom Bus – Channel 0 – Input Data Bus pin number 7: The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. If STS-1 Telecom Bus (Channel 0) has been enabled – STS-1 Transmit Telecom Bus – Input Data Bus pin number 7: STS1TXA_D7_0: This input pin along with "STS1TXA_D_0[6:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 0. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_0". ***Note:*** *This input pin functions as the MSB (Most Significant Bit) of the Transmit (Add) Telecom Bus, for Channel 0.* If STS-1 Telecom Bus (Channel 0) is disabled – TXHDLCDAT_7_0 **(Transmit HDLC block data – Channel 0 – Input data pin 7)** **TXAISEN_0 (Transmit AIS Enable – Channel 0)**
B23	STS1TXA_D0_1 TXHDLCDAT_0_1 TXGFCMSB_1	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 0:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 0: This input pin along with "STS1TXA_D_1[7:1]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. This particular input pin functions as the LSB (Least Significant Bit) input pin on the Transmit (Add) Telecom Bus – Input Data Bus. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". The LSB of any byte, which is being input into the "STS-1 Transmit Telecom Bus – Data Bus (for Channel 1) should be input via this pin. **TXHDLCDAT_0_1 (Transmit HDLC block data – Channel 1 – Input data pin 0)** **TXGFCMSB_1 (Transmit GFC MSB Indicator – Channel 1)**

C20	STS1TXA_D1_1 TXHDLCDAT_1_1 TXGFC_1	I	TTL	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 1:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 1: This input pin along with "STS1TXA_D_1[7:2]" and "STS1TXA_D0_1 function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **TXHDLCDAT_1_1 (Transmit HDLC block data – Channel 1 – Input data pin 1)** **TXGFC_1 (Transmit GFC data – Channel 1)**
B22	STS1TXA_D2_1 TXHDLCDAT_2_1 TXCELLTXED_1	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 2:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 2: STS1TXA_D2_1 This input pin along with "STS1TXA_D_1[7:3]" and "STS1TXA_D_1[1:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **TXHDLCDAT_2_1 (Transmit HDLC block data – Channel 1 – Input data pin 2)** **TXCELLTXED_1 (Cell Transmitted – Channel 1)**
E18	STS1TXA_D3_1 TXHDLCDAT_3_1 SSI_POS	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 3:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 3: STS1TXA_D3_1: This input pin along with "STS1TXA_D_1[7:4]" and "STS1TXA_D_1[2:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **TXHDLCDAT_3_1 (Transmit HDLC block data – Channel 1 – Input data pin 3)** **SSI_POS (Slow Speed Interface Data Positive for Ingress Path)**

A23	STS1TXA_D4_1 TXHDLCDAT_4_1 TXDS3OHIND_1	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 4:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 4: STS1TXA_D4_1: This input pin along with "STS1TXA_D_1[7:5]" and "STS1TXA_D_1[3:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **If STS-1 Telecom Bus (Channel 1) has NOT been enabled:** If STS-1 Telecom Bus (Channel 1) has not been enabled, then this particular pin can be configured to function in either of the following roles **TXHDLCDAT_4_1 (Transmit HDLC block data – Channel 1 – Input data pin 4)** This input pin will function as a part of the "Transmit HDLC Controller" byte-wide data input bus, whenever the user configures the DS3/E3 Framer block (associated with Channel 1) to operate in the "High-Speed HDLC Controller" Mode. This pin will function as Data Input Pin # 4. **TXDS3OHIND_1 (Transmit DS3 Overhead Indicator – Channel 1)** This output pin will pulse "high" one bit-period prior to the time that the DS3/E3 Frame Generator block (within Channel 1) will be processing an Overhead bit. The purpose of this outpout pin is to warn the Terminal Equipment that, during the very next bit-period, the DS3/E3 Frame Generator block is going to be processing an Overhead Bit and will be ignoring any data that is applied to to the TxSer input pin. *NOTE: The user can ignore this output pin provide that that either the Primary or Secondary Frame Synchronizer block is always "up-stream" from the DS3/E3 Frame Generator block.*

C19	STS1TXA_D5_1 TXHDLCDAT_5_1 TXDS3FP_1	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 5:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 5: STS1TXA_D5_1: This input pin along with "STS1TXA_D_1[7:6]" and "STS1TXA_D_1[4:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **TXHDLCDAT_5_1 (Transmit HDLC block data – Channel 1 – Input data pin 5)** **TXDS3FP_1 (Transmit DS3 Frame Pulse – Channel 1)**
D18	STS1TXA_D6_1 TXHDLCDAT_6_1 TXDS3DATA_1	I	TTL	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 6:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 6: STS1TXA_D6_1: This input pin along with "STS1TXA_D7_1" and "STS1TXA_D_1[5:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". **TXHDLCDAT_6_1 (Transmit HDLC block data – Channel 1 – Input data pin 6)** **TXDS3DATA_1 (Transmit DS3 Data – Channel 1)**

B21	STS1TXA_D7_1 TXHDLCDAT_7_1 TXAISEN_1	I	TTL	**Transmit STS-1 Telecom Bus – Channel 1 – Input Data Bus pin number 7:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 7: STS1TXA_D7_1: This input pin along with "STS1TXA_D_1[6:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 1. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_1". ***Note:*** *This input pin functions as the MSB (Most Significant Bit) of the Transmit (Add) Telecom Bus, for Channel 1.* **TXHDLCDAT_7_1 (Transmit HDLC block data – Channel 1 – Input data pin 7)** **TXAISEN_1 (Transmit AIS Enable – Channel 1)**
AE15	STS1TXA_D0_2 TXHDLCDAT_0_2 TXGFCMSB_2	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 0:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 0: This input pin along with "STS1TXA_D_2[7:1]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. This particular input pin functions as the LSB (Least Significant Bit) input pin on the Transmit (Add) Telecom Bus – Input Data Bus. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". The LSB of any byte, which is being input into the "STS-1 Transmit Telecom Bus – Data Bus (for Channel 2) should be input via this pin. **TXHDLCDAT_0_2 (Transmit HDLC block data – Channel 2 – Input data pin 0)** **TXGFCMSB_2 (Transmit GFC MSB Indicator – Channel 2)**

AD15	STS1TXA_D1_2 TXHDLCDAT_1_2 TXGFC_2	I	TTL	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 1:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 1: This input pin along with "STS1TXA_D_2[7:2]" and "STS1TXA_D0_2 function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_1_2 (Transmit HDLC block data – Channel 2 – Input data pin 1)** **TXGFC_2 (Transmit GFC data – Channel 2)**
AC15	STS1TXA_D2_2 TXHDLCDAT_2_2 TXCELLTXED_2	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 2:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled –** STS-1 Transmit Telecom Bus – Input Data Bus pin number 2: STS1TXA_D2_2 This input pin along with "STS1TXA_D_2[7:3]" and "STS1TXA_D_2[1:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_2_2 (Transmit HDLC block data – Channel 2 – Input data pin 2)** **TXCELLTXED_2 (Cell Transmitted – Channel 2)**
AG16	STS1TXA_D3_2 TXHDLCDAT_3_2 SSI_NEG	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 3:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 3: STS1TXA_D3_2: This input pin along with "STS1TXA_D_2[7:4]" and "STS1TXA_D_2[2:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_3_2 (Transmit HDLC block data – Channel 2 – Input data pin 3)** **SSI_NEG (Slow Speed Interface Data Negative for Ingress Path)**

AG17	STS1TXA_D4_2 TXHDLCDAT_4_2 TXDS3OHIND_2	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 4:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 4: STS1TXA_D4_2: This input pin along with "STS1TXA_D_2[7:5]" and "STS1TXA_D_2[3:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_4_2 (Transmit HDLC block data – Channel 2 – Input data pin 4)** **TXDS3OHIND_2 (Transmit DS3 Overhead Indicator – Channel 2)**
AF16	STS1TXA_D5_2 TXHDLCDAT_5_2 TXDS3FP_2	I/O	TTL/CMOS	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 5:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 5: STS1TXA_D5_2: This input pin along with "STS1TXA_D_2[7:6]" and "STS1TXA_D_2[4:0] function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_5_2 (Transmit HDLC block data – Channel 2 – Input data pin 5)** **TXDS3FP_2 (Transmit DS3 Frame Pulse – Channel 2)**
AG18	STS1TXA_D6_2 TXHDLCDAT_6_2 TXDS3DATA_2	I	TTL	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 6:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 6: STS1TXA_D6_2: This input pin along with "STS1TXA_D7_2" and "STS1TXA_D_2[5:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". **TXHDLCDAT_6_2 (Transmit HDLC block data – Channel 2 – Input data pin 6)** **TXDS3DATA_2 (Transmit DS3 Data – Channel 2)**

AE16	STS1TXA_D7_2 TXHDLCDAT_7_2 TXAISEN_2	I	TTL	**Transmit STS-1 Telecom Bus – Channel 2 – Input Data Bus pin number 7:** The exact function of this pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Transmit Telecom Bus – Input Data Bus pin number 7: STS1TXA_D7_2: This input pin along with "STS1TXA_D_2[6:0]" function as the "STS-1 Transmit (Add) Telecom Bus – Input Data Bus for Channel 2. The STS-1 Telecom Bus interface will sample and latch this pin upon the falling edge of "STS1TXA_CK_2". ***Note:*** *This input pin functions as the MSB (Most Significant Bit) of the Transmit (Add) Telecom Bus, for Channel 2.* **If STS-1 Telecom Bus (Channel 2) has NOT been enabled:** If STS-1 Telecom Bus (Channel 2) has not been enabled, then this particular pin can be configured to function in either of the following roles. **TXHDLCDAT_7_2 (Transmit HDLC block data – Channel 2 – Input data pin 7 – High Speed HDLC Controller Mode Only)** This input pin will function as a part of the "Transmit HDLC Controller" byte-wide data input bus, whenever the user configures the DS3/E3 Framer block (associated with Channel 2) to operate in the "High-Speed HDLC Controller" Mode. This pin will function as Data Input Pin # 2. **TXAISEN_2 (Transmit AIS Enable – Channel 2)** This input pin permits the user to command the DS3/E3 Frame Generator block (associated with Channel 2) to transmit the DS3/E3 AIS indicator. Pulling this input pin "high" configures the DS3/E3 Frame Generator block to generate and transmit the DS3/E3 AIS indicator. Pulling this input pin "low" configures the DS3/E3 Frame Generator block to transmit normal DS3/E3 data-streams. *NOTE: The user should pull this pin to "GND" for normal operation*
RECEIVE SYSTEM SIDE INTERFACE PINS				

B15 C23 AG13	RxOH_0 RxOH_1 RxOH_2	O	CMOS	**Receive Overhead Data Output Interface – output** This output pin functions as the "Receive Overhead Data" output for the receive system side interface when the XRT94L33 is configured to operate in DS3/E3 mode, however, it functions as the "Receive STS-1 Overhead Data" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** All overhead bits, which are received via the "Receive Section" of the channel, will be output via this output pin, upon the rising edge of "RxOHClk_n". **When configured to operate in STS-1 mode:** These output pins, along with "RxOHEnable_n", "RxOHClk_n" and "RxOHFrame_n" function as the "Receive STS-1 TOH and POH Output Port". Each bit, within the TOH and POH bytes (within the incoming STS-1 data stream) is updated upon the falling edge of "RxOHClk_n". As a consequence, external circuitry receiving this data, should sample this data upon the rising edge of "RxOHClk_n". ***Notes:*** *1. The external circuitry can determine whether or not it is receiving POH or TOH data via this output pin. The "RxOHEnable_n" output pin will be "high" anytime POH data is being output via these output pins. Conversely, the "RxOHEnable_n" output pin will be "low" anytime TOH data is being output via these output pins.* *2. TOH and POH data, associated with Receive STS-1 TOH and POH Processor Block – Channel 0 will be output via the "RxOH_0, and so on.*

C15 D21 AF13	RxOHENABLE_0 RxOHENABLE_1 RxOHENABLE_2	O	CMOS	**Receive Overhead Data Output Interface – Enable Output** This output pin functions as the "Receive Overhead Enable" output for the receive system side interface when the XRT94L33 is configured to operate in DS3/E3 mode, however, it functions as the "Receive STS-1 Overhead Data" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** The channel will assert this output signal for one "RxOHClk_n" period when it is safe for the local terminal equipment to sample the data on the "RxOH_n" output pin. **When configured to operate in STS-1 mode:** These output pins, along with "RxOHClk_n", "RxOHFrame_n" and "RxOH_n" function as the "Receive STS-1 TOH and POH Output Port". These output pins indicate whether POH or TOH data is being output via the "RxOH_n" output pins. These output pins will toggle "high" coincident with when POH data is being output via the "RxOH_n" output pins. Conversely, these output pins will toggle "low" coincident with when TOH data is being output via the "RxOH_n" output pins. These output pins are updated upon the falling edge of "RxOHClk_n". As a consequence, external circuitry, receiving this data, should sample this data upon the rising edge of "RxOHClk_n".
D15 E20 AE13	RxOHCLK_0 RxOHCLK_1 RxOHCLK_2	O	CMOS	**Receive Overhead Data Output Interface – clock** This output pin functions as the "Receive Overhead Clock" output for the receive system side interface when the XRT94L33 is configured to operate in DS3/E3 mode, however, it functions as the "Receive STS-1 Overhead Clock" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** The channel will output the overhead bits (within the incoming DS3 or E3 frames) via the RxOH_n output pin, upon the falling edge of this clock signal. As a consequence, the user's local terminal equipment should use the rising edge of this clock signal to sample the data on both the "RxOH" and "RxOHFrame" output pins. ***Note:*** *This clock signal is always active.* **When configured to operate in STS-1 mode:** These output pins, along with "RxOH_n", "RxOHFrame_n", and "RxOHEnable_n" function as the "Receive STS-1 TOH and POH Output Port". These output pins function as the "Clock Output" signals for the Receive STS-1 TOH and POH Output Port. The "RxOH_n", "RxSTS1Frame_n" and "RxOHEnable_n" output pins are updated upon the falling edge of this clock signal.

E15 D22 AD13	RxOHFRAME_0 RxOHFRAME_1 RxOHFRAME_2	O	CMOS	**Receive Overhead Data Interface – Framing Pulse indicator** This output pin functions as the "Receive Overhead Clock" output for the receive system side interface when the XRT94L33 is configured to operate in DS3/E3 mode, however, it functions as the "Receive STS-1 Overhead Clock" output when the device is configured to operate in the STS-1 mode. **When configured to operate in DS3/E3 mode:** This output pin pulses "high" whenever the Receive Overhead Data Output Interface block outputs the first overhead bit of a new DS3 or E3 frame. **When configured to operate in STS-1 mode:** These output pins, along with "RxOH_n", "RxOHEnable_n" and "RxOHClk_n" function as the "Receive STS-1 TOH and POH Output Port". These output pins will pulse "high" coincident with either of the following events. When the very first TOH byte (A1), of a given STS-1 frame, is being output via the corresponding "RxOH_n" output pin. When the very first POH byte (J1), of a given STS-1 frame, is being output via the corresponding "RxOH_n" output pin. The external circuitry can determine whether these output pins are pulsing high for the first TOH or POH byte by checking the state of the corresponding "RxOHEnable_n" output pin.
Y26	RxPERR	O	CMOS	**For mapper applications, Please let this pin "float".**
AB27	RxPEOP	O	CMOS	**For mapper applications, Please let this pin "float".**
AA26	RxPDVAL	O	CMOS	**For mapper applications, Please let this pin "float".**
V24	RxMOD_0	O	CMOS	**For mapper applications, Please let this pin "float".**
V25	RxUPRTY/ RxPPRTY	O	CMOS	**For mapper applications, Please let this pin "float".**

U23 W26 U24 AA27 Y27 U25 V26 W27 T23 T24 U26 T25 V27 T26 U27 T27	RxUDATA_0/ RxPDATA_0 RxUDATA_1/ RxPDATA_1 RxUDATA_2/ RxPDATA_2 RxUDATA_3/ RxPDATA_3 RxUDATA_4/ RxPDATA_4 RxUDATA_5/ RxPDATA_5 RxUDATA_6/ RxPDATA_6 RxUDATA_7/ RxPDATA_7 RxUDATA_8/ RxPDATA_8 RxUDATA_9/ RxPDATA_9 RxUDATA_10/ RxPDATA_10 RxUDATA_11/ RxPDATA_11 RxUDATA_12/ RxPDATA_12 RxUDATA_13/ RxPDATA_13 RxUDATA_14/ RxPDATA_14 RxUDATA_15/ RxPDATA_15	O	CMOS	**For mapper applications, Please let these pins "float".**
R23 R24 R25 R26 R27	RxUADDR_0 RxUADDR_1 RxUADDR_2 RxUADDR_3 RxUADDR_4	I	TTL	**For mapper applications, Please connect these pins to GND**
P27	RxUClav/ RxPPA	O	CMOS	**For mapper applications, Please let this pin "float".**
P25	RxUSOC/ RxPSOP/ RxPSOC	O	CMOS	**For mapper applications, Please let this pin "float".**
P23	RxTSX/ RXPSOF	O	CMOS	**For mapper applications, Please let this pin "float".**
P24	RXUENB_L/ RXPENB_L	I	TTL	**For mapper applications, Please connect this pin to VDD**
P26	RXUCLKO/ RXPCLKO	O	CMOS	**For mapper applications, Please let this pin "float".**

N27	RXUCLK/ RXPCLK	I	TTL	**For mapper applications, Please connect this pin to GND**
A16 J23 AC13	EXTLOS_0 EXTLOS_1 EXTLOS_2	I	TTL	**Receive LOS (Loss of Signal) Indicator Input (from the XRT94L33 DS3/E3/STS-1 LIU IC):** This input pin, is intended to be connected to each of the RLOS (Receive Loss of Signal) output pins of the XRT94L33 DS3/E3/STS-1 LIU IC. The user can monitor the state of this input pin by reading the state of Bit 0 (RLOS) within the Line Interface Scan Register (Address = 0xXX, 0xXX). If this input pin is "Low", then it means that the corresponding channel (within the XRT94L33) is currently NOT declaring an LOS condition. However, if this input pin is "high", then it means that this particular channel is currently declaring an LOS condition. ***Note:*** *Asserting this input pin will cause the XRT94L33 Framer/UNI IC to declare an "LOS (Loss of Signal) condition. Therefore, this input pin should not be used as a General Purpose Input pin.*
A14 D20 AE14	RxOOF_0 RxOOF_1 RxOOF_2	O	CMOS	**Receive STS-1/DS3/E3 Out of Frame Indicator** The STS-1/DS3/E3 Receive DS3 Framer will assert this output signal whenever it has declared an "Out of Frame" (OOF) condition with the incoming DS3 frames. This signal is negated when the framer correctly locates the F- and M-bits and regains synchronization with the DS3 frame.
A15 B24 AG14	RxLOS_0 RxLOS_1 RxLOS_2	O	CMOS	**STS-1/DS3/E3 Framer - Loss of Signal Output Indicator:** This pin is asserted when the Receive Section of the channel encounters 180 consecutive 0's (for DS3 applications) or 32 consecutive 0's (for E3 applications) via the RxPOS_n and RxNEG pins. For STS-1 applications, users can set the LOS threshold value in the Receive LOS Threshold register. (RxSTOH_LOS_TH, Address Location: 0xN02E – 0xN02F) This pin will be negated once the Receive DS3/E3 Framer has detected at least 60 "1s" out of 180 consecutive bits (for DS3 applications) or has detected at least four consecutive 32 bit strings of data that contain at least 8 "1s" in the receive path.

STS-1 TELECOM BUS INTERFACE – RECEIVE DIRECTION				
A21	STS1RXD_CK_0 RXVALIDFCS_0 RXGFCCLK_0	O	CMOS	**Receive STS-1/STS-3 Telecom Bus Clock Output – Channel 0;** The exact function of this input pin depends upon whether the "STS-1 Telecom Bus Interface associated with Channel 0" is enabled or not, as described below. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus Clock Output – Channel 0; STS1RXD_CK_0: All signals, which are output via the "Receive Telecom Bus – Channel 0" are clocked out upon the rising edge of this clock signal. This includes the following signals. • STS1RXD_D_0[7:0] • STS1RXD_ALARM_0 • STS1RXD_DP_0 • STS1RXD_PL_0 • STS1RXD_C1J1_0 This clock signal will operate at 19.44MHz (For STS-3 mode) or 6.48MHz (Fro STS-1 mode) **RXVALIDFCS_0 (Receive HDLC block valid FCS Indicator – Channel 0)** **RXGFCCLK_0 (Receive ATM GFC clock signal – Channel 0)**

H24	STS1RXD_CK_1 RXVALIDFCS_1 RXGFCCLK_1 TxP_STPA	O	CMOS	**Receive STS-1 Telecom Bus Clock Output – Channel 1;** The exact function of this input pin depends upon whether the "STS-1 Telecom Bus Interface associated with Channel 1" is enabled or not, as described below. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus Clock Output – Channel 1; STS1RXD_CK_1: All signals, which are output via the "Receive Telecom Bus – Channel 1" are clocked out upon the rising edge of this clock signal. This includes the following signals. • STS1RXD_D_1[7:0] • STS1RXD_ALARM_1 • STS1RXD_DP_1 • STS1RXD_PL_1 • STS1RXD_C1J1_1 This clock signal will operate at 19.44MHz. (For STS-3 mode) or 6.48MHz (Fro STS-1 mode) **RXVALIDFCS_1 (Receive HDLC block valid FCS Indicator – Channel 1)** **RXGFCCLK_1 (Receive ATM GFC clock signal – Channel 1)** **TxP_STPA (Transmit PPP Level 2 Selected Channel Packet Available)**
AG8	STS1RXD_CK_2 RXVALIDFCS_2 RXGFCCLK_2	O	CMOS	**Receive STS-1 Telecom Bus Clock Output – Channel 2;** The exact function of this input pin depends upon whether the "STS-1 Telecom Bus Interface associated with Channel 2" is enabled or not, as described below. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus Clock Output – Channel 2; STS1RXD_CK_2: All signals, which are output via the "Receive Telecom Bus – Channel 2" are clocked out upon the rising edge of this clock signal. This includes the following signals. • STS1RXD_D_2[7:0] • STS1RXD_ALARM_2 • STS1RXD_DP_2 • STS1RXD_PL_2 • STS1RXD_C1J1_2 This clock signal will operate at 19.44MHz. (For STS-3 mode) or 6.48MHz (Fro STS-1 mode) **RXVALIDFCS_2 (Receive HDLC block valid FCS Indicator – Channel 2)** **RXGFCCLK_2 (Receive ATM GFC clock signal – Channel 2)**

A20	STS1RXD_PL_0 RXIDLE_0 RXLCD_0	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output Signal – Channel 0:** The exact function of this output pin depends upon whether the user has enabled or disabled the "STS-1 Telecom Bus Interface block" associated with Channel 0. **If the STS-1 Telecom Bus Interface (associated with Channel 0) is enabled** – STS-1/STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output – STS1RXD_PL_0: This output pin indicates whether or not Transport Overhead bytes are being output via the "STS1RXD_D_0[7:0]" output pins. This output pin is pulled "low" for the duration that the STS-1 Receive Telecom Bus is transmitting a Transport Overhead byte via the "STS1RXD_D_0[7:0]" output pins. Conversely, this output pin is pulled "high" for the duration that the STS-1 Receive Telecom Bus is transmitting something other than a Transport Overhead byte via the "STS1RXD_D_0[7:0]" output pins. **RXIDLE_0 (Receive HDLC block idle indicator – Channel 0)** **RXLCD_0 (Receive Cell Processor Loss of Cell Delineation – Channel 0)**
D26	STS1RXD_PL_1 RXIDLE_1 RXLCD_1	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output Signal – Channel 1:** The exact function of this output pin depends upon whether the user has enabled or disabled the "STS-1 Telecom Bus Interface block" associated with Channel 1. **If the STS-1 Telecom Bus Interface (associated with Channel 1) is enabled –** STS-1/STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output – STS1RXD_PL_1: This output pin indicates whether or not Transport Overhead bytes are being output via the "STS1RXD_D_1[7:0]" output pins. This output pin is pulled "low" for the duration that the STS-1 Receive Telecom Bus is transmitting a Transport Overhead byte via the "STS1RXD_D_1[7:0]" output pins. Conversely, this output pin is pulled "high" for the duration that the STS-1 Receive Telecom Bus is transmitting something other than a Transport Overhead byte via the "STS1RXD_D_1[7:0]" output pins. **RXIDLE_1 (Receive HDLC block idle indicator – Channel 1)** **RXLCD_1 (Receive Cell Processor Loss of Cell Delineation – Channel 1)**

AE11	STS1RXD_PL_2 RXIDLE_2 RXLCD_2	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output Signal – Channel 2:** The exact function of this output pin depends upon whether the user has enabled or disabled the "STS-1 Telecom Bus Interface block" associated with Channel 2. **If the STS-1 Telecom Bus Interface (associated with Channel 2) is enabled** – STS-1/STS-1 Receive (Drop) Telecom Bus – Payload Indicator Output – STS1RXD_PL_2: This output pin indicates whether or not Transport Overhead bytes are being output via the "STS1RXD_D_2[7:0]" output pins. This output pin is pulled "low" for the duration that the STS-1 Receive Telecom Bus is transmitting a Transport Overhead byte via the "STS1RXD_D_2[7:0]" output pins. Conversely, this output pin is pulled "high" for the duration that the STS-1 Receive Telecom Bus is transmitting something other than a Transport Overhead byte via the "STS1RXD_D_2[7:0]" output pins. **RXIDLE_2 (Receive HDLC block idle indicator – Channel 2)** **RXLCD_2 (Receive Cell Processor Loss of Cell Delineation – Channel 2)**

C17	STS1RXD_C1J1_0 TXDS3LINECLK_0	O	CMOS	**STS-1 Receive Telecom Bus - C1/J1 Byte Phase Indicator Output Signal – Channel 0:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – C1/J1 Byte Phase Indicator Output Signal: This output pin pulses "high" under the following two conditions. Whenever the C1 byte is being output via the "STS1RXD_D_0[7:0]" output, and Whenever the J1 byte is being output via the "STS1RXD_D_0[7:0]" output. ***Notes:*** *1. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 0) will indicate that it is transmitting the C1 byte (via the STS1RXD_D_0[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "STS1RXD_CK_0") and keeping the "STS1RXD_PL_0" output pin pulled "LOW.* *2. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 0) will indicate that it is transmitting the J1 byte (via the STS1RXD_D_0[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "STS1RXD_CK_0") while the "STS1TXD_PL_0" output pin is pulled "HIGH".* **TXDS3LINECLK_0 (Transmit DS3/E3/STS-1 line clock to LIU – Channel 0)**

E25	STS1RXD_C1J1_1 TXDS3LINECLK_1	O	CMOS	**STS-1 Receive Telecom Bus - C1/J1 Byte Phase Indicator Output Signal – Channel 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – C1/J1 Byte Phase Indicator Output Signal: This output pin pulses “high” under the following two conditions. Whenever the C1 byte is being output via the “STS1RXD_D_1[7:0]” output, and Whenever the J1 byte is being output via the “STS1RXD_D_1[7:0]” output. ***Notes:*** *1. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 1) will indicate that it is transmitting the C1 byte (via the STS1RXD_D_1[7:0] output pins), by pulsing this output pin “HIGH” (for one period of “STS1RXD_CK_1”) and keeping the “STS1RXD_PL_1” output pin pulled “LOW.* *2. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 1) will indicate that it is transmitting the J1 byte (via the STS1RXD_D_1[7:0] output pins), by pulsing this output pin “HIGH” (for one period of “STS1RXD_CK_1”) while the “STS1TXD_PL_1” output pin is pulled “HIGH”.* **TXDS3LINECLK_1 (Transmit DS3/E3/STS-1 line clock to LIU – Channel 1)**

AF10	STS1RXD_C1J1_2 TXDS3LINECLK_2	O	CMOS	**STS-1 Receive Telecom Bus - C1/J1 Byte Phase Indicator Output Signal – Channel 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – C1/J1 Byte Phase Indicator Output Signal: This output pin pulses "high" under the following two conditions. Whenever the C1 byte is being output via the "STS1RXD_D_2[7:0]" output, and Whenever the J1 byte is being output via the "STS1RXD_D_2[7:0]" output. ***Notes:*** *1. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 2) will indicate that it is transmitting the C1 byte (via the STS1RXD_D_2[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "STS1RXD_CK_2") and keeping the "STS1RXD_PL_2" output pin pulled "LOW.* *2. The STS-1 Receive (Drop) Telecom Bus (associated with Channel 2) will indicate that it is transmitting the J1 byte (via the STS1RXD_D_2[7:0] output pins), by pulsing this output pin "HIGH" (for one period of "STS1RXD_CK_2") while the "STS1TXD_PL_2" output pin is pulled "HIGH".* **TXDS3LINECLK_2 (Transmit DS3/E3/STS-1 line clock to LIU – Channel 2)**

B18	STS1RXD_DP_0 TXDS3POS_0	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Parity Output pin – Channel 0:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Parity Output pin: This output pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits which are output via the "STS1RXD_D_0[7:0]" output pins. The EVEN or ODD parity value of the bits which are being output via the "STS1RXD_D_0[7:0]" output pins and the states of the "STS1RXD_PL_0" and "STS1RXD_C1J1_0" output pins. This output pin will ultimately be used (by "drop-side" circuitry) to verify the verify of the data which is output via the "STS-1 Telecom Bus Interface associated with Channel 0. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Telecom Bus Control" Register (Address Location = 0x013B).* **TXDS3POS_0 (Transmit DS3/E3/STS-1 line data positive to LIU– Channel 0)**
G24	STS1RXD_DP_1 TXDS3POS_1	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Parity Output pin – Channel 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Parity Output pin: This output pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits output via the "STS1RXD_D_1[7:0]" output pins. The EVEN or ODD parity value of the bits being output via the "STS1RXD_D_1[7:0]" output pins and the states of the "STS1RXD_PL_1" and "STS1RXD_C1J1_1" output pins. This output pin will ultimately be used (by "drop-side" circuitry) to verify of the data which is output via the "STS-1 Telecom Bus Interface associated with Channel 1. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Telecom Bus Control" Register (Address Location = 0x013A).* **TXDS3POS_1 (Transmit DS3/E3/STS-1 line data positive to LIU– Channel 1)**

AG9	STS1RXD_DP_2 TXDS3POS_2	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Parity Output pin – Channel 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Parity Output pin: This output pin can be configured to function as one of the following. The EVEN or ODD parity value of the bits output via the "STS1RXD_D_2[7:0]" output pins. The EVEN or ODD parity value of the bits being output via the "STS1RXD_D_2[7:0]" output pins and the states of the "STS1RXD_PL_2" and "STS1RXD_C1J1_2" output pins. This output pin will ultimately be used (by "drop-side" circuitry) to verify the verify of the data which is output via the "STS-1 Telecom Bus Interface associated with Channel 2. ***Note:*** *The user can make any one of these configuration selections by writing the appropriate value into the "Telecom Bus Control" Register (Address Location = 0x0139).* **TXDS3POS_2 (Transmit DS3/E3/STS-1 line data positive to LIU– Channel 2)**
A19	STS1RXD_ALARM_0 TXDS3NEG_0/	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Alarm Indicator Output signal – Channel 0:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 0 has been enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Alarm Indicator Output signal: This output pin pulses "high", coincident with any STS-1 signal (that is being output via the "STS1RXD_D_0[7:0]" output pins) that is carrying an AIS-P indicator. This output pin is "low" for all other conditions. **TXDS3NEG_0 (Transmit DS3/E3 line data negative to LIU – Channel 0)**

H23	STS1RXD_ALARM_1 TXDS3NEG_1/	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Alarm Indicator Output signal – Channel 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 1 has been enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Alarm Indicator Output signal: This output pin pulses "high", coincident with any STS-1 signal (that is being output via the "STS1RXD_D_1[7:0]" output pins) that is carrying an AIS-P indicator. This output pin is "low" for all other conditions. **TXDS3NEG_1 (Transmit DS3/E3 line data negative to LIU – Channel 1)**
AB12	STS1RXD_ALARM_2 TXDS3NEG_2/	O	CMOS	**STS-1 Receive (Drop) Telecom Bus – Alarm Indicator Output signal – Channel 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface for Channel 2 has been enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Alarm Indicator Output signal: This output pin pulses "high", coincident with any STS-1 signal (that is being output via the "STS1RXD_D_2[7:0]" output pins) that is carrying an AIS-P indicator. This output pin is "low" for all other conditions. **TXDS3NEG_2 (Transmit DS3/E3 line data negative to LIU – Channel 2)**
F16	STS1RXD_D0_0 RXHDLCDAT_0_0 RXGFCMSB_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 0:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 0: STS1RXD_D0_0 This output pin along with "STS1RXD_D_0[7:1]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. ***Note:*** *This input pin functions as the LSB (Least Significant Bit) of the Receive (Drop) Telecom Bus for Channel 0.* **RXHDLCDAT_0_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 0)** **RXGFCMSB_0 (Receive GFC MSB Indicator – Channel 0)**

E16	STS1RXD_D1_0 RXHDLCDAT_1_0 RXGFC_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 1: STS1RXD_D1_0 This output pin along with "STS1RXD_D_0[7:2]" and "STS1RXD_D0_0 function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. **RXHDLCDAT_1_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 1):** **RXGFC_0 (Receive GFC output data – Channel 0)**
D16	STS1RXD_D2_0 RXHDLCDAT_2_0 RXCELLRXED_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled –** STS-1 Receive Telecom Bus – Output Data bus pin number 2: STS1RXD_D2_0 This output pin along with "STS1RxD_D_0[7:3]" and "STS1RxD_D_0[1:0]" function as the "STS-3/STM-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-3/STM-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RxD_CK_0:. **RXHDLCDAT_2_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 2)** **RXCELLRXED_0 (Receive cell received indicator – Channel 0)**

B17	STS1RXD_D3_0 RXHDLCDAT_3_0 SSE_CLK	O O IO O	CMOS CMOS TTL/CMOS CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 3:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 3: STS1RXD_D3_0 This output pin along with "STS1RXD_D_0[7:4]" and "STS1RXD_D_0[2:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. **RXHDLCDAT_3_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 3)** **SSE_CLK (Slow Speed Clock Interface for Egress Path)**
C16	STS1RXD_D4_0 RXHDLCDAT_4_0 RXOHIND_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 4:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 4: STS1RXD_D4_0 This output pin along with "STS1RXD_D_0[7:5]" and "STS1RXD_D_0[3:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. **RXHDLCDAT_4_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 4)** **RXOHIND_0 (Receive Overhead Indicator – Channel 0)**
A18	STS1RXD_D5_0 RXHDLCDAT_5_0 RXDS3FP_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 5:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 5: STS1RXD_D5_0 This output pin along with "STS1RXD_D_0[7:6]" and "STS1RXD_D_0[4:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. **RXHDLCDAT_5_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 5)** **RXDS3FP_0 (Receive DS3 frame pulse – Channel 0)**

B16	STS1RXD_D6_0 RXHDLCDAT_6_0 RXDS3DATA_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 6:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 6: STS1RXD_D6_0 This output pin along with "STS1RXD_D7_0" and "STS1RXD_D_0[5:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. **RXHDLCDAT_6_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 6)** **RXDS3DATA_0 (Receive DS3 data – Channel 0)**
A17	STS1RXD_D7_0 RXHDLCDAT_7_0 RXDS3CLK_0	O	CMOS	**Receive STS-1 Telecom Bus – Channel 0 – Output Data Bus pin number 7:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 0 is enabled or not. **If STS-1 Telecom Bus (Channel 0) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 7: STS1RXD_D7_0 This output pin along with "STS1RXD_D_0[6:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 0. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_0:. ***Note:*** *This output pin functions as the MSB (Most Significant Bit) for the STS-1 Receive (Drop) Telecom Bus Interface – Output Data Bus (Channel 0).* **RXHDLCDAT_7_0 (Receive HDLC block data output – Channel 0 – Output Data Bus pin 7)** **RXDS3CLK_0 (Receive DS3 clock – Channel 0)**

F24	STS1RXD_D0_1 RXHDLCDAT_0_1 RXGFCMSB_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 0**: The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 0: STS1RXD_D0_1 This output pin along with "STS1RXD_D_1[7:1]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. ***Note:*** *This input pin functions as the LSB (Least Significant Bit) of the Receive (Drop) Telecom Bus for Channel 1.* **RXHDLCDAT_0_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 0)** **RXGFCMSB_1 (Receive GFC MSB Indicator – Channel 1)**
H22	STS1RXD_D1_1 RXHDLCDAT_1_1 RXGFC_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 1: STS1RXD_D1_1 This output pin along with "STS1RXD_D_1[7:2]" and "STS1RXD_D0_1 function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_1_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 1)** **RXGFC_1 (Receive GFC output data – Channel 1)**

D25	STS1RXD_D2_1 RXHDLCDAT_2_1 RXCELLRXED_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 2: STS1RXD_D2_1 This output pin along with "STS1RXD_D_1[7:3]" and "STS1RXD_D_1[1:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_2_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 2)** **RXCELLRXED_1 (Receive cell received indicator – Channel 1)**
G23	STS1RXD_D3_1 RXHDLCDAT_3_1 SSE_POS	O O IO O	CMOS CMOS TTL/CMOS CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 3:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 3: STS1RXD_D3_1 This output pin along with "STS1RXD_D_1[7:4]" and "STS1RXD_D_1[2:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_3_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 3)** **SSE_POS (Slow Speed Interface Data Positive for Egress Path)**

D23	STS1RXD_D4_1 RXHDLCDAT_4_1 RXOHIND_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 4:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 4: STS1RXD_D4_1 This output pin along with "STS1RXD_D_1[7:5]" and "STS1RXD_D_1[3:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_4_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 4)** **RXOHIND_1 (Receive Overhead Indicator – Channel 1)**
E21	STS1RXD_D5_1 RXHDLCDAT_5_1 RXDS3FP_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 5:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 5: STS1RXD_D5_1 This output pin along with "STS1RXD_D_1[7:6]" and "STS1RXD_D_1[4:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_5_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 5)** **RXDS3FP_1 (Receive DS3 frame pulse – Channel 1)**
C24	STS1RXD_D6_1 RXHDLCDAT_6_1 RXDS3DATA_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 6:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 6: STS1RXD_D6_1 This output pin along with "STS1RXD_D7_1" and "STS1RXD_D_1[5:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. **RXHDLCDAT_6_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 6):** **RXDS3DATA_1 (Receive DS3 data – Channel 1):**

F20	STS1RXD_D7_1 RXHDLCDAT_7_1 RXDS3CLK_1	O	CMOS	**Receive STS-1 Telecom Bus – Channel 1 – Output Data Bus pin number 7:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 1 is enabled or not. **If STS-1 Telecom Bus (Channel 1) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 7: STS1RXD_D7_1 This output pin along with "STS1RXD_D_1[6:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 1. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_1. ***Note:*** *This output pin functions as the MSB (Most Significant Bit) for the STS-1 Receive (Drop) Telecom Bus Interface – Output Data Bus (Channel 1).* **RXHDLCDAT_7_1 (Receive HDLC block data output – Channel 1 – Output Data Bus pin 7)** **RXDS3CLK_1 (Receive DS3 clock – Channel 1)**
AC12	STS1RXD_D0_2 RXHDLCDAT_0_2 RXGFCMSB_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 0:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 0: STS1RXD_D0_2 This output pin along with "STS1RXD_D_2[7:1]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. ***Note:*** *This input pin functions as the LSB (Least Significant Bit) of the Receive (Drop) Telecom Bus for Channel 2.* **RXHDLCDAT_0_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 0)** **RXGFCMSB_2 (Receive GFC MSB Indicator – Channel 2)**

AD12	STS1RXD_D1_2 RXHDLCDAT_1_2 RXGFC_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 1:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 1: STS1RXD_D1_2 This output pin along with "STS1RXD_D_2[7:2]" and "STS1RXD_D0_2 function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_1_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 1)** **RXGFC_2 (Receive GFC output data – Channel 2)**
AF11	STS1RXD_D2_2 RXHDLCDAT_2_2 RXCELLRXED_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 2:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 2: STS1RXD_D2_2 This output pin along with "STS1RXD_D_2[7:3]" and "STS1RXD_D_2[1:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_2_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 2)** **RXCELLRXED_2 (Receive cell received indicator – Channel 2)**

AE12	STS1RXD_D3_2 RXHDLCDAT_3_2 SSE_NEG	O O IO O	CMOS CMOS TTL/CMOS CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 3:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 3: STS1RXD_D3_2 This output pin along with "STS1RXD_D_2[7:4]" and "STS1RXD_D_2[2:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_3_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 3)** **SSE_NEG (Slow Speed Interface Data Negative for Egress Path)**
AG10	STS1RXD_D4_2 RXHDLCDAT_4_2 RXOHIND_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 4:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 4: STS1RXD_D4_2 This output pin along with "STS1RXD_D_2[7:5]" and "STS1RXD_D_2[3:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_4_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 4)** **RXOHIND_2 (Receive Overhead Indicator – Channel 2)**

AF12	STS1RXD_D5_2 RXHDLCDAT_5_2 RXDS3FP_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 5:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 5: STS1RXD_D5_2 This output pin along with "STS1RXD_D_2[7:6]" and "STS1RXD_D_2[4:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_5_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin5):** This output pin along with RxHDLCDat_ **RXDS3FP_2 (Receive DS3 frame pulse – Channel 2)**
AG11	STS1RXD_D6_2 RXHDLCDAT_6_2 RXDS3DATA_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 6:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 6: STS1RXD_D6_2 This output pin along with "STS1RXD_D7_2" and "STS1RXD_D_2[5:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. **RXHDLCDAT_6_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 6)** **RXDS3DATA_2 (Receive DS3 data – Channel 2)**

AG12	STS1RXD_D7_2 RXHDLCDAT_7_2 RXDS3CLK_2	O	CMOS	**Receive STS-1 Telecom Bus – Channel 2 – Output Data Bus pin number 7:** The exact function of this output pin depends upon whether the STS-1 Telecom Bus Interface, associated with Channel 2 is enabled or not. **If STS-1 Telecom Bus (Channel 2) has been enabled** – STS-1 Receive Telecom Bus – Output Data bus pin number 7: STS1RXD_D7_2 This output pin along with "STS1RXD_D_2[6:0]" function as the "STS-1 Receive (Drop) Telecom Bus – Output Data Bus for Channel 2. The STS-1 Telecom Bus Interface will update the data via this output upon the rising edge of "STS1RXD_CK_2. ***Note:*** *This output pin functions as the MSB (Most Significant Bit) for the STS-1 Receive (Drop) Telecom Bus Interface – Output Data Bus (Channel 2).* **RXHDLCDAT_7_2 (Receive HDLC block data output – Channel 2 – Output Data Bus pin 7)** **RXDS3CLK_2 (Receive DS3 clock – Channel 2)**

RECEIVE TRANSPORT OVERHEAD INTERFACE				
AD5	RxTOHClk	O	CMOS	**Receive TOH Output Port – Clock Output:** This output pin, along with "RxTOH, RxTOHValid and "RxTOHFrame" function as the "Receive TOH Output Port". The Receive TOH Output Port permits the user to obtain the value of the TOH Bytes, within the incoming STS-3/STM-1 signal. This output pin provides the user with a clock signal. If the "RxTOHValid" output pin is "HIGH", then the contents of the "TOH" bytes, within the incoming STS-3 data-stream will be serially output via the "RxTOH" output. This data will be updated upon the falling edge of this clock signal. Therefore, the user is advised to sample the data (at the "RxTOH" output pin) upon the rising edge of this clock output signal.
AC7	RxTOHValid	O	CMOS	**Receive TOH Output Port – TOH Valid (or READY) indicator:** This output pin, along with "RxTOH" and "RxTOHFrame" function as the "Receive TOH Output Port". This output pin will toggle "HIGH" whenever valid "TOH" data is being output via the "RxTOH" output pin.
AE4	RxTOH	O	CMOS	**Receive TOH Output port – Output pin:** This output pin, along with "RxTOHClk", RxTOHValid" and "RxTOHFrame" function as the "Receive TOH Output port. All TOH data that resides within the incoming STS-3 data-stream will be output via this output pin. The "RxTOHValid" output pin will toggle high, coincident with anytime a bit (from the Receive STS-3 TOH data) is being output via this output pin. The "RxTOHFrame" output pin will pulse "high" (for eight periods of "RxTOHClk") coincident to when the A1 byte is being output via this output pin. Data, on this output pin, is updated upon the falling edge of "RxTOHClk".
AB8	RxTOHFrame	O	CMOS	**Receive TOH Output Port – STS-3/STM-1 Frame Indicator:** This output pin, along with the "RxTOHClk", "RxTOHValid" and "RxTOH" output pins function as the "Receive TOH Output port". This output pin will pulse "high", for one period of "RxTOHClk", one "RxTOHClk" period prior to the very first "TOH" bit (of a given STS-3 frame) being output via the "RxTOH" output pin.

AD7	RxLDCCVAL	O	CMOS	**Receive – Line DCC Output Port – DCC Value Indicator Output pin:** This output pin, along with the "RxTOHClk" and the "RxLDCC" output pins function as the "Receive Line DCC" output port of the XRT94L33. This output pin pulses "High" coincident to when the "Receive Line DCC" output port outputs a DCC bit via the "RxLDCC" output pin. This output pin is updated upon the falling edge of "RxTOHClk". The Line DCC HDLC Controller circuitry that is interfaced to this output pin, the "RxLDCC" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of this output pin upon the rising edge of "RxTOHClk". Anytime the "Line DCC HDLC" circuitry samples this output pin being "HIGH", it should sample and latch the data on the "RxLDCC" output pin (as a valid Line DCC bit) into the "Line DCC HDLC" circuitry.
AE5	RxLDCC	O	CMOS	**Receive – Line DCC Output Port – Output Pin:** This output pin, along with "RxLDCCVAL" and the "RxTOHClk" output pins function as the "Receive Line DCC" output port of the XRT94L33. This pin outputs the contents of the Line DCC (e.g., the D4, D5, D6, D7, D8, D9, D10, D11 and D12 bytes), within the incoming STS-3 data-stream. The Receive Line DCC Output port will assert the "RxLDCCVAL" output pin, in order to indicate that the data, residing on the "RxLDCC" output pin is a valid Line DCC byte. The Receive Line DCC output port will update the "RxLDCCVAL" and the "RxLDCC" output pins upon the falling edge of the "RxTOHClk" output pin. The Line DCC HDLC circuitry that is interfaced to this output pin, the "RxLDCCVAL" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of the "RxLDCCVAL" output pin upon the rising edge of "RxTOHClk". Anytime the "Line DCC HDLC" circuitry samples the "RxLDCCVAL" output pin "HIGH", it should sample and latch the contents of this output pin (as a valid Line DCC bit) into the "Line DCC HDLC" circuitry.
AD8	RxE1F1E2FP	O	CMOS	**Receive – Order-Wire Output Port – Frame Boundary Indicator:** This output pin, along with "RxE1F1E2", "RxE1F1E2Val" and the "RxTOHClk" output pins function as the "Receive Order-Wire Output port of the XRT94L33. This output pin pulses "high" (for one period of "RxTOHClk") coincident to when the very first bit (of the E1 byte) is being output vi the "RxE1F1E2" output pin.

AC9	RxE1F1E2	O	CMOS	**Receive – Order-Wire Output Port – Output Pin:** This output pin, along with "RxE1F1E2Val", "RxE1F1F2FP, and the "RxTOHClk" output pins function as the "Receive Order-Wire Output Port of the XRT94L33. This pin outputs the contents of the "Order-Wire" bytes (e.g., the E1, F1 and E2 bytes) within the incoming STS-3 data-stream. The Receive Order-Wire Output port will pulse the "RxE1F1E2FP" output pin "high" (for one period of "RxTOHClk") coincident to when the very first bit (of the E1 byte) is being output via the "RxE1F1E2" output pin. Additionally, the Receive Order-Wire Output port will also assert the "RxE1F1E2Val" output pin, in order to indicate that the data, residing on the "RxE1F1E2" output pin is valid "Order-Wire" byte. The Receive Order-Wire output port will update the "RxE1F1E2Val", the "RxE1F1E2FP" and the "RxE1F1E2" output pins upon the falling edge of the "RxTOHClk" output pin. The "Receive Order-Wire" circuitry that is interfaced to this output pin, and the "RxE1F1E2Val", the "RxE1F1E2" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of the "RxE1F1E2Val" and "RxE1F1E2FP" output pins upon the rising edge of "RxTOHClk". Anytime the "Order-wire" circuitry samples the "RxE1F1E2Val" and "RxE1F1E2FP output pins "HIGH, it should begin to sample and latch the contents of this output pin (as a valid "Order-Wire" bit) into the "Order-Wire" circuitry. The "Order-Wire" circuitry should continue to sample and latch the contents of the output pin until the "RxE1F2E2Val" output pin is sampled "low".
AC8	RxSDCC	O	CMOS	**Receive – Section DCC Output Port – Output Pin:** This output pin, along with "RxSDCCVAL" and the "RxTOHClk" output pins function as the "Receive Section DCC" output port of the XRT94L33. This pin outputs the contents of the Section DCC (e.g., the D1, D2 and D3 bytes), within the incoming STS-3 data-stream. The Receive Section DCC Output port will assert the "RxSDCCVAL" output pin, in order to indicate that the data, residing on the "RxSDCC" output pin is a valid Section DCC byte. The Receive Section DCC output port will update the "RxSDCCVAL" and the "RxSDCC" output pins upon the falling edge of the "RxTOHClk" output pin. The Section DCC HDLC circuitry that is interfaced to this output pin, the "RxSDCCVAL" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of the "RxSDCCVAL" output pin upon the rising edge of "RxTOHClk". Anytime the "Section DCC HDLC" circuitry samples the "RxSDCCVAL" output pin "HIGH", it should sample and latch the contents of this output pin (as a valid Section DCC bit) into the "Section DCC HDLC" circuitry.

AD6	RxSDCCVAL	O	CMOS	**Receive – Section DCC Output Port – DCC Value Indicator Output pin:** This output pin, along with the "RxTOHClk" and the "RxSDCC" output pins function as the "Receive Section DCC" output port of the XRT94L33. This output pin pulses "High" coincident to when the "Receive Section DCC" output port outputs a DCC bit via the "RxSDCC" output pin. This output pin is updated upon the falling edge of "RxTOHClk". The Section DCC HDLC Controller circuitry that is interfaced to this output pin, the "RxSDCC" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of this output pin upon the rising edge of "RxTOHClk". Anytime the "Section DCC HDLC" circuitry samples this output pin being "HIGH", it should sample and latch the data on the "RxSDCC" output pin (as a valid Section DCC bit) into the "Section DCC HDLC" circuitry.
AF4	RxE1F1E2VAL	O	CMOS	**Receive – Order Wire Output Port – E1F1E2 Value Indicator Output Pin:** This output pin, along with the "RxTOHClk", "RxE1F1E2FP", "RxE1F1E2" and "RxTOHClk" output pins function as the "Receive – Order Wire Output Port" of the XRT94L33. This output pin pulses "high" coincident to when the "Receive – Order Wire" output port outputs the contents of an E1, F1 or E2 byte, via the "RxE1F1E2" output pin. This output pin is updated upon the falling edge of "RxTOHClk". The "Receive Order-Wire" circuitry, that is interfaced to this output pin, the "RxE1F1E2" and the "RxTOHClk" pins is suppose to do the following. It should continuously sample and monitor the state of this output pin upon the rising edge of "RxTOHClk". Anytime the "Receive Order-Wire" circuitry samples this output pin being "high", it should sample and latch the data on the "RxE1F1E2" output pin (as a valid Order-wire bit) into the "Receive Order-Wire" circuitry.
AE6	RXPOH	O	CMOS	**Receive AU-4/VC-4/STS-3c Mapper POH Processor Block – Path Overhead Output Port – Output Pin:** This output pin, along with the "RxPOHClk", "RxPOHFrame" and "RxPOHValid" function as the "AU-4/VC-4 Mapper POH Processor block – POH Output port. These pins serially output the POH data that have been received by the Receive AU-4/VC-4 Mapper POH Processor block (via the "incoming" STS-3 data-stream). Each bit, within the POH bytes is updated (via these output pins) upon the falling edge of "RxPOHClk". As a consequence, external circuitry receiving this data, should sample this data upon the rising edge of "RxPOHClk".

AG4	RXPOHCLK	O	CMOS	**Receive AU-4/VC-4/STS-3c Mapper POH Processor Block – Path Overhead Output Port – Clock Output Signal:** This output pin, along with "RxPOH", "RxPOHFrame" and "RxPOHValid" function as the "AU-4/VC-4 Mapper POH Processor block – POH Output Port. These output pins function as the "Clock Output" signals for the "AU-4/VC-4 Mapper POH Processor Block– POH Output Port. The "RxPOH", "RxPOHFrame" and "RxPOHValid" output pins are updated upon the falling edge of this clock signal. As a consequence, the external circuitry should sample these signals upon the rising edge of this clock signal.
AE7	RXPOHFRAME	O	CMOS	**Receive AU-4/VC-4/STS-3c Mapper POH Processor Block – Path Overhead Output Port – Frame Boundary Indicator:** These output pins, along with the "RxPOH", RxPOHClk" and "RxPOHValid" output pins function as the "AU-4/VC-4 Mapper POH Processor Block – Path Overhead Output Port. These output pins will pulse "high" coincident with the very first POH byte (J1), of a given STS-1 frame, is being output via the corresponding "RxPOH" output pin.
AD9	RXPOHVALID	O	CMOS	**Receive AU-4/VC-4/STS-3c Mapper POH Processor Block – Path Overhead Output Port – Valid POH Data Indicator:** These output pins, along with "RxPOH", "RxPOHClk" and "RxPOHFrame" function as the "AU-4/VC-4 Mapper POH Processor block – Path Overhead Output port. These output pins will toggle "high" coincident with when valid POH data is being output via the "RxPOH" output pins. This output is updated upon the falling edge of RxPOHClk. Hence, external circuitry should sample these signals upon rising edge of "RxPOHClk".
AF5 AG5 AF8	RxPOH_0 RxPOH_1 RxPOH_2	O	CMOS	**Receive SONET POH Processor Block – Path Overhead Output Port – Output Pin:** These output pins, along with the "RxPOHClk_n", "RxPOHFrame_n" and "RxPOHValid_n" function as the "Receive SONET POH Processor block – POH Output port. These pins serially output the POH data that have been received by each of the Receive SONET POH Processor blocks (via the "incoming" STS-3 data-stream). Each bit, within the POH bytes is updated (via these output pins) upon the falling edge of "RxPOHClk_n". As a consequence, external circuitry receiving this data, should sample this data upon the rising edge of "RxPOHClk_n".

AE8 AE9 AG6	RxPOHClk_0 RxPOHClk_1 RxPOHClk_2	O	CMOS	**Receive SONET POH Processor Block – Path Overhead Output Port – Clock Output Signal:** These output pins, along with "RxPOH_n", "RxPOHFrame_n" and "RxPOHValid_n" function as the "Receive SONET POH Processor block – POH Output Port. These output pins function as the "Clock Output" signals for the "Receive SONET POH Processor block – POH Output Port. The "RxPOH_n", "RxPOHFrame_n" and "RxPOHValid_n" output pins are updated upon the falling edge of this clock signal. As a consequence, the external circuitry should sample these signals upon the rising edge of this clock signal.
AF6 AD10 AE10	RxPOHFrame_0 RxPOHFrame_1 RxPOHFrame_2	O	CMOS	**Receive SONET POH Processor Block – Path Overhead Output Port – Frame Boundary Indicator:** These output pins, along with the "RxPOH_n", RxPOHClk_n" and "RxPOHValid_n" output pins function as the "Receive SONET POH Processor Block – Path Overhead Output Port. These output pins will pulse "high" coincident with the very first POH byte (J1), of a given STS-1 frame, is being output via the corresponding "RxPOH_n" output pin.
AC10 AF7 AC11	RxPOHValid_0 RxPOHValid_1 RxPOHValid_2	O	CMOS	**Receive SONET POH Processor Block – Path Overhead Output Port – Valid POH Data Indicator:** These output pins, along with "RxPOH_n", "RxPOHClk_n" and "RxPOHFrame_n" function as the "Receive SONET POH Processor block – Path Overhead Output port. These output pins will toggle "high" coincident with when valid POH data is being output via the "RxPOH_n" output pins. This output is updated upon the falling edge of RxPOHClk_n. Hence, external circuitry should sample these signals upon rising edge of "RxPOHClk_n".
AD11	LOF	O	CMOS	**Receive STS-3 LOF (Loss of Frame) Indicator:** This output pin indicates whether or not the Receive STS-3 TOH Processor block (within the device) is currently declaring the LOF defect condition as described below. LOW – Indicates that the Receive STS-3 TOH Processor block is NOT currently declaring the LOF defect condition. HIGH – Indicates that the Receive STS-3 TOH Processor block is currently declaring the LOF defect condition.
AF9	SEF	O	CMOS	**Receive STS-3 SEF (Severed Errored Frame) Indicator:** This output pin indicates whether or not the Receive STS-3 TOH Processor block (within the device) is currently declaring the SEF defect condition as described below. LOW – Indicates that the Receive STS-3 TOH Processor block is NOT currently declaring the SEF defect condition. HIGH – Indicates that the Receive STS-3 TOH Processor block is currently declaring the SEF defect condition.

AG7	LOS	O	CMOS	**Receive STS-3 LOS (Loss of Signal) Defect Indicator:** This output pin indicates whether or not the Receive STS-3 TOH Processor block (within the device) is currently declaring the LOS defect condition as described below. LOW – Indicates that the Receive STS-3 TOH Processor block is NOT currently declaring the LOS defect condition. HIGH – Indicates that the Receive STS-3 TOH Processor block is currently declaring the LOS defect condition.
GENERAL PURPOSE INPUT/OUTPUT				
W25	GPI0_0	I/O	TTL/CMOS	**General Purpose Input/Output pin:** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 0 (GPIO_DIR_0), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin the state of this pin can be monitored by reading the state of Bit 0 (GPIO_0) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin the state of this pin can be controlled by writing the appropriate value into Bit 0 (GPIO_0) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).
AC27	GPIO_1	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 1 (GPIO_DIR_1), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 1 (GPIO_1) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 1 (GPIO_1) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).

V23	GPIO_2	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 2 (GPIO_DIR_2), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 2 (GPIO_2) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 2 (GPIO_2) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).
AB26	GPIO_3	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 3 (GPIO_DIR_3), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 3 (GPIO_3) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 3 (GPIO_3) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).
Y25	GPIO_4	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 4 (GPIO_DIR_4), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 4 (GPIO_4) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 4 (GPIO_4) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).

AC26	GPIO_5	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 35(GPIO_DIR_5), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 5 (GPIO_5) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 5 (GPIO_5) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).
W24	GPIO_6	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 6 (GPIO_DIR_6), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 6 (GPIO_6) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 6 (GPIO_6) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).
AA25	GPIO_7	I/O	TTL/CMOS	**General Purpose Input/Output pin** This input pin can be configured to function as either an input or output pin by writing the appropriate value into Bit 7 (GPIO_DIR_7), within the "Operation General Purpose Input/Output Direction Register – 0" (Address Location = 0x014B). If this pin is configured to be an input pin, then the state of this pin can be monitored by reading the state of Bit 7 (GPIO_7) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147). If this pin is configured to be an output pin, then the state of this pin can be controlled by writing the appropriate value into Bit 7 (GPIO_7) within the "Operation General Purpose Input/Output Register – Byte 0" (Address Location = 0x0147).

CLOCK INPUTS				
E7	REFCLK34	I	TTL	**E3 Reference Clock Input for the Jitter Attenuator within the DS3/E3 Mapper Block:** To operate any of the channels (within the XRT94L33) in the E3 Mode, apply a clock signal with a frequency of 34.368±20ppm to this input pin. This input pin functions as the timing reference for the DS3/E3/STS-1 Jitter Attenuator (within the DS3/E3 Mapper block) for E3 applications. ***Note:*** *Connect this pin to GND if none of the channels of the XRT94L33 are to be operated in the E3 or if the XRT94L33 is to be operated in the SFM mode.*
D5	REFCLK51	I	TTL	**STS-1 Reference Clock Input for the Jitter Attenuator within the DS3/E3 Mapper Block.** To operate any of the channels (within the XRT94L33) in the STS-1/STM-0 Mode, apply a clock signal with a frequency of 51.84MHz±20ppm to this input pin. This input pin functions as the timing reference for the DS3/E3/STS-1 Jitter Attenuator (within the DS3/E3 Mapper block) for STS-1 applications. ***Notes:*** *If the user intends to operate the XRT94L33 in the SFM Mode, apply a 12.288MHz±20ppm clock signal to this input pin.* *If the user does not intend to operate any of the channels in the STS-1/STM-0 Mode, connect this input pin to GND*.
F7	REFCLK45	I	TTL	**DS3 Reference Clock Input for the Jitter Attenuator within the DS3/E3 Mapper Block:** To operate any of the channels of the XRT94L33 in the DS3 Mode, apply a clock signal with a frequency of 44.736±20ppm to this input pin. This input pin functions as the timing reference for the DS3/E3/STS-1 Jitter Attenuator (within the DS3/E3 Mapper block) for DS3 applications. If the user does not intend to operate any of the three (3) channels within the XRT94L33 in the DS3 Mode, or if the user intends to configure the XRT94L33 to operate in the SFM Mode, then tie this input pin to GND.
BOUNDARY SCAN				
F5	TDO	O	CMOS	**Test Data Out: Boundary Scan Test data output**
F4	TDI	I	TTL	**TEST Data In: Boundary Scan Test data input** ***Note:*** *This input pin should be pulled "Low" for normal operation.*
D3	TRST	I	TTL	**JTAG Test Reset Input**

E4	TCK	I	TTL	**Test clock: Boundary Scan clock input** ***Note:*** *This input pin should be pulled "Low" for normal operation.*
E5	TMS	I	TTL	**Test Mode Select: Boundary Scan Mode Select input** ***Note:*** This *input pin should be pulled "Low" for normal operation.*
FILTERING CAPACITORS				
U6	RXCAPP	I	ANALOG	**External Loop Capacitor for Receive PLL:** This pin connects to the positive side of the external capacitor, which is used to minimize jitter peaking.
U5	RXCAPN	I	ANALOG	**External Loop Capacitor for Receive PLL:** This pin connects to the negative side of the external capacitor, which is used to minimize jitter peaking.
W6	RXCAPP_R	I	ANAL0OG	**External Redundant Loop Capacitor for Receive PLL:** This pin connects to the positive side of the external capacitor, which is used to minimize jitter peaking.
W5	RXCAPN_R	I	ANALOG	**External Redundant Loop Capacitor for Receive PLL:** This pin connects to the negative side of the external capacitor, which is used to minimize jitter peaking.
MISCELLANEOUS PINS				
H5	REFSEL_L	I	TTL	**Clock Synthesizer Block Select:** This input pin permits the user to configure the "Transmit SONET" circuitry (within the XRT94L33) to use either of the following clock signals as its timing source. a. The "Directly-Applied" 19.44MHz clock signal, which is applied to the REFTTL input pin (P1) or, b. The output of the "Clock Synthesizer" block (within the chip). Setting this input pin "HIGH" configures the "Transmit SONET" circuitry within the XRT94L33 to use the "Clock Synthesizer" block as its timing source. In this mode, the user can supply either a 19.44MHz, 38.88MHz, 51.84MHz or 77.76MHz clock signal to the REFTTL input pin. Setting this input pin "LOW" by-passes the "Clock Synthesizer" block. In this case, the user MUST supply a 19.44MHz clock signal to the REFTTL input pin in order to insure proper performance.

K4	SFM	I	TTL	**Single Frequency Mode (SFM) Select:** This input pin permits the user to configure the three Jitter Attenuator blocks within the XRT94L33 to operate in the Single-Frequency Mode (SFM). If the XRT94L33 has been configured to operate in the SFM Mode, then the user only needs to supply a 12.288MHz clock signal to the REFCLK51 input pin. In this case, the user does not need to supply a 44.736MHz clock signal to the REFCLK45 input pin, nor a 34.368MHz clock signal to the REFCLK34 input pin. The SFM PLL (within the XRT94L33) will internally synthesize the appropriate 44.736MHz, 34.368MHz or 51.84MHz clock signals, and will route these signals to the appropriate channels (within the chip) depending upon the data rate that they are configured to operate in. Setting this input pin to a logic "LOW" disables the Single-Frequency Mode. In this mode, the user must supply all of the appropriate frequencies to the REFCLK34, REFCLK45 and REFCLK51 input pins. Setting this input pin to a logic "HIGH" configures the XRT94L33 to operate in the Single-Frequency Mode.
J3	Test Mode	I	TTL	**Test Mode Input Pin:** User should connect this input pin "LOW" for normal operation.
G2	FL_TSTCLK	O	CMOS	**JA Testing Clock:** This pin is used for JA testing purposes.
J2	ANALOG	O	ANALOG	**Analog Output Pin:** This output analog pin is used for testing purposes.
N1	VDCTST1	O	ANALOG	**DC Test Pin:** This pin is used for internal DC test, for example, it can be used to test for DC current, DC voltage.
N2	VDCTST2	O	ANALOG	**DC Test Pin:** This pin is used for internal DC test, for example, it can be used to test for DC current, DC voltage.

NO-CONNECT PINS				
K1	N/C			
AA1	N/C			
V3	N/C			
AB1	N/C			
AA2	N/C			
AC1	N/C			
R1	N/C			
AB2	N/C			
AC2	N/C			
T1	N/C			
AC4	N/C			
AB5	N/C			
AD4	N/C			
AC5	N/C			
AB7	N/C			
AC6	N/C			
AC22	N/C			
AD24	N/C			
AB21	N/C			
AC23	N/C			
AB23	N/C			
AC24	N/C			
AA23	N/C			
E24	N/C			
F23	N/C			
D24	N/C			
E23	N/C			
F21	N/C			
E22	N/C			

VDD (3.3V)				
N23 N25 V5 H2 L2 K3 H1 L5 U4 N3 T5 M5	Analog VDD Pins	_		
U3 R2 R6 C2 C1 J6 K6 W3 Y3 AE1 AE2 AF3 AB9 AB10 AB11 AB17 AB18 AB19 AF25 AE26 W22 V22 U22 L22 K22 J22 C27 C26 B25 A25 F19 F18 F17 F11 F10	Digital VDD			

F9 A3 B3 D4 C4				

GROUND				
G6 C3 A1 B1 AF1 AF2 AA6 AB6 AE3 AG1 AG2 AB13 AB14 AB15 AG26 AF26 AB22 AA22 AE25 AG27 AF27 T22 R22 P22 N22 M22 B27 B26 G22 F22 C25 A27 A26 F15 F14 F13 A2 B2 F6 V2 W4 Y6 Y5 Y4 E6 V4 R5	Digital Ground			

R3 P4				
V6 L6 T4 N24 N26 R4 F1 K2 G1 L1 M6 N4 T6 J1	Analog Ground			

1.0 XRT94L33 ARCHITECTURE

The XRT94L33 can be configured to operate in any of the following modes for ATM and PPP Applications

- The 1-Channel STS-3c and up to 2 Channel DS3/E3 ATM UNI/PPP Mode
- The 1-Channel STS-3 (e.g., 3 Channel of ATM UNI/PPP over STS-1, which is in-turn mapped into STS-3 Mode
- The 1-Channel STS-3/STM-1 (e.g., 3 Channels of ATM/PPP over DS3/E3, which is turn mapped into STS-3 Mode.

The basic functional architecture of the XRT94L33, when it is configured to operate in either of these modes will be presented below.

1.1 FUNCTIONAL ARCHITECTURE OF THE XRT94L33 – 1 CHANNEL STS-3C ATM UNI/PPP MODE

If the XRT94L33 has been configured to operate in the "1 Channel STS-3c ATM UNI/PPP" Mode, then it will have the "Functional Architecture" as is presented below in Figure 1.

Figure 1: The Functional Block Diagram of the XRT94L33 when it has been configured to operate in the 1-Channel STS-3c ATM UNI/PPP Mode

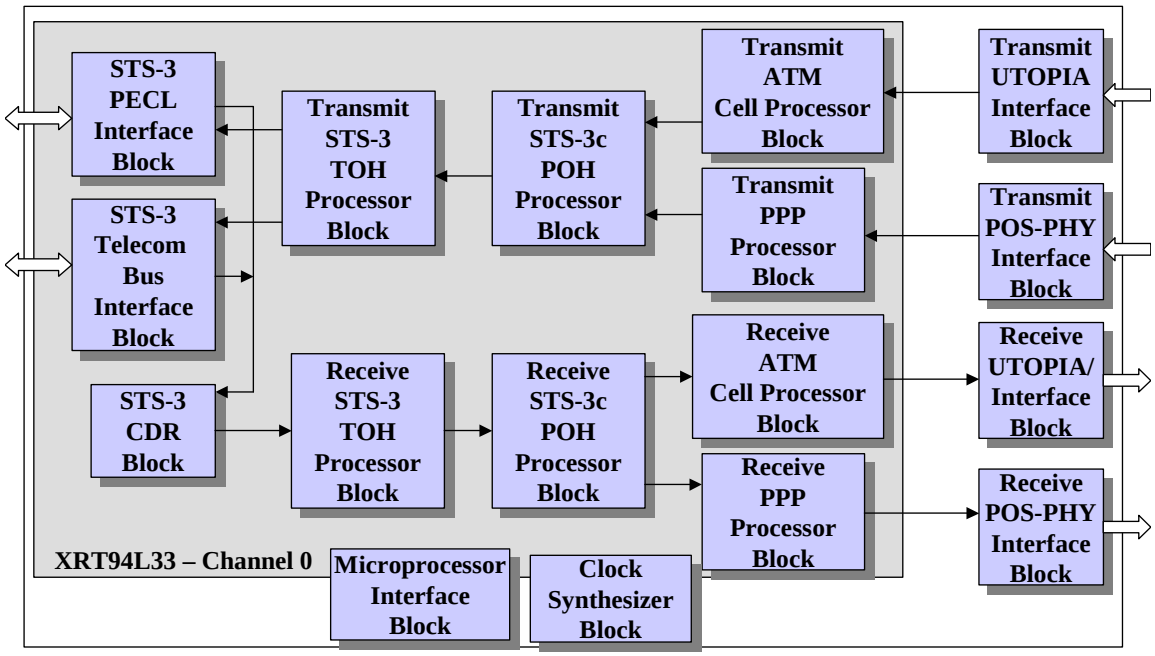


Figure 1 indicates that the XRT94L33 consists of the following functional blocks.

- The Receive STS-3 PECL Interface Block
- The Receive STS-3 Telecom Bus Interface Block
- The Receive STS-3 TOH Processor Block
- The Receive STS-3c POH Processor Block
- The Receive ATM Cell Processor Block
- The Receive UTOPIA Interface Block
- The Receive PPP Packet Processor Block
- The Receive POS-PHY Interface Block
- The Transmit POS-PHY Interface Block
- The Transmit ATM Cell Processor Block
- The Transmit PPP Packet Processor Block
- The Transmit STS-3c POH Processor Block
- The Transmit STS-3 TOH Processor Block
- The Transmit STS-3 Telecom Bus Interface Block
- The Transmit STS-3 PECL Interface Block

Each of these functional blocks is briefly discussed below. These functional blocks will be discussed in considerable detail throughout this data sheet.

1.1.1 THE CLOCK SYNTHESIZER BLOCK

The purpose of the Clock Synthesizer block is to synthesize a 19.44MHz and a 155.52MHz clock signal from an externally supplied 19.44MHz reference clock signal.

The Transmit STS-3 TOH and the Transmit STS-3c POH Processor blocks will use these clock signals as its timing source, for transmitting the outbound STS-3 data either via the PECL interface (e.g., to the optical transceiver) or via the Telecom Bus Interface, to the remote terminal.

1.1.2 THE RECEIVE STS-3 PECL INTERFACE AND CDR BLOCK

The purpose of the Receive STS-3 PECL Interface and CDR Block is to perform the following functions.

- To receive an STS-3 electrical signal (which is of the PECL format) from either a system back-plane or from an optical transceiver.

- As the Receive STS-3 PECL Interface block receives this electrical (data) signal, it will route this data-stream to the “STS-3 Clock and Data Recovery (CDR) Block. This STS-3 CDR block will then generate a 155.52MHz clock and corresponding data signal, which will be routed to the Receive STS-3 TOH Processor block for further processing.

1.1.3 THE RECEIVE STS-3 TOH PROCESSOR BLOCK

The purpose of the “Receive STS-3 TOH Processor” block is to perform the following functions.

- To receive an STS-3 signal from the remote terminal via optical fiber and a PECL interface, or via the “Receive STS-3 Telecom Bus Interface.
- To declare and clear the LOS, SEF, LOF and AIS-L defect conditions.
- To declare and clear the RDI-L, SD and SF defect conditions.
- To optionally transmit the AIS-P indicator (downstream, towards the Receive STS-3c POH Processor block) upon declaration of the AIS-L, LOS, LOF, SD or SF defect conditions.
- To compute and verify the B1 and B2 bytes of the incoming STS-3 signal.
- To detect and increment performance monitor registers anytime it detects any B1 and B2 byte errors.
- To receive and process Section Trace messages via the J0 byte.
- To terminate the Transport Overhead (TOH) within the incoming STS-3 signal.
- To detect and increment performance monitor registers anytime it detects any REI-L events.
- To receive and process messages via the J0 byte.
- To terminate the Transport Overhead (TOH) within the incoming STS-3 signal.
- To the resulting STS-3c SPE data-stream to the Receive STS-3c POH Processor block.

1.1.4 THE RECEIVE STS-3C POH PROCESSOR BLOCK

The purpose of the “Receive STS-3c POH Processor” block is to perform the following functions.

- To receive the STS-3c signal (originally extracted from the incoming STS-3 signal) to terminate the Path Overhead (POH).
- To declare and clear LOP-P, AIS-P, UNEQ-P, PLM-P, TIM-P, and the RDI-P defect conditions.
- To optionally transmit the AIS-P indicator, in the down-stream direction (towards the Receive ATM Cell or Receive PPP Packet Processor Blocks) anytime (and for the duration that) the Receive STS-3c POH Processor declares the AIS-P, LOP-P, UNEQ-P, PLM-P or TIM-P defect conditions
- To declare and clear the LOP-C and AIS-C defect condition.
- To compute and verify the B3 byte of the incoming STS-3 SPE.
- To detect and increment Performance Monitor registers anytime it detects B3 byte errors in the incoming STS-3c SPE data-stream.
- To detect and increment performance monitor registers anytime it detects any REI-P events.
- To receive and process Path Trace messages via the J1 byte.
- To route the STS-3c SPE data to the Receive ATM Cell Processor or Receive PPP Packet Processor blocks for further processing.

1.1.5 THE RECEIVE ATM CELL PROCESSOR BLOCK

The purpose of the "Receive ATM Cell Processor" block is to extract out the data (being carried by the incoming STS-3c SPE data-stream) and to perform the following operations on it.

- Cell Delineation
- HEC Byte Verification
- Idle Cell Filtering
- User Cell Filtering
- To receive cells, with "user-specified" header bytes and to load them into the "Receive Cell Extraction Memory Buffer" (where they can be read out and accessed by the Microprocessor Interface)
- To read out a "user-specified" ATM cell (which is residing in the Receive Cell Insertion Buffer) and to insert this cell into the "Receive ATM Cell" traffic.
- To route all filtered cells to the "Receive ATM Cell Buffer" (where it will be made available to the ATM Layer Processor via the Receive UTOPIA Interface block).

1.1.6 THE RECEIVE UTOPIA INTERFACE BLOCK

The purpose of the Receive UTOPIA Interface Block is to provide a standard UTOPIA Level 1, 2 or 3 interface to the ATM Layer Processor; for reading out the contents of all ATM cells that are written into the "Receive ATM Cell Buffer".

The Receive UTOPIA Interface block can be configured to operate with either an 8 or 16-bit wide "Receive UTOPIA Data" bus.

Note: *The Receive UTOPIA Interface Block supports "UTOPIA Level 3" from a signaling stand-point. The Receive UTOPIA Interface Block within the XRT94L33 still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates up to 50MHz (not 100MHz).*

1.1.7 THE TRANSMIT UTOPIA INTERFACE BLOCK

The purpose of the Transmit UTOPIA Interface Block is to provide a standard UTOPIA Level 1, 2 or 3 interface to the ATM Layer Processor, for writing in the contents of all "Valid" ATM cells, into the "Transmit Cell Buffer".

The Transmit UTOPIA Interface block can be configured to operate with either an 8 or 16-bit wide "Transmit UTOPIA Data" bus.

Note: *The Transmit UTOPIA Interface Block supports "UTOPIA Level 3" from a signaling stand-point. The Receive UTOPIA Interface Block within the XRT94L33 still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates up to 50MHz (not 100MHz).*

1.1.8 THE TRANSMIT ATM CELL PROCESSOR BLOCK

The purpose of the "Transmit ATM Cell Processor" block is to read out the contents of "user" cells that have been written into the "TxFIFO" (via the Transmit UTOPIA Interface block); and perform the following functions.

- Optionally Compute and Verify the HEC byte of each cell written into the "Transmit Cell Buffer"
- To optionally discard all incoming ATM cells that contain HEC byte errors.
- Optionally Compute and Insert the HEC byte into the fifth octet position, within each cell written into the "Transmit ATM Cell Buffer".
- To optionally filter User Cells (that are read out from the TxFIFO) by either discarding these User Cells, or by replicating them and routing the copies of these cells to the Transmit Cell Extraction Buffer.
- To insert cells (residing within the Transmit Cell Insertion Buffer) into the Transmit Data Path anytime the TxFIFO is depleted of user cells.
- Generate Idle Cells, anytime the TxFIFO and the Transmit Cell Insertion Buffer are depleted of "User" cells.
- To route the composite stream of "valid" and "idle" cells to the Transmit STS-3c POH Processor Block.

1.1.9 THE TRANSMIT STS-3C POH PROCESSOR BLOCK

The purpose of the "Transmit STS-3c POH Processor" block is to perform the following functions.

- To receive ATM Cell or PPP data from the Transmit ATM Cell Processor or Transmit PPP Packet Processor blocks and to map this data into an STS-3c SPE data-stream.
- To automatically transmit the RDI-P indicator (to the remote PTE) upon detection of the following conditions, by the Receive STS-3c POH Processor block.
    a. AIS-P
    b. LOP-P
    c. UNEQ-P
    d. PLM-P
    e. TIM-P
- To automatically transmit the RDI-P indicator (to the remote PTE) anytime (and for the duration that) the corresponding Receive ATM Cell Processor block declares the LCD-P (Loss of Cell Delineation) defect condition.
- To transmit the RDI-P indicator upon software control.
- To transmit the REI-P indicator (to the remote PTE) upon detection of B3 byte errors by the corresponding Receive STS-3c POH Processor block.
- To transmit the REI-P indicator upon software control.
- To compute the BIP-8 value over the entire STS-3c SPE and to insert this value into the "B3-byte" position within the very next outbound STS-3c SPE.
- To transmit the AIS-P indicator (under software control).

1.1.10 THE TRANSMIT STS-3/12 TOH PROCESSOR BLOCK

The purpose of the "Transmit STS-3 TOH Processor" block is to perform the following functions.

- To generate and insert the TOH (for the "outbound" STS-3c signal) prior to transmission to the remote LTE.
- To automatically transmit the RDI-L indicator (to the remote LTE) upon detection of the following defect conditions by the corresponding Receive STS-3 TOH Processor block.
    a. LOS
    b. LOF
    c. AIS-L
    d. SD
    e. SF
- To transmit the RDI-L indicator upon software control.
- To automatically transmit the REI-L indicator (to the remote LTE) upon detection of B2 byte errors by the Receive STS-3 TOH Processor block.
- To transmit the REI-L indicator upon software control.
- To transmit the AIS-L indicator (to the remote LTE) upon software command.

1.1.11 THE TRANSMIT STS-3 PECL INTERFACE BLOCK

The purpose of the Transmit STS-3 PECL Interface block is to accept STS-3 data from the Transmit STS-3 TOH Processor block and to perform the following functions on this signal.

- Converting this "outbound" CMOS-level signal into the LVPECL format.
- To output this STS-3 PECL signal to either the System Back-plane (for transmission to a "Concentrator" Board) or to an Optical Transceiver, for transmission to remote terminal equipment.

1.2 FUNCTIONAL ARCHITECTURE OF THE XRT94L33 – 1 CHANNEL STS-3 ATM UNI/PPP MODE

If the XRT94L33 has been configured to operate in the "1-Channel STS-3 ATM UNI/PPP" Mode, then it will have the "Functional Architecture" as is presented below in Figure 2

Figure 2 The Functional Block Diagram of the XRT94L33 when it has been configured to operate in the 1-Channel STS-3 ATM UNI/PPP Mode

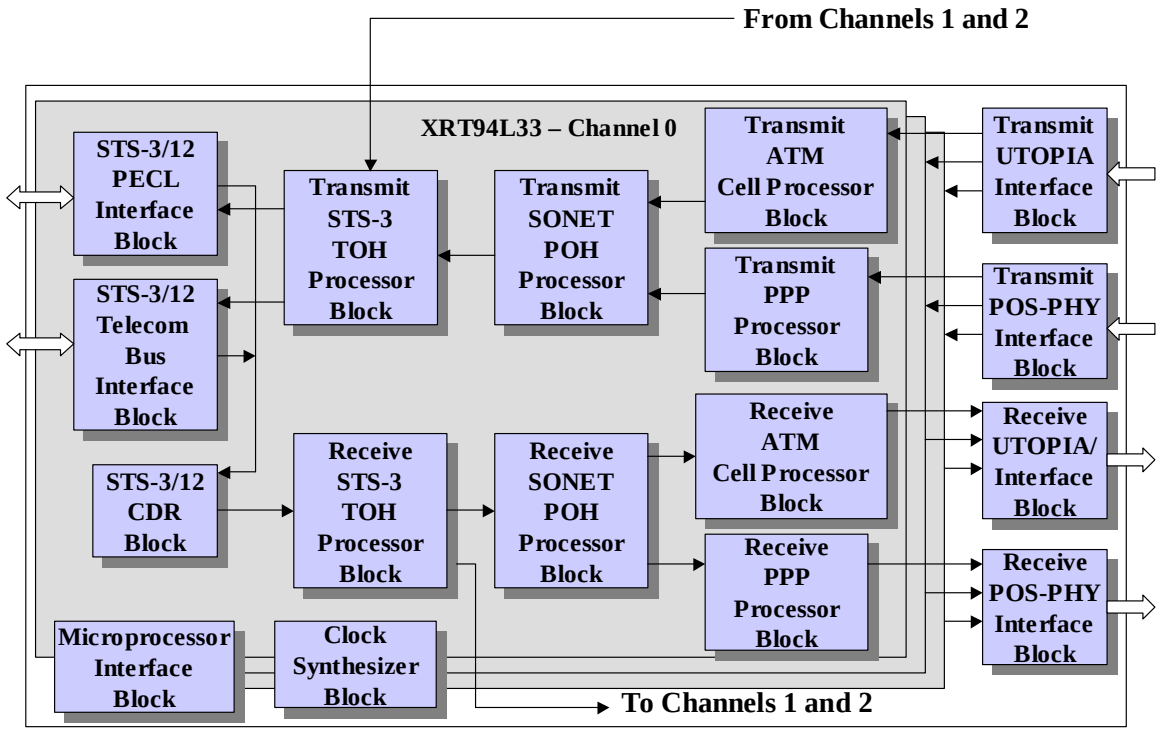


Figure 2 indicates that the XRT94L33 consists of the following functional blocks.

- The Receive STS-3 PECL Interface Block
- The Receive STS-3 Telecom Bus Interface Block
- The Receive STS-3 TOH Processor Block
- The Receive SONET POH Processor Block
- The Receive ATM Cell Processor Block
- The Receive UTOPIA Interface Block
- The Receive PPP Packet Processor Block
- The Receive POS-PHY Interface Block
- The Transmit POS-PHY Interface Block
- The Transmit ATM Cell Processor Block
- The Transmit PPP Packet Processor Block
- The Transmit SONET POH Processor Block
- The Transmit STS-3 TOH Processor Block
- The Transmit STS-3 Telecom Bus Interface Block
- The Transmit STS-3 PECL Interface Block

Each of these functional blocks is briefly discussed below. These functional blocks will be discussed in considerable detail throughout this data sheet.

1.2.1 THE CLOCK SYNTHESIZER BLOCK

The purpose of the Clock Synthesizer block is to synthesize a 19.44MHz and a 155.52MHz clock signal from an externally supplied 19.44MHz reference clock signal.

The Transmit STS-3 TOH and the Transmit STS-3c POH Processor blocks will use these clock signals as its timing source, for transmitting the outbound STS-3 data either via the PECL interface (e.g., to the optical transceiver) or via the Telecom Bus Interface, to the remote terminal.

1.2.2 THE RECEIVE STS-3 PECL INTERFACE AND CDR BLOCK

The purpose of the Receive STS-3 PECL Interface and CDR Block is to perform the following functions.

- To receive an STS-3 electrical signal (which is of the PECL format) from either a system back-plane or from an optical transceiver.
- As the Receive STS-3 PECL Interface block receives this electrical (data) signal, it will route this data-stream to the "STS-3 Clock and Data Recovery (CDR) Block. This STS-3 CDR block will then generate a 155.52MHz clock and corresponding data signal, which will be routed to the Receive STS-3 TOH Processor block for further processing.

1.2.3 THE RECEIVE STS-3 TOH PROCESSOR BLOCK

The purpose of the "Receive STS-3 TOH Processor" block is to perform the following functions.

- To receive an STS-3 signal from the remote terminal via optical fiber and a PECL interface, or via the "Receive STS-3 Telecom Bus Interface.
- To declare and clear the LOS, SEF, LOF and AIS-L defect conditions.
- To declare and clear the RDI-L, SD and SF defect conditions.
- To optionally transmit the AIS-P indicator (downstream, towards the Receive STS-3c POH Processor block) upon declaration of the AIS-L, LOS, LOF, SD or SF defect conditions.
- To compute and verify the B1 and B2 bytes of the incoming STS-3 signal.
- To detect and increment performance monitor registers anytime it detects any B1 and B2 byte errors.
- To receive and process Section Trace messages via the J0 byte.
- To terminate the Transport Overhead (TOH) within the incoming STS-3 signal.
- To detect and increment performance monitor registers anytime it detects any REI-L events.
- To receive and process messages via the J0 byte.
- To terminate the Transport Overhead (TOH) within the incoming STS-3 signal.
- To byte de-interleave the STS-3 signal into 3 STS-1 SPE data-streams, and to route the resulting STS-1 SPE data-stream to each of the three (3) Receive SONET POH Processor block.

1.2.4 THE RECEIVE SONET POH PROCESSOR BLOCK

The purpose of the "Receive SONET POH Processor" block is to perform the following functions.

- To receive one of the three STS-1 signal (originally extracted from the incoming STS-3 signal) to terminate the Path Overhead (POH).
- To declare and clear LOP-P, AIS-P, UNEQ-P, PLM-P, TIM-P, and the RDI-P defect conditions.

- To optionally transmit the AIS-P indicator, in the down-stream direction (towards the Receive ATM Cell or Receive PPP Packet Processor Blocks) anytime (and for the duration that) the Receive SONET POH Processor declares the AIS-P, LOP-P, UNEQ-P, PLM-P or TIM-P defect conditions
- To declare and clear the LOP-C and AIS-C defect condition.
- To compute and verify the B3 byte of the incoming STS-3 SPE.
- To detect and increment Performance Monitor registers anytime it detects B3 byte errors in the incoming STS-1 SPE data-stream.
- To detect and increment performance monitor registers anytime it detects any REI-P events.
- To receive and process Path Trace messages via the J1 byte.
- To route the STS-1 SPE data to the Receive ATM Cell Processor or Receive PPP Packet Processor blocks for further processing.

1.2.5 THE RECEIVE ATM CELL PROCESSOR BLOCK

The purpose of the "Receive ATM Cell Processor" block is to extract out the data (being carried by its corresponding incoming STS-1 SPE data-stream) and to perform the following operations on it.

- Cell Delineation
- HEC Byte Verification
- Idle Cell Filtering
- User Cell Filtering
- To receive cells, with "user-specified" header bytes and to load them into the "Receive Cell Extraction Memory Buffer" (where they can be read out and accessed by the Microprocessor Interface)
- To read out a "user-specified" ATM cell (which is residing in the Receive Cell Insertion Buffer) and to insert this cell into the "Receive ATM Cell" traffic.
- To route all filtered cells to the "Receive ATM Cell Buffer" (where it will be made available to the ATM Layer Processor via the Receive UTOPIA Interface block).

1.2.6 THE RECEIVE UTOPIA INTERFACE BLOCK

The purpose of the Receive UTOPIA Interface Block is to provide a standard UTOPIA Level 1, 2 or 3 interface to the ATM Layer Processor; for reading out the contents of all ATM cells that are written into the "Receive ATM Cell Buffer".

The Receive UTOPIA Interface block can be configured to operate with either an 8 or 16-bit wide "Receive UTOPIA Data" bus.

Note: *The Receive UTOPIA Interface Block supports "UTOPIA Level 3" from a signaling stand-point. The Receive UTOPIA Interface Block within the XRT94L33 still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates up to 50MHz (not 100MHz).*

1.2.7 THE TRANSMIT UTOPIA INTERFACE BLOCK

The purpose of the Transmit UTOPIA Interface Block is to provide a standard UTOPIA Level 1, 2 or 3 interface to the ATM Layer Processor, for writing in the contents of all "Valid" ATM cells, into the "Transmit Cell Buffer".

The Transmit UTOPIA Interface block can be configured to operate with either an 8 or 16-bit wide "Transmit UTOPIA Data" bus.

Note: *The Transmit UTOPIA Interface Block supports "UTOPIA Level 3" from a signaling stand-point. The Receive UTOPIA Interface Block within the XRT94L33 still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates up to 50MHz (not 100MHz).*

1.2.8 THE TRANSMIT ATM CELL PROCESSOR BLOCK

The purpose of the "Transmit ATM Cell Processor" block is to read out the contents of "user" cells that have been written into the "TxFIFO" (via the Transmit UTOPIA Interface block); and perform the following functions.

- Optionally Compute and Verify the HEC byte of each cell written into the "Transmit Cell Buffer"
- To optionally discard all incoming ATM cells that contain HEC byte errors.
- Optionally Compute and Insert the HEC byte into the fifth octet position, within each cell written into the "Transmit ATM Cell Buffer".
- To optionally filter User Cells (that are read out from the TxFIFO) by either discarding these User Cells, or by replicating them and routing the copies of these cells to the Transmit Cell Extraction Buffer.
- To insert cells (residing within the Transmit Cell Insertion Buffer) into the Transmit Data Path anytime the TxFIFO is depleted of user cells.
- Generate Idle Cells, anytime the TxFIFO and the Transmit Cell Insertion Buffer are depleted of "User" cells.
- To route the composite stream of "valid" and "idle" cells to the Transmit SONET POH Processor Block.

1.2.9 THE TRANSMIT SONET POH PROCESSOR BLOCK

The purpose of the "Transmit SONET POH Processor" block is to perform the following functions.

- To receive ATM Cell or PPP data from its corresponding Transmit ATM Cell Processor or Transmit PPP Packet Processor blocks and to map this data into an STS-1 SPE data-stream.
- To automatically transmit the RDI-P indicator (to the remote PTE) upon detection of the following conditions, by the corresponding Receive SONET POH Processor block.
    a. AIS-P
    b. LOP-P
    c. UNEQ-P
    d. PLM-P
    e. TIM-P
- To automatically transmit the RDI-P indicator (to the remote PTE) anytime (and for the duration that) the corresponding Receive ATM Cell Processor block declares the LCD-P (Loss of Cell Delineation) defect condition.
- To transmit the RDI-P indicator upon software control.
- To transmit the REI-P indicator (to the remote PTE) upon detection of B3 byte errors by the corresponding Receive SONET POH Processor block.
- To transmit the REI-P indicator upon software control.
- To compute the BIP-8 value over the entire STS-1 SPE and to insert this value into the "B3-byte" position within the very next outbound STS-1 SPE.
- To transmit the AIS-P indicator (under software control).

1.2.10 THE TRANSMIT STS-3 TOH PROCESSOR BLOCK

The purpose of the "Transmit STS-3 TOH Processor" block is to perform the following functions.

- To generate and insert the TOH (for the "outbound" STS-3 signal) prior to transmission to the remote LTE.
- To automatically transmit the RDI-L indicator (to the remote LTE) upon detection of the following defect conditions by the corresponding Receive STS-3 TOH Processor block.
    a. LOS
    b. LOF
    c. AIS-L
    d. SD
    e. SF
- To transmit the RDI-L indicator upon software control.
- To automatically transmit the REI-L indicator (to the remote LTE) upon detection of B2 byte errors by the Receive STS-3 TOH Processor block.
- To transmit the REI-L indicator upon software control.
- To transmit the AIS-L indicator (to the remote LTE) upon software command.

1.2.11 THE TRANSMIT STS-3 PECL INTERFACE BLOCK

The purpose of the Transmit STS-3 PECL Interface block is to accept STS-3 data from the Transmit STS-3 TOH Processor block and to perform the following functions on this signal.

- Converting this "outbound" CMOS-level signal into the LVPECL format.
- To output this STS-3 PECL signal to either the System Back-plane (for transmission to a "Concentrator" Board) or to an Optical Transceiver, for transmission to remote terminal equipment.

1.3 THE MICROPROCESSOR INTERFACE

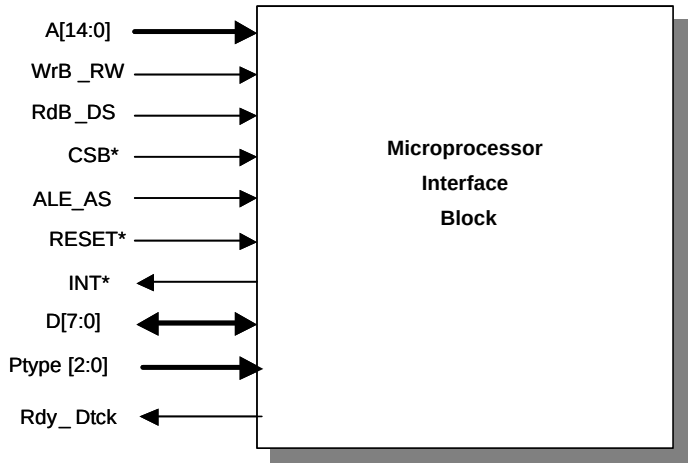
The purpose of the Microprocessor Interface block is to support the following operations between the user's Microprocessor/Microcontroller and the XRT94L33.

- Writing/Loading configuration information into the "on-chip" Command Registers; within the XRT94L33.
- Reading out the state of various bit-fields, reflecting Alarm conditions.
- Writing in the contents of "outbound" J0 (Section Trace) Messages into the "Transmit J0 (Section Trace) Message" buffer.
- Reading out the contents of "inbound" J0 Messages from the "Receive J0 (Section Trace) Message" buffer.
- Writing in the contents of "outbound" J1 (Path Trace) Messages into the "Transmit J1 (Path Trace) Message" buffer.
- Reading out the contents of "inbound" J1 Messages from the "Receive J1 (Path Trace) Message" buffer.
- Writing the contents of "outbound" ATM (OAM) cells into the Transmit Cell Insertion Memory within the Transmit ATM Cell Processor block
- Reading the contents of certain "outbound" ATM cells from the Transmit Cell Extraction Memory within the Transmit ATM Cell Processor block.
- Reading the contents of "inbound" ATM (OAM) cells from the Receive Cell Extraction Memory within the Receive ATM Cell Processor block.

- Servicing of various interrupts.

Each of these operations (between the microprocessor and the XRT94L33) will be discussed in some detail, throughout this data sheet.

Figure 3 Simple Block Diagram of Microprocessor Interface block of XRT94L33



1.3.1 THE MICROPROCESSOR INTERFACE SIGNALS

The XRT94L33 may be configured into a wide variety of different operating modes and have its performance monitored by software through a microprocessor, using data, address and control signals.

The μC/μP configures the XRT94L33 (into a desired operating mode) by writing data into specific addressable, on-chip "Read/Write" registers; or on-chip RAM. The microprocessor interface provides the signals which are required for a general purpose microprocessor to read or write data into these registers. The Microprocessor Interface also supports "polled" and interrupt driven environments. These interface signals are described below in Tables 1, 2, and 3. The microprocessor interface within the XRT94L33 can be configured to operate in two basic modes.

- The Intel Asynchronous Mode, and
- The Power PC 403 Mode

Each of these Microprocessor Interface Modes is described in detail below.

1.3.1.1 THE INTEL ASYNCHRONOUS MODE:

Whenever the Microprocessor Interface has been configured to operate in the "Intel Asynchronous" mode, then some of the these control signals function in a manner as required by the "Intel 80xx" family of Microprocessors.

presents a brief description of each of the Microprocessor Interface pins whenever the Microprocessor Interface has been configured to operate in the "Intel Asynchronous" Mode.

Table 1 Description of the Microprocessor Interface Pins whenever the Microprocessor Interface has been configured to operate in the Intel Asynchronous Mode

PIN NAME	PIN #	TYPE	DESCRIPTION
D[7:0]		I/O	**Bi-directional data bus Input/Output pins for on-chip register and buffer READ or WRITE operation.** This byte-wide data bus carries all data that is being written into or read form the XRT94L33.
A[15:0]		I	**Sixteen Bit Address Bus Input pins:** This 16-bit Address Bus permits the user to select an on-chip register or buffer location to be the "target" address for the current READ or WRITE operation.
CS*		I	**Chip Select Input Pin:** This "active-low" signal selects the Microprocessor Interface of the XRT94L33 and enables READ/WRITE operations with on-chip registers and buffer locations.
INT*		O	**Interrupt Request Output Pin:** This open-drain/active-low output signal will inform the local Microprocessor/Microcontroller that the XRT94L33 has a pending interrupt request. Under normal conditions this output pin will be at a logic "high" level. However, this output pin will toggle "low" whenever the XRT94L33 has a pending interrupt request that needs servicing. This output pin will remain "low" until all pending interrupt requests have been serviced. Please see Section _ for more information on servicing interrupts.
ALE_AS		I	**Address-Latch Enable**: This "active-high" signal is used to latch the contents on the address bus, A[15:0]. The contents of the Address Bus are latched into the A[15:0] inputs on the falling edge of ALE_AS. The contents of the Address Bus will only be latched into the XRT94L33 circuitry if the CS* input pin is asserted (e.g., pulled "low").
RD_DS		I	**Read Strobe Signal**: This "active-low" input functions as the read signal from the local μP. When this signal goes "low", the XRT94L33 Microprocessor Interface will respond by placing the contents of the addressed register on the Data Bus pins (D[7:0]). The Data Bus will be "tri-stated" once this input signal returns "high".
WR_RW		I	**Write Strobe Signal**: This "active-low" input functions as the write signal from the local μP. The contents of the Data Bus (D[7:0]) will be written into the addressed register (via A[15:0]), on the rising edge of this signal.

The XRT94L33 supports a wide variety of Microprocessor types, including Intel and Motorola types of Microprocessors. When the Microprocessor Interface is operating in the "Motorola" mode, then some of the control signals function in a manner as required by the Motorola 68000 family of microprocessors. Likewise, when the Microprocessor Interface is operating in the "Intel" Mode, then some of these Control Signals function in a manner as required by the Intel 80xx family of microprocessorsTable 2 lists and describes those Microprocessor Interface signals whose role is

constant across the two modes. Error! Reference source not found. **describes the role of some of these signals when the Microprocessor Interface is operating in the Intel Mode. Likewise**

Table 4 describes the role of these signals when the Microprocessor Interface is operating in the Motorola Mode.

Table 2 Description of the Microprocessor Interface Signals that exhibit constant roles in both the "Intel" and "Motorola" Modes.

PIN NAME	TYPE	DESCRIPTION
D[7:0]	I/O	**Bi-Directional Data Bus for register read or write operations.** This byte wide carries all data that is being written into or read from the XRT94L33.
A[14:0]	I	**Fourteen Bit Address Bus input:** This nine bit Address Bus is provided to allow the user to select an on-chip register or on-chip RAM location.
CS*	I	**Chip Select input.** This "active low" signal selects the Microprocessor Interface of the XRT94L33 and enables read/write operations with the on-chip registers/on-chip RAM.
INT*	O	**Interrupt Request Output** This "open-drain/active-low" output signal will inform the local μP that the XRT94L33 has an interrupt condition that needs servicing.

Table 3 Pin Description of Microprocessor Interface Signals - While the Microprocessor Interface is Operating in the Intel Mode.

PIN NAME	EQUIVALENT PIN IN INTEL ENVIRONMENT	TYPE	DESCRIPTION
ALE_AS	ALE	I	**Address-Latch Enable:** This "active-high" signal is used to latch the contents on the address bus, A[8:0]. The contents of the Address Bus are latched into the A[8:0] inputs on the falling edge of ALE_AS. Additionally, this signal can be used to indicate the start of a burst cycle.
RdB_DS	RD*	I	**Read Signal:** This "active-low" input functions as the read signal from the local μP. When this signal goes "low", the XRT94L33 Microprocessor Interface will place the contents of the addressed register on the Data Bus pins (D[7:0]). The Data Bus will be "tri-stated" once this input signal returns "high".
WRB_RW	WR*	I	**Write Signal:** This "active-low" input functions as the write signal from the local μP. The contents of the Data Bus (D[7:0]) will be written into the addressed register (via A[8:0]), on the rising edge of this signal.
Rdy_Dtck	READY*	O	**Ready Output:** This "active-low" signal is provided by the XRT94L33, and indicates that the current read or write cycle is to be extended until this signal is asserted. The local μP will typically insert "WAIT" states until this signal is asserted. This output will toggle "low" when the device is ready for the next Read or Write cycle.

Table 4 Pin Description of the Microprocessor Interface Signals while the Microprocessor Interface is operating in the Motorola Mode

PIN NAME	EQUIVALENT PIN IN MOTOROLA ENVIRONMENT	TYPE	DESCRIPTION
ALE_AS	AS*	I	**Address Strobe:** This "active-low" signal is used to latch the contents on the address bus input pins: A[14:0] into the Microprocessor Interface circuitry. The contents of the Address Bus are latched into the XRT94L33 on the rising edge of the ALE_AS signal. This signal can also be used to indicate the start of a burst cycle.
RdB_DS	DS*	I	**Data Strobe:** This signal latches the contents of the bi-directional data bus pins into the Addressed Register within the XRT94L33 during a Write Cycle.
WRB_RW	R/W*	I	**Read/Write* Input:** When this pin is "high", it indicates a Read Cycle. When this pin is "low", it indicates a Write cycle.
Rdy_Dtck	DTACK*	O	**Data Transfer Acknowledge:** The XRT94L33 asserts DTACK* in order to inform the CPU that the present READ or WRITE cycle is nearly complete. The 68000 family of CPUs requires this signal from its peripheral devices, in order to quickly and properly complete a READ or WRITE cycle.

1.3.2 INTERFACING THE XRT94L33 TO THE μC/μP OVER VIA THE MICROPROCESSOR INTERFACE BLOCK

The Microprocessor Interface block, within the XRT94L33 is very flexible and provides the following options to the user.

- To interface the XRT94L33 to a μC/μP over an 8-bit-wide bi-directional data bus.
- To interface the XRT94L33 to a wide variety of Microprocessor Interface types
- To transfer data (between the XRT94L33 IC and the μC/μP) via the Programmed I/O Mode.

Each of the options is discussed in detail below. Section _ will discuss Data Access (e.g., Programmed I/O and Burst) Mode when interfaced to both Motorola-type and Intel-type μC/μP.

1.3.2.1 SELECTING THE APPROPRIATE MICROPROCESSOR INTERFACE MODE

The user can configure the Microprocessor Interface, within the XRT94L33, to support a wide-variety of Microprocessor Interface Modes. The user can accomplish this by setting the PType[2:0] input pins to the appropriate setting as listed below.

Table 5 Settings for the PType[2:0] and the Corresponding Microprocessor Interface Modes

PTYPE[2:0]	MICROPROCESSOR INTERFACE MODE
000	68HC11, 8051, 80C188
001	Motorola – 68000 Family
010	Intel X86 Family
011	Intel i960
100	IDT3051/52
101	Power PC 403

This revision of the XRT94L33 Data Sheet discusses the Motorola and Intel X86 Modes in detail. The remaining Microprocessor Interface Modes will be discussed in a later revision of this data sheet.

1.3.2.2 DATA ACCESS MODES

As mentioned earlier, the Microprocessor Interface block supports data transfer between the XRT94L33 and the μC/μP (e.g., "Read" and "Write" operations) via two modes: the "Programmed I/O" and the "Burst" Modes. Programmed I/O access is discussed within this revision of the data sheet. Burst Mode access will be discussed in the next revision of this data sheet.

1.3.2.3 DATA ACCESS USING PROGRAMMED I/O

"Programmed I/O" is the conventional manner in which a microprocessor exchanges data with a peripheral device. However, it is also the slowest method of data exchange between the XRT94L33 and the μC/μP; as will be described in this text.

The next two sections present detailed information on Programmed I/O Access, when the XRT94L33 is operating in the "Intel Mode" and in the "Motorola Mode".

1.3.3 PROGRAMMED I/O ACCESS IN THE "INTEL" MODE

If the XRT94L33 is interfaced to an "Intel-type" μC/μP (e.g., the 80x86 family, etc.), then it should be configured to operate in the "Intel" mode (by tying the "MOTO" pin to ground). Intel-type "Read" and "Write" operations are described below.

1.3.3.1 THE INTEL MODE READ CYCLE

Whenever an Intel-type μC/μP wishes to read the contents of a register or some location within the Transmit or Receive Extraction Memory or the J0/J1 Message Buffers, within the XRT94L33, it should do the following.

1. Place the address of the "target" register or buffer location (within the UNI) on the Address Bus input pins A[14:0].

2. While the μC/μP is placing this address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the CS* (Chip Select) pin of the XRT94L33, by toggling it "low". This action enables further communication between the μC/μP and the XRT94L33 Microprocessor Interface block.

3. Toggle the ALE_AS (Address Latch Enable) input pin "high". This step enables the "Address Bus" input drivers, within the Microprocessor Interface block of the XRT94L33.

4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address" Data Setup time"), the μC/μP should toggle the ALE_AS pin "low". This step causes the XRT94L33 to "latch" the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer locations within the XRT94L33, has now been selected.

5. Next, the μC/μP should indicate that this current bus cycle is a "Read" Operation by toggling the RdB_DS (Read Strobe) input pin "low". This action also enables the bi-directional data bus output drivers of the XRT94L33. At this point, the "bi-directional" data bus output drivers will proceed to drive the contents of the "latched addressed" register (or buffer location) onto the bi-directional data bus, D[7:0].

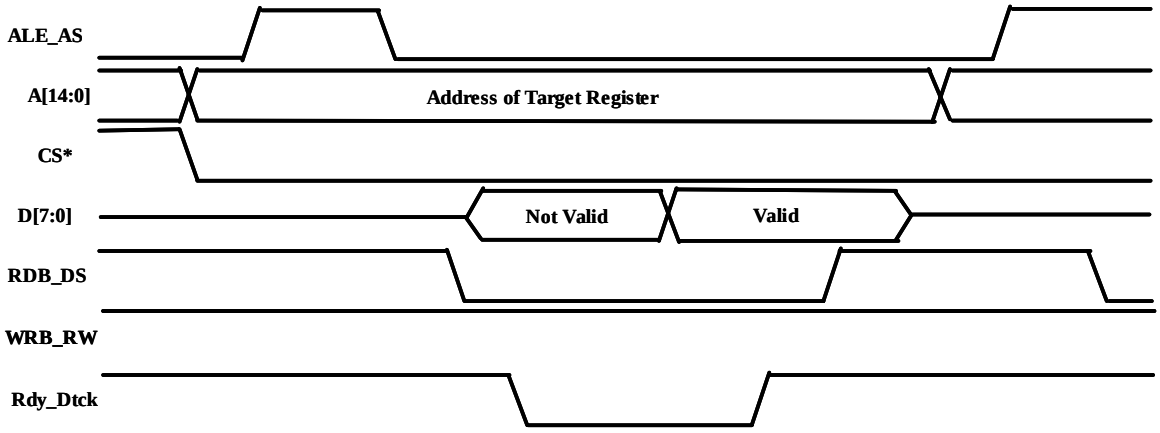
6. Immediately after the μC/μP toggles the "Read Strobe" signal "low", the XRT94L33 will toggle the Rdy_Dtck output pin "low". The XRT94L33 does this in order to inform the μC/μP that the data (to be read from the data bus) is "NOT READY" to be "latched" into the μC/μP.

7. After some settling time, the data on the "bi-directional" data bus will stabilize and can be read by the μC/μP. The XRT94L33 will indicate that this data can be read by toggling the Rdy_Dtck (READY) signal "high".

8. After the μC/μP detects the Rdy_Dtck signal (from the XRT94L33 UNI), it can then terminate the Read Cycle by toggling the RdB_DS (Read Strobe) input pin "high".

Figure 4 presents a timing diagram which illustrates the behavior of the Microprocessor Interface signals, during an "Intel-type" Programmed I/O Read Operation.

Figure 4 Behavior of Microprocessor Interface signals during an "Intel-type" Programmed I/O Read Operation



1.3.3.2 THE INTEL MODE WRITE CYCLE

Whenever an Intel-type μC/μP wishes to write a byte or word of data into a register or buffer location, within the XRT94L33, it should do the following.

1. Assert the ALE_AS (Address Latch Enable) input pin by toggling it "high". When the μC/μP asserts the ALE_AS input pin, it enables the "Address Bus Input Drivers" within the XRT94L33 chip.

2. Place the address of the "target" register or buffer location within the XRT94L33, on the Address Bus input pins, A[14:0].

3. While the μC/μP is placing this address value onto the Address Bus, the Address Decoding circuitry (within the user's system) should assert the CS* input pin of the XRT94L33 by toggling it "low". This step enables further communication between the μC/μP and the XRT94L33 Microprocessor Interface block.

4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time); the μC/μP should toggle the ALE_AS input pin "low". This step causes the XRT94L33 to "latch" the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer location within the XRT94L33, has now been selected.

5. Next, the μC/μP should indicate that this current bus cycle is a "Write" Operation; by toggling the WRB_RW (Write Strobe) input pin "low". This action also enables the "bi-directional" data bus input drivers of the XRT94L33.

6. The μC/μP should then place the byte or word that it intends to write into the "target" register, on the bi-directional data bus, D[7:0].

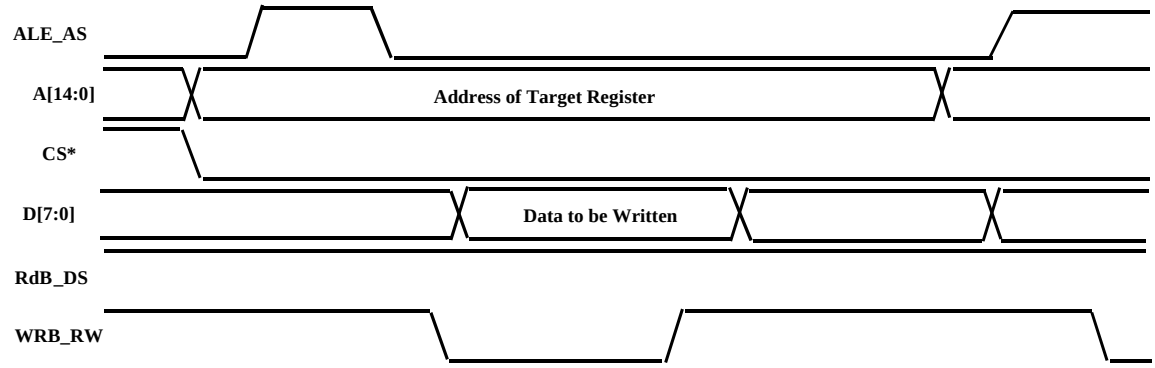
7. After waiting the appropriate amount of time, for the data (on the bi-directional data bus) to settle; the μC/μP should toggle the WRB_RW (Write Strobe) input pin "high". This action accomplishes two things:

a. It latches the contents of the bi-directional data bus into the XRT94L33 Microprocessor Interface block.

b. It terminates the write cycle.

Figure 5 presents a timing diagram which illustrates the behavior of the Microprocessor Interface signals, during an "Intel-type" Programmed I/O Write Operation.

Figure 5 Behavior of the Microprocessor Interface Signals, during an "Intel-type" Programmed I/O Write Operation



1.3.4 PROGRAMMED I/O ACCESS IN THE MOTOROLA MODE

If the XRT94L33 is interfaced to a "Motorola-type" μC/μP (e.g., the MC680X0 family, etc.); it should be configured to operate in the "Motorola" mode. Motorola-type Programmed I/O "Read" and "Write" operations are described below.

1.3.4.1 THE MOTOROLA MODE READ CYCLE

Whenever a "Motorola-type" μC/μP wishes to read the contents of a register or some location within the Receive J0 or J1 Message Buffer, within the XRT94L33 it should do the following.

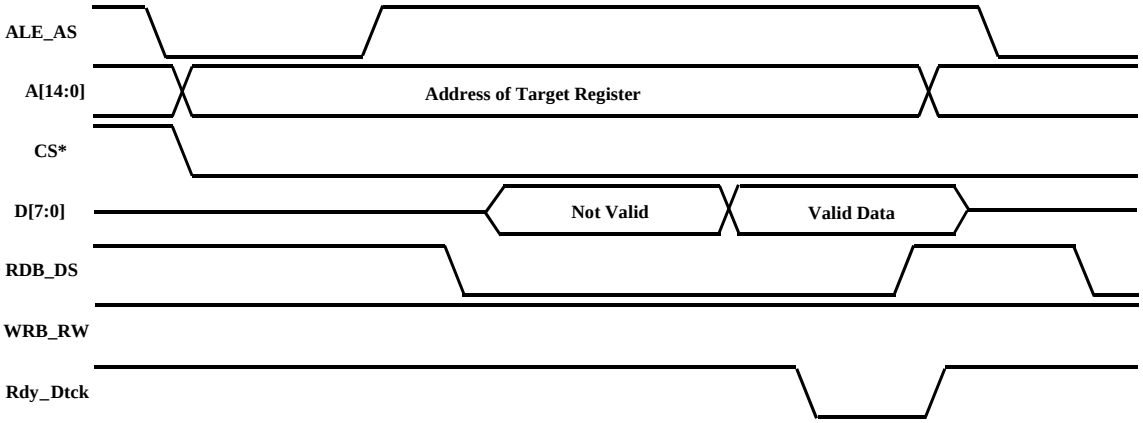
1. Assert the ALE_AS (Address-Strobe) input pin by toggling it low. This step enables the Address Bus input drivers, within the Microprocessor Interface Block of the XRT94L33 IC.
2. Place the address of the "target" register (or buffer location) within the XRT94L33, on the Address Bus input pins, A[14:0].
3. At the same time, the Address Decoding circuitry (within the user's system) should assert the CS* (Chip Select) input pin of the XRT94L33, by toggling it "low". This action enables further communication between the μC/μP and the XRT94L33 Microprocessor Interface block.
4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time), the μC/μP should toggle the ALE_AS input pin "high". This step causes the XRT94L33 to latch the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer location within the XRT94L33 has now been selected.
5. Further, the μC/μP should indicate that this cycle is a "Read" cycle by setting the WRB_RW (R/W*) input pin "high".
6. Next the μC/μP should initiate the current bus cycle by toggling the RdB_DS (Data Strobe) input pin "low". This step enables the bi-directional data bus output drivers, within the XRT94L33 UNI device. At this point, the bi-directional data bus output drivers will proceed to driver the contents of the "Address" register onto the bi-directional data bus, D[7:0].

After some settling time, the data on the "bi-directional" data bus will stabilize and can be read by the μC/μP. The XRT94L33 UNI will indicate that this data can be read by asserting the Rdy_Dtck (DTACK) signal.

After the μC/μP detects the Rdy_Dtck signal (from the XRT94L33 UNI) it will terminate the Read Cycle by toggling the "RdB_DS" (Data Strobe) input pin "high".

Figure 6 presents a timing diagram which illustrates the behavior of the Microprocessor Interface signals during a "Motorola-type" Programmed I/O Read Operation.

Figure 6 Illustration of the Behavior of Microprocessor Interface signals, during a "Motorola-type" Programmed I/O Read Operation



1.3.4.2 THE MOTOROLA MODE WRITE CYCLE

Whenever a Motorola-type μC/μP wishes to write a byte or word of data into a register or buffer location, within the UNI, it should do the following.

1. Assert the ALE_AS (Address Select) input pin by toggling it "low". This step enables the "Address Bus" input drivers (within the UNI chip).

2. Place the address of the "target" register or buffer location (within the UNI), on the Address Bus input pins, A[14:0].

3. While the μC/μP is placing this address value onto the Address Bus, the Address-Decoding circuitry (within the user's system) should assert the CS* (Chip Select) input pins of the UNI by toggling it "low". This step enables further communication between the μC/μP and the UNI Microprocessor Interface block.

4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time), the μC/μP should toggle the ALE_AS input pin "high". This step causes the UNI device to "latch" the contents of the "Address Bus" into its own circuitry. At this point, the Address of the register or buffer location (within the UNI), has now been selected.

5. Further, the μC/μP should indicate that this current bus cycle is a "Write" operation by toggling the WRB_RW (R/W*) input pin "low".

6. The μC/μP should then place the byte or word that it intends to write into the "target" register, on the bi-directional data bus, D[15:0].

7. Next, the μC/μP should initiate the bus cycle by toggling the RdB_DS (Data Strobe) input pin "low". When the XRT94L33 senses that the WRB_RW (R/W*) input pin is "high" and that the RdB_DS (Data Strobe) input pin has toggled "low", it will enable the "input drivers" of the bidirectional data bus, D[7:0].

8. After waiting the appropriate time, for this newly placed data to settle on the bi-directional data bus (e.g., the "Data Setup" time) the UNI will assert the Rdy_Dtck output signal.

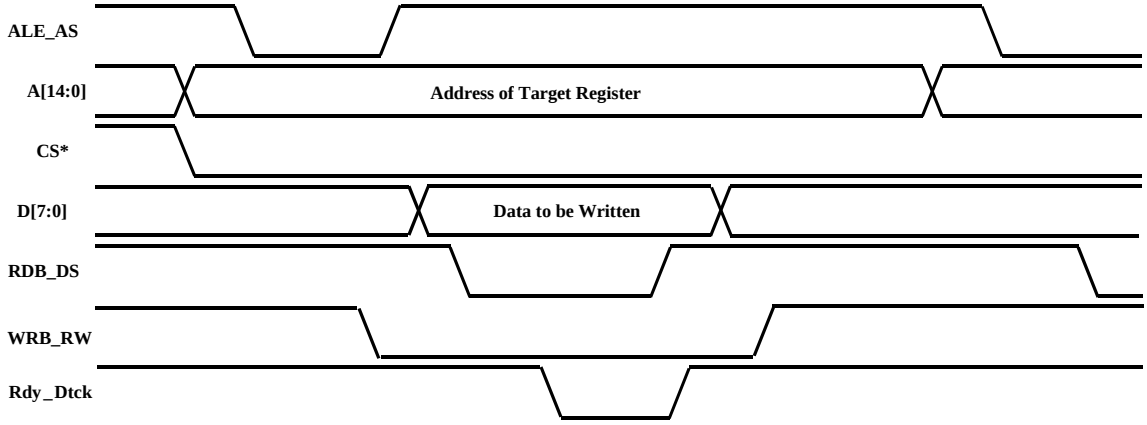
9. After the μC/μP detects the Rdy_Dtck signal (from the UNI), the μC/μP should toggle the RdB_DS input pin "high". This action accomplishes two things.

a. It latches the contents of the bi-directional data bus into the XRT94L33 Microprocessor Interface block.

b. It terminates the "Write" cycle.

Figure 7 presents a timing diagram which illustrates the behavior of the Microprocessor Interface signals, during a "Motorola-type" Programmed I/O Write Operation.

Figure 7 Illustration of the Behavior of the Microprocessor Interface signal, during a "Motorola-type" Programmed I/O Write Operation



1.4 INTERRUPT STRUCTURE WITHIN THE XRT94L33 ATM UNI/PPP IC

The XRT94L33 is equipped with a sophisticated Interrupt Servicing Structure. This Interrupt Structure includes an Interrupt Request output pin, (INT*), numerous Interrupt Enable Registers and numerous Interrupt Status Registers. The Interrupt Servicing Structure, within the XRT94L33 IC contains three levels of hierarchy. The top level is at the "Operational block level". The second level is at the "Functional Block" level (e.g., the Operation Control Block, Receive ATM Cell Processor Block, Receive PPP Processor Block, the Receive STS-3 TOH Processor Block, the Receive STS-3c/SONET POH Processor Block). The lower hierarchical level is at the individual interrupt or "source" level. Each hierarchical level consists of a complete set of Interrupt Status Registers/bits and Interrupt Enable Registers/bits, as will be discussed below.

Most of the functional blocks within the XRT94L33 are capable of generating Interrupt Requests to the μC/μP. The XRT94L33 Interrupt Structure has been carefully designed to allow the user to quickly determine the exact source of the interrupt (with a minimum number of read operations, and, in-turn, minimal latency) which will aid the μC/μP in determining the appropriate interrupt service routine to call up in order to either eliminate, or properly respond to the condition(s) causing the interrupt.

Table 6 lists all of the possible conditions that can generate interrupts in each functional block of the XRT94L33.

Table 6 List of all of the Possible Conditions that can Generate Interrupts within the XRT94L33 ATM UNI/PPP Device

Functional Block	Interrupt Condition
Operation Control Block	• STS-3 Telecom Bus – Detection of Parity Error
Receive STS-3 TOH Processor Block	• Change in the LOS (Loss of Signal) Defect Condition • Change in the LOF (Loss of Frame) Defect Condition • Change in the SEF (Severely Erred Frame) Defect Condition • Change in the AIS-L (Line AIS) Defect Condition • Change in the RDI-L (Line Remote Defect Indicator) Condition • Change in the SD (Signal Degrade) Defect Condition • Change in the SF (Signal Failure) Defect Condition • Detection of REI-L (Line Remote Error Indicator) Defect Condition • Detection of B1 Byte Error • Detection of B2 Byte Error • Detection of New S1 Byte • Change in S1 Byte Unstable Condition • Change in J0 (Section Trace) Message Unstable Condition • Detection of New J0 (Section Trace) Message • Change in J0 – Section Trace Message Mismatch Condition • TOH (Buffer) Capture Complete • Change in K1, K2 Byte Unstable Condition • Detection of New K1, K2 Byte
Receive STS-3c/SONET POH Processor Block	• Detection of B3 Byte Error • Detection of New Pointer • Detection of an Unknown Pointer • Detection of Pointer Decrement Event • Detection of Pointer Increment Event • Detection of NDF Pointer • Change in the Loss of Pointer (LOP-P) Defect Condition • Change in the AIS-P (Path – AIS) Defect Condition • Detection of New J1 (Path Trace) Message • Detection of REI-P (Path Remote Error Indicator) Event • Change in the UNEQ-P Defect Condition • Change in the PLM-P Defect Condition • Detection of New C2 Byte value

	• Change of “C2 Byte” Unstable Condition • Change in the TIM-P Defect Condition • Change of RDI-P (Path Remote Defect Indicator) Condition • Change in the “RDI-P Unstable” condition • Change in AIS-C (AIS Concatenation) condition • Change in LOP-C (LOP Concatenation) condition
Receive ATM Cell Processor Block	• Receive Cell Extraction Event • Receive Cell Insertion Event • Receive Cell Insertion Memory Overflow Condition • Receive Cell Extraction Memory Overflow Condition • RxFIFO Overflow • Detection of Correctable HEC Byte Error • Detection of Uncorrectable HEC Byte Error • Change of LCD (Loss of Cell Delineation) Condition • Change of OCD (Out of Cell Delineation) Condition
Transmit ATM Cell Processor Block	• Transmit Cell Extraction Event • Transmit Cell Insertion Event • Transmit Cell Insertion Memory Overflow Condition • Transmit Cell Extraction Memory Overflow Condition • TxFIFO Overflow • Detection of HEC Byte Error • Detection of Transmit UTOPIA Parity Error
Receive PPP Processor Block	• Detection of Receive FIFO Overflow Condition • Detection of FCS (Frame Check Sequence) Error • Detection of ABORT Sequence • Detection of RUNT Packet
Transmit PPP Processor Block	• Detection of Transmit FIFO Underflow Condition • Detection of Parity Error

The XRT94L33 ATM UNI/PPP comes equipped with the following registers to support the servicing of this wide array of potential “interrupt request” sources.

Table 7 A Listing of the XRT94L33 ATM UNI/PPP Device Interrupt Block Registers

REGISTER	ADDRESS LOCATION
Operation Interrupt Status Register – Byte 0	0x010B
Operation Interrupt Enable Register – Byte 0	0x010F
Operation Block Interrupt Status Register – Byte 1	0x0112
Operation Block Interrupt Status Register – Byte 0	0x0113
Operation Block Interrupt Enable Register – Byte 1	0x0116
Operation Block Interrupt Enable Register – Byte 0	0x0117
Receive STS-3 Transport Interrupt Status Register – Byte 2	0x1109
Receive STS-3 Transport Interrupt Status Register – Byte 1	0x110A
Receive STS-3 Transport Interrupt Status Register – Byte 0	0x110B
Receive STS-3 Transport Interrupt Enable Register – Byte 2	0x110D
Receive STS-3 Transport Interrupt Enable Register – Byte 1	0x110E
Receive STS-3 Transport Interrupt Enable Register – Byte 0	0x110F
Receive STS-3c Path – Interrupt Status Register – Byte 2	0x1189
Receive STS-3c Path – Interrupt Status Register – Byte 1	0x118A
Receive STS-3c Path – Interrupt Status Register – Byte 0	0x118B
Receive STS-3c Path – Interrupt Enable Register – Byte 2	0x118D
Receive STS-3c Path – Interrupt Enable Register – Byte 1	0x118E
Receive STS-3c Path – Interrupt Enable Register – Byte 0	0x118F
Redundant Receive STS-3 Transport Interrupt Status Register – Byte 2	0x1709
Redundant Receive STS-3 Transport Interrupt Status Register – Byte 1	0x170A
Redundant Receive STS-3 Transport Interrupt Status Register – Byte 0	0x170B
Redundant Receive STS-3 Transport Interrupt Enable Register – Byte 2	0x170D
Redundant Receive STS-3 Transport Interrupt Enable Register – Byte 1	0x170E
Redundant Receive STS-3 Transport Interrupt Enable Register – Byte 0	0x170F
Receive SONET POH Interrupt Status Register – Byte 2 – Channel N-1	0xN189
Receive SONET POH Interrupt Status Register – Byte 1 – Channel N-1	0xN18A
Receive SONET POH Interrupt Status Register – Byte 0 – Channel N-1	0xN18B
Receive SONET POH - Path Interrupt Enable Register – Byte 2 – Channel N-1	0xN18D
Receive SONET POH - Path Interrupt Enable Register – Byte 1 – Channel N-1	0xN18E

Table 8 A Listing of the XRT94L33 ATM UNI/PPP Device Interrupt Block Registers

REGISTER	ADDRESS LOCATION
Receive SONET POH - Path Interrupt Enable Register – Byte 0 – Channel N-1	0xN18F
DS3/E3 Framer Block – Block Interrupt Enable Register – Channel N-1	0xN304
DS3/E3 Framer Block – Block Interrupt Status Register – Channel N-1	0xN305
DS3/E3 Framer Block – RxDS3/E3 Interrupt Enable Register – Channel N-1	0xN312
DS3/E3 Framer Block – RxDS3 Interrupt Status Register – Channel N-1 DS3/E3 Framer Block – RxE3 Interrupt Enable Register # 2 – Channel N-1	0xN313
DS3/E3 Framer Block – RxE3 Interrupt Status Register # 1 – Channel N-1	0xN314
DS3/E3 Framer Block – RxE3 Interrupt Status Register # 2 – Channel N-1	0xN315
DS3/E3 Framer Block –RxDS3 FEAC Interrupt Enable/Status Register – Channel N-1	0xN317
DS3/E3 Framer Block – Receive PLCP Interrupt Enable Register – Channel N-1	0xN391
DS3/E3 Framer Block – Receive PLCP Interrupt Status Register – Channel N-1	0xN392
Receive ATM Interrupt Status Register – Byte 1 – Channel N-1	0xN70A
Receive ATM Interrupt Status Register – Byte 0 – Channel N-1 Receive PPP Interrupt Status Register – Channel N-1	0xN70B
Receive ATM Interrupt Enable Register – Byte 1 – Channel N-1	0xN70E
Receive ATM Interrupt Enable Register – Byte 0 – Channel N-1 Receive PPP Interrupt Enable Register – Channel N-1	0xN70F
Receive Mapper Interrupt Status Register – Byte 0 – Channel N-1	0xNB0B
Receive Mapper Interrupt Enable Register – Byte 0 – Channel N-1	0xNB0F
Transmit ATM Interrupt Status Register – Channel N-1 Transmit PPP Interrupt Status Register – Channel N-1	0xNF0B
Transmit ATM Interrupt Enable Register – Channel N-1 Transmit PPP Interrupt Enable Register- Channel N-1	0xNF0F

Note: *In Table _, the value of N ranges from 1 (which corresponds with Channel 0) to 3 (which corresponding with Channel 2).*

1.4.1 GENERAL FLOW OF XRT94L33 ATM UNI/PPP DEVICE INTERRUPT SERVICING

When any of the conditions, presented in Figure 6 occurs (if their Interrupt is enabled), then the XRT94L33 will generate an interrupt request to the μP/μC by asserting the active-low interrupt request output pin, INT*. Shortly after the μP/μC has detected the activated INT* signal, it will enter into the appropriate "user-supplied" interrupt service routine. The first task, for the μP/μC, while running this interrupt service routine, may be to isolate the source of the interrupt request down to the device level (e.g., the XRT94L33 ATM UNI/PPP device), if multiple peripheral devices exist in the user's system.

However, once the "interrupting peripheral" device has been identified and determined to be the XRT94L33, the next task for the μP/μC is to identify the functional block within the XRT94L33 requested the interrupt. Finally, the μP/μC will need to proceed further and identify the exact condition(s) causing the interrupt to be generated by the XRT94L33.

The procedure for servicing the "XRT94L33" Interrupts is best achieved by executing the following steps.

STEP 1 – DETERMINE THE FUNCTIONAL BLOCK(S) REQUESTING THE INTERRUPT

If the interrupting device turns out to be the XRT94L33 ATM UNI/PPP IC, then the μC/μP must determine which "functional block" requested the interrupt. Hence, upon reaching this state, one of the very first things that the μC/μP must do within the user supplied "XRT94L33" interrupt service routine, is to perform a read of both of the following registers.

- Operation Block Interrupt Status Register – Byte 1 (Address = 0x0112)
- Operation Block Interrupt Status Register – Byte 0 (Address = 0x0113)

The bit-format of each of these registers is presented below.

Operation Block Interrupt Status Register – Byte 1 (Address Location = 0x0112)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Op Control Block Interrupt Status	DS3/E3 Mapper Block Interrupt Status	Unused	Receive STS-1 TOH Processor Block Interrupt Status	Receive STS-1 POH Processor Block Interrupt Status	DS3/E3 Framer Block Interrupt Status	Receive Line Interface Block Interrupt Status	Unused
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Operation Block Interrupt Status Register – Byte 0 (Address Location = 0x0113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Status	Receive STS-3 TOH Processor Block Interrupt Status	Receive SONET/ STS-3c POH Processor Block Interrupt Status	Receive PPP Processor Block Interrupt Status	Transmit ATM Cell Processor Block Interrupt Status	Unused		Transmit PPP Processor Block Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Each of the "Operation Block Interrupt Status" Registers presents the "interrupt-request" status of each functional block, within the chip. The purpose of these two registers is to help the μC/μP identify which functional block(s) has requested the interrupt. Whichever bit(s) are asserted, in this register, identifies which block(s) have experienced an "interrupt-generating" condition as presented in Table _. Once the μC/μP has read this register, it can determine which "branch" within the interrupt service routine that it must follow in order to properly service this interrupt.

The XRT94L33 ATM UNI/PPP IC further supports the Operational Block hierarchy by providing the Operation Block Interrupt Enable Register – Bytes 1 and 0. The bit format of these two registers are identical to that for the Operation Block Interrupt Status Registers – Bytes 1 and 0, and are presented below for the sake of completeness.

Operation Block Interrupt Enable Register – Byte 1 (Address Location = 0x0116)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Op Control Block Interrupt Enable	DS3/E3 Mapper Block Interrupt Enable	Unused	Receive STS-1 TOH Processor Block Interrupt Enable	Receive STS-1 POH Processor Block Interrupt Enable	DS3/E3 Framer Block Interrupt Enable	Receive Line Interface Block Interrupt Enable	Unused
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/O
0	0	0	0	0	0	0	0

Operation Block Interrupt Enable Register – Byte 0 (Address Location = 0x0117)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Enable	Receive STS-3 TOH Processor Block Interrupt Enable	Receive SONET/ STS-3c POH Processor Block Interrupt Enable	Receive PPP Processor Block Interrupt Enable	Transmit ATM Cell Processor Block Interrupt Enable	Unused		Transmit PPP Processor Block Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	0	0	0	0	0	0

These Operation Block Interrupt Enable registers permit the user to individually enable or disable the interrupt requesting capability of the functional blocks within the XRT94L33. If a particular bit-field, within this register contains the value "0", then the corresponding functional block has been disabled for generating any interrupt requests. Conversely, if that bit-field contains the value "1"; then the corresponding functional block has been enabled for interrupt generation (e.g., those potential interrupts, within the "enabled functional block" that are enabled at the source level are now enabled). The user should be aware of the fact that each functional block, within the XRT94L33 contains multiple potential interrupt sources. Each of these lower level interrupt sources contain their own set of interrupt enable bits and interrupt status bits, existing in various on-chip registers.

STEP 2 – INTERRUPT SERVICE ROUTING BRANCHING: AFTER READING THE OPERATION BLOCK INTERRUPT STATUS REGISTERS

The contents of the Operation Block Interrupt Status Registers permit the user to identify which of the seven (7) functional blocks (within the XRT94L33 IC) have requested interrupt service. The μP/μC should use this information in order to determine where, within the Interrupt Service Routine, program control should branch to. The following table can be viewed as an "interrupt service routine" guide. It lists each of the Functional Blocks that contain bit-field in the Operation Block Interrupt Status and Enable registers. Additionally, this table also presents a list and addresses of the corresponding on-chip Registers that the Interrupt Service Routine should branch to and read; based upon the Interrupt Functional Block.

Table 9 Interrupt Service Routine Guide for the XRT94L33

INTERRUPTING FUNCTIONAL BLOCK	THE NEXT REGISTER TO BE READ DURING THE INTERRUPT SERVICE ROUTINE	ADDRESS LOCATION
Operation Control Block	Operation Interrupt Status Register – Byte 0	0x010B
Receive ATM Cell Processor Block	Operation Channel Interrupt Indicator – Receive ATM Cell Processor Block	0x0128
Receive SONET/STS-3c POH Processor Block	Operation Channel Interrupt Indicator – Receive SONET POH Processor Block	0x0120
Receive STS-3 TOH Processor Block	Receive STS-3 Transport Interrupt Status Register – Byte 2	0x1109
	Receive STS-3 Transport Interrupt Status Register – Byte 1	0x110A
	Receive STS-3 Transport Interrupt Status Register – Byte 0	0x110B
Receive PPP Packet Processor Block	Operation Channel Interrupt Indicator – Receive PPP Packet Processor Block	0x012A
Transmit ATM Cell Processor Block	Operation Channel Interrupt Indicator – Transmit ATM Cell Processor Block	0x0127
Transmit PPP Packet Processor Block	Operation Channel Interrupt Indicator – Transmit PPP Packet Processor Block	0x0129
DS3/E3 Mapper Block	Operation Channel Interrupt Indicator – DS3/E3 Mapper Block	0x0126
DS3/E3 Framer Block	Operation Channel Interrupt Indicator – DS3/E3 Framer Block	0x0122

Note: *Registers associated within each functional block are specified in ascending order (based upon the on-chip Address Location). No other inferences should be made regarding the order in which these registers are presented in this table.*

Once the μC/μP has read out the contents of the appropriate register (as listed above in Table 9); then there may (or may not) be additional "interrupt status" registers to read; as described below.

Interrupt Servicing for the "Operation Control" Block

If the interrupt service routine is currently servicing an "Operation Control" Block Interrupt, then reading out the contents of the corresponding register (as presented in Table 9) should result in the following occurrences.

1. The μC/μP will uniquely identify the source or condition causing the interrupt request.
2. The "asserted interrupt status" bit-fields within this register will be reset upon read.
3. The "asserted" bit-field(s), within the Operation Block Interrupt Status Register will be reset.

4. The XRT94L33 will negate the "INT*" (Interrupt Request) output pin.

Once these events have occurred, then (as far as the XRT94L33 is concerned) the interrupt has been serviced.

1.4.1.1 INTERRUPT SERVICING FOR THE "RECEIVE STS-3 TOH PROCESSOR" BLOCK

If the interrupt service routine is currently servicing an "Receive STS-3 TOH Processor" Block Interrupt, then reading out the contents of the corresponding register (as presented in Table _) should result in the following occurrences.

1. The μC/μP will uniquely identify the source or condition causing the interrupt request.
2. The "asserted interrupt status" bit-fields within these registers will be reset upon read.
3. The "asserted" bit-field(s), within the Operation Block Interrupt Status Register will be reset.
4. The XRT94L33 will negate the "INT*" (Interrupt Request) output pin.

Once these events have occurred, then (as far as the XRT94L33 is concerned) the interrupt has been serviced.

The remainder of the user's interrupt service routine should now execute the appropriate steps to respond to the conditions causing this interrupt request.

1.4.1.2 INTERRUPT SERVICING FOR THE REMAINING BLOCKS WITHIN THE XRT94L33

If the interrupt service routine is currently servicing interrupts associated with the following blocks.

- Receive ATM Cell Processor Block
- Receive PPP Packet Processor Block
- Receive STS-3c/SONET POH Processor Block
- Transmit ATM Cell Processor Block
- Transmit PPP Packet Processor Block
- DS3/E3 Mapper Block
- DS3/E3 Framer Block

Then there are still more steps that the μP/μC must take in order to fully service these interrupts.

STEP 3 – IDENTIFY THE INTERRUPTING CHANNEL

For each of the "above-mentioned" blocks, the user was advised (in Table 9) to read out the contents of a "Channel Interrupt Indicator" Register. This register will uniquely identify the "interrupting" channel.

For example, the bit-format of the "Operation Channel Interrupt Indicator – Receive STS-3c/SONET POH Processor Block" register is presented below.

Operation Channel Interrupt Indicator – Receive STS-3c/SONET POH Processor Block (Address = 0x0120)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Receive STS-3c POH Processor Block – Channel Interrupt Indication Status	Unused	Receive SONET POH Processor Block - Channel 2 Interrupt Indication Status	Receive SONET POH Processor Block - Channel 1 Interrupt Indication Status	Receive SONET POH Processor Block - Channel 0 Interrupt Indication Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

If a given bit-field is set to "1", then the "Receive STS-3c/SONET POH Processor" block, (within the corresponding channel) is generating an interrupt request. Conversely, if a given bit-field is set to "0", then the "Receive STS-3c/SONET POH Processor" blocks (within the corresponding channel) is not generating an interrupt request.

The "Operation Channel Interrupt Indicator" registers for the other functional blocks are virtually identical to that presented for the "Receive STS-3c/SONET POH Processor" block.

STEP 4 – PROCEED TO READ OUT THE APPROPRIATE "SOURCE-LEVEL" INTERRUPT STATUS REGISTERS; CORRESPONDING TO THE "INTERRUPTING" CHANNEL.

Now that the "interrupting" functional block and channel has been identified, the μP/μC should now go off and read out the contents of the appropriate "source-level" interrupt status register.

Table 10 presents a list of "functional blocks" and the corresponding "Source-Level" interrupt status registers that the μC/μP should read, once the "interrupting" channel (N) has been identified.

Table 10 List of "Source-Level" Interrupt Status Registers that should be read, once "Interrupting" Channel has been identified

INTERRUPTING FUNCTIONAL BLOCK	THE NEXT REGISTER TO BE READ DURING THE INTERRUPT SERVICE ROUTING	ADDRESS LOCATION
Receive ATM Cell Processor Block	Receive ATM Interrupt Status Register – Byte 1 – Channel N-1	0xN70A
	Receive ATM Interrupt Status Register – Byte 0 – Channel N-1	0xN70B
Receive PPP Packet Processor Block	Receive PPP Interrupt Status Register – Channel N-1	0xN70B
Receive SONET POH Processor Block	Receive SONET POH Interrupt Status Register – Byte 2 – Channel N-1	0xN189
	Receive SONET POH Interrupt Status Register – Byte 1 – Channel N-1	0xN18A
	Receive SONET POH Interrupt Status Register – Byte 0 – Channel N-1	0xN18B
Receive STS-3c POH Processor Block	Receive STS-3c Path Interrupt Status Register – Byte 2	0x1189
	Receive STS-3c Path Interrupt Status Register – Byte 1	0x118A
	Receive STS-3c Path Interrupt Status Register – Byte 0	0x118B
Transmit ATM Cell Processor Block	Transmit ATM Interrupt Status Register – Channel N-1	0xNF0B

Note: *The value "N" (within the address locations of these registers) ranges in value from 1 to 3, and corresponding to "interrupting" channel N-1.*

1.5 AN OVERVIEW OF POSSIBLE CONFIGURATION OPTIONS IN THE XRT94L33

The XRT94L33 can be configured to function a wide variety of operating modes. This section briefly summarizes and describes the procedure that one should employ in order to configure the XRT94L33 into each of these modes.

In all, the XRT94L33 can be configured to function in any of the following modes.

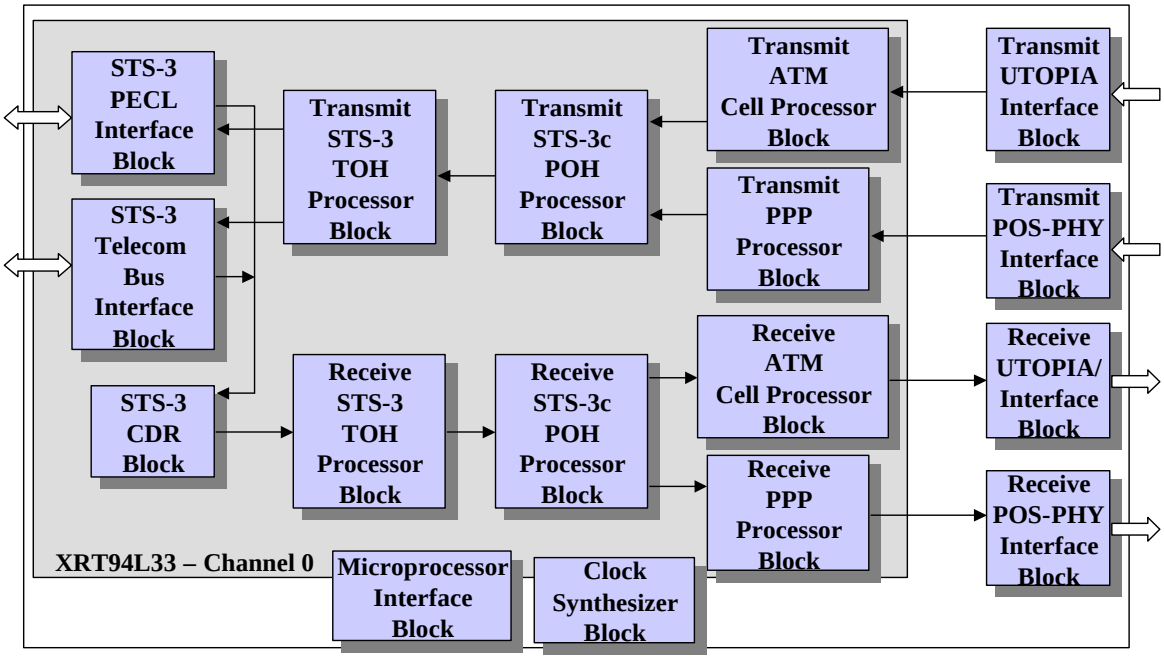
- 1 Channel STS-3c and up to 2 Channel DS3/E3 ATM UNI/PPP Mode
- 1-Channel STS-3 (e.g., 3 Channels of ATM UNI/PPP over STS-1, which is in-turn is mapped into STS-3) Mode
- 1-Channel STS-3 (e.g., 3 Channels of ATM UNI/PPP over DS3/E3, which is in-turn, mapped into STS-3) Mode

Each of these operating modes is described below.

1.5.1 1-CHANNEL STS-3C AND UP TO 2 CHANNELS DS3/E3 ATM UNI/PPP MODE

If the user configures the XRT94L33 to operate in this mode, then the XRT94L33 will function as an ATM UNI/PPP device that supports ATM cell or PPP packet transmission and reception over one STS-3c channel and possibly as many as two additional DS3/E3 channels, in parallel. Since the UTOPIA and POS-PHY interfaces, within the XRT94L33 support Multi-PHY Operation, the ATM Layer Processor will have no problem writing ATM Cell/PPP Packet data into and reading ATM Cell/PPP Packet data from the UTOPIA/POS-PHY Interface.

Figure 8 presents the functional block diagram of the XRT94L33, if it is configured to operate in this mode.



SOME NOTES ABOUT THE ABOVE FIGURE

Figure 8 indicates that the XRT94L33 is functioning as a 1-Channel STS-3c ATM UNI/PPP device. The existence of the two DS3/E3 channels are not depicted in this illustration. Nonetheless, whenever the XRT94L33 is operating in this mode, these additional DS3/E3 channels are available for the transmission of ATM cell or PPP packet.

Configuring the XRT94L33 to operate in the "1-Channel STS-3c (with 0 to 2-Channel DS3/E3) ATM UNI/PPP Mode.

To configure the XRT94L33 to operate in this mode execute the following two steps.

STEP 1 - SET BITS 1 AND 0 (CONF[1:0]), WITHIN THE OPERATION CONTROL REGISTER – BYTE 3; TO [0, 1]; AS DEPICTED BELOW.

Operation Control Register – Byte 3 (Address = 0x0100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CH_MODE_SEL[3:0]				Dual Bus	Unused	CONF[1:0]	
R/W	R/W	R/W	R/W	R/W	R/O	R/W	R/W
X	X	X	X	X	0	0	0

STEP 2 – SET BIT 2 (PPP/ATM*), WITHIN THE "OPERATION CONTROL REGISTER- BYTE 0" TO THE APPROPRIATE VALUE AS DEPICTED BELOW.

Operation Control Register – Byte 0 (Address = 0x0103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit UTOPIA PLL OFF	Receive UTOPIA PLL OFF	Unused			PPP/ATM*	Unused	SW RESET
R/W	R/W	R/O	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	X	0	0

Setting this bit-field to "0" configures the XRT94L33 to operate in the ATM UNI Mode. Conversely, setting this bit-field to "1" configures the XRT94L33 to operate in the PPP Mode.

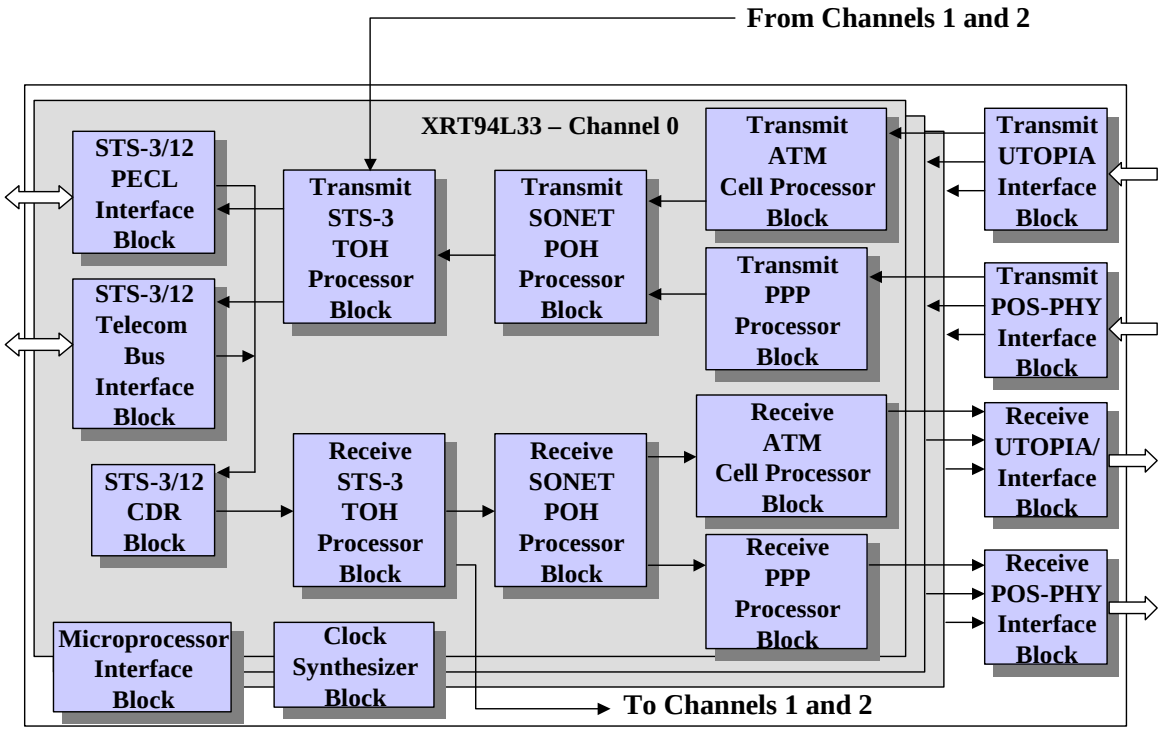
Please refer to Section 4.0 for a description on how to configure the XRT94L33 to operate in the ATM UNI Mode. Further, please refer to Section 5.0 for a description on how to configure the XRT94L33 to operate in the PPP Mode.

1.5.2 1-CHANNEL STS-3 ATM UNI/PPP MODE

If the user configures the XRT94L33 to operate in this mode, then the XRT94L33 will function as a 1-Channel STS-3 ATM UNI/PPP device.

Figure 9 presents the functional block diagram of the XRT94L33, if it is configured to operate in this mode.

Figure 9 Illustration of the XRT94L33, when it is configured to operate in the 1-Channel STS-3 ATM UNI/PPP Mode



Configuring the XRT94L33 to operate in the "1-Channel STS-3 ATM UNI/PPP Mode

To configure the XRT94L33 to operate in this mode execute the following steps.

STEP 1 – SET BITS 1 AND 0 (CONF[1:0]), WITHIN THE OPERATION CONTROL REGISTER – BYTE 3; TO [1, 1]; AS DEPICTED BELOW.

Operation Control Register – Byte 3 (Address = 0x0100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CH_MODE_SEL[3:0]				Dual Bus	Unused	CONF[1:0]	
R/W	R/W	R/W	R/W	R/W	R/O	R/W	R/W
X	X	X	X	X	0	1	1

STEP 2 – SET BIT 2 (PPP/ATM*), WITHIN THE "OPERATION CONTROL REGISTER- BYTE 0" TO THE APPROPRIATE VALUE AS DEPICTED BELOW.

Operation Control Register – Byte 0 (Address = 0x0103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit UTOPIA PLL OFF	Receive UTOPIA PLL OFF	Unused			PPP/ATM*	Unused	SW RESET
R/W	R/W	R/O	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	X	0	0

Setting this bit-field to "0" configures the XRT94L33 to operate in the ATM UNI Mode. Conversely, setting this bit-field to "1" configures the XRT94L33 to operate in the PPP Mode.

Please refer to Section 4.0 for a description on how to configure the XRT94L33 to operate in the ATM UNI Mode. Further, please refer to Section 5.0 for a description on how to configure the XRT94L33 to operate in the PPP Mode.

2.0 ATM UNI MODE OPERATION

If the XRT94L33 (or a given channel within the XRT94L33) has been configured to operate in the ATM UNI Mode, then the following functional blocks will become active.

- The Transmit UTOPIA Interface Block
- The Transmit ATM Cell Processor Block
- The Transmit STS-3c/SONET POH Processor Block
- The Transmit STS-3 TOH Processor Block
- The Receive STS-3 TOH Processor Block
- The Receive STS-3c/SONET POH Processor Block
- The Receive ATM Cell Processor Block
- The Receive UTOPIA Interface Block

Each of these functional blocks will be discussed in considerable detail, over the next few sections.

2.1 CONFIGURING THE XRT94L33 TO OPERATE IN THE ATM UNI MODE

The user can configure a given channel within the XRT94L33 to operate in the ATM UNI Mode by setting Bit 2 (PPP/ATM*) within the Operation Control Register – Byte 0; to "0" as depicted below.

Operation Control Register – Byte 0 (Address = 0x0103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx UTOPIA Clock De-Skew PLL OFF	Rx UTOPIA Clock De-Skew PLL OFF	Transmit STS-3 RESYNC	Reserved		PPP/ ATM*	Reserved	Software Reset
R/W	R/W	R/W	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user executes this step, then the entire XRT94L33 will be configured to operate in the ATM UNI Mode.

2.2 THE TRANSMIT DIRECTION

If a given channel (or the entire device) is configured to operate in the ATM Mode, then the purpose of the Transmit section within the XRT94L33 1-Channel STS-3c/STS-3 ATM UNI device is to allow a local ATM Layer (or ATM Adaptation Layer) processor to transmit ATM Cell data to a remote piece of equipment via an OC-3c or OC-3 transport medium

For ATM UNI Applications, the Transmit Section of the XRT94L33 chip consists of the following functional blocks:

- Transmit UTOPIA Interface Block
- Transmit ATM Cell Processor Block
- Transmit STS-3c POH Processor Block (for STS-3c Applications)
- Transmit SONET POH Processor Block (for STS-3 Applications)
- Transmit STS-3 TOH Processor Block

The ATM Layer processor will write ATM Cell Data to the Transmit UTOPIA Interface Block of the XRT94L33. The Transmit UTOPIA Interface block provides the industry standard ATM/PHY interface functions. The Transmit UTOPIA Interface Block will ultimately write this cell data to an internal FIFO (referred to as TxFIFO throughout this document); where it can be read and further processed by the Transmit ATM Cell Processor Block. The Transmit UTOPIA Interface block will also provide signaling to support data-flow control between the ATM Layer Processor and the Transmit UTOPIA Interface block.

The Transmit ATM Cell Processor block will read in the ATM cell data from the TxFIFO. As the Transmit ATM Cell Processor block reads in this ATM cell data (from the Tx FIFO) it will perform some parity checking on this data. If the Transmit ATM Cell Processor block detects any parity errors in these incoming ATM cells, then these ATM cells are optionally discarded. Afterwards, all "error-free" ATM cell data will then be processed by a "user-defined" (e.g., "Transmit User") cell filter. All cells (with header bytes) that do not comply with the "user-defined" filtering requirements will be (optionally) discarded. Additionally, this Transmit User Cell Filter can be configured to copy and route "replicate" cells (that contain "header-byte" patterns that match the "User-defined" Cell Filtering criteria) to the "Transmit Cell Extraction" Buffer/Processor where these ATM cells can be read out and processed via the Microprocessor Interface. Continuing along the transmit output path, the Transmit ATM Cell Processor block will then (optionally) proceed to take the first four octets of a given cell and compute the HEC (Header Error Check) byte from these bytes. Afterwards the Transmit ATM Cell Processor block will insert this HEC byte into the 5th octet position within the cell. The Transmit ATM Cell Processor block will also (optionally) scramble the payload portion of the cell (bytes 6 through 53) in order to prevent user data from mimicking framing or control bits/bytes. Once the cell has gone through this process it will then be transferred to either of the Transmit STS-3c POH Processor block or the corresponding Transmit SONET POH Processor blocks for further processing.

If the TxFIFO (within the Transmit UTOPIA Interface block) is depleted and has no (user) cells available, then the Transmit ATM Cell Processor block will automatically read out the contents of all cells residing within the "Transmit Cell Insertion" Buffer, and will transmit these cells into the outbound ATM cell traffic via the "Transmit Data Path". Once the "Transmit Cell Insertion Buffer" has been depleted, then the Transmit ATM Cell Processor block will generate, process and transmit Idle cells, in the exact same manner as with user cells. This generation and transmission of Idle cells is also known as cell-rate decoupling (e.g., Idle cells are generated in order to fill up the bandwidth of the PMD carrier requirements – 155.52 Mbps in this case). The Transmit ATM Cell Processor block has provisions to allow the user to generate and transmit certain "user-defined" cells, such as OAM cells via the "Transmit Cell Insertion" Processor block.

Note: *These special or OAM cells will be subjected to the same processing as are User and Idle cells (e.g., HEC Byte Calculation and Insertion, Cell Payload Scrambling).*

The Transmit SONET POH Processor or Transmit STS-3c POH Processor block will accept these ATM cells and will map them into either an STS-1 or STS-3c SPE, respectively. Additionally, the Transmit SONET POH Processor block or the Transmit STS-3c POH Processor block also has responsibility of generating the POH bytes for these SPEs, prior to routing this data to the Transmit STS-3 TOH Processor Block.

The Transmit STS-3 TOH Processor Block will take the STS-1 SPE data (from the Transmit SONET POH Processor block) or the STS-3c SPE (from the Transmit STS-3c POH Processor block) and will generate and overhead the TOH bytes within each outbound STS-3 frame.

The following sections discuss the blocks comprising the Transmitter Portion of the XRT94L33 in detail.

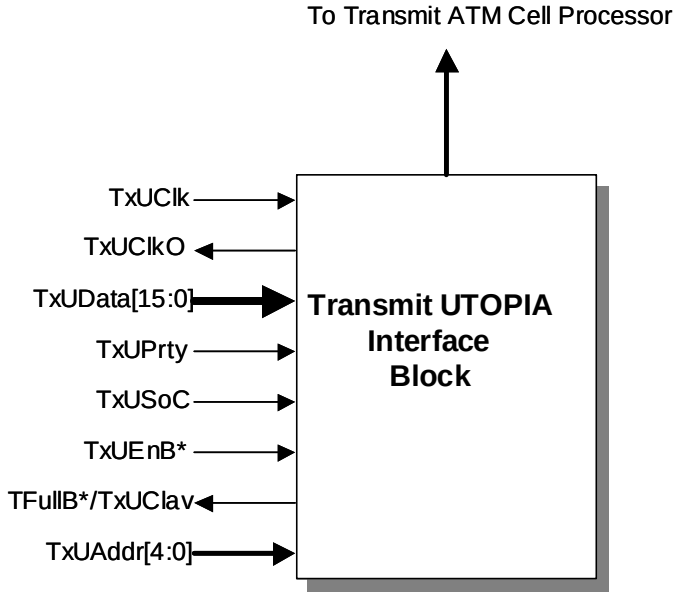
2.2.1 TRANSMIT UTOPIA INTERFACE BLOCK

The Transmit UTOPIA input interface complies with UTOPIA Level 1, 2 and 3 standard interface (e.g., the Transmit UTOPIA can support both Single-PHY and Multi-PHY operations.) Additionally, the XRT94L33 provides the user with the option of varying the following features associated with the Transmit UTOPIA Bus Interface.

- Operating the Transmit UTOPIA Data Bus per the UTOPIA Levels 1, 2 or 3 standards
- Transmit UTOPIA Data Bus width of 8 or 16 bits
- The cell size (e.g., the number of octets being processed per cell via the UTOPIA bus)
- Assigning a UTOPIA Address to a given STS-3c port.
- Whether the Transmit UTOPIA Clock De-Skewing PLL is enabled or not.

Figure 10 presents a simplistic illustration of the Transmit UTOPIA Interface block, along with its external input pins.

Figure 10 A Simple Illustration of the Transmit UTOPIA Interface block



2.2.1.1 THE PINS OF THE TRANSMIT UTOPIA BUS INTERFACE

The ATM Layer processor will interface to the Transmit UTOPIA Interface block via the following pins.

- TxUData[15:0] - Transmit UTOPIA Data Bus Input pins
- TxUAddr[4:0] - Transmit UTOPIA Address Bus Input pins
- TxUClk Transmit UTOPIA Interface block clock input pin
- TxUClkO Transmit UTOPIA Interface block clock output pin
- TxUSoC Transmit "Start of Cell" indicator input pin
- TxUPrty Transmit UTOPIA - Odd Parity Input pin
- TxUEnB* Transmit UTOPIA Data Bus - Write Enable input pin
- TxUClav/TFullB* TxFIFO Cell Available

Each of these signals is briefly discussed below.

TxUData[15:0] - Transmit UTOPIA Data Bus inputs

The ATM Layer Processor will write its ATM Cell Data into the Transmit UTOPIA Interface block, by placing it, in a byte-wide (or 16-bit word-wide) manner on these input pins. The Transmit UTOPIA Data Bus can be configured to operate in the "8-bit wide" or "16-bit wide" mode (See Section _). If the "8-bit wide" mode is selected, then only the TxUData[7:0] input pins are active and capable of accepting ATM cell data from the ATM Layer Processor. If the "16-bit wide" mode is selected, the all 16 input pins (e.g., TxUData[15:0]) are active. The Transmit UTOPIA Data bus is tri-stated while the active-low TxUEnB* (Transmit UTOPIA Data Bus - Write Enable) input signal is "high". Therefore, the ATM Layer processor must assert this signal (e.g., toggling TxUEnB* "low") in order write the cell data, on the Transmit UTOPIA Data bus, into the Transmit UTOPIA Interface Block. The data on the Transmit UTOPIA Data Bus is sampled and latched into the Transmit UTOPIA Interface block, on the rising edge of the Transmit UTOPIA Interface Block Clock signal, TxUClk.

Additionally, the Transmit UTOPIA Interface block will only process one cell worth of data (e.g., 52, 53 or 54 bytes, as configured via the Cell_Size_Sel[1:0] option - See Section _), following the latest assertion of the TxUSoC (Transmit-Start of Cell) pin. Afterwards, the Transmit UTOPIA Data bus will become tri-stated and will cease to process any more data from the ATM Layer Processor until the next assertion of the TxUSoC pin. Once the Transmit UTOPIA Interface block reaches this condition, it will ignore the assertions of the TxUEnB* pin, and will keep the Transmit UTOPIA Data bus input pins tri-stated until the ATM Layer Processor pulses the TxUSoC input pin, once again.

If the Transmit UTOPIA Interface block detects a "runt" cell (e.g., if the amount of data that is read into the TxUFIFO is less than that configured via the "Cell_Size_Sel[1:0]" option), then the Transmit UTOPIA Interface block will discard this cell, and resume normal operation.

TxUAddr[4:0] - Transmit UTOPIA Address Bus inputs

These input pins are only used when the XRT94L33 is operating in the Multi-PHY mode. Therefore, for more information on the Transmit UTOPIA Address Bus, please see Section _.

TxUClk - Transmit UTOPIA Interface Block - Clock signal input pin

The Transmit UTOPIA Interface block uses this signal to sample and latch the data on the Transmit UTOPIA Data bus into the Transmit UTOPIA Address block (for Multi-PHY operation) into the Transmit UTOPIA Interface block. This clock signal can run at frequencies of up to 50 MHz.

TxUClkO – Transmit UTOPIA Interface Block Clock Output signal

If the "Transmit UTOPIA Clock De-Skewing" PLL is enabled, then the ATM Layer Processor can use this signal to clock out the contents of the "outbound" ATM cells, and to assert the UTOPIA Address (during Multi-PHY operation). Similar to the "TxUClk" signal, this signal can run at frequencies up to 50MHz.

TxUEnB* - Transmit UTOPIA Data Bus - Write Enable input

The Transmit UTOPIA Data Bus is tri-stated while this input signal is negated. Therefore, the ATM Layer Processor must assert this "active-low" signal (toggle it "low") in order to write the byte (or word) on the Transmit UTOPIA Data Bus, into the Transmit UTOPIA Interface block.

TxUPrty - Transmit UTOPIA - Odd Parity Bit input pin

The ATM Layer Processor is expected to compute the odd-parity value of each byte (or word) of ATM Cell data that it intends to place on the Transmit UTOPIA Data bus. The ATM Layer Processor is then expected to apply this parity value at the TxUPrty pin, while the corresponding byte (or word) is present on the Transmit UTOPIA Data Bus. This signal, along with the corresponding byte (or word) of ATM cell data will be routed to the Transmit ATM Cell Processor block for "Transmit UTOPIA parity verification".

TxUSoC - Transmit UTOPIA - "Start of Cell" Indicator

The ATM Layer processor is expected to pulse this input signal "high", for one clock period of TxUClk, when the first byte (or word) of a given ATM cell is present on the Transmit UTOPIA Data Bus. This signal must be kept "low" at all other times.

Note: *Once the ATM Layer Processor has pulsed the TxUSoC pin "high", the Transmit UTOPIA Interface Block will proceed to read in and process only one cell of data (e.g., 52, 53, or 54 bytes, as configured via the "Cell_Size_Sel[1:0]" option - See Section _) via the Transmit UTOPIA Data Bus. Afterwards, the Transmit UTOPIA Interface block will cease to process any more data from the ATM Layer Processor until the TxUSoC pin has been pulsed "high" once again. This phenomenon is more clearly defined in "Example - 1" below.*

Example -1

For example, if the user configures the Transmit UTOPIA Interface block to process 53 bytes per cell; then following the assertion of the TxUSoC pin (which is coincident with the placement of the first byte of the cell on the Transmit UTOPIA Data bus), the Transmit UTOPIA Interface block will read in and process 52 more bytes of data via the Transmit UTOPIA data bus resulting in a total of 53 bytes being processed. After the Transmit UTOPIA Interface block has read in the 53rd byte, it will no longer read in any more data from the ATM Layer Processor, (even if the "TxUEnb*" input pin is pulled "low") until the TxUSoC pin has been asserted.

TxUClav/TFullB* - Tx FIFO Cell Available/TxFIFO Full*

This output signal is used to provide some data flow control between the ATM Layer processor and the Transmit UTOPIA Interface block. Please See Section _ for more information regarding this signal.

2.2.1.2 CONFIGURATION OPTIONS WITH THE TRANSMIT UTOPIA INTERFACE BLOCK

SELECTING THE UTOPIA LEVEL

The XRT94L33 permits the user to configure the Transmit UTOPIA Interface block in either of the following "UTOPIA Levels".

- UTOPIA Level 3
- UTOPIA Level 1 or 2

The user can configure the Transmit UTOPIA Interface block within the XRT94L33 to operate in the appropriate UTOPIA Level, by writing the appropriate value into Bit 7 (UTOPIA Level) within the "Transmit UTOPIA Control Register", as depicted below.

Transmit UTOPIA Control Register – Byte 0, Address = 0x0583

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA/POS-PHY Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	0	0	0	X	X	1	1

Setting this bit-field to "0" configures the Transmit UTOPIA Interface block to support "UTOPIA Level 3" signaling. Conversely, setting this bit-field to "1" configures the Transmit UTOPIA Interface block to support the "UTOPIA Levels 1 and 2" form of signaling. A description of the operation of the Transmit UTOPIA Interface block, for UTOPIA Level 1, 2 and 3 operation is presented below.

2.2.1.3 UTOPIA LEVEL 1 AND 2 OPERATION OF THE TRANSMIT UTOPIA INTERFACE BLOCK

This section presents an in-depth write up of the UTOPIA Level 1 and 2 protocols.

When the Transmit UTOPIA Interface block has been configured to operate in the "UTOPIA Level 2" Mode, then it will either be configured to operate in the "Single-PHY" or "Multi-PHY" mode, as described below.

2.2.1.3.1 Selecting the UTOPIA Data Bus Width

The user can configure the width of the Transmit UTOPIA Data bus to be either 8 or 16 bits by writing the appropriate data into Bits 3 and 2 (Transmit UTOPIA Data Bus Width[1:0]) within the "Transmit UTOPIA Control" Register, as depicted below.

Transmit UTOPIA Control Register – Byte 0, Address = 0x0583

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	X	X	1	1

If the user chooses a UTOPIA Data Bus width of 8 bits, then only the Transmit UTOPIA Data inputs: TxUData[15:8] will be active. (The input pins: TxUData[7:0] will not be active). If the user chooses a UTOPIA Data bus width of 16 bits, then all of the Transmit UTOPIA Data inputs: TxUData[15:0] will be active. The following table relates the value of Bits 2 and 3 (Transmit UTOPIA Data Bus Width[1:0]) within the Transmit UTOPIA Control Register, to the corresponding width of the Transmit UTOPIA Data bus.

Note: *This configuration setting does not apply to the Receive UTOPIA Interface block. The user will still need to specify the width of the Receive UTOPIA Data Bus, separately, as described in Section _.*

Table 11 The Relationship between the contents of "Transmit UTOPIA Data Bus Width[1:0] within the Transmit UTOPIA Control Register and the operating width of the Transmit UTOPIA Data bus

TRANSMIT UTOPIA DATA BUS WIDTH[1:0]	WIDTH OF TRANSMIT UTOPIA DATA BUS
00	In-active:
01	8 bits
10	16 bits
11	Not valid (do not use)

2.2.1.3.2 Selecting the Cell Size (Number of Octets per Cell)

The XRT94L33 permits the user to select the number of octets per cell that the Transmit UTOPIA Interface block will process, following each assertion of the TxUSoC input pin. Specifically, the user has the following cell size options.

- If the UTOPIA Data Bus width is set to 8 bits then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  53 bytes (with either a dummy or actual HEC byte in the cell)

- If the UTOPIA Data Bus width is set to 16 bits then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  54 bytes (with either a dummy or actual HEC byte, and a stuff byte in the cell)

The user makes his/her selection by writing the appropriate data into bits 1 and 0 (Cell_Size_Sel[1:0]) within the Transmit UTOPIA Control Register, as depicted below.

Transmit UTOPIA Control Register – Byte 0, Address = 0x0583

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	X	X

The following table presents the relationship between the value of these bits and the number of octets/cell that the Transmit UTOPIA Interface block will process.

Table 12 The Relationship between the contents of Bits 1 and 0 (Cell_Size_Sel[1:0]) within the Transmit UTOPIA Control Register, and the number of octets per cell that will be processed by the Transmit UTOPIA Interface blocks per assertion of TxUSOC

CELL_SIZE_SEL[1:0]	NUMBER OF BYTES/CELLS
00	52 bytes/cell
01	53 bytes/cell (only valid if the Transmit UTOPIA Data Bus Width = 8 bits)
10	54 bytes/cell
11	Unused

Once the user has implemented his/her selection for the cell size, then the Transmit UTOPIA Interface block will be configured to accept the "Cell Size" number of octets, per each assertion of the "TxUSoC" input pin. Once the Transmit UTOPIA Interface block has accepted "Cell Size" number of bytes then it will not accept any more octets until the "TxUSoC" input pin has been pulsed "high" again.

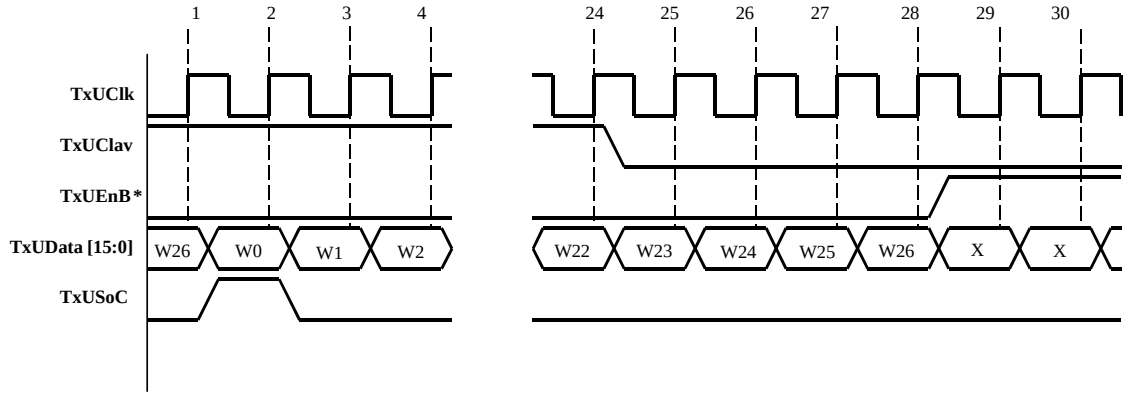
Note: *In this case the Transmit UTOPIA Interface block will cease accepting more octets, even if the "TxUEnB* input pin is pulled "low". These additional bytes are simply ignored by the "Transmit UTOPIA Interface" block.*

2.2.1.3.3 Cell-Level Handshaking

ATM Forum documentation refers to both "Cell Level" and "Octet-Level" handshaking. However, the XRT94L33 only supports the "Cell-Level" Handshaking mode. Octet-level handshaking is NOT supported. In the "Cell-Level" Handshaking mode, when the XRT94L33 sets the TxUClav output pin to a logic "1", it means that the Tx FIFO has enough remaining empty space for it to receive at least one more full cell of data from the ATM Layer processor. However, when TxUClav toggles from "high" to "low", it indicates that the very next cell (following the one that is currently being written into the Transmit UTOPIA Interface block) cannot be accepted by the Tx FIFO. Conversely, once the TxUClav output pin has returned to the logic "1" level, it indicates that at least one more full cell may be written into the TxFIFO by the ATM Layer processor. The ATM Layer processor is expected to poll the state of the TxUClav output pin towards the end of transmission of the cell currently being written and to proceed with writing the next ATM cell into the Transmit UTOPIA Interface block only if TxUClav is at a logic "high".

Figure 11 presents a timing diagram that illustrates the behavior of various Transmit UTOPIA Interface block signals, when the Transmit UTOPIA Interface block is operating in the "Cell-Level" Handshaking Mode.

Figure 11 Timing Diagram of various Transmit UTOPIA Interface block signals, when the Transmit UTOPIA Interface block is operating in the "Cell Level Handshaking" Mode



Notes regarding Figure 11:

1. The Transmit UTOPIA Data Bus is configured to be 16 bits wide. Hence, the data, which the ATM Layer processor places on the Transmit UTOPIA Data Bus, is expressed in terms of 16-bit words: W0 - W26.

2. The Transmit UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, figure 11 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

In Figure 11, the ATM Layer processor starts to write in a new ATM cell, into the Transmit UTOPIA Interface block, during clock edge #2. However, shortly after the ATM Layer processor has written in word W22 (at clock edge # 24), the TxUClav output pin toggles "low". In the "Cell-Level" Handshaking mode, this means that ATM Layer processor is not permitted to write in the subsequent ATM cell (e.g., the ATM cell which is to follow the one that is currently being written into the Transmit UTOPIA Interface block) into the Transmit UTOPIA Interface block. Hence, the ATM Layer processor must complete writing in the current cell, and then halt with any further write operations to the Transmit UTOPIA Interface block. Therefore, the ATM Layer processor will proceed to write in Words W23 through W26 and then negate the TxUEnB* signal after clock edge #28. At this point, the ATM Layer processor must wait until the TxUClav output pin toggles "high" once again; before writing in the next ATM cell.

USING THE "TRANSMIT UTOPIA DE-SKEWING" PLL

2.2.1.3.4 UTOPIA Modes of Operation (Single PHY and Multi-PHY operation)

The XRT94L33 can be configured to support either Single-PHY or Multi-PHY operation. Each of these operating modes is discussed below.

2.2.1.3.5 Single PHY Operation

The XRT94L33 permits the user to configure it to operate in either the "Single-PHY" or "Multi-PHY" Mode. The user can configure the chip to operate in the "Single-PHY" Mode by setting Bit 6 (Multi-PHY Mode) to "0"; as illustrated below.

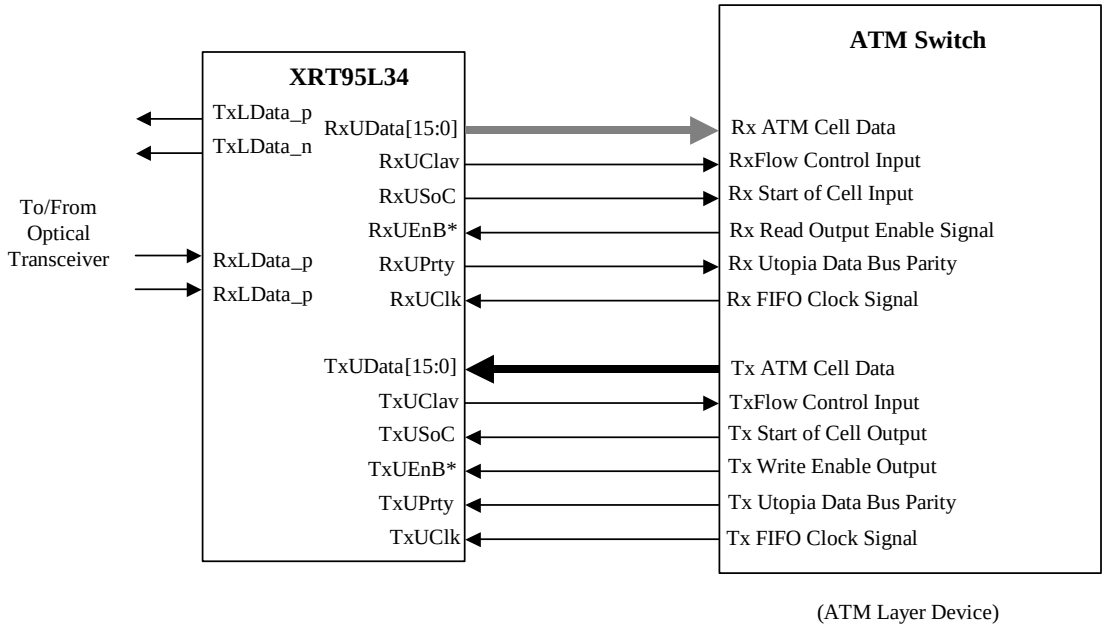
Transmit UTOPIA Control Register – Byte 0, Address = 0x0583

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	X	X

Note: *This configuration setting does not apply to the Receive UTOPIA Interface block. Therefore the user will also need to configure the Receive UTOPIA Interface block into the Single-PHY Mode, as described in Section _.*

In Single-PHY Mode operation, the ATM layer processor is pumping data into and receiving data from only one PHY-Layer device, as depicted below in Figure 22.

Figure 12 Simple Illustration of Single - PHY Mode Operation



This section presents a detailed description of the Transmit UTOPIA Interface block operating in the "Single-PHY" mode. A description of the Receive UTOPIA Interface block operating in the "Single-PHY" mode is presented in Section _. Whenever the Transmit UTOPIA Interface block has been configured to operate in the Single-PHY Mode, and whenever the ATM Layer Processor wishes to write one or a series of ATM cells to the Transmit UTOPIA Interface block, it must do the following.

1. ***Check the level of the TxUClav output pin upon each rising edge of TxUClk.***

If the TxUClav output pin is at a logic "high" then there is available space in the Tx FIFO for more ATM cell data and the ATM Layer Processor may begin writing cell data to the Transmit UTOPIA Interface block. However, if the TxUClav pin is "low", then the Tx FIFO is too full to accept anymore data and the ATM Layer Processor must wait until TxUClav toggles "high" before writing any cell data to the Transmit UTOPIA Interface block.

2. ***Apply the first byte (or word) of the new cell to the Transmit UTOPIA Data Bus.***

The ATM Layer processor must designate this byte (or word) as the beginning of a new cell, by pulsing the TxUSoC input pin "high" for one period of TxUClk.

Note: *The Transmit UTOPIA Interface block will sample and latch the state of the TxUSoC input pin upon the rising edge of TxUClk. Therefore, if the TxUClk to TxUSoC output delay (of the ATM Layer Processor) is 1ns or more, then the ATM Layer Processor should update the state of the "TxUSoC" input pin (of the XRT94L33) upon the rising edge of TxUClk.*

3. ***Apply the Odd-Parity value of this first byte (or word), currently residing on the Transmit UTOPIA Data Bus, to the TxUPrty input pin.***

This should be done concurrently with pulsing the TxUSoC input pin "high".

Note: *The Transmit UTOPIA Interface block will sample and latch the state of the "TxUPrty" input pin upon the rising edge of TxUClk. Therefore, if the TxUClk to TxUPrty output delay (of the ATM Layer Processor) is 1ns or more, then the ATM Layer Processor should update the state of the "TxUPrty" input pin (of the XRT94L33) upon the rising edge of "TxUClk".*

4. ***Assert the "Transmit UTOPIA Data Bus" - Write Enable Signal, TxUEnb*.***

This step should also be done concurrently with pulsing the TxUSoC input pin "high".

When writing the subsequent bytes (word) of the cell, the ATM Layer Processor must repeatedly exercise Steps 3 and 4, of the above list.

Since the XRT94L33 is always operating in the Cell-Level Handshake mode, then the ATM Layer Processor should check the level of the TxUClav signal, as it nears completion of writing in a given cell. If "TxUClav" is "high" then the ATM Layer Processor can finish writing the current cell into the "Transmit UTOPIA Interface" block. Afterwards, the ATM Layer Processor may proceed to write in the contents of the very next "outbound" ATM cell into the Transmit UTOPIA Interface block.

Conversely, if "TxUClav" is "low" then the ATM Layer Processor should finish writing the current ATM cell data into the "Transmit UTOPIA Interface" block. Afterwards, the ATM Layer Processor must negate the "TxUEnb*" input pin and halt writing any more ATM cell data into the "Transmit UTOPIA Interface" block, until the TxUClav output pin toggles "high" again.

The above-mentioned procedure is also depicted in Flow-Chart Form in Figure 13 and in Timing Diagram form in Figure 14.

Figure 13 Flow Chart depicting the approach that the ATM Layer Processor should take when writing ATM Cell Data into the Transmit UTOPIA Interface block, when the XRT94L33 is operating in the Single PHY Mode

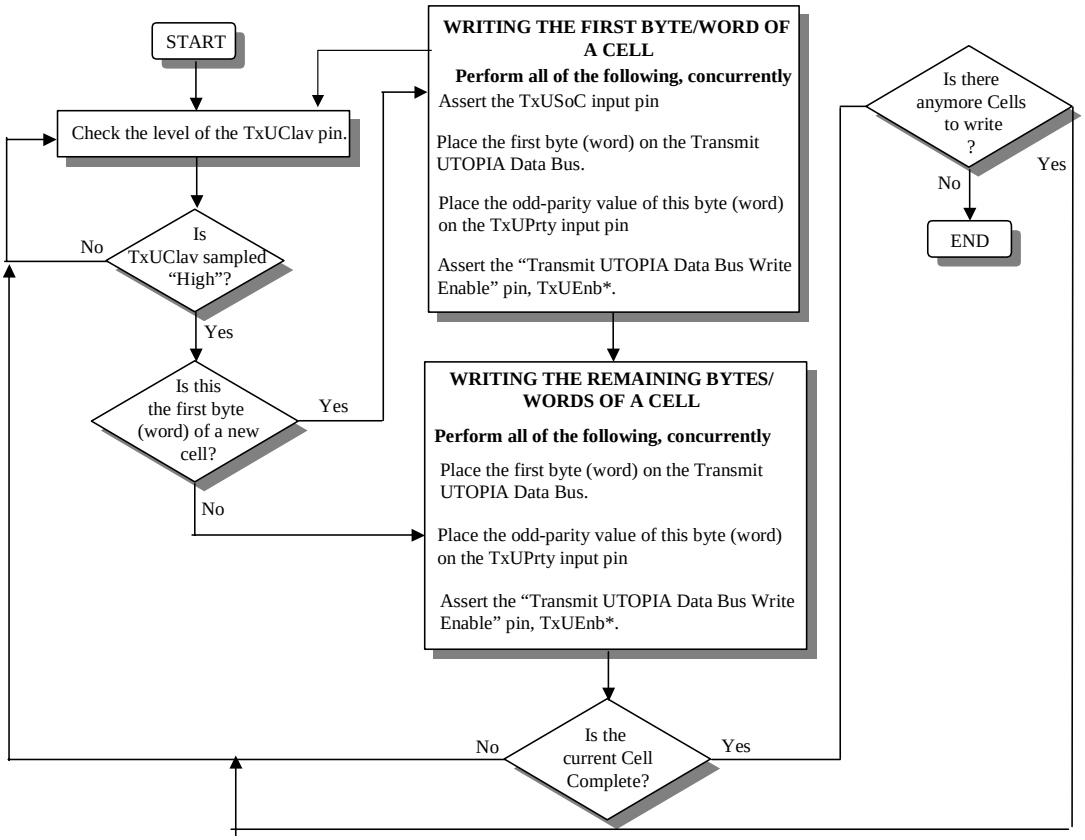
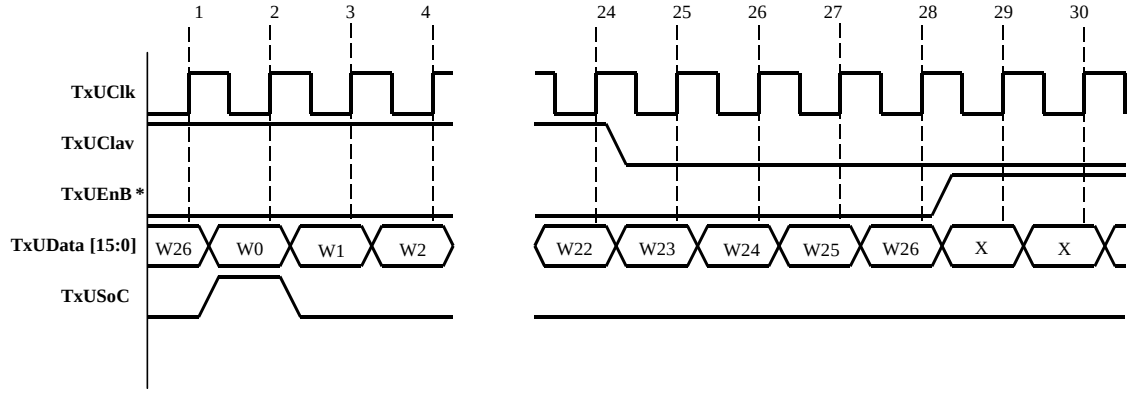


Figure 14 Timing Diagram of ATM Layer Processor writing ATM Cell data into the Transmit UTOPIA Data Bus, (Single - PHY Mode)



Notes regarding Figure 14**:**

1. The Transmit UTOPIA Data Bus is configured to be 16 bits wide. Hence, the data, which the ATM Layer processor places on the Transmit UTOPIA Data Bus, is expressed in terms of 16-bit words: W0 - W26.

2. The Transmit UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, Figure 24 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

Final Comments on Single-PHY Operation

The important thing to note about the Single-PHY mode is that the TxUClav pin is used as a data flow control pin, and has a role somewhat similar to RTS (Request To Send) in RS-232 based data transmission. The TxUClav pin will have a slightly different role when the XRT94L33 is operating in the Multi-PHY mode.

The ATM Layer processor is expected to poll the TxUClav output pin during the last four "TxUClk" periods (while writing a given ATM cell into the Transmit UTOPIA Interface block) prior to writing the next ATM cell to the TxFIFO.

2.2.1.3.6 Multi PHY Operation

The XRT94L33 permits the user to configure it to operate in the "Multi-PHY" Mode. This can be accomplished by setting Bit 6 (Multi-PHY Mode), within the "Transmit UTOPIA Control Register – Byte 0" to "1" as depicted below.

Transmit UTOPIA Control Register – Byte 0, Address = 0x0483

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	0	0	1	1	X	X

Note: *This configuration setting does not apply to the Receive UTOPIA Interface block. Therefore, the user will also need to configure the Receive UTOPIA Interface block into the "Multi-PHY" Mode, as described in Section _.*

In the "Multi-PHY" operating mode, the ATM Layer processor may be writing data into and reading data from several UNI (e.g., PHY-Layer) devices in parallel. Figure 12 presents an illustration of a simple "Multi-PHY System" consisting of a single ATM Layer Processor being interfaced to two (2) UNI devices. When the

XRT94L33 is operating in the Multi-PHY mode, the Transmit UTOPIA Interface block will support two kinds of operations with the ATM Layer processor:

- Polling for "available" UNI (PHY Layer) devices.
- Selecting which UNI (out of several possible UNI devices) to write ATM cell data to.

Each of these operations is discussed in the sections below. However, prior to discussing each of these operations, the reader must understand the following.

"Multi-PHY" operation involves the use of one (1) ATM Layer processor and several UNI (or PHY-Layer) devices, within a system. The ATM Layer processor is expected to read/write ATM cell data from/to these UNI devices. Hence, "Multi-PHY" operation requires, at a minimum, some means for the ATM Layer processor to uniquely identify a particular UNI device (among all of the UNI devices within the "Multi-PHY" system) that it wishes to "poll", write ATM cell data to, or read ATM cell data from. Actually, "Multi-PHY" operation provides an addressing scheme allows the ATM Layer processor to uniquely identify "UTOPIA Interface Blocks" (e.g., Transmit and Receive) within all of the UNI devices, operating in the "Multi-PHY" system. In order to uniquely identify a given "UTOPIA Interface block", within a "Multi-PHY" system, each "Transmit and Receive UTOPIA Interface Blocks are assigned a 5-bit "UTOPIA address" value. The user assigns this address value to a particular "Transmit UTOPIA Interface block" by writing this address value into the "Transmit UTOPIA Address Register" (Address = 0x0593) and appropriate data into the "Transmit UTOPIA Port Address" Register (Address = 0x0597); within its "host" XRT94L33; per the procedure (as presented below). The bit-format of the "Transmit UTOPIA Address Register" and "Transmit Port Address" Register is presented below.

Transmit UTOPIA Address Register (Address = 0x0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

Transmit UTOPIA Port Number Register (Address = 0x0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

2.2.1.3.7 Assigning Transmit UTOPIA Addresses to Each STS-3c Channel within the XRT94L33

The XRT94L33 can be configured to function as a three channel ATM UNI over either an STS-3 signal or an STS-3c. As a consequence, for Multi-PHY Operation, the XRT94L33 can be assigned as many as four "Transmit UTOPIA Addresses" at a given time (one for each STS-1 channel).

The user can assign a "Transmit UTOPIA Address" value to a given channel, within the XRT94L33 by executing the following steps.

STEP 1 – Assign a "Transmit UTOPIA Address" to Channel 0

The user can accomplish this by doing the following.

STEP 1a – Set Bits 0 through 4 ("Tx_UTOPIA_Port_Number[4:0]") within the "Transmit UTOPIA Port Number Register" to "0x00"; as depicted below.

Transmit UTOPIA Port Number Register (Address = 0x0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures an internal "Transmit UTOPIA Address assignment" pointer to point to "STS-3c Channel 0" within the XRT94L33.

STEP 1b – Set Bits 0 through 4 ("Tx_UTOPIA_Addr[4:0]") within the "Transmit UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [a4, a3, a2, a1, a0]; as depicted below.

Transmit UTOPIA Address Register (Address = 0x0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	A4	A3	A2	A1	A0

This step configures the Transmit UTOPIA Address, of the value [A4, A3, A2, A1, A0] to be assigned to Channel 0.

Notes:

During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Transmit UTOPIA Interface Addresses within a "Multi-PHY" system.

Use of the value "0x1F" is NOT permitted.

These steps do not assign a Receive UTOPIA Address value to the STS-3c Channel 0. This assignment must be performed separately per the "Receive UTOPIA Address" assignment instructions, presented in Section _.

Once the user has executed STEPS 1a and 1b, then STS-3c Channel 0 within the XRT94L33 has been assigned the "Transmit UTOPIA Address" of [a4, a3, a2, a1, a0].

STEP 2 – Assign a "Transmit UTOPIA Address" to STS-3c Channel 1

The user can accomplish this by doing the following.

STEP 2a – Set Bits 0 through 4 ("Tx_UTOPIA_Port_Number[4:0]") within the "Transmit UTOPIA Port Number Register" to "0x01"; as depicted below.

Transmit UTOPIA Port Number Register (Address = 0x0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This step configures an internal "Transmit UTOPIA Address assignment" pointer to point to "STS-3c Channel 1" within the XRT94L33.

STEP 2b – Set Bits 0 through 4 ("Tx_UTOPIA_Addr[4:0]") within the "Transmit UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [b4, b3, b2, b1, b0]; as depicted below.

Transmit UTOPIA Address Register (Address = 0x0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	B4	B3	B2	B1	B0

This step configures the Transmit UTOPIA Address, of the value [B4, B3, B2, B1, B0] to be assigned to Channel 1.

Notes:

1. *During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Transmit UTOPIA Addresses within a "Multi-PHY" system.*
2. *Use of the value "0x1F" is NOT permitted.*
3. *These steps do not assign a Receive UTOPIA Address value to STS-3c Channel 1. This assignment must be performed separately per the "Receive UTOPIA Address Assignment" instructions presented in Section _.*

Once the user has executed STEPS 2a and 2b, then STS-3c Channel 1 within the XRT94L33 has been assigned the "Transmit UTOPIA Address" of [b4, b3, b2, b1, b0].

STEP 3 – Assign a "Transmit UTOPIA Address" to STS-3c Channel 2

The user can accomplish this by doing the following.

STEP 3a – Set Bits 0 through 4 ("Tx_UTOPIA_Port_Number[4:0]") within the "Transmit UTOPIA Port Number Register" to "0x02"; as depicted below.

Transmit UTOPIA Port Number Register (Address = 0x0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This step configures an internal "Transmit UTOPIA Address assignment" pointer to point to "STS-3c Channel 2" within the XRT94L33.

STEP 3b – Set Bits 0 through 4 ("Tx_UTOPIA_Addr[4:0]") within the "Transmit UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [c4, c3, c2, c1, c0]; as depicted below.

Transmit UTOPIA Address Register (Address = 0x0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	C4	C3	C2	C1	C0

This step configures the Transmit UTOPIA Address, of the value [C4, C3, C2, C1, C0] to be assigned to STS-3c Channel 2.

Notes:

1. *During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Transmit UTOPIA Interface addresses within a "Multi-PHY" system.*
2. *Use of the value "0x1F" is NOT permitted.*
3. *These steps do not assign a Receive UTOPIA Address value to STS-3c Channel 2. This assignment must be performed separately per the "Receive UTOPIA Address Assignment" instructions presented in Section _.*

Once the user has executed STEPS 3a and 3b then STS-3c Channel 2 within the XRT94L33 has been assigned the "Transmit UTOPIA Address" of [c4, c3, c2, c1, c0].

STEP 4 – Assign a "Transmit UTOPIA Address" to STS-3c Channel 3

The user can accomplish this by doing the following.

STEP 4a – Set Bits 0 through 4 ("Tx_UTOPIA_Port_Number[4:0]") within the "Transmit UTOPIA Port Number Register" to "0x03"; as depicted below.

Transmit UTOPIA Port Number Register (Address = 0x0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

This step configures an internal "Transmit UTOPIA Address assignment" pointer to point to "STS-3c Channel 3" within the XRT94L33.

STEP 4b – Set Bits 0 through 4 ("Tx_UTOPIA_Addr[4:0]") within the "Transmit UTOPIA Address" Register to the "Multi-PHY" Address value for this channel [d4, d3, d2, d1, d0]; as depicted below.

Transmit UTOPIA Address Register (Address = 0x0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	D4	D3	D2	D1	D0

This step configures the Transmit UTOPIA Address, of the value [D4, D3, D2, D1, D0] to be assigned to STS-3c Channel 3.

Notes:

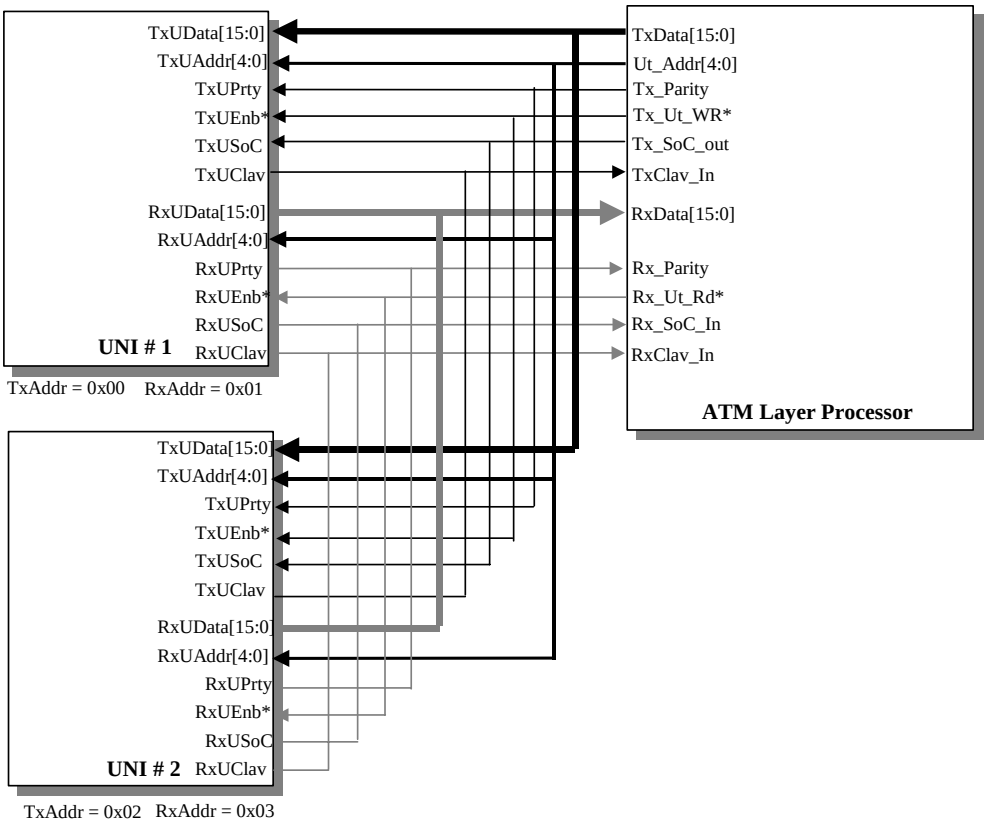
1. *During this step, the user can write in any value, between 0x00 and 0x1E provided this "Multi-PHY Address" is unique among all of the Transmit UTOPIA Interface Addresses within a "Multi-PHY" system.*
2. *Use of the value "0x1F" is NOT permitted.*
3. *These steps do not assign a Receive UTOPIA Address value to STS-3c Channel 3. This assignment must be performed separately per the "Receive UTOPIA Address Assignment" instructions presented in Section _.*

Once the user has executed STEPS 4a and 4b then STS-3c Channel 3 within the XRT94L33 has been assigned the "Transmit UTOPIA Address" of [d4, d3, d2, d1, d0].

2.2.1.3.8 ATM Layer Processor "polling" of the UNIs, in the Multi-PHY Mode

In this section, the various Multi-PHY Operations (e.g., polling and selection for writing) will be first discussed for a "Conceptual Multi-PHY" System, and then later, specifically for the XRT94L33. When the XRT94L33 is operating in the "Multi-PHY" mode, then the Transmit UTOPIA Interface block will automatically be configured to support "polling". "Polling" allows an ATM Layer processor (which is interfaced to several UNI devices) to determine which UNIs are capable of receiving and handling additional ATM cell data, at any given time. The manner in which the ATM Layer processor "polls" its UNI devices, (per the "Conceptual Multi-PHY" system) follows.

Figure 15 An Illustration of the "Conceptual Multi-PHY System consisting of UNI Devices #1 and #2



2.2.1.3.9 ATM Layer Processor "polling" in a Conceptual Multi-PHY System

Figure 26 depicts a "Multi-PHY" system consisting of a single ATM Layer processor and two (2) UNI devices, which are designated as "UNI #1" and "UNI #2". In this figure, both of the UNIs are connected to the ATM Layer processor via a common "Transmit UTOPIA" Data Bus, a common "Receive UTOPIA" Data Bus, a common "TxUClav" line, a common "RxUClav" line, as well as common TxUEnB*, RxUEnB*, TxUSoC and RxUSoC lines. The ATM Layer processor will also be addressing both the Transmit and Receive UTOPIA Interface blocks via a common "UTOPIA" address bus (Ut_Addr[4:0]) Therefore, the Transmit and Receive UTOPIA Interface Blocks, within a given UNI might have different addresses; as depicted in Figure 26.

The UTOPIA Address values, that have been assigned to each of the Transmit and Receive UTOPIA Interface blocks, within Figure 15, are listed below in Table 13.

Table 13 UTOPIA Address Values of the UTOPIA Interface blocks illustrated in Figure 26.

BLOCK	UTOPIA ADDRESS VALUE
Transmit UTOPIA Interface block - UNI #1	0x00
Receive UTOPIA Interface block - UNI #1	0x01
Transmit UTOPIA Interface block - UNI #2	0x02
Receive UTOPIA Interface block - UNI #2	0x03

Recall, that the Transmit UTOPIA Interface blocks were assigned these addresses by writing these values into registers that are similar to the "Transmit UTOPIA Port Number (Address = 0x0597) and the Transmit UTOPIA Address Register" (Address = 0x0593) within these UNI devices. The discussion of the Receive UTOPIA Interface blocks, within UNIs #1 and #2 is presented in Section _.

Polling Operation

Consider that the ATM Layer processor is currently writing a continuous stream of ATM cell data into UNI #1. While writing this cell data into UNI #1, the ATM Layer processor can also "poll" UNI #2 for "availability" (e.g., tries to determine if the ATM Layer processor can write any more ATM cell data into the "Transmit UTOPIA Interface block" within UNI #2).

The ATM Layer processor's role in the "polling" operation

The ATM Layer processor accomplishes this "polling" operation by executing the following steps.

1. Assert the TxUEnB* input pin (if it is not asserted already).

The UNI device (being "polled") will know that this is only a "polling" operation, if the TxUEnB* input pin is asserted, prior to detecting its UTOPIA Address on the "UTOPIA Address" bus (TxUAddr[4:0]).

2. The ATM Layer processor places the address of the Transmit UTOPIA Interface Block of UNI #2 onto the UTOPIA Address Bus, Ut_Addr[4:0],

3. The ATM Layer processor will then check the value of its "TxUClav_in" input pin (see Figure 24).

The ATM Layer Processor is suppose to check the state of the "TxUClav" signal, one "TxUClk" period after placing the UTOPIA Address (corresponding to a particular UNI device) on the "TxUAddr[4:0]" input pins. If "TxUClav" is sampled "high" then this means that this particular UNI device contains sufficient empty space within its TxFIFO to accept another ATM cell from the ATM Layer Processor. Conversely, if TxUClav is sampled "low" then this means that this particular UNI device does not contain enough empty space within its TxFIFO to accept another ATM cell from the ATM Layer Processor.

The UNI devices role in the "polling" operation

UNI #2 will sample the signal levels placed on its Tx UTOPIA Address input pins (TxUAddr[4:0]) on the rising edge of its "Transmit UTOPIA Interface block" clock input signal, TxUClk. Afterwards, UNI #2 will compare the value of these "Transmit UTOPIA Address Bus input pin" signals with that of the contents of its "Tx UTOPIA Address Register (Address = 0x0593).

If these values do not match, (e.g., TxUAddr[4:0] $\neq$ 0x02) then UNI #2 will keep its "TxUClav" output signal "tri-stated"; and will continue to sample its "Transmit UTOPIA Address bus input" pins; with each rising edge of TxUClk.

If these two values do match, (e.g., TxUAddr[4:0] = 0x02) then UNI #2 will drive its "TxUClav" output pin to the appropriate level, reflecting its TxFIFO "fill-status". Since the UNI is only operating in the "Cell Level Handshaking" mode, the UNI will drive the TxUClav output signal "high" if it is capable of receiving at least one more complete cell of data from the ATM Layer processor. Conversely, the UNI will drive the "TxUClav"

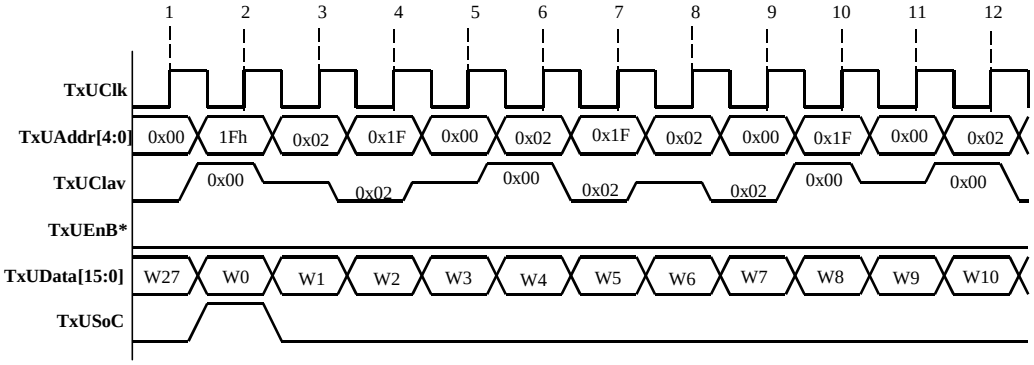
output signal “low” if its TxFIFO is too full and is incapable of receiving one more complete cell of data from the ATM Layer processor.

When UNI #2 has been selected for “polling”, UNI #1 will continue to keeps its “TxUClav” output signal “tri-stated”. Therefore, when UNI #2 is driving its “TxUClav” output pin to the appropriate level; it will be driving the entire “TxUClav” line, within the “Multi-PHY” system. Consequently, UNI#2 will also be driving the “TxUClav_in” input pin of the ATM Layer processor (see Figure 26).

If UNI #2 drives the “TxUClav” line “low”, upon the application of its address on the UTOPIA Address Bus, then the ATM Layer processor will “learn” that it cannot write any more cell data to this UNI device; and will deem this device “unavailable”. However, if UNI #2 drives the TxUClav line “high” (during “polling”), then the ATM Layer processor will know that it can write more ATM cell data into the Transmit UTOPIA Interface block, of UNI # 2.

Figure 16 presents a timing diagram, that depicts the behavior of the ATM Layer processor’s and the UNI’s signals during polling.

Figure 16:Timing Diagram illustrating the Behavior of various signals from the ATM Layer processor and the UNI, during Polling



Notes regarding Figure 16:

1. The Transmit UTOPIA Data Bus is configured to be 16 bits wide. Hence, the data, which the ATM Layer processor places on the Transmit UTOPIA Data bus, is expressed in terms of 16-bit words: (e.g., W0 - W26.)

2. The Transmit UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, Figure 27 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

3. The ATM Layer processor is currently writing ATM cell data to the Transmit UTOPIA Interface Block, within UNI #1 (TxUAddr[4:0] = 0x00) during this “polling process”.

4. The TxFIFO, within UNI#2’s Transmit UTOPIA Interface block (TxUAddr[4:0] = 0x02) is incapable of receiving any additional ATM cell data from the ATM Layer processor. Hence, the TxUClav line will be driven “low” whenever this particular Transmit UTOPIA Interface block is “polled”.

5. The Transmit UTOPIA Address of 0x1F (e.g., the NULL address), is not associated with any UNI device, within this “Multi-PHY” system. Hence, the TxUClav line is tri-stated whenever this address is “polled”.

Note: *Although Figure 26 depicts connections between the Receive UTOPIA Interface block pins and the ATM Layer processor; the Receive UTOPIA Interface block operation, in the Multi-PHY mode, will not be discussed in this section. Please see Section _ for a discussion on the Receive UTOPIA Interface block during Multi-PHY operation.*

2.2.1.3.10 ATM Layer Processor “polling” with the XRT94L33

In Figure 15, a simple illustration of the “Conceptual Multi-PHY” system consisting of two single-channel UNI devices was presented. In reality, a given Multi-PHY system can or will consist of numerous “multi-channel” UNI devices. The XRT94L33 is an example of this, being a “4-channel” UNI device. Therefore, the

XRT94L33 can be thought of as consisting of four ATM UNIs within a single package (one for each STS-3c port within the device).

It is important to note that although the XRT94L33 consists of a single "Transmit UTOPIA Interface" block, the XRT94L33 can be assigned four unique (4) Transmit UTOPIA Addresses as presented in Section 4.2.1.2.1.4.2.1.

2.2.1.3.11 Writing ATM Cell Data into a Different UNI

After the ATM Layer processor has "polled" each of the UNI devices, within its system, it must now select a UNI, and begin writing ATM cell data to that device. The ATM Layer processor makes its selection and begins the writing process by:

1. Applying the UTOPIA Address of the "target" UNI on the "Transmit UTOPIA Address Bus" (TxUAddr[4:0]).

Negate the TxUEnB* signal. This step causes the "addressed" UNI to recognize that it has been selected to receive the next set of ATM cell data from the ATM Layer processor.

Assert the TxUEnB* signal.

Assert the TxUSoC input pin.

Begin applying the ATM Cell data in a byte-wide (or word-wide) manner to the Transmit UTOPIA Data Bus (TxUData[15:0]).

At this point, ATM cell data is now being written into the TxFIFO associated with the newly selected UNI.

Figure 17 presents a flow-chart that depicts the "UNI Device Selection and Write" process in Multi-PHY operation.

Figure 17: Flow-Chart of the "UNI Device Selection and Write Procedure" for the Multi-PHY Operation

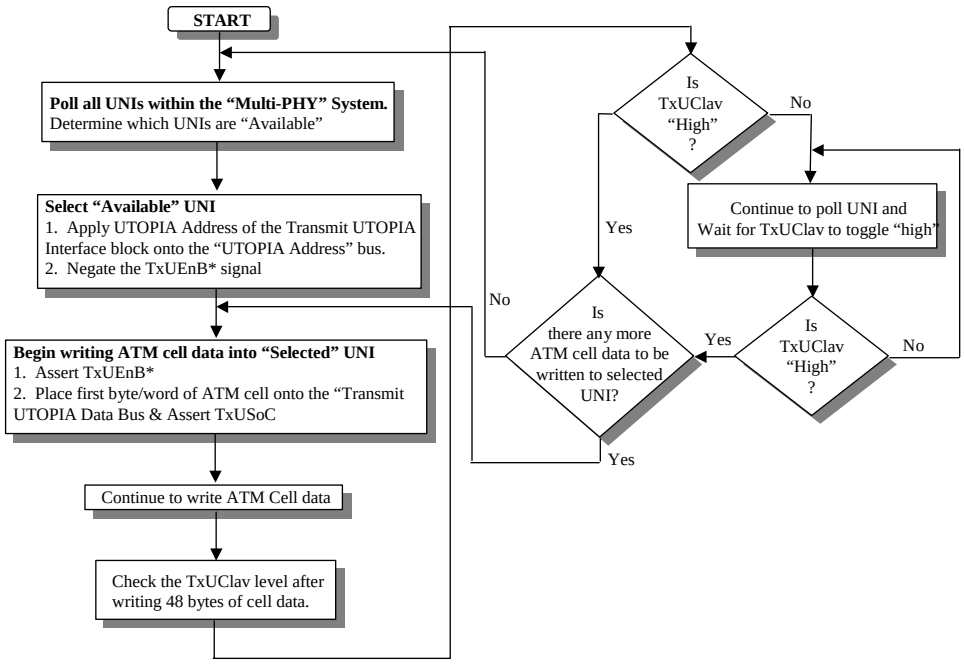
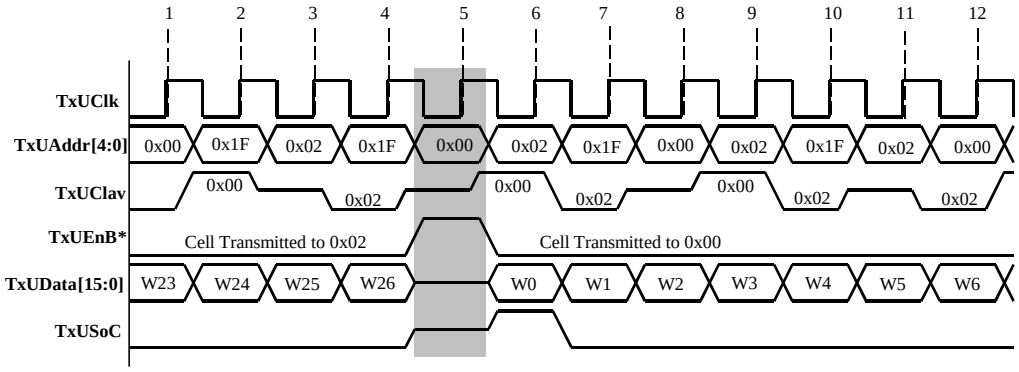


Figure 18 presents a timing diagram that illustrates the behavior of various "Transmit UTOPIA Interface block" signals; during the "Multi-PHY" UNI Device Selection and Write operation.

Figure 18: Timing Diagram of the Transmit UTOPIA Data and Address Bus signals, during the "Multi-PHY" UNI Device Selection and Write Operations



Notes regarding Figure 18:

1. The Transmit UTOPIA Data bus is configured to be 16 bits wide. Hence, the data, which the ATM Layer processor places on the Transmit UTOPIA Data bus, is expressed in terms of 16-bit words (e.g., W0 - W26).

2. The Transmit UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, Figure 29 illustrates the ATM Layer processor writing 27 words (e.g., W0 through W26) for each ATM cell.

In Figure 29, the ATM Layer processor is initially writing ATM cell data to the Transmit UTOPIA Interface block within UNI #2 (TxUAddr[4:0] = 0x02). However, the ATM Layer processor is also polling the Transmit UTOPIA Interface block within UNI #1 (TxAddr[4:0] = 0x00) and the "NULL" address of 0x1F. The ATM Layer processor completes its writing of the cell to UNI #1 at clock edge #4. Afterwards, the ATM Layer processor will cease to write any more cell data to UNI #1, and will begin to write this data into UNI #2 (TxUAddr[4:0] = 0x02). The ATM Layer processor will indicate its intentions to select a new UNI device for writing by negating the TxUEnB* signal, at clock edge #5 (see the shaded portion of Figure 29). At this time, UNI #1 will notice two things:

1. The UTOPIA Address for the Transmit UTOPIA Interface block, within UNI #1 is on the Transmit UTOPIA Address bus (TxUAddr[4:0] = 0x00).

2. The TxUEnB* signal has been negated.

UNI #1 will interpret this signaling as an indication that the ATM Layer processor is going to be performing write operations to it. Afterwards, the ATM Layer processor will begin to write ATM cell data into Transmit UTOPIA Interface block, within UNI #1.

2.2.1.4 UTOPIA LEVEL 3 OPERATION OF THE TRANSMIT UTOPIA INTERFACE BLOCK

BACK-TO-BACK POLLING

If the Transmit UTOPIA Interface block has been configured to operate in the UTOPIA Level 3 mode, then it can be configured to support "Back-to-Back" polling.

UTOPIA Level 2 specifications mandate that the ATM Layer Processor interleave the application of UTOPIA addresses with the "NULL" address of 0x1F.

If "Back-to-Back" polling is selected, then the ATM Layer Processor does not need to interleave the application of UTOPIA addresses with the "NULL" address. Instead, the ATM Layer Processor can poll each of the UNIs, by applying a different "UTOPIA Address" value, with each cycle of "TxUClk" (e.g., in a "Back-to-Back" Manner).

The user can configure the "Transmit UTOPIA Interface" block to support "Back-to-Back" polling by setting bit 5 (Back-to-Back Polling Enable) within the "Transmit UTOPIA Control" Register, to "1" as depicted below.

Transmit UTOPIA Control Register – Byte 0, Address = 0x0483

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Transmit UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	0	1	1	X	X

2.2.2 TRANSMIT ATM CELL PROCESSOR BLOCK

The next functional block, within the Transmit Path of the XRT94L33 is the Transmit ATM Cell Processor Block. Figure 19 presents an illustration of the XRT94L33 Functional Block Diagram, with the "Transmit ATM Cell Processor Block" highlighted.

Figure 19: Illustration of the XRT94L33 Functional Block Diagram, with the Transmit ATM Cell Processor block highlighted

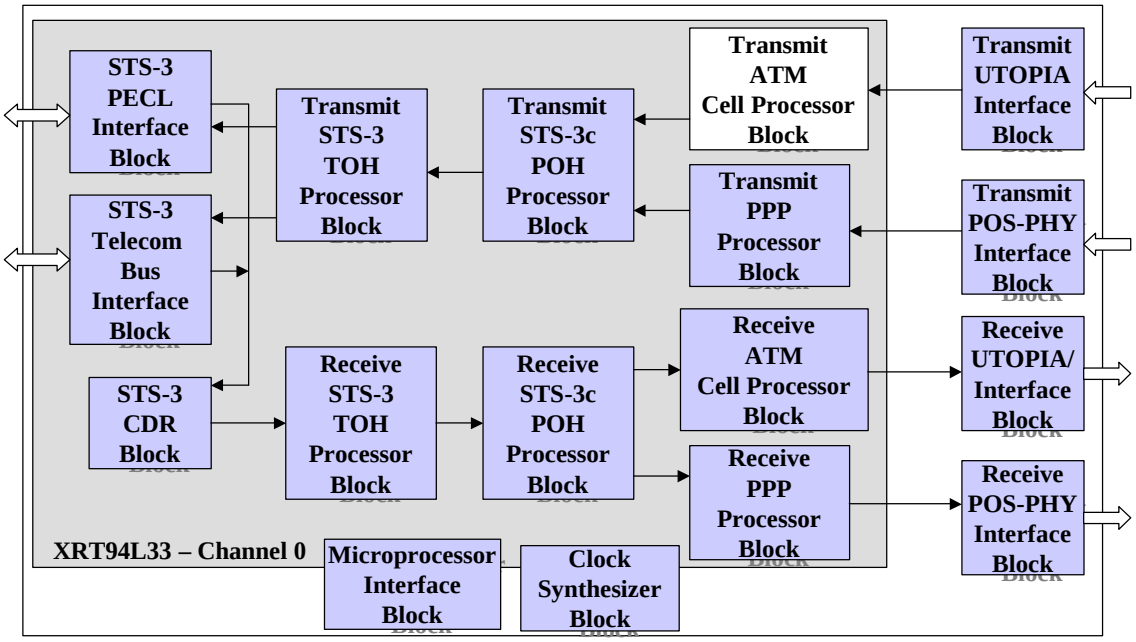
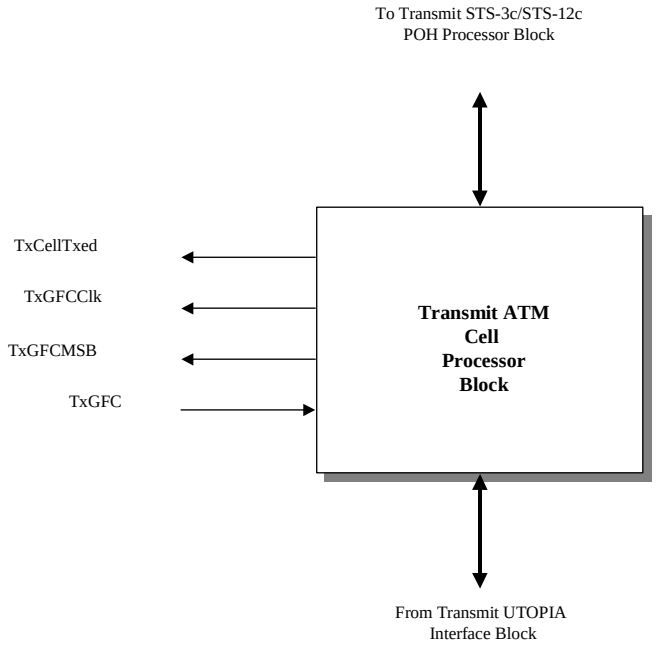


Figure 20 presents a simple block diagram of the Transmit ATM Cell Processor block (with the external pins indicated).

Figure 20: Simple Illustration of the Transmit ATM Cell Processor Block and the Associated External Pins



A very brief description of the Transmit ATM Cell Processor block

The Transmit ATM Cell Processor block reads in ATM cells from the Transmit UTOPIA FIFO (TxFIFO) within the Transmit UTOPIA Interface block. Immediately after reading in a given cell from the TxFIFO, the Transmit ATM Cell Processor will compute and verify the parity value of each byte or (16-bit) word of this ATM cell. If the Transmit ATM Cell Processor block detects a parity error within one of these bytes then (or 16-bit words) then the entire cell will be (optionally) discarded. Afterwards, each ATM cell (which is not discarded due to parity error) will be processed through a "user-configurable" filter. If the cell does not meet the "user-cell" filtering criteria, then it will be discarded. Next, the ATM cell will be processed through both the "Transmit Data Path" and the "Transmit Cell Extraction" Processor, in parallel. The manner in which this ATM cell is handled (via each of these paths) is briefly described below.

2.2.3 THE TRANSMIT CELL EXTRACTION PROCESSOR

The User Cell Filter blocks (within the Transmit ATM Cell Processor block) permit the user to configure it to copy (e.g., replicate) certain ATM cells; and route these "copies" of ATM cells to the "Transmit Cell Extraction Buffer". The "Transmit Cell Extraction Processor permits the user to read out the contents of these "copied" ATM cells via the Microprocessor Interface.

2.2.4 THE TRANSMIT DATA PATH

All ATM cells that successfully pass through "Parity Checking" and "User Cell Filtering" will be processed via the "HEC Byte Calculation & Insertion" Block. In this case, the Transmit ATM Cell Processor block will optionally compute and insert the HEC byte into the fifth octet position within each outbound ATM cell and, optionally scrambles the cell payload bytes.

When the TxFIFO does not contain a full cell, the Transmit ATM Cell Processor block will then proceed to read out the contents of the "Transmit Cell Insertion" Buffer. Once the "Transmit Cell Insertion" Buffer has been depleted, then the "Idle Cell Generator" block will begin to generate and transmit "Idle Cells" via the Transmit Data Path. Finally, the Transmit ATM Cell Processor is also equipped with a serial input port that permits the user to externally insert the value of the GFC (Generic Flow Control) field for each outbound cell. Figure 30 presents an illustration of the functional block diagram of the Transmit ATM Cell Processor block and the associated external pins.

A detailed description of the Transmit ATM Cell Processor block is presented below.

2.2.4.1 FUNCTIONAL DESCRIPTION OF THE TRANSMIT ATM CELL PROCESSOR

This section presents an in-depth functional description of the Transmit ATM Cell Processor block. Additionally, this section presents all of the configuration options associated with the Transmit ATM Cell Processor block.

The Transmit ATM Cell Processor consists of the following functional blocks.

- Parity Checker Block
- Transmit User Cell Filter Block
- HEC Byte Calculation & Insertion Block
- The Transmit Cell Insertion Buffer/Processor
- The Transmit Cell Extraction Buffer/Processor
- Cell Payload Scrambler Block
- IDLE Cell Generator Block
- "Transmit GFC Nibble-field" serial input port

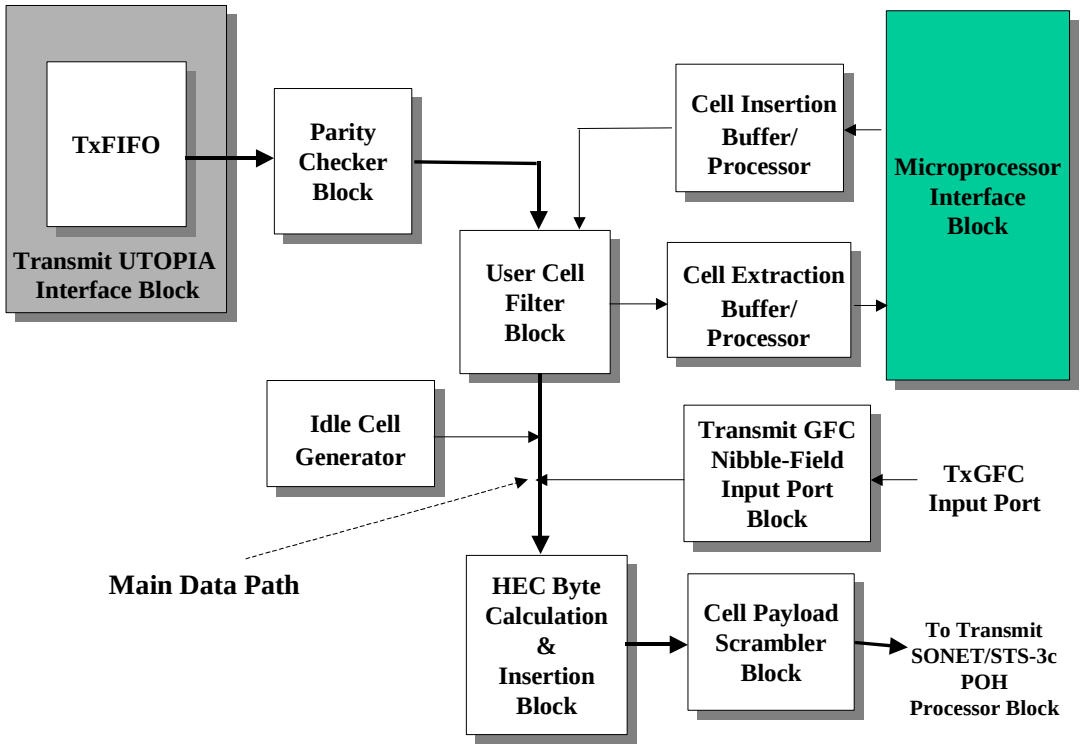
Figure 21 presents an illustration of the Functional Block Diagram of the Transmit ATM Cell Processor block; with each of these "above-mentioned" functional blocks noted.

Each of these "sub-blocks" will be discussed in some detail below. However, before we get too much into the detailed functional description of the Transmit ATM Cell Processor block; the user MUST note that the Transmit ATM Cell Processor block will NOT even function unless the user enables the "Transmit ATM Cell Processor" block for operation. The user can enable the Transmit ATM Cell Processor block by setting Bit 0 (Transmit ATM Cell Processor Enable), within the "Transmit ATM Cell Processor Block – Transmit ATM Control Register – Byte 2" to "1" as depicted below.

Transmit ATM Cell Processor Block – Transmit ATM Control Register – Byte 2 (Address = 0xNF01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Transmit ATM Cell Processor Enable
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	1

Figure 21: Functional Block Diagram of the Transmit ATM Cell Processor Block



2.2.5 THE PARITY CHECKER BLOCK

Whenever the ATM Layer Processor writes ATM cell data into the Transmit UTOPIA Interface block; it will do so by placing a sequence of bytes or (16-bit) words unto the Transmit UTOPIA Interface data bus (TxUData[15:0]). As the ATM Layer Processor is placing each byte or word (or a given ATM cell); into the Transmit UTOPIA Data bus, it is expected to compute the odd parity of each of these bytes or words. Additionally, the ATM Layer Processor is expected to apply this parity value to the "TxUPrty" input pin, coincident with placing the corresponding byte or word (of ATM cell data) onto the Transmit UTOPIA Data Bus.

The Transmit UTOPIA Interface block will latch the contents of a given byte or word (of ATM cell data) upon the rising edge of "TxUClk". As the Transmit UTOPIA Interface block samples and latches each byte or word

of ATM cell data, it is also sampling and latching the value of the corresponding parity bit (via the "TxUPrty" input pin). All sampled byte/words and their corresponding sampled parity value will be loaded into the "TxFIFO" for further processing by the Transmit ATM Cell Processor block.

As the Transmit ATM Cell Processor block reads out the contents of the ATM cells (and corresponding parity information) from the TxFIFO, it will route this data to the "Parity-Checker" block. The Parity Check block will then compute the parity value of each byte or 16-bit word (of ATM cell data that is read from the TxFIFO). Afterwards, the Parity Checker block will then compare the value of this "locally-computed" parity bit, with the corresponding sampled parity value (originating from the "TxUPrty" input pin).

If the "locally-computed" and the "ATM Layer Processor" supplied parity values match then this byte (or 16-bit word) of ATM cell data is deemed to be "error-free" and the parity-checker block will proceed to check and verify the parity value of the very next byte (or 16-bit word) of ATM cell data. On the other hand, if there is a discrepancy between the "locally-computed" the "ATM Layer Processor" supply parity values, then the "Parity-Checker" block will declare a "Transmit UTOPIA Interface – Parity Error" event, and this entire ATM cell is subject to being discarded (depending upon user configuration).

2.2.5.1 HOW THE TRANSMIT ATM CELL PROCESSOR BLOCK VERIFIES PARITY

The Transmit ATM Cell Processor block provides the user with the following parity-checking features.

- To disable parity-checking altogether.
- To verify odd parity
- To verify even parity
- To retain cells that contain parity errors
- To discard cells that contain parity errors

Each of these "Transmit ATM Cell Processor" block parity options are discussed below.

2.2.5.2 ENABLING/DISABLING PARITY CHECKING WITHIN THE TRANSMIT CELL PROCESSOR BLOCK

The Transmit ATM Cell Processor block permits the user to either enable or disable the "Parity Verification" of all bytes/words of ATM cells that have been read out of the TxFIFO. The user can accomplish this by writing the appropriate data into Bit 5 (Parity Check Enable) within the Transmit ATM Control – Byte 0 Register; as depicted below.

Transmit ATM Control – Byte 0 Register (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	X	0	0	0	0	0

Setting this bit-field to "1" configures the Transmit ATM Cell Processor block to check and verify parity of all incoming "user" cells. Conversely, setting this bit-field to "0" disables parity checking by the Transmit ATM Cell Processor block.

2.2.5.3 CONFIGURING THE TRANSMIT ATM CELL PROCESSOR BLOCK TO CHECK ODD OR EVEN PARITY

The Transmit ATM Cell Processor block can be configured to check and verify either even or odd parity. The user can implement this selection by writing the appropriate value into Bit 3 (Odd Parity) within the "Transmit ATM Control – Byte 0" Register; as depicted below.

Transmit ATM Control – Byte 0 Register (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	1	0	X	0	0	0

Setting this bit-field to "1" configures the Transmit Cell Processor block to check and verify "ODD" Parity. Conversely, setting this bit-field to "0" configures the Transmit Cell Processor block to check and verify "EVEN" parity.

Note: *This bit-field is ignored if Bit 5 (Parity Check Enable) is set to "0".*

2.2.5.4 CONFIGURING THE TRANSMIT ATM CELL PROCESSOR BLOCK TO DISCARD CELLS UPON PARITY ERRORS

The Transmit ATM Cell Processor block permits the user to configure it to discard all cells that contain Parity errors. The user can implement this selection by writing the appropriate value into Bit 4 (Discard Cell upon Parity Error) within the "Transmit ATM Control – Byte 0" Register; as depicted below.

Transmit ATM Control – Byte 0 Register (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	1	X	X	0	0	0

Setting this bit-field to "1' configures the Transmit ATM Cell Processor block to discard all cells that contain parity errors. Conversely, setting this bit-field to "0" configures the Transmit ATM Cell Processor block to retain all cells (even if they contain parity errors).

Note: *This bit-field is ignored if Bit 5 (Parity Check Enable) is set to "0".*

2.2.5.5 INTERRUPTS DUE TO DETECTION OF "TRANSMIT UTOPIA INTERFACE" PARITY ERRORS

The Transmit ATM Cell Processor block can be configured to generate an interrupt, anytime it detects a parity error, within a given cell. The user can accomplish this by setting Bit 0 (Detection of Parity Error Interrupt Enable), within the Transmit ATM Cell Processor – Interrupt Enable Register to "1"; as depicted below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Once the user has enabled the "Detection of Parity Error" Interrupt, then the Transmit ATM Cell Processor block will generate an interrupt anytime it detects a Parity Error within a given byte (or 16-bit word) in a given incoming "User Cell". Whenever the Transmit ATM Cell Processor block generates this interrupt, it will indicate so by doing all of the following.

1. By toggling the "INT*" output pin "LOW.
2. By setting Bit 0 (Detection of Parity Error Interrupt Status), within the "Transmit ATM Cell Processor – Interrupt Status Register" to "1", as depicted below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

2.2.5.6 COUNTING PARITY ERRORS

For Performance Monitoring purposes, the user can keep track of the number of parity errors that have been detected over time, by periodically polling and reading out the contents of the "Transmit ATM Cell Processor – Parity Error Count" Register. This register is a 32-bit wide, RESET-upon-READ register, that is incremented each time the Transmit ATM Cell Processor block detects a parity error in an incoming User Cell. This register will increment up until it reaches the value of "0xFFFFFFFF"; at this point it will saturate at this value (e.g., cease to increment), until read again.

Transmit ATM Cell Processor – Parity Error Count Register – Byte 3 (Address = 0xNF34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Number of Parity Errors[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Cell Processor – Parity Error Count Register – Byte 2 (Address = 0xNF35)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Number of Parity Errors[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Cell Processor – Parity Error Count Register – Byte 3 (Address = 0xNF36)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Number of Parity Errors[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

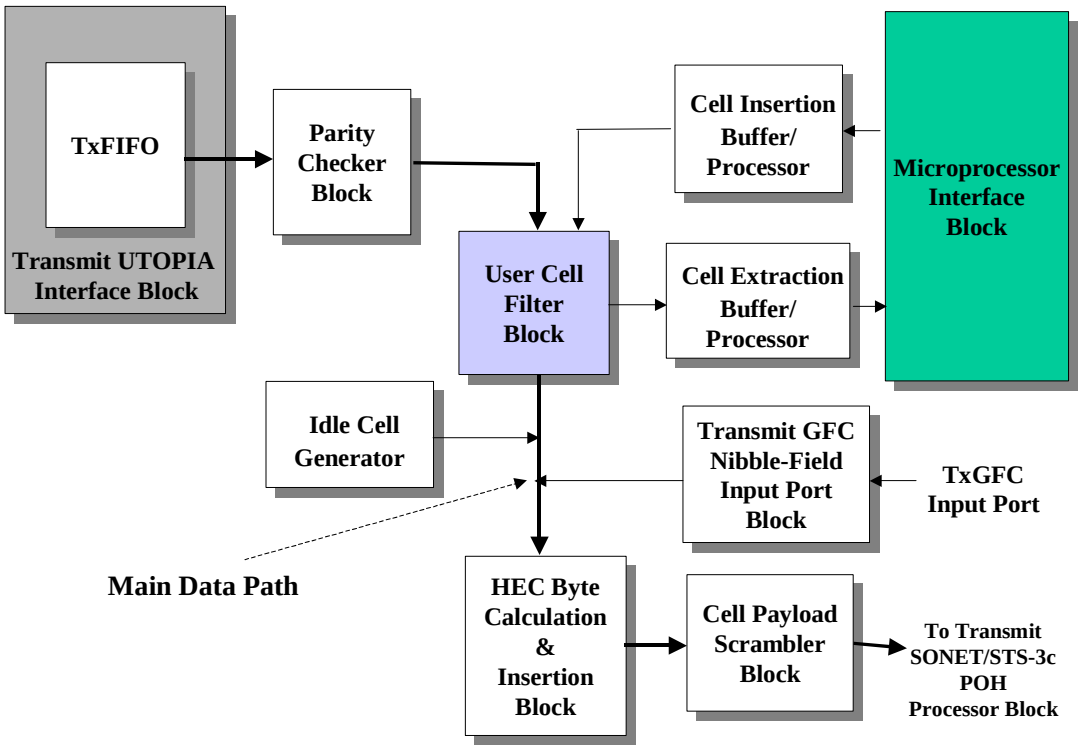
Transmit ATM Cell Processor – Parity Error Count Register – Byte 3 (Address = 0xNF37)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Number of Parity Errors[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

2.2.6 THE TRANSMIT USER CELL FILTER BLOCK

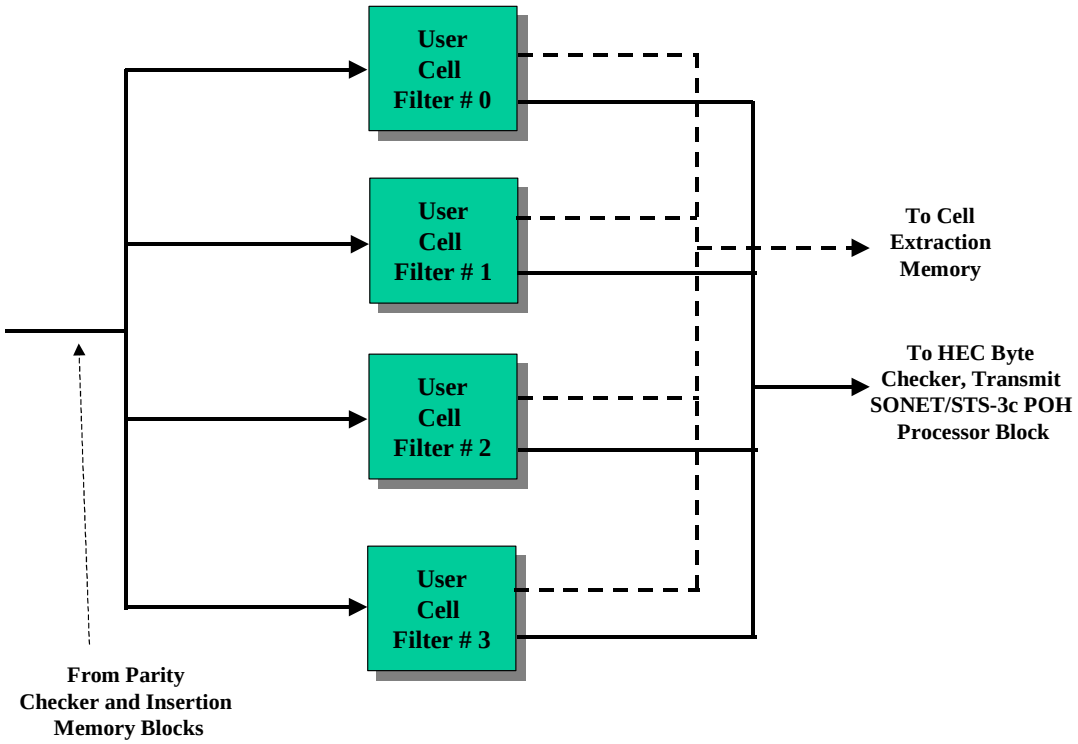
Once a given ATM cell has successfully passed through the “Parity-Checker” block, or has been read out of the “Transmit Cell Insertion” Memory buffer, it will now be processed by the “Transmit User Cell Filter” block. Figure 22 presents the functional block diagram of the Transmit ATM Cell Processor block with the “Transmit User Cell Filter” block highlighted.

Figure 22: Illustration of the "Transmit ATM Cell Processor" Functional Block Diagram with the "Transmit User-Cell Filter" block highlighted



The Transmit ATM Cell Processor block consists of four parallel User Cell Filters. These User Cell Filters are connected in parallel, in the sense that the entire "transmit" ATM cell traffic will be presented to the inputs of each of the four "user-cell" filters, at the same time. Figure 23 attempts to clarify this phenomenon by presenting an illustration as to how each of these user cell filters is connected to the ATM traffic.

Figure 23: An Illustration of the Configuration of the Four Transmit User Cell Filters within the Transmit ATM Cell Processor block



Each of these four (4) User Cell filters can be configured to analyze all of the header bytes, within a given ATM cell. Based upon the value of these header bytes (within a given cell), each of the four User Cell Filter blocks can be configured to perform either of the following functions.

- To filter (e.g., discard) the cell
- To NOT filter (e.g., permit the cell to pass through, along the Data Path) the cell
- To replicate (or copy) the cell, and route the replicate cell to the "Transmit ATM Cell Processor" block – Extraction Processor block.
- To NOT replicate the cell.

Instructions on how to implement the following User Cell Filter options are presented below.

- Enabling/Disabling the User Cell Filter
- Defining the User Cell Filter Action
- Defining the User Cell Filter Sense
- Specifying the User Cell Filter Values

2.2.6.1 ENABLING/DISABLING THE TRANSMIT USER CELL FILTER

The Transmit ATM Cell Processor block permits the user to either enable or disable each of the four Transmit User Cell Filters. The user can accomplish this by writing the appropriate data into Bit 3 (User Cell Filter Enable), within the "Transmit ATM Filter Control – Byte 0" register; as depicted below.

Transmit ATM Filter # 0, # 1, # 2, # 3 Control – Byte 0 (Address = 0xNF43, 0xNF53, 0xNF63, 0xNF73)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	0	0	0

Note: *As the bit-format table for the "Transmit ATM Filter Control – Byte 0" register implies, each of the four (4) Transmit User Cell Filters can be individually enabled or disabled. Further, each of these four Transmit User Cell Filters can be individually configured to either "Copy" or "Discard" cells (or both).*

Setting this bit-field to "1" enables the corresponding Transmit User Cell Filter. Conversely, setting this bit-field to "0" disables the "Transmit User Cell Filter".

If a given Transmit User Cell Filter is enabled, then it will perform action on all incoming ATM cells, based upon the settings of the remaining register bits within this register; and that within both the "Transmit ATM Filter Control – Pattern" and "Transmit ATM Filter Control – Check" registers. Conversely, if the given Transmit User Cell Filter is disabled, then all User Cells will pass through the Transmit User Cell Filter, and towards the "HEC Byte Calculation & Insertion" block without any such actions performed on these cells.

2.2.6.2 SELECTING THE TRANSMIT USER CELL FILTER ACTION

The Transmit ATM Cell Processor block permits the user to specify the action that the User Cell Filter will take on each cell, that meets certain "User-Defined" Filtering requirements. The user can accomplish this by setting Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable), within the Transmit ATM Filter Control – Byte 0" Register; to the appropriate values. These two bit-fields are highlighted below.

Transmit ATM Filter # 0, # 1, # 2, # 3 Control – Byte 0 (Address = 0xNF43, 0xNF53, 0xNF63, 0xNF73)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell Enable	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	X	X	X

Bit 2 – Copy Cell Enable

This bit-field permits the user to configure the "Transmit User Cell" filter (within the Transmit ATM Cell Processor block) to either replicate (copy) or not replicate a given cell that complies with the "user-defined" filtering requirements. All copied cells are routed to the "Transmit ATM Cell Processor Extraction Memory", where their contents can be read out by the user via the Microprocessor Interface.

Setting this bit-field to "1" configures the Transmit User Cell Filter to copy all cells that comply with the "user-cell" filter requirements, and to route these cells to the "Transmit ATM Cell Processor – Extraction Memory".

Conversely, setting this bit-field to "0" configures the Transmit User Cell Filter to NOT copy these cells, and NOT route these cells to the "Transmit ATM Cell Processor – Extraction Memory.

Bit 1 – Discard Cell Enable

This bit-field permits the user to configures the "Transmit User Cell" filter (within the Transmit ATM Cell Processor block) either discard or not discard a given cell that complies with the "user-defined" filtering requirements. Cells that are discarded will not be routed to the "Transmit STS-3c POH Processor" block. Cells that are NOT discarded will proceed on through the remainder of the Transmit ATM Cell Processor block circuitry, for further processing.

Setting this bit-field to "1" configures the Transmit User Cell Filter to discard all cells that comply with the "user-cell" filter requirements.

Conversely, setting this bit-field to "0" configures the Transmit User Cell Filter to NOT discard the cells that comply with the "user-cell" filter requirements.

2.2.6.3 DEFINING THE USER CELL FILTER SENSE

The Sense of a given Transmit User Cell Filter is defined by the state of Bit 0 (User Cell Filter Sense) within the "Transmit ATM Filter Control – Byte 0" Register; as described below.

Bit 0 - User Cell Filter Sense

Transmit ATM Filter # 0 , # 1, # 2, # 3 Control – Byte 0 (Address = 0xNF43, 0xNF53, 0xNF63, 0xNF73)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell Enable	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	X	X	X

This bit-field controls the "Filter Mode". If the user sets this bit-field to "1" then the "Transmit User-Cell" Filter will act, per the settings within Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable) on cells with header byte patterns MATCHING the "user-cell" filtering criteria. Conversely, setting this bit-field to "0" configures the "Transmit User-Cell" Filter to act, per the settings within Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable) on cells with header byte patterns NOT MATCHING the "user-cell" filtering criteria.

2.2.6.4 SPECIFYING THE TRANSMIT USER CELL FILTERING CRITERIA

As described above, each of the four Transmit User Cell Filters (within the Transmit ATM Cell Processor block) can be configured to perform a variety of actions (e.g., copy cells, discard cells, etc); based upon whether the Header Byte Patterns of User Cells MATCH; or DO NOT MATCH a particular "User-Defined" Filter criteria.

The "User-Defined" Filter criteria (for each of the four Transmit User Cell Filter blocks, within the Transmit ATM Cell Processor block) are ultimately defined by the values residing within a total of eight (8) registers. Four of these registers are referred to as "User Cell Filter – Pattern" Registers; and the remaining four registers are referred to as "User Cell Filter – Check" Registers. Each of these register types are defined below.

2.2.6.5 TRANSMIT USER CELL FILTER – PATTERN REGISTERS

The four User Cell Filter – Pattern Registers permit the user to specify the Header Byte Pattern for the Transmit User Cell Filter. There are four User Cell Filter – Pattern Registers (one for each of the four (4) header bytes, within an ATM cell). The bit-format of these "User Cell Filter – Pattern Registers" is presented below.

Transmit ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 1 (Address = 0xNF44, 0xNF54, 0xNF64, 0xNF74)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 2 (Address = 0xNF45, 0xNF55, 0xNF65, 0xNF75)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 3 (Address = 0xNF46, 0xNF56, 0xNF66, 0xNF76)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 3[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 4 (Address = 0xNF47, 0xNF57, 0xNF67, 0xNF77)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 4[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *Each of the four Transmit User Cell Filters (within the Transmit ATM Cell Processor block) contains their own set of "User Cell Filter – Pattern" registers.*

2.2.6.6 TRANSMIT USER CELL FILTER – CHECK REGISTERS

The four User Cell Filter – Check Registers permit the user to specify which bits (within the Header bytes of User Cells) will be checked and compared with the contents of the "User Cell Filter – Pattern Registers". There are four (4) "User Cell Filter-Check" Registers (one for each of the four header bytes within an ATM cell). Setting a particular bit-field to "0" configures the "Transmit User Cell Filter" to NOT compare the corresponding bit-field (within the header bytes of the incoming user cell) to the corresponding bit-value, within the User Cell Filter – Pattern Register; during User Cell pattern checking. Conversely, setting a particular bit-field to "1" configures the "Transmit User Cell Filter" to compare the corresponding bit-field (within the header bytes of the incoming user cell) to the corresponding bit-value, within the User Cell Filter – Pattern Register; during User Cell pattern checking.

For example, if the user were write the value "0x00" into a given "User Cell Filter – Check Register", then the Transmit User Cell Filter will NOT check and compare the value of the corresponding header byte, within an incoming User Cell; to the contents of the corresponding "User Cell Filter – Pattern" register during User Cell pattern checking. Conversely, if the user were to write the value "0xFF" into a given "User Cell Filter – Check Register", then the User Cell Filter will check and compare all eight bits (within the corresponding byte) within an incoming User Cell; to the contents of the corresponding "User Cell Filter – Pattern" register during User Cell pattern checking.

The bit-format of the four "User Cell Filter – Check Registers" is presented below.

Transmit ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 1 (Address = 0xNF48, 0xNF58, 0xNF68, 0xNF78)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 2 (Address = 0xNF49, 0xNF59, 0xNF69, 0xNF79)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 3 (Address = 0xNF4A, 0xNF5A, 0xNF6A, 0xNF7A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 3[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 4 (Address = 0xNF4B, 0xNF5B, 0xNF6B, 0xNF7B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 4[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *Each of the four User Cell Filters (within the Transmit ATM Cell Processor block) contains their own set of "User Cell Filter – Check" registers.*

2.2.6.7 COUNTING FILTERED CELLS

The user can keep account of the number of User Cells that have been filtered out by the Transmit User Cell Filter by periodically reading out the contents of the "Transmit ATM Filter Count" Registers. Each time a user cell is filtered (e.g., either discarded or copied) by the "Transmit User Cell" Filter, then this counter will be incremented by the value of "1". This counter will increment until it reaches the value of 0xFFFFFFFF. At this time, the counter will "saturate" and will not increment any further until read. These registers are 32 bits, in width and are "RESET-upon-READ. The bit-format for these registers (for each of the four filters) is presented below.

Transmit ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 3 (Address = 0xNF4C, 0xNF5C, 0xNF6C, 0xNF7C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 2 (Address = 0xNF4D, 0xNF5D, 0xNF6D, 0xNF7D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 1 (Address = 0xNF4E, 0xNF5E, 0xNF6E, 0xNF7E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 0 (Address = 0xNF4F, 0xNF5F, 0xNF6F, 0xNF7F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Note: *Each of the four User Cell Filters (within the Transmit ATM Cell Processor block) contains their own set of "Filtered Cell Counter" registers.*

The Procedure for Reading Out the Contents of the Transmit ATM Filter – Filtered Cell Counter Register

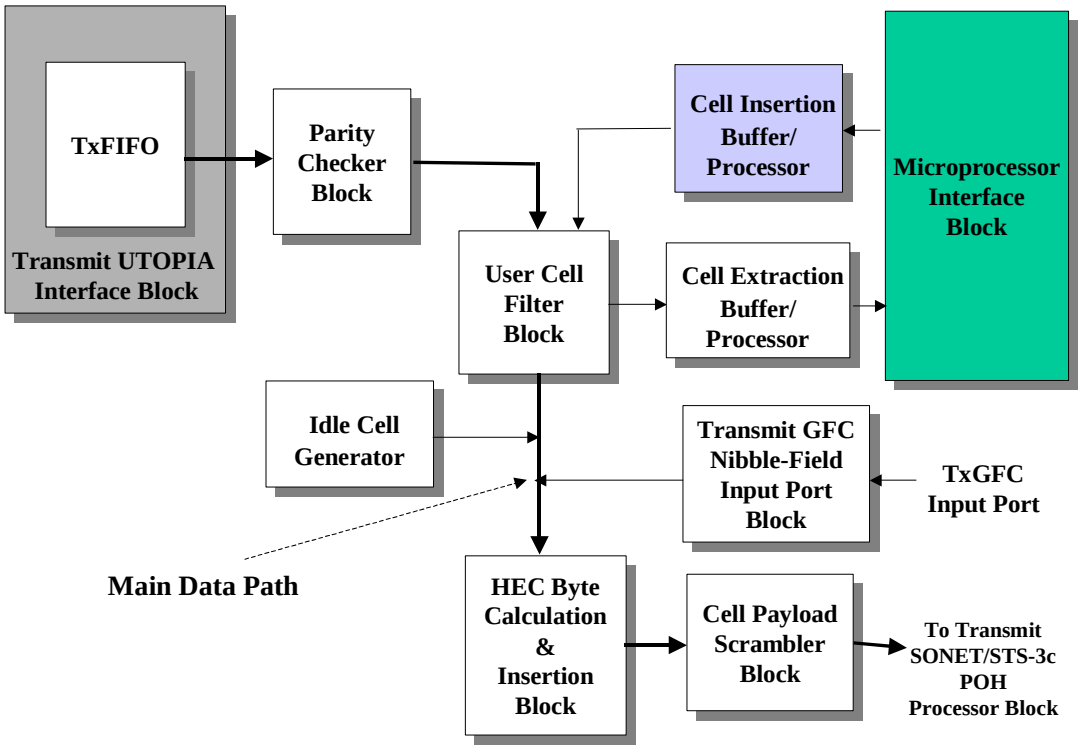
In order to properly read out the contents of these registers, the user must read these registers in the following order.

- Transmit ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 3 (Address = 0xNF4C, 0xNF5C, 0xNF6C, 0xNF7C)
- Transmit ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 2 (Address = 0xNF4D, 0xNF5D, 0xNF6D, 0xNF7D)
- Transmit ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 1 (Address = 0xNF4E, 0xNF5E, 0xNF6E, 0xNF7E)
- Transmit ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 0 (Address = 0xNF4F, 0xNF5F, 0xNF6F, 0xNF7F)

2.2.6.8 THE TRANSMIT CELL INSERTION BUFFER/PROCESSOR

The Transmit ATM Cell Processor block consists of a "Transmit Cell Insertion Buffer/Processor" block. Figure 24 presents the functional block diagram of the Transmit ATM Cell Processor block with the "Transmit Cell Insertion Buffer/Processor" block highlighted.

Figure 24: Illustration of the Transmit ATM Cell Processor block Functional Block Diagram, with the "Transmit Cell Insertion Buffer/Processor" block highlighted



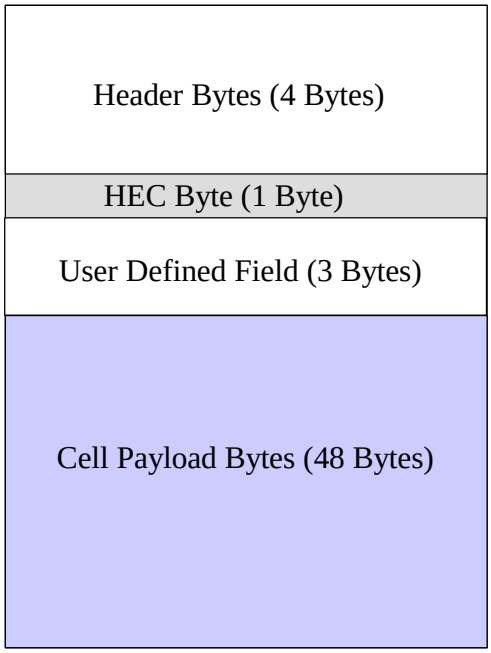
The Transmit Cell Insertion Buffer/Processor block permits the user to load the contents of an "outbound" ATM cell into the "Transmit Cell Insertion Buffer" via the Microprocessor Interface. Once this cell has been loaded into the "Transmit Cell Insertion Buffer", then it will be transmitted to the remote terminal equipment (via the Transmit Data Path) whenever the "TxFIFO" (within the Transmit UTOPIA Interface block) is depleted of user cells. An example of an application where this feature would be useful is in the loading and transmission of OAM cells.

The Format of ATM Cell Data that is written into the "Transmit Cell Insertion" Buffer

As the user loads the contents of an ATM cell into the "Transmit Cell Insertion" Buffer (via the Microprocessor Interface), they will be expected to write this ATM cell data into a 32 bit wide register/buffer interface. As a consequence, the user must write in 56-byte size ATM cells into the "Transmit Cell Insertion" buffer.

The byte format of this 56 byte ATM cell is as illustrated below in Figure 25.

Figure 25: Byte-Format of the ATM Cell that is to be loaded into the "Transmit Cell Insertion" Memory



As a consequence, the user must write in a total of 14 "32-bit words" into the "Transmit Cell Insertion" buffer for each ATM cell that is written into the "Transmit Cell Insertion" Buffer.

Procedure for Writing a Cell into the "Transmit Cell Insertion Buffer"

The user can write an ATM cell into the "Transmit Cell Insertion Buffer" within the Transmit ATM Cell Processor block, by executing the following steps.

STEP 1 – Flush the contents of the "Transmit Cell Insertion Buffer"

This is accomplished by executing a "Transmit Cell Insertion Buffer" RESET. The user can perform this "Transmit Cell Insertion Buffer" RESET by doing the following.

STEP 1a – Write a "0" into Bit 2 (Insertion Memory RESET*), within the "Transmit ATM Cell – Memory Control" register; as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1->0	1	0

STEP 1b – Write a "1" into Bit 2 (Insertion Memory RESET*), within the "Transmit ATM Cell – Memory Control" Register; as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0->1	1	0

Note: *This step should typically be performed upon power-up, prior to writing in any ATM cell data into the "Transmit Cell Insertion Buffer". This step is not necessary after the first cell has been written into the "Transmit Cell Insertion Buffer" following a power cycle to the chip.*

STEP 2 – Check and Verify that there is sufficient space available (within the Transmit Cell Insertion Buffer) to handle this cell.

This can be accomplished by one of two approaches.

- Polling approach
- Interrupt-driven approach.

Each of these approaches is described below.

Executing STEP 2 using the Polling Approach

The user can determine whether or not there is room (to write another ATM cell of data) in to the "Transmit Cell Insertion" Buffer" by polling the state of Bit 1 (Insertion Memory ROOM) within the "Transmit ATM Cell – Memory Control Register" as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	X	0

If Bit 1 (Insertion Memory ROOM) is set to "1" then the "Transmit Cell Insertion Buffer" is NOT too full to accept another cell. At this point, the Microprocessor can now move onto STEP 3.

Conversely, if Bit 1 is set to "0" then the "Transmit Cell Insertion Buffer" is too full to accept another cell. The Microprocessor Interface should continue to poll the state of this bit-field and wait until this bit-field toggles to "1".

Executing STEP 2 using the Interrupt-Driven Approach

In order to reduce or eliminate the Microprocessor Overhead of continuously polling the state of Bit 1, the user can use the "Transmit Cell Insertion" Interrupt feature, within the chip. If the Microprocessor invokes this feature, then the XRT94L33 will generate an interrupt anytime a cell (residing in the Transmit Cell Insertion Buffer) has been inserted into the "Transmit Output Data Path" (thereby freeing up some space within the Transmit Cell Insertion Buffer).

The user can enable the "Transmit Cell Insertion" Interrupt by setting Bit 4 (Cell Insertion Interrupt Enable), within the "Transmit ATM Cell Processor – Interrupt Enable" Register to "1" as indicated below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once a cell (residing within the Transmit Cell Insertion Buffer) has been inserted into the "Transmit Output Data Path", then the XRT94L33 will do all of the following:

- It will toggle the "INT*" output pin "LOW".
- It will set Bit 4 (Cell Insertion Interrupt Status), within the Transmit ATM Cell Processor – Interrupt Status Register; to "1" as depicted below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

At this point, the user can now proceed on with STEP 3.

STEP 3 – Inform the "Transmit Cell Insertion Processor" that the very next 32-bit word to be written into the "Transmit Cell Insertion Buffer" is the first word of a new ATM cell.

This is accomplished by writing the value "1" into Bit 0 (Insertion Memory Write SoC), within the "Transmit ATM Cell – Memory Control Register" as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	1

STEP 4 – Write the very first 32-bit word of this new ATM cell into the "Transmit Cell Insertion Buffer".

This is accomplished by executing the following four sub-steps.

STEP 4a - Write the contents of first byte (of this new ATM cell) into the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 3; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 3 (Address = 0xNF14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A7	A6	A5	A4	A3	A2	A1	A0

Note: *In this case, the contents of the first byte (within this ATM cell) is of the value [A7, A6, A5, A4, A3, A2, A1, A0]*

STEP 4b – Write the contents of the second byte (of this new ATM cell) into the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B7	B6	B5	B4	B3	B2	B1	B0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [B7, B6, B5, B4, B3, B2, B1, B0]*

STEP 4c – Write the contents of the third byte (of this new ATM cell) into the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 1; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF16)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
C7	C6	C5	C4	C3	C2	C1	C0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [C7, C6, C5, C4, C3, C2, C1, C0]*

STEP 4d – Write the contents of the fourth byte (of this new ATM cell) into the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 0; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D7	D6	D5	D4	D3	D2	D1	D0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [D7, D6, D5, D4, D3, D2, D1, D0]*

STEP 5 – Inform the "Cell Insertion Processor" that the very next 32-bit word to be written into the "Cell Insertion Buffer" is NOT the first word of a new ATM cell.

This is accomplished by writing the value "0" into Bit 0 (Insertion Memory Write SoC), within the "Transmit ATM Cell – Memory Control Register" as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	0

STEP 6 – Continue to write the contents of the remaining bytes of this ATM cell into the "Cell Insertion Memory".

This is accomplished by executing the procedure, outlined in STEP 4, repeatedly for 13 times.

Notes:

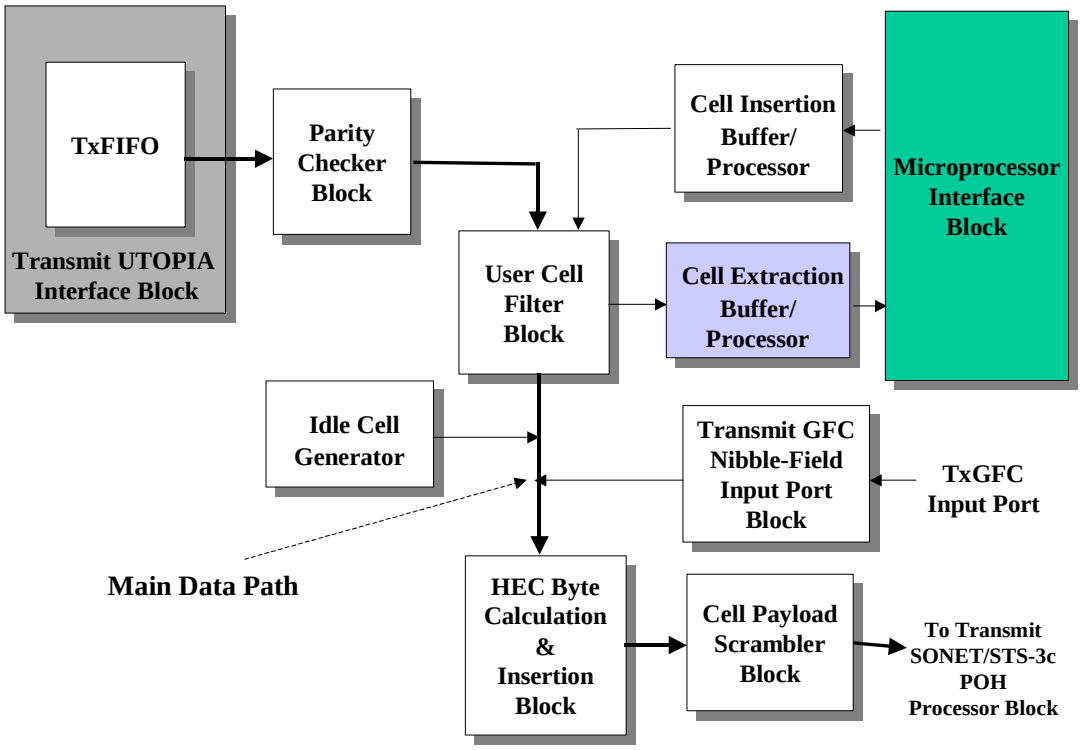
As the user writes the contents of an ATM cell into the "Transmit Cell Insertion Memory" they must follow the mandated ATM cell byte format, as depicted in

1. .
2. If the user performs a READ operation to the "Transmit ATM Cell – Insertion/Extraction Memory Register – Bytes 3 through 0" they will NOT be reading out the contents of the Transmit Cell Insertion Buffer. Instead, the user will be reading out data from the "Transmit Cell Extraction" buffer.

2.2.6.9 THE TRANSMIT CELL EXTRACTION BUFFER/PROCESSOR

The Transmit ATM Cell Processor block consists of a "Transmit Cell Extraction Buffer/Processor" block. Figure 26 presents the functional block diagram of the "Transmit ATM Cell Processor block with the "Transmit Cell Extraction Buffer/Processor" block highlighted.

Figure 26: Illustration of the Transmit ATM Cell Processor block Functional Block Diagram, with the "Transmit Cell Extraction Buffer/Processor" block highlighted

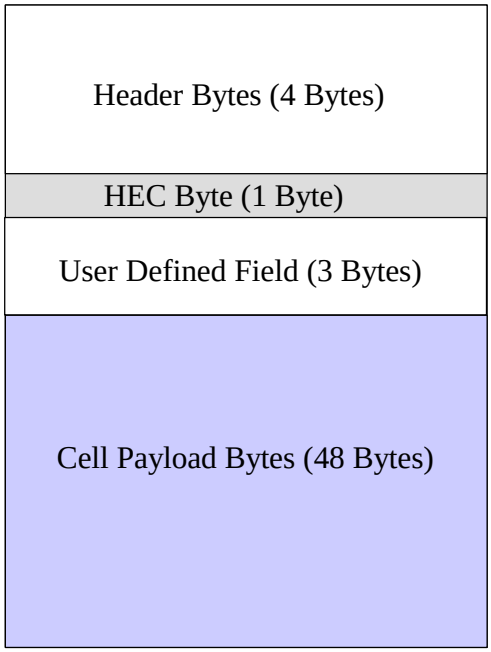


The Transmit Cell Extraction Buffer/Processor block permits the user to read out the contents of an "inbound" ATM cell via the Microprocessor Interface. If the user configures the "Transmit User Cell Filter" appropriately, the "Transmit User Cell Filter" will copy (e.g., replicate) the contents of certain cells (which comply with the user cell filtering requirements). These copied cells will be routed to the "Transmit Cell Extraction Buffer/Processor" block where it can be read out and processed by the Microprocessor Interface.

The Format of the ATM Cell that is Read from the "Transmit Cell Extraction" Buffer

As the user reads out the contents of an ATM cell from the "Transmit Cell Extraction" Buffer (via the Microprocessor Interface), they will be expected to read this ATM cell data via a 32-bit wide register/buffer interface. As a consequence, the user must read out 56 byte size ATM cells from the "Transmit Cell Extraction" Buffer. The byte format of this 56-byte ATM cell is as illustrated below.

Figure 27: Byte Format of the ATM Cells that are read out from the "Transmit Cell Extraction" Memory



As a consequence, the user must read out a total of 14 "32-bit" words from the "Transmit Cell Extraction" buffer for each ATM cell that is read out of the Transmit Cell Extraction Buffer.

Procedure for Reading a Cell from the "Transmit Cell Extraction Buffer"

The user can read an ATM cell from the "Transmit Cell Extraction Buffer" within the Transmit ATM Cell Processor block, by executing the following steps.

STEP 1 – Flush the contents of the "Transmit Cell Extraction Buffer"

This is accomplished by executing a "Transmit Cell Extraction Buffer" RESET. The user can perform this "Cell Extraction Buffer" RESET by doing the following.

STEP 1a – Write a "0" into Bit 4 (Extraction Memory RESET*) within the "Transmit ATM Cell – Memory Control" Register; as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1->0	0	1	1	0

STEP 1b – Write a "1" into Bit 4 (Extraction Memory RESET*), within the "Transmit ATM Cell – Memory Control" Register; as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0->1	0	1	1	0

Note: *This step should typically be performed upon power-up, prior to processing any ATM cell traffic through the XRT94L33. This step is not necessary after the first cell has been read from the "Transmit Cell Extraction Buffer" following a power cycle to the chip.*

STEP 2 – Check and see if an ATM cell exists in the "Transmit Cell Extraction Buffer"

This can be accomplished by one of two approaches.

- Polling approach
- Interrupt-driven approach

Each of these approaches is described below.

Executing STEP 2 using the Polling Approach

The user can determine whether or not a cell is available, within the "Transmit Cell Extraction Buffer" by testing the state of Bit 3 (Extraction Memory CLAV) within the Transmit ATM Cell – Memory Control Register; as depicted below.

Transmit ATM Cell – Memory Control Register (Address = 0xNF13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	X	1	1	0

If Bit 3 is set to "1", then the "Transmit Cell Extraction Buffer" contains an ATM cell that needs to be read. At this point, the user should proceed on to STEP 3.

Conversely, if Bit 3 is set to "0", then the "Transmit Cell Extraction Buffer" does not contain an ATM cell that needs to be read. At this point, the Microprocessor Interface should continue to poll the state of this bit-field and wait until this bit-field toggles to "1".

Executing STEP 2 using the Interrupt-Driven Approach

In order to reduce or eliminate the Microprocessor Overhead of continuously polling the state of Bit 3, the user can use the "Transmit Cell Extraction" Interrupt feature, within the chip. If the Microprocessor invokes this feature, then the XRT94L33 will generate an interrupt anytime a new cell has been received and loaded into the "Transmit Cell Extraction Buffer".

The user can enable the "Cell Extraction" Interrupt by setting Bit 5 (Cell Extraction Interrupt Enable), within the "Transmit ATM Cell Processor – Interrupt Enable" Register to "1" as indicated below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

Once the "Cell Extraction Buffer" receives a "COPIED" cell from the "Transmit User Cell Filter", then the XRT94L33 will do all of the following:

- It will toggle the "INT*" output pin "LOW".
- It will set Bit 5 (Cell Extraction Interrupt Status) within the Transmit ATM Cell Processor – Interrupt Status Register, to "1" as depicted below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

At this point, the user can now proceed onto STEP 3.

STEP 3 – Read out the very first 32-bit word of this new ATM cell from the "Transmit Cell Extraction Buffer".

This is accomplished by executing the following four sub-steps.

STEP 3a – Read the contents of the first byte (of this newly received ATM cell) from the "Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 3; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 3 (Address = 0xNF14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A7	A6	A5	A4	A3	A2	A1	A0

Note: *In this case, the contents of the first byte (within this ATM cell) is of the value [A7, A6, A5, A4, A3, A2, A1, A0]*

STEP 3b – Read the contents of the second byte (of this new ATM cell) from the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B7	B6	B5	B4	B3	B2	B1	B0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [B7, B6, B5, B4, B3, B2, B1, B0]*

STEP 3c – Read the contents of the third byte (of this new ATM cell) from the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 1; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF16)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
C7	C6	C5	C4	C3	C2	C1	C0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [C7, C6, C5, C4, C3, C2, C1, C0]*

STEP 3d – Read the contents of the fourth byte (of this new ATM cell) from the Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 0; as depicted below.

Transmit ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xNF17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D7	D6	D5	D4	D3	D2	D1	D0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [D7, D6, D5, D4, D3, D2, D1, D0]*

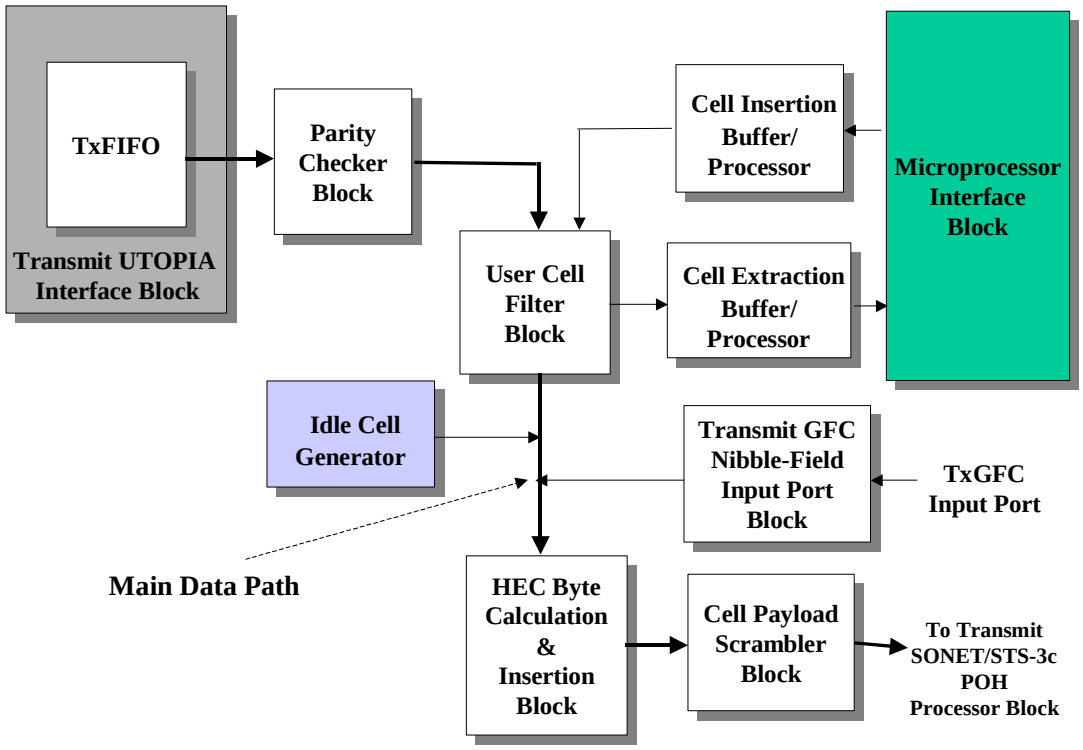
STEP 4 – Continue to read out the contents of the remaining bytes of this ATM cell from the "Transmit Cell Extraction Memory".

This is accomplished by executing the procedure, outlined in STEP 3, repeatedly for 13 more times.

2.2.6.10 THE IDLE CELL GENERATOR

The Transmit ATM Cell Processor block consists of a "Idle Cell Generator" block. Figure 28 presents the functional block diagram of the "Transmit ATM Cell Processor block with the "Idle Cell Generator" block highlighted.

Figure 28: Illustration of the Transmit ATM Cell Processor block Functional Block Diagram, with the "Idle Cell Generator" block highlighted



Whenever the TxFIFO (within the Transmit UTOPIA Interface block) does not contain a complete cell, the Transmit ATM Cell Processor will first read out any ATM cell data that resides within the "Transmit Cell Insertion Buffer" and will insert this (or these) ATM cells into the "Transmit Data Path". Once both the "TxFIFO" and the "Transmit Cell Insertion Buffer" are depleted, then the Transmit ATM Cell Processor block will automatically fill in the STS-3 SPE bandwidth by generating and transmitting Idle Cells via the "Transmit Data Path". By default, the Transmit ATM Cell Processor block will generate Idle Cells that contain header byte patterns which conform to the ATM Forum recommendations. However, the XRT94L33 does contain some registers that permit the user to "customize" the header byte and payload byte pattern of these Idle cells. The procedure for configuring the Idle Cell Generator to generate and transmit Idle Cells with "customized" header and payload bytes is presented below.

The Procedure for configuring the Transmit ATM Cell Processor Block to transmit Idle Cells with "user-specified" header and payload bytes

The user can configure the Transmit ATM Cell Processor block to generate and transmit Idle Cells with "user-specified" header and payload bytes, by executing the following steps.

STEP 1 – Write the "desired value" for header byte 1 (within these Idle Cells) into the "Transmit ATM Cell – Idle Cell Header Byte 1 Register; as depicted below.

Transmit ATM Cell – Idle Cell Header Byte 1 Register (Address = 0xNF18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Idle_Cell_Header_Byte_1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A7	A6	A5	A4	A3	A2	A1	A0

STEP 2 – Write the "desired value" for header byte 2 (within these Idle Cells) into the "Transmit ATM Cell – Idle Cell Header Byte 2 Register; as depicted below.

Transmit ATM Cell – Idle Cell Header Byte 2 Register (Address = 0xNF19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Idle_Cell_Header_Byte_2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B7	B6	B5	B4	B3	B2	B1	B0

STEP 3 – Write the "desired value" for header byte 3 (within these Idle Cells) into the "Transmit ATM Cell – Idle Cell Header Byte 3 Register; as depicted below.

Transmit ATM Cell – Idle Cell Header Byte 3 Register (Address = 0xNF1A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Idle_Cell_Header_Byte_3[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
C7	C6	C5	C4	C3	C2	C1	C0

STEP 4 – Write the "desired value" for header byte 4 (within these Idle Cells) into the "Transmit ATM Cell – Idle Cell Header Byte 4 Register; as depicted below.

Transmit ATM Cell – Idle Cell Header Byte 4 Register (Address = 0xNF1B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Idle_Cell_Header_Byte_4[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D7	D6	D5	D4	D3	D2	D1	D0

STEP 5 – Write the "desired value" for the payload byte (within these Idle Cells) into the "Transmit ATM Cell – Idle Cell Payload Byte Register; as depicted below.

Transmit ATM Cell – Idle Cell Payload Byte Register (Address = 0xNF1F)

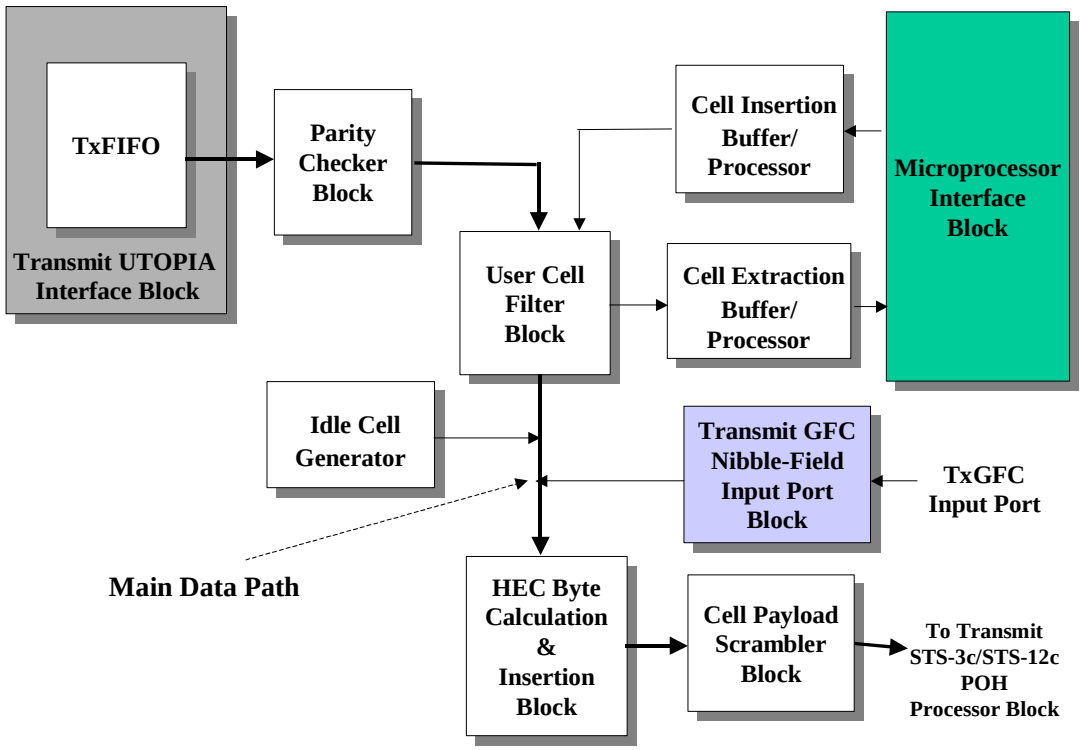
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Idle_Cell_Payload_Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
P7	P6	P5	P4	P3	P2	P1	P0

Note: *Each of the 48 bytes, within the Idle Cell Payload will be set to the value written into this register.*

2.2.6.11 THE TRANSMIT GFC NIBBLE-FIELD SERIAL INPUT PORT

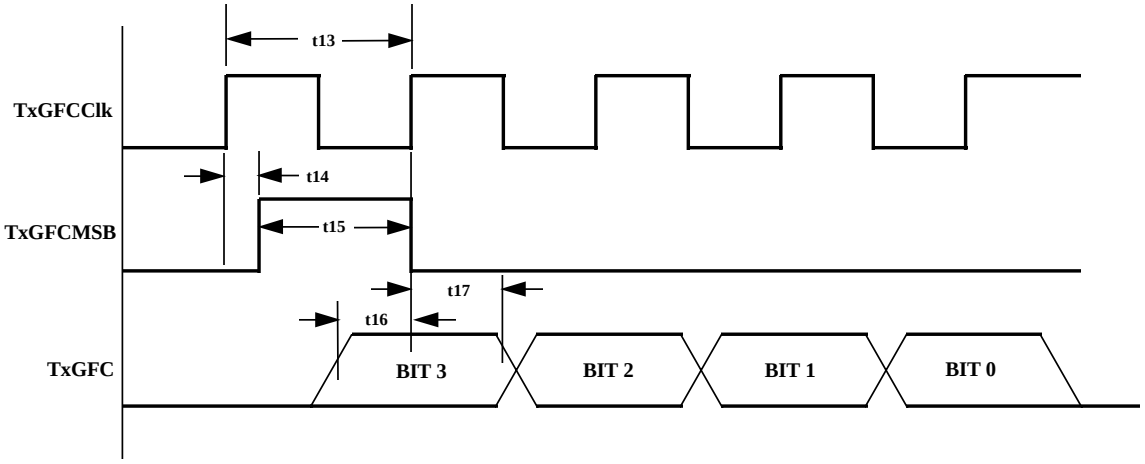
The Transmit ATM Cell Processor block consists of a "Transmit GFC Nibble-Field" Input port. Figure 29 presents the Functional Block Diagram of the "Transmit ATM Cell Processor" block with the "Transmit GFC Nibble-Field Serial Input Port" block highlighted.

Figure 29: An Illustration of the Functional Block Diagram of the Transmit ATM Cell Processor Block, with the "Transmit GFC Nibble-Field Serial Input Port" block highlighted



As the name of this port implies, this port permits the user to insert his/her value for the GFC nibble into each outbound ATM Cell.

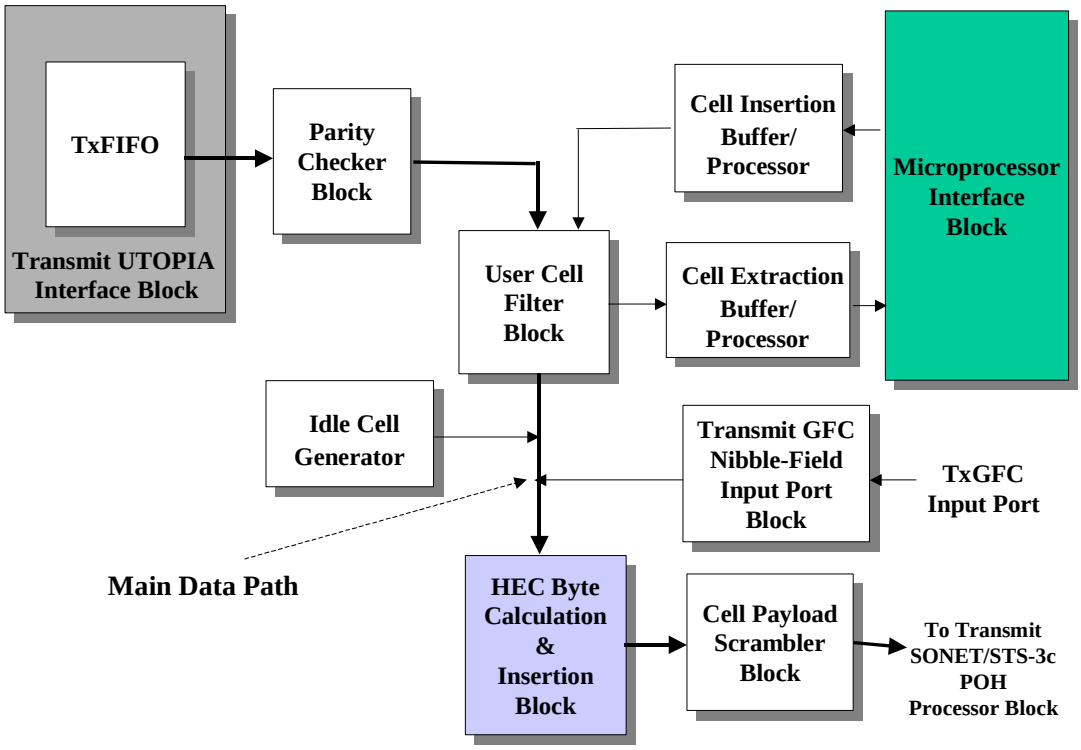
Figure 30: Behavior of TxGFC, TxGFCClk, and TxGFCMSB during GFC insertion into the "Outbound"



2.2.6.12 THE HEC BYTE CALCULATION AND INSERTION BLOCK

The Transmit ATM Cell Processor block consists of a "HEC Byte Calculation and Insertion" block. Figure 31 presents the functional block diagram of the "Transmit ATM Cell Processor" block with the "HEC Byte Calculation and Insertion" block highlighted.

Figure 31: Illustration of the Transmit ATM Cell Processor block Functional Block Diagram with the HEC Byte Calculator and Insertion" block highlighted



The "HEC Byte Calculation & Insertion" block receives either filtered User Cells or Idle cells from the upstream circuitry, within the Transmit ATM Cell Processor block. As it receives these cells, the "HEC Byte Calculation & Insertion" block can be configured to perform any of the following actions on these cells.

- To compute and verify the HEC bytes of incoming cells from upstream circuitry within the Transmit ATM Cell Processor block
- To regenerate the HEC byte (e.g., to compute and insert the HEC bytes of incoming cells)
- To add the Coset Polynomial to the HEC byte, prior to transmission via the "Transmit Data Path"
- A detailed description of the HEC Byte Calculation & Insertion block is presented below.

Computing and Verifying HEC Bytes of incoming cells

As the HEC Byte Calculation & Insertion block receives ATM cells, it takes the first four bytes of each cell and computes a CRC-8 value via the generating polynomial $x^8 + x^2 + x + 1$.

Note: *The user has the option to have the coset polynomial $x^6 + x^4 + x^2 + 1$ modulo-2 added to the CRC-8 byte and, insert this newly modified CRC-8 value into the fifth octet position within the cell before transmission. This option will be discussed later in this section.*

Configuring the HEC Byte Calculation & Insertion block to Check for HEC Byte Errors

If the user wishes to configure the Transmit ATM Cell Processor block to compute, verify and flag HEC byte errors in cells originating from upstream circuitry, then they must write a "1" into Bit 6 (HEC Byte Check Enable) within the "Transmit ATM Control – Byte 0, as depicted below.

Transmit ATM Control – Byte 0 (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	1	0	X	X	0	0	X

If the user implements this configuration option, then the HEC Byte Calculation & Insertion Block will compute and verify the HEC byte, within all ATM cells that it receives from the Idle Cell Generator and the Transmit User Cell Filter blocks. If the HEC Byte Calculation & Insertion Block detects a HEC byte error, then the Transmit ATM Cell Processor block will generate the "Detection of HEC Byte Error" Interrupt. The Transmit ATM Cell Processor block will indicate that it is generating this interrupt by doing all of the following.

- It will toggle the "INT*" output pin "LOW".
- The Transmit ATM Cell Processor block will set Bit 1 (Detection of HEC Byte Error Interrupt Status), within the "Transmit ATM Cell Processor – Interrupt Status Register" as depicted below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR

0	0	0	0	0	0	1	0

- Finally, the Transmit ATM Cell Processor block will increment the "Transmit ATM Cell – HEC Byte Error Count" register. The "Transmit ATM Cell" HEC Byte Error Count Register is a 32-bit "RESET-upon-READ" register. The bit-format for this register is presented below.

Transmit ATM Cell – HEC Byte Error Count Register – Byte 3 (Address = 0xNF30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_HEC_Byte_Error_Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Cell – HEC Byte Error Count Register – Byte 2 (Address = 0xNF31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_HEC_Byte_Error_Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Cell – HEC Byte Error Count Register – Byte 1 (Address = 0xNF32)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_HEC_Byte_Error_Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Transmit ATM Cell – HEC Byte Error Count Register – Byte 0 (Address = 0xNF33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_HEC_Byte_Error_Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

In general, if the HEC Byte Calculation & Insertion block detects ATM cells that contains "Single-Bit" or "Multi-Bit" errors (within the Header bytes), then it will NOT correct these Header byte errors. Further, the Transmit ATM Cell Processor block will automatically discard these erred cells unless it has been configured to recalculate and insert a new HEC byte into the 5th octet position of each ATM cell (as discussed below in Section 4.2.2.7.2).

Note: *The Transmit ATM Cell – HEC Byte Error Count Register will be incremented by the value "1" each time the Transmit ATM Cell Processor detects a HEC byte error in the incoming ATM cell stream. This 32-bit register will continue to be incremented until it reaches the value "0xFFFFFFFF". At this point, this RESET-upon-READ register will saturate and will not increment any further, until the Microprocessor reads out the contents of these registers.*

If the user does not wish to configure the "HEC Byte Calculation & Insertion" block to check for HEC byte errors in the incoming ATM cell data-stream, then they should set Bit 6 (HEC Byte Check Enable), within the Transmit ATM Control – Byte 0 register to "0" as indicated below.

Transmit ATM Control – Byte 0 (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	0	X	X	0	0	X

In this mode, all ATM cells (even those with HEC byte errors) will pass through the "HEC Byte Calculation & Insertion" block, without any flagging of HEC byte errors.

Configuring the HEC Byte Calculation & Insertion block to Regenerate (e.g., recompute and insert) the HEC byte, prior to transmission via the "Transmit Data Path"

In addition to being capable of detecting and flagging HEC byte errors, the "HEC Byte Calculation & Insertion" block can be configured to recompute and insert a new HEC byte into the fifth octet position within all ATM cells that are processed by the "HEC Byte Calculation & Insertion" block. The user can implement this configuration by writing a "1" into Bit 0 (Re-Calculate HEC Byte Enable), within the "Transmit ATM Control – Byte 1" Register, as depicted below.

Transmit ATM Control – Byte 1 (Address = 0xNF02)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Test Cell Mode Enable	Test Cell Generator – One Shot	GFC_Enable[3:0]				Coset Addition	Re-Calculate HEC Byte Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Notes:

Setting this bit-field to "1" configures the "HEC Byte Calculation & Insertion" block to unconditionally compute a new HEC byte (based upon the value of the first four header bytes of a given ATM cell) and insert this value into the 5th octet position within each outbound ATM cell. Further, this configuration setting also configures the "HEC Byte Calculation & Insertion" block to NOT discard any ATM cells that it receives from upstream circuitry, even those cells that it determines to contain HEC byte errors.

This bit-field is ignored if Bit 6 (HEC Byte Check Enable) within the "Transmit ATM Control – Byte 0" register (Address = 0xNF03) is set to "0".

Other Options available via the "HEC Byte Calculation & Insertion" block

The user has the following additional options via the "HEC Byte Calculation & Insertion" block.

- To modulo-2 add the Coset Polynomial to the HEC byte, prior to transmission.
- To invert the HEC byte

Each of these options is discussed below.

Adding the Coset Polynomial to the HEC Byte

The "HEC Byte Calculation & Insertion" block permits the user to configure it to modulo-2 add the Coset Polynomal (e.g., $x^6 + x^4 + x^2 + 1$) to the CRC-8 value (e.g., the HEC byte). Afterwards, this "newly computed" HEC byte would be written back into the fifth octet position within each outbound ATM cell.

The user can implement this configuration option by writing a "1" into Bit 1 (Coset Addition) within the "Transmit ATM Control – Byte 1" Register, as depicted below.

Transmit ATM Control – Byte 1 (Address = 0xNF02)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Test Cell Mode Enable	Test Cell Generator – One Shot	GFC_Enable[3:0]				Coset Addition	Re-Calculate HEC Byte Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

Note: *In order to configure the "HEC Byte Calculation & Insertion" block to add the Coset polynomial to the HEC byte, it is imperative that the user also set Bit 0 (Re-Calculate HEC Byte Enable), within the same register; to "1" as depicted above.*

Inverting the HEC Byte

The "HEC Byte Calculation & Insertion" block permits the user to configure it to invert the value of each of the eight bits within the HEC byte of each outbound ATM cell. The user can implement this configuration option by executing the following steps.

STEP 1 – Configure the "HEC Byte Calculation & Insertion" block to compute and insert the HEC byte into the fifth octet position, within each "outbound" ATM cell.

This is accomplished by setting Bit 0 (Re-Calculate HEC Byte Enable), within the "Transmit ATM Control – Byte 1" register to "1", as depicted below.

Transmit ATM Control – Byte 1 (Address = 0xNF02)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Test Cell Mode Enable	Test Cell Generator – One Shot	GFC_Enable[3:0]				Coset Addition	Re-Calculate HEC Byte Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	X	1

STEP 2 – Configure the "HEC Calculation & Insertion" Block to now invert the contents of the HEC byte that it is computing.

This is accomplished by setting Bit 7 (HEC Byte Invert), within the "Transmit ATM Control – Byte 0" Register to "1" as depicted below.

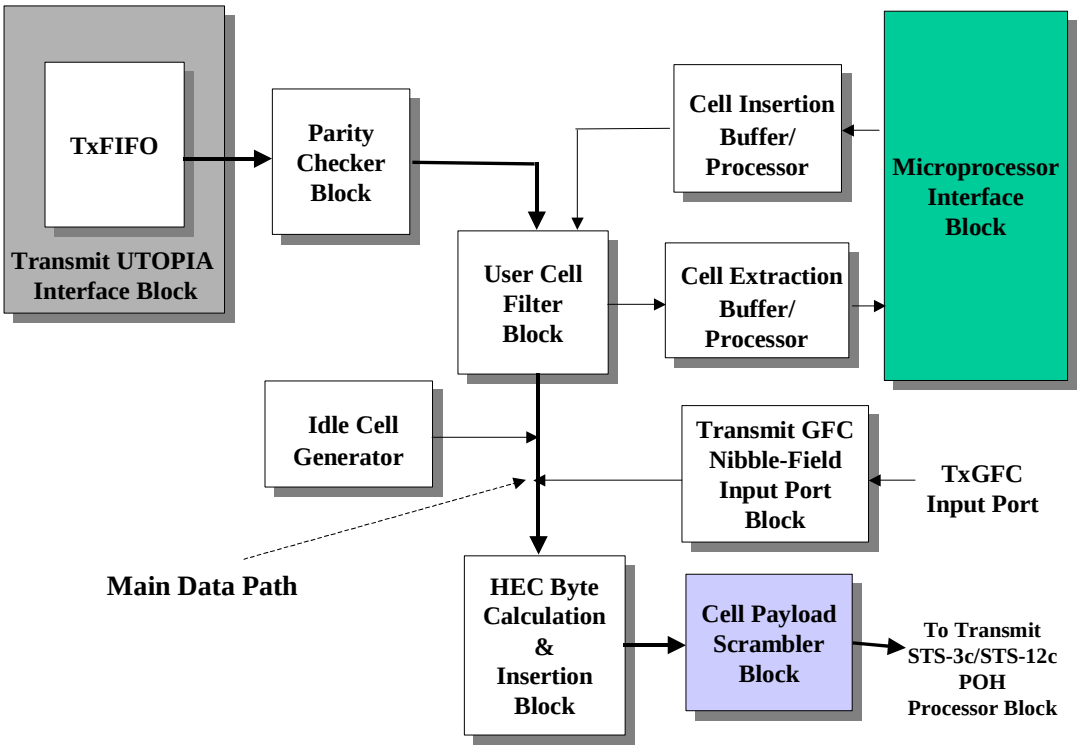
Transmit ATM Control – Byte 0 (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
1	0	0	X	X	0	0	X

2.2.6.13 THE CELL PAYLOAD SCRAMBLER BLOCK

The Transmit ATM Cell Processor block consists of a "Cell Scrambler" block. Figure 32 presents the functional block diagram of the "Transmit ATM Cell Processor block" with the "Cell Scrambler" block highlighted.

Figure 32: Illustration of the Transmit ATM Cell Processor block Functional Block Diagram with the "Cell Scrambler" block highlighted



If the "Cell Payload Scrambler" block is enabled, then it will scramble the payload bytes (within each outbound ATM cell) per the polynomial $x^{43} + 1$. Historically, the purpose of the Cell Payload Scrambler block is to prevent the contents of payload bytes from mimicking certain framing alignment bytes, within a given standard transport medium. However, this rationale is not very important in SONET applications, in which all bytes (with the exception of the A1 and A2 bytes) within the "outbound" STS-N or STS-Nc data-stream are also scrambled.

If should be noted that the "Cell Payload Scrambler" block only scrambles the payload bytes. The "Cell Payload Scrambler" block does not alter the value of the five (5) header bytes within each outbound ATM cell.

The user can enable the "Cell Payload Scrambler" block by setting Bit 1 (Cell Payload Scramble Enable), within the "Transmit ATM Control – Byte 0" Register; as depicted below.

Transmit ATM Control – Byte 0 (Address = 0xNF03)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Invert	HEC Byte Check Enable	Parity Check Enable	Discard Cell upon Parity Error	Odd Parity	Unused		Cell Payload Scramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
X	X	0	X	X	0	0	1

Note: *This particular setting does not enable nor disable the Cell De-Scrambler within the Receive ATM Cell Processor block. The user will need to separately enable or disable the Cell De-Scrambler per the instructions presented in Section _.*

2.2.6.14 THE TEST CELL GENERATOR BLOCK

The Transmit ATM Cell Processor block consists of a Test Cell Generator block. Similarly, the Receive ATM Cell Processor block has a corresponding Test Cell Receiver.

2.2.6.15 TRANSMIT ATM CELL PROCESSOR BLOCK INTERRUPTS

The Transmit ATM Cell Processor block is capable of generating the following interrupts.

- The “Transmit Cell Extraction” Interrupt
- The “Transmit Cell Insertion” Interrupt
- The “Transmit Cell Extraction Memory Overflow” Interrupt
- The “Transmit Cell Insertion Memory Overflow” Interrupt
- The “Detection of HEC Byte Error” Interrupt
- The “Detection of Parity Error” Interrupt

This section describes the following aspects of these interrupts.

- The conditions causing these interrupts to be declared
- Instructions on how to enable or disable interrupts
- Instructions on how to services these interrupts

The Transmit Cell Extraction Interrupt

The Transmit ATM Cell Processor block will generate the “Transmit Cell Extraction” interrupt anytime the “Cell Extraction” buffer receives a new “copied” ATM cell from the User Cell Filter. The purpose of this interrupt is to notify the Microprocessor that the “Cell Extraction” buffer contains an ATM cell that needs to be read out via the Microprocessor Interface.

Enabling the Transmit Cell Extraction Interrupt

The user can enable the "Transmit Cell Extraction" Interrupt by executing the following steps.

STEP 1 – Write the value "1" into Bit 3 (Transmit ATM Cell Processor Block Interrupt Enable), within the "Operation Block Interrupt Enable Register – Byte 0", as depicted below.

Operation Block Interrupt Enable Register – Byte 0 (Address = 0x0017)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Enable	Receive STS-3 TOH Block Interrupt Enable	Receive SONET/ VC-3 POH Block Interrupt Enable	Receive PPP Processor Block Interrupt Enable	Transmit ATM Cell Processor Block Interrupt Enable	Unused		Transmit PPP Processor Block Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	0	0	1	0	0	0

Note: *This step enables the "Transmit ATM Cell Processor" block for Interrupt Generation, at the "Block Level".*

STEP 2 – Set Bit 5 (Cell Extraction Interrupt Enable), within the "Transmit ATM Cell Processor – Interrupt Enable Register" to "1" as depicted below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

This step enables the "Transmit Cell Extraction" Interrupt, at the "Source-Level". Once the user executes this write operation, then the Transmit ATM Cell Processor block will generate an interrupt to the Microprocessor anytime that the User Cell Filter copies a cell, and routes the "replicate" cell to the "Transmit Cell Extraction Buffer".

Servicing the Transmit Cell Extraction Interrupt

Once the XRT94L33 generates an interrupt, then the user must develop the Interrupt Service routine such that it executes the following steps.

STEP 1 – Read out the contents of both the Operation Block Interrupt Status Register – Byte 1 and Byte 0. The bit-format of these two registers is presented below.

Operation Block Interrupt Status Register – Byte 1 (Address = 0x0012)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Op Control Block Interrupt Status	Unused						
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Operation Block Interrupt Status Register – Byte 0 (Address = 0x0013)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Status	Receive STS-3 TOH Block Interrupt Status	Receive SONET/VC-3 POH Block Interrupt Status	Receive PPP Processor Block Interrupt Status	Transmit ATM Cell Processor Block Interrupt Status	Unused		Transmit PPP Processor Block Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

Note: *If the Transmit ATM Cell Processor block is the source of the interrupt, then Bit 3 (Transmit ATM Cell Processor Block Interrupt Status), within the "Operation Block Interrupt Status Register – Byte 0" will be set to "1", as depicted above.*

STEP 2 – After the Microprocessor has determined that the Transmit ATM Cell Processor block is the "Interrupting" Functional block, within the XRT94L33, it must now identify which of the four channels has generated this interrupt.

The Microprocessor can accomplish this by reading out the contents of the "Operation Channel Interrupt Indicator – Transmit ATM Cell Processor block. The bit-format of the "Operation Channel Interrupt Indicator – Transmit ATM Cell Processor Block" is presented below.

Operation Channel Interrupt Indicator – Transmit ATM Cell Processor Block (Address = 0x0122)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Transmit ATM Cell Processor Block - Channel 3 Interrupt Indication Status	Transmit ATM Cell Processor Block - Channel 2 Interrupt Indication Status	Transmit ATM Cell Processor Block - Channel 1 Interrupt Indication Status	Transmit ATM Cell Processor Block Channel 0 Interrupt Indication Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

This register permits the user to identify which of the four Transmit ATM Cell Processor blocks within the XRT94L33 are generating this particular interrupt request.

STEP 3 – After the Microprocessor has determined which of the four Transmit ATM Cell Processor blocks is the "Interrupting" block within the XRT94L33, then it should read out the contents of the corresponding "Transmit ATM Cell Processor – Interrupt Status Register".

This will permit the Microprocessor to identify the exact cause of the interrupt request, from the Transmit ATM Cell Processor block. The bit-format of "Transmit ATM Cell Processor – Interrupt Status" Register is presented below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

If the cause of this interrupt is the "Transmit Cell Extraction" Interrupt, then Bit 5 (Cell Extraction Interrupt Status) within the "Transmit ATM Cell Processor – Interrupt Status Register" will be set to "1" as depicted above.

Recommended Subsequent Action

Once the Microprocessor Interface has identified this particular interrupt as being the "Transmit Cell Extraction" Interrupt, then the user is advised to read out the contents of the "Transmit Cell Extraction" Buffer. The procedure for reading out the contents of the "Transmit Cell Extraction" Buffer is presented in Section 4.2.2.4.

The Transmit Cell Insertion Interrupt

The Transmit ATM Cell Processor block will generate the "Transmit Cell Insertion" interrupt anytime the "Transmit Cell Insertion" Processor transmits an ATM cell into the "Transmit Data Path" (thereby "freeing up" space for another ATM cell to be written into the "Cell Insertion" Buffer). The purpose of this interrupt is to notify the Microprocessor that the "Transmit Cell Insertion" Buffer contains sufficient available space to accept at least one more ATM cell from the Microprocessor.

Enabling the "Transmit Cell Insertion" Interrupt

The user can enable the "Cell Insertion" Interrupt by executing the following steps.

STEP 1 – Write the value "1" into Bit 3 (Transmit ATM Cell Processor Block Interrupt Enable), within the "Operation Block Interrupt Enable" Register – Byte 0" as depicted below.

Operation Block Interrupt Enable Register – Byte 0 (Address = 0x0017)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Enable	Receive STS-3 TOH Block Interrupt Enable	Receive SONET/ VC-3 POH Block Interrupt Enable	Receive PPP Processor Block Interrupt Enable	Transmit ATM Cell Processor Block Interrupt Enable	Unused		Transmit PPP Processor Block Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	0	0	1	0	0	0

Note: *This step enables the "Transmit ATM Cell Processor" block for Interrupt Generation, at the "Block Level".*

STEP 2 – Set Bit 4 (Cell Insertion Interrupt Enable), within the "Transmit ATM Cell Processor – Interrupt Enable Register" to "1" as depicted below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit Cell Insertion" Interrupt at the "Source-Level". Once the user executes this write operation, then the Transmit ATM Cell Processor block will generate an interrupt to the Microprocessor anytime that the "Transmit Cell Insertion" Processor inserts a cell into the "Transmit Data Path" and "frees up" space for a new ATM cell to be written into the "Cell Insertion" buffer.

This interrupt is most useful, when the Microprocessor is attempting to load a cell into the "Transmit Cell Insertion" buffer, but determines that the "Transmit Cell Insertion" buffer is currently full.

Servicing the Transmit Cell Insertion Interrupt

Once the XRT94L33 generates an interrupt, then the user must develop the Interrupt Service routine such that it executes the following steps.

STEP 1 – Read out the contents of both the Operation Block Interrupt Status Register – Byte 1 and Byte 0. The bit-format of these two registers is presented below.

Operation Block Interrupt Status Register – Byte 1 (Address = 0x0012)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Op Control Block Interrupt Status	Unused						
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Operation Block Interrupt Status Register – Byte 0 (Address = 0x0013)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Status	Receive STS-3 TOH Block Interrupt Status	Receive SONET/VC-3 POH Block Interrupt Status	Receive PPP Processor Block Interrupt Status	Transmit ATM Cell Processor Block Interrupt Status	Unused		Transmit PPP Processor Block Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

Note: *If the Transmit ATM Cell Processor block is the source of the interrupt, then Bit 3 (Transmit ATM Cell Processor Block Interrupt Status), within the "Operation Block Interrupt Status Register – Byte 0" will be set to "1", as depicted above.*

STEP 2 – After the Microprocessor has determined that the Transmit ATM Cell Processor block is the "Interrupting" Functional block, within the XRT94L33, it must now identify which of the four channels has generated this interrupt.

The Microprocessor can accomplish this by reading out the contents of the "Operation Channel Interrupt Indicator – Transmit ATM Cell Processor block. The bit-format of the "Operation Channel Interrupt Indicator – Transmit ATM Cell Processor Block" is presented below.

Operation Channel Interrupt Indicator – Transmit ATM Cell Processor Block (Address = 0x0122)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Transmit ATM Cell Processor Block - Channel 3 Interrupt Indication Status	Transmit ATM Cell Processor Block - Channel 2 Interrupt Indication Status	Transmit ATM Cell Processor Block - Channel 1 Interrupt Indication Status	Transmit ATM Cell Processor Block Channel 0 Interrupt Indication Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

This register permits the user to identify which of the four Transmit ATM Cell Processor blocks within the XRT94L33 are generating this particular interrupt request.

STEP 3 – After the Microprocessor has determined which of the four Transmit ATM Cell Processor blocks is the "Interrupting" block within the XRT94L33, then it should read out the contents of the corresponding "Transmit ATM Cell Processor – Interrupt Status Register".

This will permit the Microprocessor to identify the exact cause of the interrupt request, from the Transmit ATM Cell Processor block. The bit-format of "Transmit ATM Cell Processor – Interrupt Status" Register is presented below.

Transmit ATM Cell Processor – Interrupt Status Register (Address = 0xNF0B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

If the cause of this interrupt is the "Transmit Cell Insertion" Interrupt, then Bit 4 (Cell Insertion Interrupt Status) within the "Transmit ATM Cell Processor – Interrupt Status Register" will be set to "1" as depicted above.

Recommended Subsequent Action

Once the Microprocessor Interface has identified this particular interrupt as being the "Transmit Cell Insertion" Interrupt, then this means that some space within the Transmit Cell Insertion Buffer has been "freed-up". As a consequence, the user can respond to this interrupt by writing in another ATM cell into the Transmit Cell Insertion Buffer. The procedure for writing the contents of an ATM cell into the "Transmit Cell Insertion" Buffer is presented in Section 4.2.2.3.

The Transmit Cell Extraction Memory Overflow Interrupt

The Transmit ATM Cell Processor block will generate the "Transmit Cell Extraction Memory Overflow" Interrupt anytime the Transmit Cell Extraction buffer is currently full, and the Transmit Cell Extraction Processor reads in another "copied" ATM cell into the "Transmit Cell Extraction" buffer. In this case, some of the data residing within the "Transmit Cell Extraction" buffer will be overwritten and will be lost.

Note: *If the "Transmit ATM Cell Processor" block generates the "Transmit Cell Extraction" Memory Overflow Interrupt this is typically the result of the Microprocessor not reading out the contents of the Transmit Cell Extraction Memory quickly or often enough before another ATM cell is "copied" by the Transmit User Cell" Filter. Additionally, this particular interrupt should serve as a warning that the "Transmit Cell Extraction" Buffer likely contains some erred data.*

Enabling the "Transmit Cell Extraction Memory Overflow" Interrupt

The user can enable the "Transmit Cell Extraction Memory Overflow" Interrupt by executing the following steps.

STEP 1 – Write the value "1" into Bit 3 (Transmit ATM Cell Processor Block Interrupt Enable), within the "Operation Block Interrupt Enable" Register – Byte 0" as depicted below.

Operation Block Interrupt Enable Register – Byte 0 (Address = 0x0017)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Enable	Receive STS-3 TOH Block Interrupt Enable	Receive SONET/ VC-3 POH Block Interrupt Enable	Receive PPP Processor Block Interrupt Enable	Transmit ATM Cell Processor Block Interrupt Enable	Unused		Transmit PPP Processor Block Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
0	0	0	0	1	0	0	0

Note: *This step enables the "Transmit ATM Cell Processor" block for Interrupt Generation, at the "Block Level".*

STEP 2 – Set Bit 3 (Cell Extraction Memory Overflow Interrupt Enable), within the "Transmit ATM Cell Processor – Interrupt Enable Register" to "1" as depicted below.

Transmit ATM Cell Processor – Interrupt Enable Register (Address = 0xNF0F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

This step enables the "Transmit Cell Extraction Memory Overflow" Interrupt at the "Source-Level". Once the user executes this write operation, then the Transmit ATM Cell Processor block will generate an interrupt to the Microprocessor anytime the "Transmit Cell Extraction Buffer" experiences an "overflow" condition.

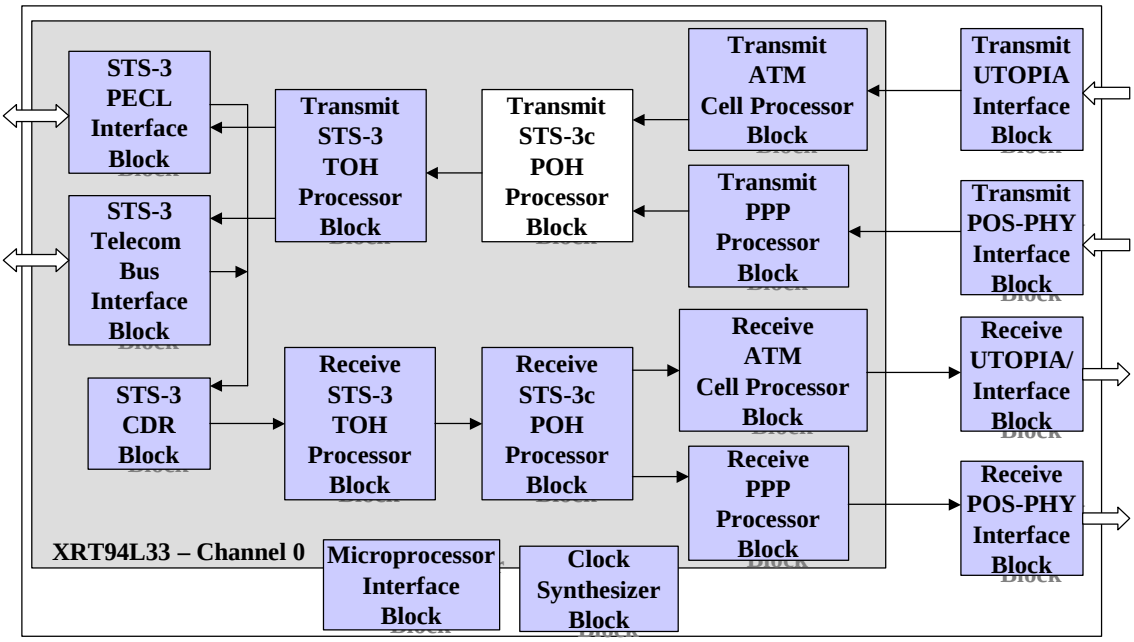
2.2.7 TRANSMIT STS-3C POH PROCESSOR BLOCK (FOR ATM/PPP OVER STS-3C APPLICATIONS)

All outbound ATM cells that exit the Transmit ATM Cell Processor block will be routed to the Transmit STS-3c POH Processor block, where they will be mapped into either STS-3c SPEs. Likewise, all outbound PPP packets that exit the Transmit PPP Packet Processor block will be routed to the Transmit STS-3c POH Processor block, where they will be mapped into STS-3c SPEs. The purpose of the Transmit STS-3c POH Processor block is to accomplish the following.

- To accept ATM cells from the Transmit ATM Cell Processor block and to map these cells into STS-3c SPE (Synchronous Payload Envelope), or
- To accept PPP packets from the Transmit PPP Packet Processor block and to map these packets into STS-3c SPEs
- To compute and insert the B3 (Path BIP-8) byte
- To deliberately transmit an STS-3c SPE with an erred B3 byte (for testing purposes)
- To source the J1, C2, G1, F2, H4, Z3, Z4 and Z5 bytes.
- To automatically transmit the RDI-P (Path – Remote Defect) Indicator whenever (and for the duration that) the corresponding Receive STS-3c POH Processor block declares the AIS-P, LOP-P, UNEQ-P, TIM-P or PLM-P defect conditions.
- To automatically transmit the RDI-P indicator whenever (and for the duration that) the corresponding Receive ATM Cell Processor block declares the LCD defect condition.
- To transmit the RDI-P Indicator upon Software Control.
- To automatically transmit the REI-P (Path – Remote Error) Indicator anytime the corresponding Receive STS-3c POH Processor block detects B3 byte errors in its incoming STS-3c SPE data-stream.
- To transmit the REI-P indicator upon Software Control.
- To transmit the AIS-P (Path – Alarm Indication Signal) Indicator upon Software Control.
- To permit the user to transmit either 1 byte, 16 byte or 64 byte Path Trace Messages to the remote PTE.
- To force “positive-stuff” pointer adjustments in the outbound STS-3c data-stream.
- To force “negative-stuff” pointer adjustments in the outbound STS-3c data-stream.
- To force single and continuous NDF (New Data Flag) events into the outbound STS-3c data-stream
- To route its output data to the Transmit STS-3c TOH Processor block for further processing.

Figure 33 presents an illustration of the block diagram of the XRT94L33 Mapper IC, with the “Transmit STS-3c POH Processor” block highlighted.

Figure 33: Illustration of the Functional Block Diagram of the XRT94L33 Mapper IC, with the Transmit STS-3c POH Processor block high-lighted



SOME BACKGROUND INFORMATION ON THE PATH OVERHEAD (POH) BYTES

For STS-3c applications, the size of the SPE is 9 rows by 261 byte columns. Figure 34 presents a simple illustration of the STS-3c SPE.

Figure 34 Simple Illustration of the STS-3c SPE

In each case, the first byte column within an STS-3c SPE is referred to as the Path Overhead (POH). Therefore, in all, the POH consists of a total of nine (9) bytes. Each of these POH bytes is briefly described below.

J1 – The Path Trace Byte

B3 – The Path BIP-8 Byte

C2 – The Payload Label Byte

G1 – The Path Status Byte

F2 – The Path User Channel Byte

H4 – The Multi-Frame Indicator (not used for ATM and PPP applications)

Z3, Z4 and Z5 – Growth Bytes

The remainder of this section describes how the Transmit STS-3c POH Processor block handles these POH bytes.

The operation of the “Transmit STS-3c POH Processor” block is discussed in some detail below.

2.2.7.1 RECEIVING ATM CELL DATA FROM THE TRANSMIT ATM CELL PROCESSOR BLOCK

The Transmit STS-3c POH Processor block receives ATM cell data from the Transmit ATM Cell Processor block; and maps this data into an STS-3c SPE.

As the Transmit STS-3c POH Processor block receives this ATM Cell data from the Transmit ATM Cell Processor block, it will perform the following functions.

- It will map these ATM cells into the payload bytes, within the STS-3c SPE.
- Compute and insert the B3 byte
- Source (per user configuration) the J1, C2, G1, F2, H4, Z3, Z4 and Z5 bytes
- To (automatically or upon software command) transmit the RDI-P (Path – Remote Defect Indicator) indicator
- To (automatically or upon software command) transmit the REI-P (Path – Remote Error Indicator) indicator
- To (upon software command) transmit the AIS-P (Path – Alarm Indication Signal) indicator
- To (upon software command) force pointer-adjustment or NDF (New Data Flag) events into the outbound STS-3c data-stream.

Each of these functions is described in detail below.

2.2.7.2 COMPUTATION AND INSERTION OF THE PATH BIP-8 (B3) BYTE

The Transmit STS-3c POH Processor block creates STS-3c SPE data and computes a BIP-8 value over this STS-3c SPE. The results of this calculation are inserted into the B3 byte-position within the very next STS-3c SPE. The Remote PTE (Path Terminating Equipment) will use this byte, in order to perform error-checking/detection on the incoming STS-3c SPE data that it receives.

TRANSMISSION OF ERRED B3 BYTES IN THE OUTBOUND STS-3c DATA-STREAM

The Transmit STS-3c POH Processor block permits the user to insert errors into the "B3 Bytes", within the outbound STS-3c SPE data-stream.

The user can accomplish this by writing a non-zero value into the "Transmit STS-3c Path – Transmitter B3 Byte Error Mask" Register. The "Transmit STS-3c POH Processor block will perform an XOR operation with the contents of the "outbound" B3 byte, and the contents of this register. The results of this calculation are written back into the B3 byte position, within the outbound STS-3c SPE data-stream.

The bit-format of the "Transmit STS-3c Path – Transmit B3 Byte Error Mask" register is presented below.

Transmit STS-3c Path – Transmitter B3 Byte Error Mask Register (Address = 0x1997)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_B3_Byte_Mask[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *For normal (e.g., un-erred) operation, the user should ensure that this register is set to "0x00" (the default value).*

2.2.7.3 TRANSMISSION OF PATH-ALARM CONDITIONS

The Transmit STS-3c POH Processor block supports the following functions.

- Transmitting the AIS-P Pattern (under Software control)
- Transmitting the RDI-P Indicator (automatically and under Software control)
- Transmitting the REI-P Indicator (automatically and under Software control)

Each of these operations is discussed in detail below.

2.2.7.3.1 TRANSMISSION OF THE AIS-P PATTERN

The Transmit STS-3c POH Processor block can be configured to generate and transmit the AIS-P (Path AIS) indicator to the remote PTE, under software control.

The user can accomplish this by setting Bit 0 (Transmit AIS-P Enable) within the "Transmit STS-3c Path – SONET Control Register – Byte 0" to "1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	1

Once the user executes this step, then the "Transmit STS-3c POH Processor" block will do the following.

- It will set all bytes, within the STS-3c SPE to an "All Ones" pattern.
- It will set the H1, H2 and H3 bytes (within the TOH) to an "All Ones" pattern.

Please note that whenever the user configures the Transmit STS-3c POH Processor block to transmit the AIS-P indicator, then no ATM cells or PPP packets will be transmitted to the remote terminal.

2.2.7.3.2 TRANSMISSION OF THE RDI-P INDICATOR

The Transmit STS-3c POH Processor block can be configured to transmit the RDI-P indicator either automatically, upon software control or via the "TxPOH_n" external input port as described below.

2.2.7.3.2.1 Configuring the Transmit STS-3c POH Processor block to automatically transmit the RDI-P Indicator

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator whenever the corresponding Receive STS-3c POH Processor block declares a PLM-P (Path – Payload Label Mismatch), UNEQ-P (Path – Unequipped), LOP-P (Path – Loss of Pointer), AIS-P, TIM-P (Path - Trace Identification Mismatch), and LCD-P (Path – Loss of Cell Delineation) condition.

The procedure for configuring the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, for each of the above-mentioned defects is presented below.

2.2.7.3.2.2 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the PLM-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor block declaring the PLM-P condition, by executing the following steps.

STEP 1 – Write the value [0, 0] into Bits 3 and 4 (RDI-P Insertion Type[1, 0]) within the "Transmit STS-3c Path – SONET Control Register – Byte 0" as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the "Transmit STS-3c POH Processor" block to automatically set bits 5 through 7 (of the G1 byte) within the outbound STS-3c SPE; to the appropriate "RDI-P value" based upon receive conditions as detected by the corresponding "Receive STS-3c POH Processor" block.

STEP 2 – Write the appropriate value into Bits 3 through 1 (PLM-P RDI-P CODE[2:0]) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 2" as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 2 (Address = 0x19C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the value that the "Transmit STS-3c POH Processor" block will set the RDI-P bit-fields (within the "outbound G1 byte) to, whenever the corresponding Receive STS-3c POH Processor block declares the PLM-P condition.

STEP 3 – Set Bit 0 (Transmit RDI-P upon PLM-P) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 2" to "1", as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 2 (Address = 0x19C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	1

This step configures the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1; within this register), anytime the corresponding Receive STS-3c POH Processor block detects the "PLM-P" condition.

2.2.7.3.2.3 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the UNEQ-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor block declaring the UNEQ-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 3 through 1 (UNEQ-P RDI-P Code[2:0]) within the "Transmit STS-3c Path – RDI-P Condition Register – Byte 1, as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 1 (Address = 0x19CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

By writing this particular value into these bit-fields the user is specifying the value that the "Transmit STS-3c POH Processor" block will set the RDI-P bit-fields (within the G1 byte of outbound STS-3c frame) to whenever the corresponding Receive STS-3c POH Processor block declares the UNEQ-P condition.

STEP 2 – Set Bit 0 (Transmit RDI-P upon UNEQ-P) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 1" to "1", as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 1 (Address = 0x19CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	1

This step configures the Transmit STS-3c POH Processor block to transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive STS-3c POH Processor" block declares the "UNEQ-P" condition.

2.2.7.3.2.4 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the LOP-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor block declaring the LOP-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 7 through 5 (LOP-P RDI-P Code[2:0]) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 0"; as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 0 (Address = 0x19CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	0	0	0	0

By writing this particular value into these three bit-fields, the user is specifying the value that the Transmit STS-3c POH Processor block will set the RDI-P bit-fields (in the G1 byte, within the "outbound" STS-3c data-stream) whenever the corresponding Receive STS-3c POH Processor block declares the LOP-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon LOP-P) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 0" to "1"; as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 0 (Address = 0x19CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive STS-3c POH Processor" block declares the "LOP-P" condition.

2.2.7.3.2.5 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the AIS-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor declaring the AIS-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 3 through 1 (AIS-P RDI-P Code[2:0]) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 0"; as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 0 (Address = 0x19CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the value that the Transmit STS-3c POH Processor block will set the RDI-P bit-fields (in the G1 byte, within the "outbound" STS-3c data-stream) whenever the corresponding Receive STS-3c POH Processor block declares the AIS-P condition.

STEP 2 – Set Bit 0 (Transmit RDI-P upon AIS-P) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 0", as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 0 (Address = 0x19CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	1

This step configures the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive STS-3c POH Processor" block declares the "AIS-P" condition.

2.2.7.3.2.6 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the TIM-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor block declaring the TIM-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 7 through 5 (TIM-P RDI-P Code[2:0]) within the "Transmit STS-3c Path – RDI-P Condition Register – Byte 1; as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 1 (Address = 0x19CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	X

By writing this particular value into these bit-fields the user is specifying the value that the "Transmit STS-3c POH Processor" block will set the RDI-P bit-fields (within the G1 byte of the outbound STS-3c frame) to whenever the corresponding Receive STS-3c POH Processor block declares the TIM-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon TIM-P) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 1"; to "1" as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 1 (Address = 0x19CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	X

This step configures the Transmit STS-3c POH Processor block to transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive STS-3c POH Processor" block declares the "TIM-P" condition.

2.2.7.3.2.7 Configuring the Transmit STS-3c POH Processor block to automatically transmit RDI-P, in response to declaration of the LCD-P Condition

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive STS-3c POH Processor block declaring the LCD-P condition, by executing the following steps.

STEP 1- Write the appropriate value into Bit 7 through 5 (LCD-P RDI-P Code[2:0]) within the "Transmit STS-3c Path – RDI-P Control Register – Byte 2" as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 2 (Address = 0x19C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the values that the "Transmit STS-3c POH Processor" block will set the RDI-P bit-fields (within the outbound G1 byte) to, whenever the corresponding Receive STS-3c POH Processor block declares the LCD-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon LCD-P) within the "Transmit STS-3c Path – RDI-P Control Registers – Byte 2" to "1", as illustrated below.

Transmit STS-3c Path – RDI-P Control Register – Byte 2 (Address = 0x19C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	0

2.2.7.3.2.8 Configuring the Transmit STS-3c POH Processor block to transmit the RDI-P indicator, upon Software Control

The user can configure the Transmit STS-3c POH Processor block to transmit the RDI-P (per software command) by executing the following steps.

STEP 1 – Write the value "[0, 1]" into Bits 3 and 4 (RDI-P Insertion Type[1:0]) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	1	0	0	0

This step configures the "Transmit STS-3c POH Processor" block to automatically read out the contents of bits 3 through 1 (of the "Transmit STS-3c Path – G1 Byte Value" register); and write the value of these bits into bits 5 through 7 (of the G1 byte) within the "outbound" STS-3c SPE. The bit-format of the "Transmit STS-3c Path – G1 Byte Value" register (with the appropriate bits "shaded") is presented below.

Transmit STS-3c Path – Transmit G1 Byte Value Register (Address = 0x199F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_G1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

In this mode, the user can transmit an "un-erred" indicator by setting Bits 3 through 1 (within the Transmit STS-3c Path – Transmit G1 Byte Value" register) to [0, 0, 0]. Conversely, the user can now transmit an "erred" indicator by setting Bits 3 through 1 (within the "Transmit STS-3c Path – Transmit G1 Byte Value" register) to some value other than [0, 0, 0].

2.2.7.3.2.9 Configuring the Transmit STS-3c POH Processor block to transmit the RDI-P Indicator, via the "TxPOH_n" input port.

The user can configure the Transmit STS-3c POH Processor block to transmit the RDI-P (per external input port) by executing the following steps.

STEP 1 – Write the value "[1, 0]" into Bits 3 and 4 (RDI-P Insertion Type[1:0]) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	1	0	0	0	0

This step configures the "Transmit STS-3c POH Processor" block to set the value of the RDI-P bit-fields (within the outbound STS-3c SPE) based upon the data that it receives via the "TxPOH_n" input port. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the RDI-P bit-fields (via the "TxPOH_n Input Port") and it will write this data into the "outbound" STS-3c SPE data-stream.

STEP 2 – Begin providing the values of the "outbound" RDI-P bit-fields to the "TxPOH_n" input port.

The procedure for applying the RDI-P bit-values to the "TxPOH_n" input port is presented below.

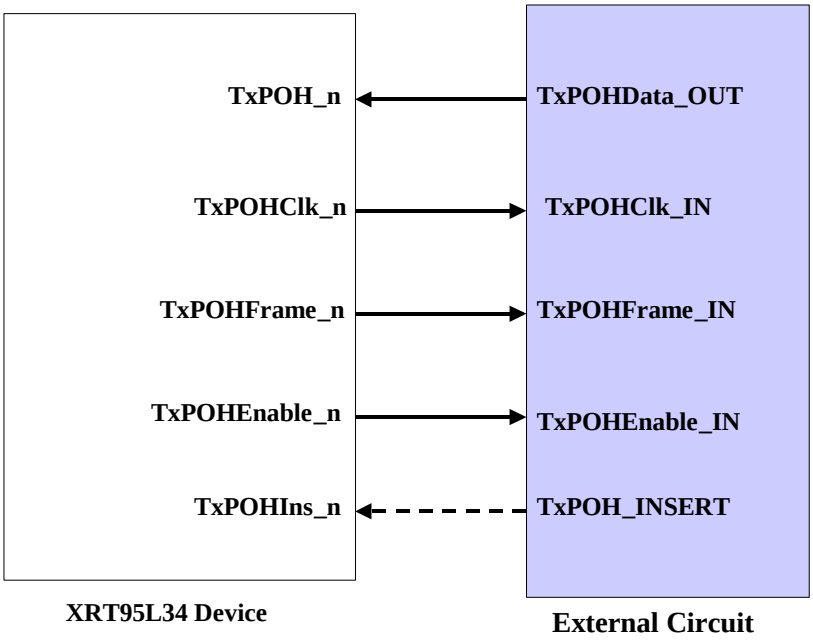
Using the "TxPOH" Input Port to insert the RDI-P bit values into the outbound STS-3c SPE data-stream

If the user intends to externally insert the RDI-P bits into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 35.

Figure 35: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (inFigure 35) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it waits for 25 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit of the "outbound" RDI-P bit-fields onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the G1 byte is the 4th byte within the POH; and bit 7 (which is the very first RDI-P bit to be latched into the External Input Interface is the second bit within the G1 byte to be processed).*

- Afterwards, the "external circuit" should serially place the remaining two bits (of the RDI-P bits) onto the "TxPOH_n" input pin, upon each of the next two falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.3 TRANSMISSION OF THE REI-P INDICATOR

The Transmit STS-3c POH Processor block can be configured to transmit the REI-P indicator either (1) automatically, (2) upon software command or (3) via the "TxPOH_n" input port as described below.

2.2.7.3.3.1 Configuring the Transmit STS-3c POH Processor block to automatically transmit the REI-P indicator

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the REI-P indicator whenever the corresponding Receive STS-3c POH Processor block detects at least one B3 byte error within its incoming STS-3c SPE data-stream.

The user can configure the Transmit STS-3c POH Processor block to automatically transmit the REI-P indicator, in response to detection of B3 byte errors, by executing the following steps.

STEP 1 – Write the value [0, 0] into Bits 5 and 6 (REI-P Insertion Type[1, 0]) within the "Transmit STS-3c Path – SONET Control Register – Byte 0" as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the "Transmit STS-3c POH Processor" block to automatically set bits 1 through 4 (of the G1 byte) within the "outbound" STS-3c SPE to the appropriate REI-P value, based upon receive conditions as detected by the corresponding Receive STS-3c POH Processor block.

STEP 2 – Indicate whether the REI-P value (transmitted to the remote PTE) reflects the number of bits (within the B3 byte) that are in error, or the number of erred STS-3c SPE that have been detected by the corresponding Receive STS-3c POH Processor block.

The XRT94L33 permits the user to (1) flag B3 byte errors, and (2) to transmit the resulting REI-P value (back out to the remote PTE) by the following means.

- By flagging and reporting the number of bits (within the B3 byte) that have be determined to be in error, within a given STS-3c SPE.
- By flagging and reporting whether or not the corresponding Receive STS-3c POH Processor block is currently receiving erred STS-3c SPE.

The user can choose between these two options by writing the appropriate value into Bit 0 (B3 Error Type) within the Receive STS-3c Path – Control Register – Byte 0"; as illustrated below.

Receive STS-3c Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	X

Setting this bit-field to "0" configures the Transmit STS-3c POH Processor block to set the REI-P bit-fields (within the G1 byte of the "outbound" STS-3c SPE); to a value the reflects the number of bits (within the B3 byte) that were determined to be in error (within the most recently received STS-3c SPE) by the corresponding Receive STS-3c POH Processor block. In this case, the REI-P bit-fields can range in value from "0" (for no B3 bit errors) to "8" (for all B3 bits being in error).

Setting this bit-field to "1" configures the Transmit STS-3c POH Processor block to set the REI-P bit-fields (within the G1 byte of the 'outbound"STS-3c SPE); to a value that indicates whether or not at least one B3 byte error was detected within the most recently received STS-3c SPE; by the corresponding Receive STS-3c POH Processor block. In this mode, the Transmit STS-3c POH Processor block will set the "REI-P" bit-fields (within the outbound STS-3c SPE) to "0" if the corresponding "Receive STS-3c POH Processor" block is not detecting any B3 byte errors, in its incoming STS-3c data-stream. Conversely, the Transmit STS-3c POH Processor block will set the "REI-P" bit-fields (within the "outbound" STS-3c SPE) to "1" if the corresponding Receive STS-3c POH Processor block receives an STS-3c SPE that contains a B3 byte error.

2.2.7.3.3.2 Configuring the Transmit STS-3c POH Processor block to transmit the REI-P Indicator, upon software control

The user can configure the Transmit STS-3c POH Processor block to transmit the REI-P indicator (per software command) by executing the following steps.

STEP 1- Write the value "[0, 1]" into Bits 5 and 6 (REI-P Insertion Type[1:0]) within the "Transmit STS-3c Path SONET Control Register – Byte 0"; as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

This step configures the "Transmit STS-3c POH Processor" block to automatically read out the contents of Bits 4 through 7 (of the Transmit STS-3c Path – G1 Byte Value" register) and write the value of these bits into bits 1 through 4 (of the G1 byte) within the "outbound" STS-3c SPE. The bit-format of the "Transmit STS-3c Path – G1 Byte Value" register (with the appropriate bits "shaded") is presented below.

Transmit STS-3c Path – Transmit G1 Byte Value Register (Address = 0x199F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_G1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

In this mode, the user can transmit an "un-erred" REI-P value by setting Bits 4 through 7 (within the Transmit STS-3c Path – Transmit G1 Byte Value" register) to [0, 0, 0, 0]. Conversely, the user can transmit an "erred" REI-P value by setting Bits 4 through 7 (within the Transmit STS-3c Path – Transmit G1 Byte Value" register) to some value between 1 and 8.

2.2.7.3.3.3 Configuring the Transmit STS-3c POH Processor block to transmit the REI-P indicator, via the "TxPOH_n" input port

The user can configure the Transmit STS-3c POH Processor block to transmit the REI-P (per the external input port) by executing the following steps.

STEP 1 – Write the value [1, 0] into Bits 3 and 4 (REI-P Insertion Type[1:0]) within the "Transmit STS-3c Path – SONET Control Register – Byte 0" as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to set the value of the REI-P bit-fields (within the outbound STS-3c SPE) based upon the data that it receives via the "TxPOH_n" input port. In this mode, the Transmit STS-3c POH Processor block will accept the value corresponding to the REI-P fields (via the "TxPOH_n Input Port") and it will write this data into the "outbound" STS-3c SPE data-stream.

STEP 2 – Begin providing the values of the "outbound" REI-P bit-fields to the "TxPOH_n" input port.

The procedure for applying the REI-P bit-values to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the REI-P bit values into the outbound STS-3c SPE data-stream

If the user intends to externally insert the REI-P bits into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHEnable_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.
- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it waits for 28 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit of the "outbound" REI-P bit-fields onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n"

2.2.7.3.4 TRANSMISSION OF PATH TRACE MESSAGES VIA THE J1 BYTE

The Transmit STS-3c POH Processor block permits the user to transmit Path Trace Messages to the remote PTE via the J1 byte. The Transmit STS-3c POH Processor block permits the user to accomplish this by either of the following options.

- Automatically set the J1 byte (within each outbound STS-3c SPE) to "0x00"
- Set and control the outbound J1 byte via on-chip register
- Set and control the outbound J1 byte via external input pin
- Use the "Transmit Path Trace Message" buffer

The details and instructions for using either of these features are presented below.

2.2.7.3.4.1 Automatically setting the J1 byte (within each outbound STS-3c SPE) to "0x00"

The XRT94L33 permits the user to configure each of the four (4) Transmit STS-3c POH Processor blocks to automatically set the contents of the J1 byte (within each outbound STS-3c SPE) to "0x00".

The user can accomplish this by writing the value "[0, 0]" into Bits 1 and 0 (J1 Type[1:0]) within the appropriate "Transmit STS-3c Path – Transmit J1 Control" Register; as depicted below.

Transmit STS-3c Path – Transmit J1 Control" Register (Address = 0x19BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	0

Once the user executes this step, then the corresponding Transmit STS-3c POH Processor block(s) will be configured to automatically set the J1 byte (within each outbound STS-3c SPE) to the value "0x00".

2.2.7.3.4.2 Using the "Transmit Path Trace Message" Buffer

The XRT94L33 contains a total of four (4) "Transmit Path Trace Message Buffers (one for each "Transmit STS-3c POH Processor Block, within the device).

The address location of the "Transmit Path Trace Message" buffer, for each of the four Transmit STS-3c POH Processor blocks is presented below.

Table 14: Address Locations of the Transmit Path Trace Message Buffers within the XRT94L33

TRANSMIT STS-3C POH PROCESSOR BLOCK - CHANNEL	ADDRESS LOCATION OF THE TRANSMIT PATH TRACE MESSAGE BUFFER
0	0x1D00 – 0x1D3F
1	0x2D00 – 0x2D3F
2	0x3D00 – 0x3D3F
3	0x4D00 – 0x4D3F

If the user wishes to use the "Path Trace Message" buffer as the means to load and transmit the "Path Trace Message" to the remote PTE, the following steps must be executed.

STEP 1 – Write the contents of this outbound "Path Trace Message" into the "Transmit Path Trace Message" buffer.

As the user writes the contents of the "outbound" Path Trace Message" into the "Transmit Path Trace" Message buffer they must make sure that the first byte of the message is written into Address location 0xNC00 (e.g., the very first address location of the "Transmit Path Trace Message" buffer).

STEP 2 – Specify the length of this "Path Trace Message" by writing the appropriate value into Bits 3 and 2 (J1 Message Length[1:0]) within the "Transmit STS-3c Path – Transmit J1 Control" Register; as depicted below.

Transmit STS-3c Path – Transmit J1 Control" Register (Address = 0x19BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	X

The relationship between the contents of "J1_Message_Length[1:0]" and the corresponding length of the "Path Trace Message" is presented below in Table _.

Table _, The Relationship Between the contents of "J1_Message_Length[1:0]" and the corresponding "Path Trace Message" Length

J1_MESSAGE_LENGTH[1:0]	PATH TRACE MESSAGE LENGTH (BYTES)
00	1
01	16
10	64
11	64

STEP 3 – Write the value "[0, 1]" into Bits 1 and 0 (J1_Type[1:0]) within the Transmit STS-3c Path – Transmit J1 Control Register; as depicted below.

Transmit STS-3c Path – Transmit J1 Control" Register (Address = 0x19BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	1

This step configures the Transmit STS-3c POH Processor block to use the contents of the "Transmit Path Trace Message" buffer as the source of the J1 byte, within each "outbound" STS-3c SPE.

2.2.7.3.4.3 Setting and Controlling the "Outbound" J1 Byte via On-Chip Register

The Transmit STS-3c POH Processor block permits the user to specify the contents of the J1 byte within the "outbound" STS-3c SPE, via software command. The user can configure the Transmit STS-3c POH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 0]" into Bits 1 and 0 (J1 Type[1:0]) within the "Transmit STS-3c Path – Transmit J1 Control" Register, as depicted below.

Transmit STS-3c Path – Transmit J1 Control" Register (Address = 0x19BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	1	0

This step configures the "Transmit STS-3c POH Processor" block to read out the contents of the "Transmit Path – Transmit J1 Byte Value" register; and load this value into the "J1 byte-field" within each outbound STS-3c SPE.

STEP 2 – Write the desired byte value (for the "outbound" J1 byte) into the "Transmit STS-3c Path – Transmit J1 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmitter J1 Value Register (Address = 0x1993)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_J1_Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

2.2.7.3.4.4 Setting and Controlling the "Outbound" J1 Byte via External Input pin

The Transmit STS-3c POH Processor block permits the user to source the contents of the J1 byte via the "TxPOH_n" input port. The user can configure the Transmit STS-3c POH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 1]" into Bits 1 and 0 (J1 Type[1:0]) within the "Transmit STS-3c Path – Transmit J1 Control" Register; as depicted below.

Transmit STS-3c Path – Transmit J1 Control" Register (Address = 0x19BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

This step configures the "Transmit STS-3c POH Processor" block to accept the value of the J1 byte, via the "TxPOH_n" input port and load this value into the J1 byte position within each outbound STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" J1 byte message to the "TxPOH_n" input port.

The procedure for applying the J1 byte to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the J1 byte value into the outbound STS-3c SPE data-stream

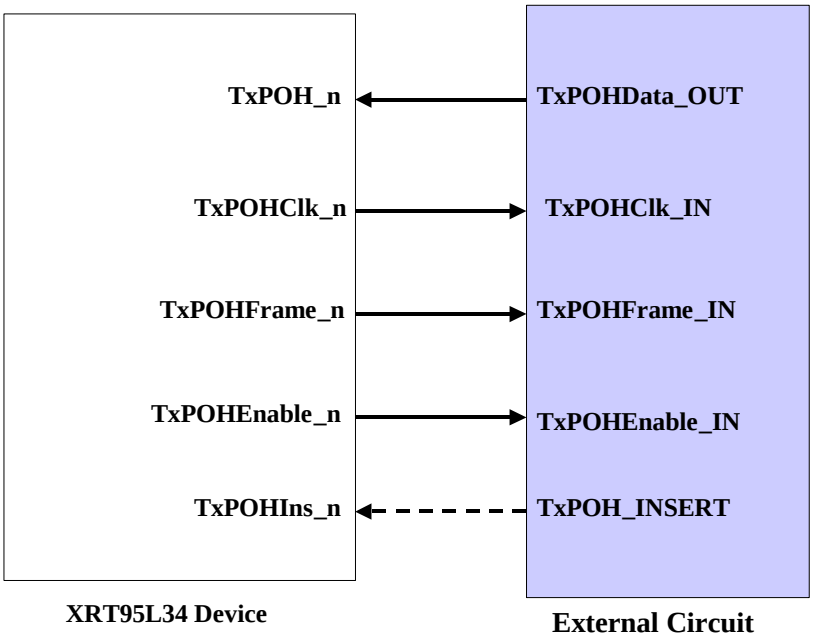
If the user intends to externally insert the J1 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in

Figure 36.

Figure 36: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in*

Figure 36) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should place the very first bit (e.g., the most significant bit) of the "outbound" J1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the J1 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.5 SUPPORT/HANDLING OF THE C2 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the C2 byte by either of the following options.

- Setting and controlling the "outbound" C2 Byte via Software
- Setting and controlling the "outbound" C2 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.5.1 Setting and Controlling the Outbound C2 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the C2 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 2 (C2 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit C2 Byte Value" register; and load this value into the C2 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound C2 byte) into the "Transmit STS-3c Path – Transmit C2 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit C2 Byte Value Register (Address = 0x199F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_C2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.5.2 Setting and Controlling the Outbound C2 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the C2 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 2 (C2 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	1	0	0

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the C2 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the C2 byte (via the "TxPOH_n" input port) and it will write this data into the C2 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" C2 byte to the "TxPOH_n" input port.

The procedure for applying the C2 byte to the "TxPOH_n" input port is presented below.

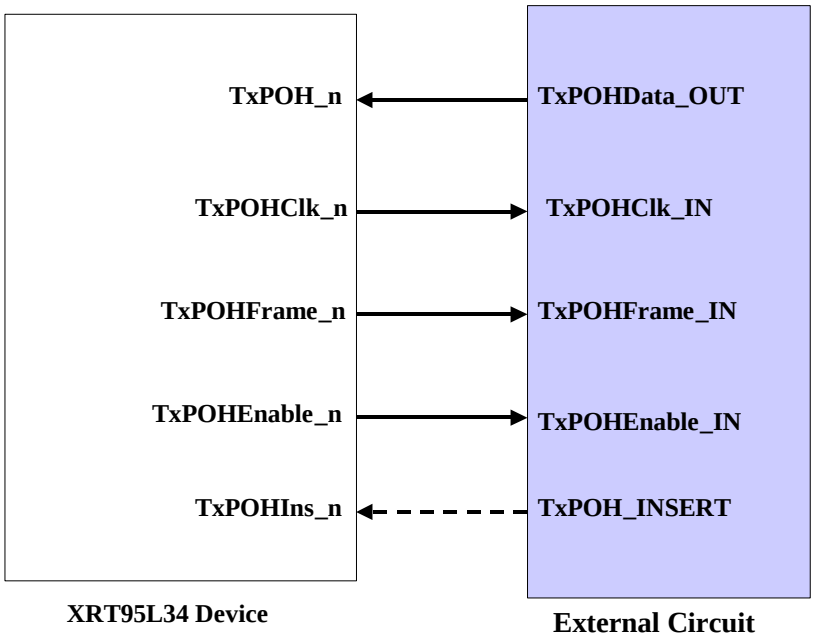
Using the "TxPOH" Input Port to insert the C2 byte value into the outbound STS-3c SPE data-stream

If the user intends to externally insert the C2 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 37.

Figure 37: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (inFigure 37) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 16 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" C2 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the C2 byte is the third byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the C2 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.6 SUPPORT/HANDLING OF THE F2 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the F2 byte by either of the following options.

- Setting and controlling the "outbound" F2 Byte via Software
- Setting and controlling the "outbound" F2 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.6.1 Setting and Controlling the Outbound F2 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the F2 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 7 (F2 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit F2 Byte Value" register; and load this value into the F2 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound F2 byte) into the "Transmit STS-3c Path – Transmit F2 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit F2 Byte Value Register (Address = 0x19A3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_F2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.6.2 Setting and Controlling the Outbound F2 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the F2 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 7 (F2 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 0", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 0 (Address = 0x1983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
1	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the F2 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the F2 byte (via the "TxPOH_n" input port) and it will write this data into the F2 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" F2 byte to the "TxPOH_n" input port.

The procedure for applying the F2 byte to the "TxPOH_n" input port is presented below.

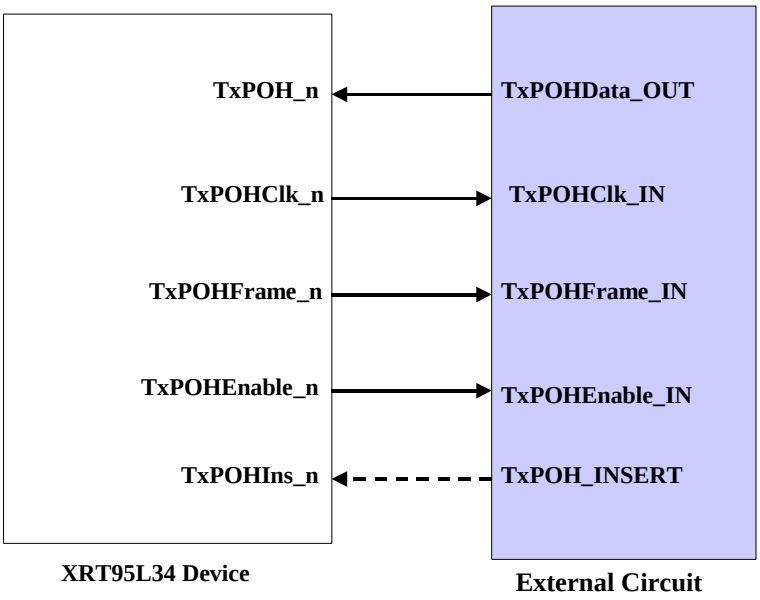
Using the "TxPOH" Input Port to insert the F2 byte value into the outbound STS-3c SPE data-stream

If the user intends to externally insert the F2 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 38.

Figure 38: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 38) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 32 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g.,

the most significant bit) of the "outbound" F2 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the F2 byte is the fifth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the F2 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 39 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the F2 byte into the "TxPOH Input Port".

Figure 39 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "F2 byte" into the "TxPOH Input Port".

2.2.7.3.7 SUPPORT/HANDLING OF THE H4 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" H4 Byte via Software
- Setting and controlling the "outbound" H4 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.7.1 Setting and Controlling the Outbound H4 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the H4 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (H4 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit H4 Byte Value" register; and load this value into the C2 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound H4 byte) into the "Transmit STS-3c Path – Transmit H4 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit H4 Byte Value Register (Address = 0x19A7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_H4_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.7.2 Setting and Controlling the Outbound H4 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the H4 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (F2 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the H4 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the H4 byte (via the "TxPOH_n" input port) and it will write this data into the H4 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" H4 byte to the "TxPOH_n" input port.

The procedure for applying the H4 byte to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the H4 byte value into the outbound STS-3c SPE data-stream

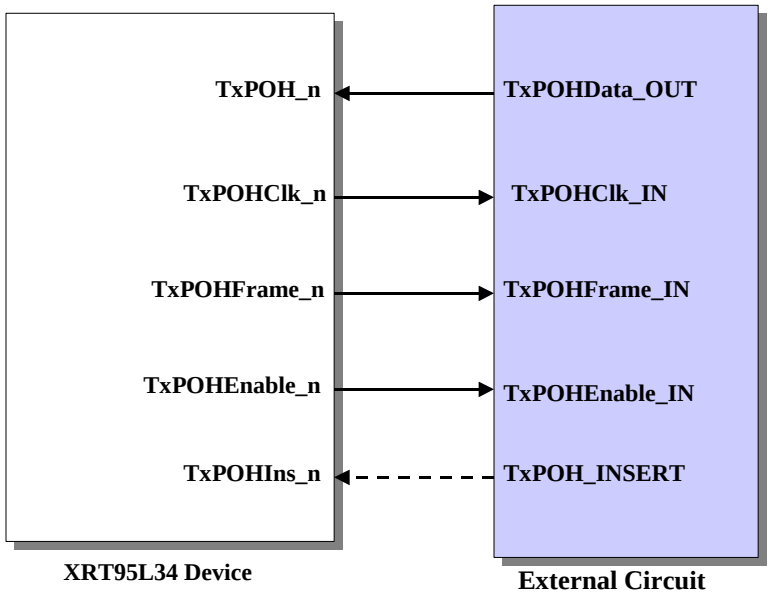
If the user intends to externally insert the H4 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in

Figure 40.

Figure 40: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in*

Figure 40) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 40 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" H4 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the H4 byte is the sixth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the H4 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 41 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the H4 byte into the "TxPOH Input Port".

Figure 41 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "H4 byte" into the "TxPOH Input Port".

2.2.7.3.8 SUPPORT/HANDLING OF THE Z3 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" Z3 Byte via Software
- Setting and controlling the "outbound" Z3 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.8.1 Setting and Controlling the Outbound Z3 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z3 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 1 (Z3 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit Z3 Byte Value" register; and load this value into the Z3 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound Z3 byte) into the "Transmit STS-3c Path – Transmit Z3 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit Z3 Byte Value Register (Address = 0x19AB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z3_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.8.2 Setting and Controlling the Outbound Z3 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z3 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 1 (Z3 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the Z3 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the Z3 byte (via the "TxPOH_n" input port) and it will write this data into the Z3 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" Z3 byte to the "TxPOH_n" input port.

The procedure for applying the Z3 byte to the "TxPOH_n" input port is presented below.

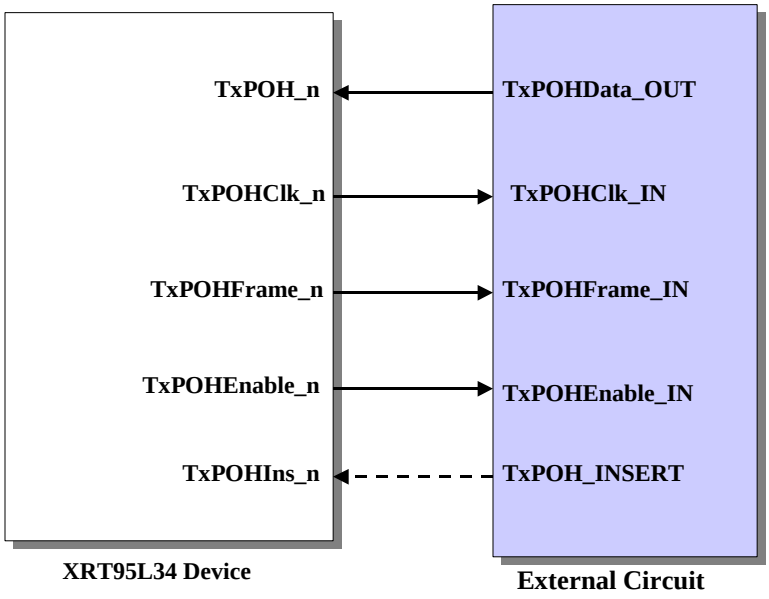
Using the "TxPOH" Input Port to insert the Z3 byte value into the outbound STS-3c SPE data-stream

If the user intends to externally insert the Z3 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 42.

Figure 42: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 42) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 48 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z3 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z3 byte is the seventh byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the Z3 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".
- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.9 SUPPORT/HANDLING OF THE Z4 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the Z4 byte by either of the following options.

- Setting and controlling the "outbound" Z4 Byte via Software
- Setting and controlling the "outbound" Z4 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.9.1 Setting and Controlling the Outbound Z4 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z4 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (Z4 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit Z4 Byte Value" register; and load this value into the Z4 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound H4 byte) into the "Transmit STS-3c Path – Transmit Z4 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit Z4 Byte Value Register (Address = 0x19AF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z4_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.9.2 Setting and Controlling the Outbound Z4 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z4 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (Z4 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	0

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the Z4 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the Z4 byte (via the "TxPOH_n" input port) and it will write this data into the Z4 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" Z4 byte to the "TxPOH_n" input port.

The procedure for applying the Z4 byte to the "TxPOH_n" input port is presented below.

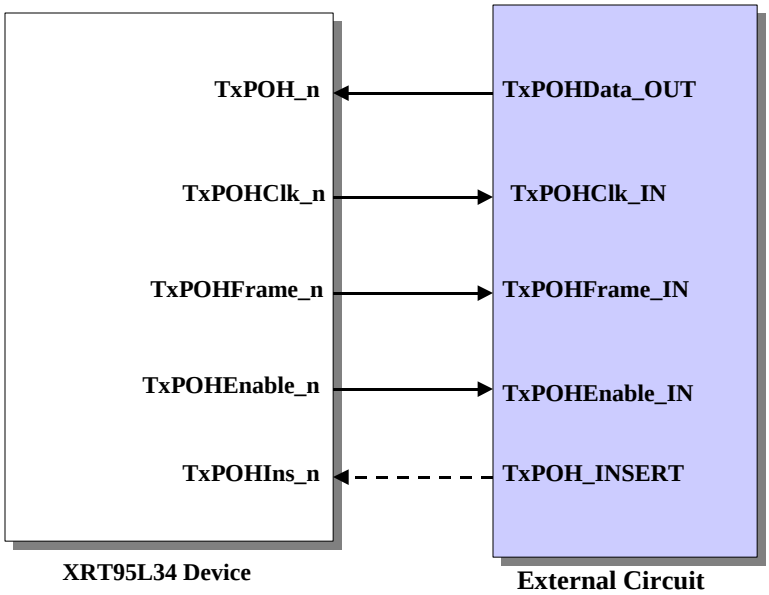
Using the "TxPOH" Input Port to insert the Z4 byte value into the outbound STS-3c SPE data-stream

If the user intends to externally insert the Z4 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 43.

Figure 43: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 43) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for _ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z4 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z4 byte is the 8th byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the H4 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.10 SUPPORT/HANDLING OF THE Z5 BYTE

The Transmit STS-3c POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" Z5 Byte via Software
- Setting and controlling the "outbound" Z5 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.7.3.10.1 Setting and Controlling the Outbound Z5 Byte via Software

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z5 byte, within the "outbound" STS-3c SPE via software command.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (Z5 Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	X

This step configures the Transmit STS-3c POH Processor block to read out the contents of the "Transmit STS-3c Path – Transmit Z5 Byte Value" register; and load this value into the Z5 byte position within each "outbound" STS-3c SPE.

STEP 2 – Write the desired byte value (for the outbound Z5 byte) into the "Transmit STS-3c Path – Transmit Z5 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3c Path – Transmit Z5 Byte Value Register (Address = 0x19B3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z5_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.7.3.10.2 Setting and Controlling the Outbound Z5 Byte via the "TxPOH_n Input Port"

The Transmit STS-3c POH Processor block permits the user to specify the contents of the Z5 byte, within the "outbound" STS-3c SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit STS-3c POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (Z5 Byte Insertion Type) within the "Transmit STS-3c Path – SONET Control Register – Byte 1", as depicted below.

Transmit STS-3c Path – SONET Control Register – Byte 1 (Address = 0x1982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

This step configures the Transmit STS-3c POH Processor block to use the "TxPOH_n" input port as the source for the Z5 byte, within each "outbound" STS-3c SPE. In this mode, the Transmit STS-3c POH Processor block will accept the value, corresponding to the Z5 byte (via the "TxPOH_n" input port) and it will write this data into the Z5 byte position, within the "outbound" STS-3c SPE.

STEP 2 – Begin providing the values of the "outbound" Z5 byte to the "TxPOH_n" input port.

The procedure for applying the Z5 byte to the "TxPOH_n" input port is presented below.

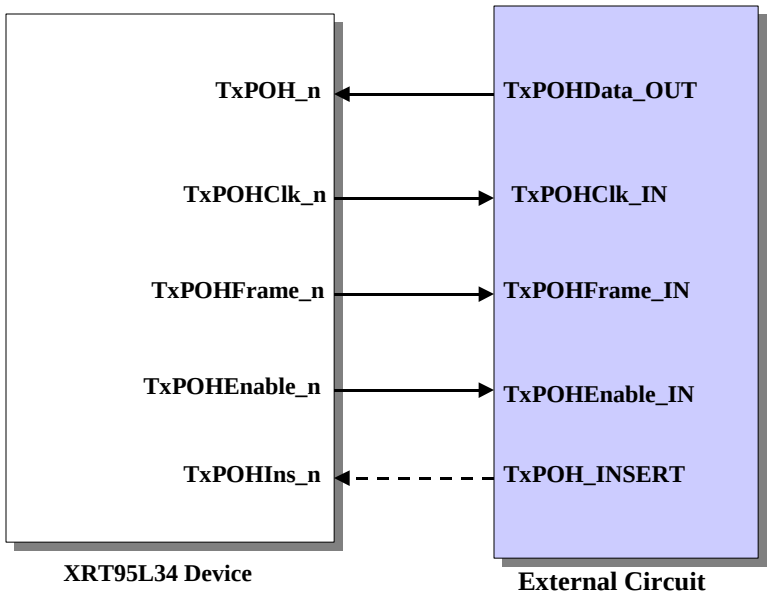
Using the "TxPOH" Input Port to insert the Z5 byte value into the outbound STS-3c SPE data-stream

If the user intends to externally insert the Z5 byte into the outbound STS-3c SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 44

Figure 44: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 44) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for _ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z5 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z5 byte is the ninth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the Z5 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.7.3.11 POINTER-ADJUSTMENT/NDF OPTIONS

FORCING POINTER ADJUSTMENTS AND NDF EVENTS VIA SOFTWARE

The "Transmit STS-3c POH Processor" block permits the user to insert pointer adjustments or NDF events into the "outbound" STS-3c data stream. Specifically, the Transmit STS-3c POH Processor block permits the user to implement the following "pointer-related" features.

- To force the pointer to shift to an "arbitrary value"

- To configure the Transmit STS-3c POH Processor block to only insert pointer-adjustment or NDF events, if no pointer-adjustment (NDF or otherwise) events have occurred within the last three (3) STS-3c framing periods.

- To force a “Positive-Stuff” pointer-adjustment event
- To force a “Negative-Stuff” pointer-adjustment event
- To force a Single NDF Event
- To force a Continuous NDF Event
- To read out and determine the current “pointer-value” within the outbound STS-3c data-stream.

The procedure for implementing each of these “pointer-related” events is presented below.

Forcing the Pointer to Shift to an Arbitrary Value

The Transmit STS-3c POH Processor block permits the user to assign the arbitrary value to the pointer bytes, within the outbound STS-3c SPE data-stream. The user can invoke this feature by executing the following steps.

STEP 1 – Write the appropriate “NDF”, “SS” and H1, H2 Pointer values into the “Transmit STS-3c Path – Transmit Arbitrary H1 Pointer” register, and the “Transmit STS-3c Path – Transmit Arbitrary H2 Pointer” register.

The bit-format for these two registers is presented below.

Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer Register (Address = 0x19BF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NDF Bits				SS Bits		H1 Pointer Value	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit STS-3c Path – Transmit Arbitrary H2 Byte Pointer Register (Address = 0x19C3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
H2 Pointer Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Induce a “0 to 1 transition” in Bit 5 (Pointer Force) within the “Transmit STS-3c Path – Transmit Path Control Register, as depicted below.

Transmit STS-3c Path – Transmit Path Control Register (Address = 0x19B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0 ->1	0	0	0	0	0

Once the user induces this “0 to 1 transition” in Bit 5, then the following events will occur, within the very next “outbound” STS-3c frame.

- The NDF bits, within the H1 byte, will be set to the value written into Bits 4 through 7 (NDF Bits) within the “Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer Register.

- The "SS" bits, within the H1 byte, will be set to the value written into Bits 2 and 3 (SS Bits) within the "Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer Register.

- The 10-bit pointer value (within bits 7 and 8, within the H1 byte, and all eight bits within the H2) will be set to the values written into Bits 1 and 0 (H1 Pointer) within the "Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer" register, and Bits 7 through 0 (H2 Pointer Value) within the "Transmit STS-3c Path – Transmit Arbitrary H2 Byte Pointer" Register.

2.2.7.3.11.1 Forcing Positive-Stuff Pointer Adjustment Events via Software

The Transmit STS-3c POH Processor block permits the user to insert a "positive-stuff" pointer adjustment event into the outbound STS-3c data-stream. This can be accomplished by inducing a "0 to 1" transition in Bit 2 (Insert Positive Stuff) within the "Transmit STS-3c Path – Transmit Path Control" Register, as depicted below.

Transmit STS-3c Path – Transmit Path Control Register (Address = 0x19B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0	0 -> 1	0	0

Once the user induces this "0 to 1 transition" in Bit 2, then the following events will occur.

- A "positive-stuff" will occur (e.g., a single stuff byte will be inserted into the STS-3c data-stream, immediately after the H3 byte position).

- The "I" bits, within the H1 and H2 bytes will be inverted (to denote an "Incrementing" Pointer Adjustment) event.

- After the "positive-stuff" event, the pointer (consisting of the H1 and H2 bytes) will be incremented by "1" and will be used as the new pointer from this point on.

Note: *The contents of Bit 2 (Insert Positive Stuff) will be automatically cleared after the user has written a "1" into this bit-field. Hence, there is no need for the user to go back and write a "0" into this bit-field.*

2.2.7.3.11.2 Forcing Negative-Stuff Pointer Adjustment Events via Software

The Transmit STS-3c POH Processor block permits the user to insert a "Positive-Stuff" pointer adjustment event into the outbound STS-3c data-stream. This can be accomplished by inducing a "0 to 1 transition" within Bit 3 (Insert Negative Stuff) within the "Transmit STS-3c Path – Transmit Path Control" register; as depicted below.

Transmit STS-3c Path – Transmit Path Control Register (Address = 0x19B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0 ->1	0	0	0

Once the user induces a "0 to 1" transition in Bit 3, then the following events will occur.

• A “negative-stuff” will occur (e.g., a single payload byte will be inserted into the “H3 byte” position, within the outbound STS-3c data-stream).

• The “D” bits, within the H1 and H2 bytes will be inverted (to denote a “Decrementing” pointer-adjustment event).

• The contents of the H1 and H2 bytes will be decremented by “1” and will be used as the new pointer from this point on.

Note: *The contents of Bit 3 (Insert Negative Stuff) will be automatically cleared after the user has written a “1” into this bit-field. Hence, there is no need for the user to go back and write a “0” into this bit-field.*

2.2.7.3.11.3 Forcing a Single NDF Event via Software

The Transmit STS-3c POH Processor block permits the user to force a single NDF event into the outbound STS-3c data-stream. This can be accomplished by executing the following steps.

STEP 1- Write the new “desired” pointer value into Bits 1 and 0 (H1 Pointer Value) within the “Transmit STS-3c Path – Transmit Arbitrary H1 Pointer Register; and Bits 7 through 0 (H2 Pointer Value) within the “Transmit STS-3c Path – Transmit Arbitrary H2 Pointer Register.

The bit-format of these two registers (with the relevant bit-fields shaded) is presented below.

Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer Register (Address = 0x19BF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NDF Bits				SS Bits		H1 Pointer Value	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit STS-3c Path – Transmit Arbitrary H2 Byte Pointer Register (Address = 0x19C3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
H2 Pointer Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Induce a “0 to 1 transition” in Bit 0 (Insert Single NDF Event) within the “Transmit STS-3c Path – Transmit Path Control” Register; as depicted below.

Transmit STS-3c Path – Transmit Path Control Register (Address = 0x19B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0	0	0	0->1

Once the user induces this “0 to 1 transition” in Bit 0, then the following events will occur.

• The “N” bits, in the H1 byte (within the very next outbound STS-3c frame) will be set to the value “1001”.

- The ten pointer value bits (within the H1 and H2 bytes) will be set to the value that was written into the "Transmit STS-3c Path – Transmit Arbitrary H1 Byte Pointer" and "Transmit STS-3c Path – Transmit Arbitrary H2 Byte Pointer" registers; during STEP 1.

- Afterwards, the N-bits will resume their normal value of "0110", and this new pointer value will be used as the new pointer, from this point on.

Note: *Once the user writes a "1" into Bit 0 (Insert Single NDF Event), the XRT94L33 will automatically clear this bit-field. Hence, there is no need to subsequently reset this bit-field to "0".*

2.2.8 TRANSMIT SONET POH PROCESSOR BLOCK (FOR ATM/PPP OVER STS-3 APPLICATIONS)

All outbound ATM cells that exit the Transmit ATM Cell Processor block will be routed to the Transmit SONET POH Processor block, where they will be mapped into either STS-1 SPEs. Likewise, all outbound PPP packets that exit the Transmit PPP Packet Processor block will be routed to the Transmit SONET POH Processor block, where they will be mapped into STS-1 SPEs. The purpose of the Transmit SONET POH Processor block is to accomplish the following.

- To accept ATM cells from the Transmit ATM Cell Processor block and to map these cells into STS-1 SPE (Synchronous Payload Envelope), or

- To accept PPP packets from the Transmit PPP Packet Processor block and to map these packets into STS-1 SPEs

- To compute and insert the B3 (Path BIP-8) byte

- To deliberately transmit an STS-1 SPE with an erred B3 byte (for testing purposes)

- To source the J1, C2, G1, F2, H4, Z3, Z4 and Z5 bytes.

- To automatically transmit the RDI-P (Path – Remote Defect) Indicator whenever (and for the duration that) the corresponding Receive SONET POH Processor block declares the AIS-P, LOP-P, UNEQ-P, TIM-P or PLM-P defect conditions.

- To automatically transmit the RDI-P indicator whenever (and for the duration that) the corresponding Receive ATM Cell Processor block declares the LCD defect condition.

- To transmit the RDI-P Indicator upon Software Control.

- To automatically transmit the REI-P (Path – Remote Error) Indicator anytime the corresponding Receive SONET POH Processor block detects B3 byte errors in its incoming STS-1 SPE data-stream.

- To transmit the REI-P indicator upon Software Control.

- To transmit the AIS-P (Path – Alarm Indication Signal) Indicator upon Software Control.

- To permit the user to transmit either 1 byte, 16 byte or 64 byte Path Trace Messages to the remote PTE.

- To force "positive-stuff" pointer adjustments in the outbound STS-1 data-stream.

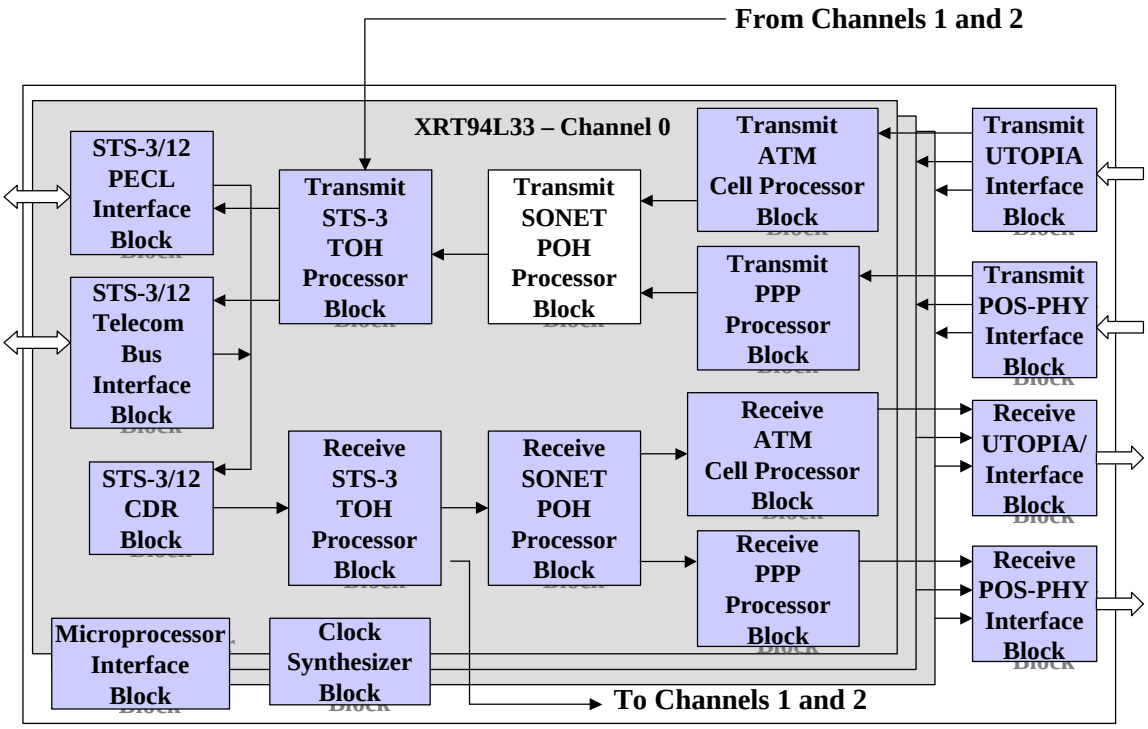
- To force "negative-stuff" pointer adjustments in the outbound STS-1 data-stream.

- To force single and continuous NDF (New Data Flag) events into the outbound STS-1 data-stream

- To route its output data to the Transmit STS-1 TOH Processor block for further processing.

Figure 46presents an illustration of the block diagram of the XRT94L33 Mapper IC, with the "Transmit SONET POH Processor" block highlighted.

Figure 45: Illustration of the Functional Block Diagram of the XRT94L33 Mapper IC, with the Transmit SONET POH Processor block high-lighted



SOME BACKGROUND INFORMATION ON THE PATH OVERHEAD (POH) BYTES

For STS-1 applications, the size of the SPE is 9 rows by 261 byte columns. Figure 46 presents a simple illustration of the STS-1 SPE.

Figure 46: Simple Illustration of the STS-1 SPE

In each case, the first byte column within an STS-1 SPE is referred to as the Path Overhead (POH). Therefore, in all, the POH consists of a total of nine (9) bytes. Each of these POH bytes is briefly described below.

J1 – The Path Trace Byte

B3 – The Path BIP-8 Byte

C2 – The Payload Label Byte

G1 – The Path Status Byte

F2 – The Path User Channel Byte

H4 – The Multi-Frame Indicator (not used for ATM and PPP applications)

Z3, Z4 and Z5 – Growth Bytes

The remainder of this section describes how the Transmit SONET POH Processor block handles these POH bytes.

The operation of the “Transmit SONET POH Processor” block is discussed in some detail below.

2.2.8.1 RECEIVING ATM CELL DATA FROM THE TRANSMIT ATM CELL PROCESSOR BLOCK

The Transmit SONET POH Processor block receives ATM cell data from the Transmit ATM Cell Processor block; and maps this data into an STS-1 SPE.

As the Transmit SONET POH Processor block receives this ATM Cell data from the Transmit ATM Cell Processor block, it will perform the following functions.

- It will map these ATM cells into the payload bytes, within the STS-1 SPE.
- Compute and insert the B3 byte
- Source (per user configuration) the J1, C2, G1, F2, H4, Z3, Z4 and Z5 bytes
- To (automatically or upon software command) transmit the RDI-P (Path – Remote Defect Indicator) indicator
- To (automatically or upon software command) transmit the REI-P (Path – Remote Error Indicator) indicator
- To (upon software command) transmit the AIS-P (Path – Alarm Indication Signal) indicator
- To (upon software command) force pointer-adjustment or NDF (New Data Flag) events into the outbound STS-1 data-stream.

Each of these functions is described in detail below.

2.2.8.2 COMPUTATION AND INSERTION OF THE PATH BIP-8 (B3) BYTE

The Transmit SONET POH Processor block creates STS-1 SPE data and computes a BIP-8 value over this STS-1 SPE. The results of this calculation are inserted into the B3 byte-position within the very next STS-1 SPE. The Remote PTE (Path Terminating Equipment) will use this byte, in order to perform error-checking/detection on the incoming STS-1 SPE data that it receives.

TRANSMISSION OF ERRED B3 BYTES IN THE OUTBOUND STS-1 DATA-STREAM

The Transmit SONET POH Processor block permits the user to insert errors into the “B3 Bytes”, within the outbound STS-1 SPE data-stream.

The user can accomplish this by writing a non-zero value into the “Transmit SONET Path – Transmitter B3 Byte Error Mask” Register. The “Transmit SONET POH Processor block will perform an XOR operation with the contents of the “outbound” B3 byte, and the contents of this register. The results of this calculation are written back into the B3 byte position, within the outbound STS-1 SPE data-stream.

The bit-format of the “Transmit SONET Path – Transmit B3 Byte Error Mask” register is presented below.

Transmit SONET Path – Transmitter B3 Byte Error Mask Register (Address = 0xN997)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_B3_Byte_Mask[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *For normal (e.g., un-erred) operation, the user should ensure that this register is set to “0x00” (the default value).*

2.2.8.3 TRANSMISSION OF PATH-ALARM CONDITIONS

The Transmit SONET POH Processor block supports the following functions.

- Transmitting the AIS-P Pattern (under Software control)
- Transmitting the RDI-P Indicator (automatically and under Software control)
- Transmitting the REI-P Indicator (automatically and under Software control)

Each of these operations is discussed in detail below.

2.2.8.3.1 TRANSMISSION OF THE AIS-P PATTERN

The Transmit SONET POH Processor block can be configured to generate and transmit the AIS-P (Path AIS) indicator to the remote PTE, under software control.

The user can accomplish this by setting Bit 0 (Transmit AIS-P Enable) within the "Transmit SONET Path – SONET Control Register – Byte 0" to "1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	1

Once the user executes this step, then the "Transmit SONET POH Processor" block will do the following.

- It will set all bytes, within the STS-1 SPE to an "All Ones" pattern.
- It will set the H1, H2 and H3 bytes (within the TOH) to an "All Ones" pattern.

Please note that whenever the user configures the Transmit SONET POH Processor block to transmit the AIS-P indicator, then no ATM cells or PPP packets will be transmitted to the remote terminal.

2.2.8.3.2 TRANSMISSION OF THE RDI-P INDICATOR

The Transmit SONET POH Processor block can be configured to transmit the RDI-P indicator either automatically, upon software control or via the "TxPOH_n" external input port as described below.

2.2.8.3.2.1 Configuring the Transmit SONET POH Processor block to automatically transmit the RDI-P Indicator

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator whenever the corresponding Receive SONET POH Processor block declares a PLM-P (Path – Payload Label Mismatch), UNEQ-P (Path – Unequipped), LOP-P (Path – Loss of Pointer), AIS-P, TIM-P (Path - Trace Identification Mismatch), and LCD-P (Path – Loss of Cell Delineation) condition.

The procedure for configuring the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, for each of the above-mentioned defects is presented below.

2.2.8.3.2.2 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the PLM-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor block declaring the PLM-P condition, by executing the following steps.

STEP 1 – Write the value [0, 0] into Bits 3 and 4 (RDI-P Insertion Type[1, 0]) within the "Transmit SONET Path – SONET Control Register – Byte 0" as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the "Transmit SONET POH Processor" block to automatically set bits 5 through 7 (of the G1 byte) within the outbound STS-1 SPE; to the appropriate "RDI-P value" based upon receive conditions as detected by the corresponding "Receive SONET POH Processor" block.

STEP 2 – Write the appropriate value into Bits 3 through 1 (PLM-P RDI-P CODE[2:0]) within the "Transmit SONET Path – RDI-P Control Register – Byte 2" as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 2 (Address = 0xN9C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the value that the "Transmit SONET POH Processor" block will set the RDI-P bit-fields (within the "outbound G1 byte) to, whenever the corresponding Receive SONET POH Processor block declares the PLM-P condition.

STEP 3 – Set Bit 0 (Transmit RDI-P upon PLM-P) within the "Transmit SONET Path – RDI-P Control Register – Byte 2" to "1", as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 2 (Address = 0xN9C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	1

This step configures the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1; within this register), anytime the corresponding Receive SONET POH Processor block detects the "PLM-P" condition.

2.2.8.3.2.3 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the UNEQ-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor block declaring the UNEQ-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 3 through 1 (UNEQ-P RDI-P Code[2:0]) within the "Transmit SONET Path – RDI-P Condition Register – Byte 1, as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 1 (Address = 0xN9CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

By writing this particular value into these bit-fields the user is specifying the value that the "Transmit SONET POH Processor" block will set the RDI-P bit-fields (within the G1 byte of outbound STS-1 frame) to whenever the corresponding Receive SONET POH Processor block declares the UNEQ-P condition.

STEP 2 – Set Bit 0 (Transmit RDI-P upon UNEQ-P) within the "Transmit SONET Path – RDI-P Control Register – Byte 1" to "1", as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 1 (Address = 0xN9CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	1

This step configures the Transmit SONET POH Processor block to transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive SONET POH Processor" block declares the "UNEQ-P" condition.

2.2.8.3.2.4 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the LOP-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor block declaring the LOP-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 7 through 5 (LOP-P RDI-P Code[2:0]) within the "Transmit SONET Path – RDI-P Control Register – Byte 0"; as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 0 (Address = 0xN9CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	0	0	0	0

By writing this particular value into these three bit-fields, the user is specifying the value that the Transmit SONET POH Processor block will set the RDI-P bit-fields (in the G1 byte, within the "outbound" STS-1 data-stream) whenever the corresponding Receive SONET POH Processor block declares the LOP-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon LOP-P) within the "Transmit SONET Path – RDI-P Control Register – Byte 0" to "1"; as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 0 (Address = 0xN9CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	0	0	0	0

This step configures the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive SONET POH Processor" block declares the "LOP-P" condition.

2.2.8.3.2.5 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the AIS-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor declaring the AIS-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 3 through 1 (AIS-P RDI-P Code[2:0]) within the "Transmit SONET Path – RDI-P Control Register – Byte 0"; as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 0 (Address = 0xN9CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the value that the Transmit SONET POH Processor block will set the RDI-P bit-fields (in the G1 byte, within the "outbound" STS-1 data-stream) whenever the corresponding Receive SONET POH Processor block declares the AIS-P condition.

STEP 2 – Set Bit 0 (Transmit RDI-P upon AIS-P) within the "Transmit SONET Path – RDI-P Control Register – Byte 0", as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 0 (Address = 0xN9CB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOP-P RDI-P Code[2:0]			Transmit RDI-P upon LOP-P	AIS-P RDI-P Code[2:0]			Transmit RDI-P upon AIS-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	1

This step configures the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive SONET POH Processor" block declares the "AIS-P" condition.

2.2.8.3.2.6 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the TIM-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor block declaring the TIM-P condition, by executing the following steps.

STEP 1 – Write the appropriate value into Bits 7 through 5 (TIM-P RDI-P Code[2:0]) within the "Transmit SONET Path – RDI-P Condition Register – Byte 1; as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 1 (Address = 0xN9CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	X

By writing this particular value into these bit-fields the user is specifying the value that the "Transmit SONET POH Processor" block will set the RDI-P bit-fields (within the G1 byte of the outbound STS-1 frame) to whenever the corresponding Receive SONET POH Processor block declares the TIM-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon TIM-P) within the "Transmit SONET Path – RDI-P Control Register – Byte 1"; to "1" as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 1 (Address = 0xN9CA)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TIM-P RDI-P Code[2:0]			Transmit RDI-P upon TIM-P	UNEQ-P RDI-P Code[2:0]			Transmit RDI-P upon UNEQ-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	X

This step configures the Transmit SONET POH Processor block to transmit the RDI-P indicator (per the values written into Bits 3 through 1, within this register); anytime the corresponding "Receive SONET POH Processor" block declares the "TIM-P" condition.

2.2.8.3.2.7 Configuring the Transmit SONET POH Processor block to automatically transmit RDI-P, in response to declaration of the LCD-P Condition

The user can configure the Transmit SONET POH Processor block to automatically transmit the RDI-P indicator, in response to the corresponding Receive SONET POH Processor block declaring the LCD-P condition, by executing the following steps.

STEP 1- Write the appropriate value into Bit 7 through 5 (LCD-P RDI-P Code[2:0]) within the "Transmit SONET Path – RDI-P Control Register – Byte 2" as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 2 (Address = 0xN9C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	0

By writing this particular value into these three bit-fields, the user is specifying the values that the "Transmit SONET POH Processor" block will set the RDI-P bit-fields (within the outbound G1 byte) to, whenever the corresponding Receive SONET POH Processor block declares the LCD-P condition.

STEP 2 – Set Bit 4 (Transmit RDI-P upon LCD-P) within the "Transmit SONET Path – RDI-P Control Registers – Byte 2" to "1", as illustrated below.

Transmit SONET Path – RDI-P Control Register – Byte 2 (Address = 0xN9C9)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LCD-P RDI-P Code[2:0]			Transmit RDI-P upon LCD-P	PLM-P RDI-P Code[2:0]			Transmit RDI-P upon PLM-P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	X	X	X	0

2.2.8.3.2.8 Configuring the Transmit SONET POH Processor block to transmit the RDI-P indicator, upon Software Control

The user can configure the Transmit SONET POH Processor block to transmit the RDI-P (per software command) by executing the following steps.

STEP 1 – Write the value "[0, 1]" into Bits 3 and 4 (RDI-P Insertion Type[1:0]) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	1	0	0	0

This step configures the "Transmit SONET POH Processor" block to automatically read out the contents of bits 3 through 1 (of the "Transmit SONET Path – G1 Byte Value" register); and write the value of these bits

into bits 5 through 7 (of the G1 byte) within the "outbound" STS-1 SPE. The bit-format of the "Transmit SONET Path – G1 Byte Value" register (with the appropriate bits "shaded") is presented below.

Transmit SONET Path – Transmit G1 Byte Value Register (Address = 0xN99F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_G1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

In this mode, the user can transmit an "un-erred" indicator by setting Bits 3 through 1 (within the Transmit SONET Path – Transmit G1 Byte Value" register) to [0, 0, 0]. Conversely, the user can now transmit an "erred" indicator by setting Bits 3 through 1 (within the "Transmit SONET Path – Transmit G1 Byte Value" register) to some value other than [0, 0, 0].

Configuring the Transmit SONET POH Processor block to transmit the RDI-P Indicator, via the "TxPOH_n" input port.

The user can configure the Transmit SONET POH Processor block to transmit the RDI-P (per external input port) by executing the following steps.

STEP 1 – Write the value "[1, 0]" into Bits 3 and 4 (RDI-P Insertion Type[1:0]) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	1	0	0	0	0

This step configures the "Transmit SONET POH Processor" block to set the value of the RDI-P bit-fields (within the outbound STS-1 SPE) based upon the data that it receives via the "TxPOH_n" input port. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the RDI-P bit-fields (via the "TxPOH_n Input Port") and it will write this data into the "outbound" STS-1 SPE data-stream.

STEP 2 – Begin providing the values of the "outbound" RDI-P bit-fields to the "TxPOH_n" input port.

The procedure for applying the RDI-P bit-values to the "TxPOH_n" input port is presented below.

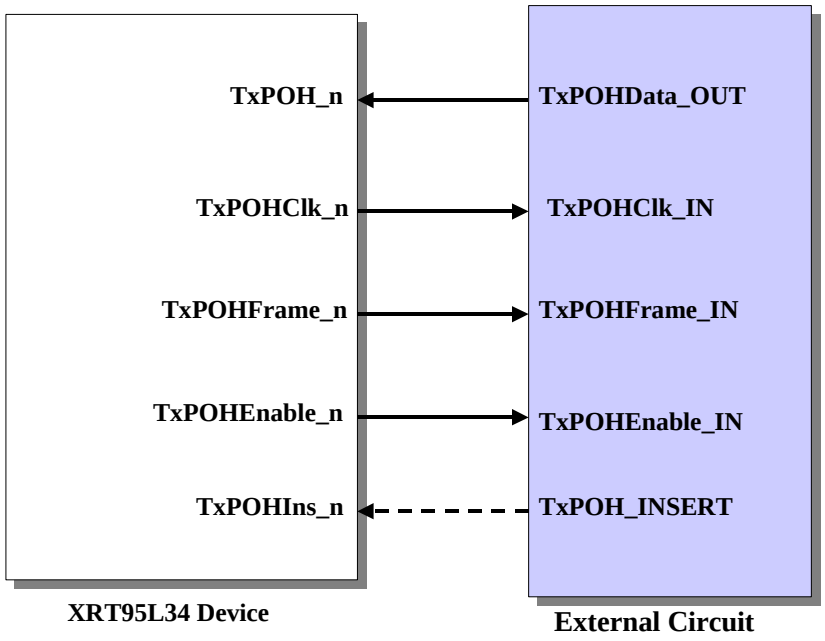
Using the "TxPOH" Input Port to insert the RDI-P bit values into the outbound STS-1 SPE data-stream

If the user intends to externally insert the RDI-P bits into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 47.

Figure 47: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 47) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it waits for 25 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit of the "outbound" RDI-P bit-fields onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the G1 byte is the 4th byte within the POH; and bit 7 (which is the very first RDI-P bit to be latched into the External Input Interface is the second bit within the G1 byte to be processed).*

- Afterwards, the "external circuit" should serially place the remaining two bits (of the RDI-P bits) onto the "TxPOH_n" input pin, upon each of the next two falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 48 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the RDI-P bits into the "TxPOH Input Port".

Figure 48 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "RDI-P bits" into the "TxPOH Input Port".

2.2.8.3.3 TRANSMISSION OF THE REI-P INDICATOR

The Transmit SONET POH Processor block can be configured to transmit the REI-P indicator either (1) automatically, (2) upon software command or (3) via the "TxPOH_n" input port as described below.

2.2.8.3.3.1 Configuring the Transmit SONET POH Processor block to automatically transmit the REI-P indicator

The user can configure the Transmit SONET POH Processor block to automatically transmit the REI-P indicator whenever the corresponding Receive SONET POH Processor block detects at least one B3 byte error within its incoming STS-1 SPE data-stream.

The user can configure the Transmit SONET POH Processor block to automatically transmit the REI-P indicator, in response to detection of B3 byte errors, by executing the following steps.

STEP 1 – Write the value [0, 0] into Bits 5 and 6 (REI-P Insertion Type[1, 0]) within the "Transmit SONET Path – SONET Control Register – Byte 0" as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the "Transmit SONET POH Processor" block to automatically set bits 1 through 4 (of the G1 byte) within the "outbound" STS-1 SPE to the appropriate REI-P value, based upon receive conditions as detected by the corresponding Receive SONET POH Processor block.

STEP 2 – Indicate whether the REI-P value (transmitted to the remote PTE) reflects the number of bits (within the B3 byte) that are in error, or the number of erred STS-1 SPE that have been detected by the corresponding Receive SONET POH Processor block.

The XRT94L33 permits the user to (1) flag B3 byte errors, and (2) to transmit the resulting REI-P value (back out to the remote PTE) by the following means.

- By flagging and reporting the number of bits (within the B3 byte) that have be determined to be in error, within a given STS-1 SPE.
- By flagging and reporting whether or not the corresponding Receive SONET POH Processor block is currently receiving erred STS-1 SPE.

The user can choose between these two options by writing the appropriate value into Bit 0 (B3 Error Type) within the Receive SONET Path – Control Register – Byte 0"; as illustrated below.

Receive SONET Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	X

Setting this bit-field to "0" configures the Transmit SONET POH Processor block to set the REI-P bit-fields (within the G1 byte of the "outbound" STS-1 SPE); to a value the reflects the number of bits (within the B3 byte) that were determined to be in error (within the most recently received STS-1 SPE) by the corresponding Receive SONET POH Processor block. In this case, the REI-P bit-fields can range in value from "0" (for no B3 bit errors) to "8" (for all B3 bits being in error).

Setting this bit-field to "1" configures the Transmit SONET POH Processor block to set the REI-P bit-fields (within the G1 byte of the 'outbound"STS-1 SPE); to a value that indicates whether or not at least one B3 byte error was detected within the most recently received STS-1 SPE; by the corresponding Receive SONET POH Processor block. In this mode, the Transmit SONET POH Processor block will set the "REI-P" bit-fields (within the outbound STS-1 SPE) to "0" if the corresponding "Receive SONET POH Processor" block is not detecting any B3 byte errors, in its incoming STS-1 data-stream. Conversely, the Transmit SONET POH Processor block will set the "REI-P" bit-fields (within the "outbound" STS-1 SPE) to "1" if the corresponding Receive SONET POH Processor block receives an STS-1 SPE that contains a B3 byte error.

2.2.8.3.3.2 Configuring the Transmit SONET POH Processor block to transmit the REI-P Indicator, upon software control

The user can configure the Transmit SONET POH Processor block to transmit the REI-P indicator (per software command) by executing the following steps.

STEP 1- Write the value "[0, 1]" into Bits 5 and 6 (REI-P Insertion Type[1:0]) within the "Transmit SONET Path SONET Control Register – Byte 0"; as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

This step configures the "Transmit SONET POH Processor" block to automatically read out the contents of Bits 4 through 7 (of the Transmit SONET Path – G1 Byte Value" register) and write the value of these bits into bits 1 through 4 (of the G1 byte) within the "outbound" STS-1 SPE. The bit-format of the "Transmit SONET Path – G1 Byte Value" register (with the appropriate bits "shaded") is presented below.

Transmit SONET Path – Transmit G1 Byte Value Register (Address = 0xN99F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_G1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

In this mode, the user can transmit an "un-erred" REI-P value by setting Bits 4 through 7 (within the Transmit SONET Path – Transmit G1 Byte Value" register) to [0, 0, 0, 0]. Conversely, the user can transmit an "erred" REI-P value by setting Bits 4 through 7 (within the Transmit SONET Path – Transmit G1 Byte Value" register) to some value between 1 and 8.

2.2.8.3.3.3 Configuring the Transmit SONET POH Processor block to transmit the REI-P indicator, via the "TxPOH_n" input port

The user can configure the Transmit SONET POH Processor block to transmit the REI-P (per the external input port) by executing the following steps.

STEP 1 – Write the value [1, 0] into Bits 3 and 4 (REI-P Insertion Type[1:0]) within the "Transmit SONET Path – SONET Control Register – Byte 0" as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	C2 Byte Auto Insert Mode Enable	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to set the value of the REI-P bit-fields (within the outbound STS-1 SPE) based upon the data that it receives via the "TxPOH_n" input port. In this mode, the Transmit SONET POH Processor block will accept the value corresponding to the REI-P fields (via the "TxPOH_n Input Port") and it will write this data into the "outbound" STS-1 SPE data-stream.

STEP 2 – Begin providing the values of the "outbound" REI-P bit-fields to the "TxPOH_n" input port.

The procedure for applying the REI-P bit-values to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the REI-P bit values into the outbound STS-1 SPE data-stream

If the user intends to externally insert the REI-P bits into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHEnable_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.
- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it waits for 28 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit of the "outbound" REI-P bit-fields onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n"

2.2.8.3.4 TRANSMISSION OF PATH TRACE MESSAGES VIA THE J1 BYTE

The Transmit SONET POH Processor block permits the user to transmit Path Trace Messages to the remote PTE via the J1 byte. The Transmit SONET POH Processor block permits the user to accomplish this by either of the following options.

- Automatically set the J1 byte (within each outbound STS-1 SPE) to "0x00"
- Set and control the outbound J1 byte via on-chip register
- Set and control the outbound J1 byte via external input pin
- Use the "Transmit Path Trace Message" buffer

The details and instructions for using either of these features are presented below.

2.2.8.3.4.1 Automatically setting the J1 byte (within each outbound STS-1 SPE) to "0x00"

The XRT94L33 permits the user to configure each of the four (4) Transmit SONET POH Processor blocks to automatically set the contents of the J1 byte (within each outbound STS-1 SPE) to "0x00".

The user can accomplish this by writing the value "[0, 0]" into Bits 1 and 0 (J1 Type[1:0]) within the appropriate "Transmit SONET Path – Transmit J1 Control" Register; as depicted below.

Transmit SONET Path – Transmit J1 Control" Register (Address = 0xN9BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	0

Once the user executes this step, then the corresponding Transmit SONET POH Processor block(s) will be configured to automatically set the J1 byte (within each outbound STS-1 SPE) to the value "0x00".

2.2.8.3.4.2 Using the "Transmit Path Trace Message" Buffer

The XRT94L33 contains a total of four (4) "Transmit Path Trace Message Buffers (one for each "Transmit SONET POH Processor Block, within the device).

The address location of the "Transmit Path Trace Message" buffer, for each of the four Transmit SONET POH Processor blocks is presented below.

Table 15: Address Locations of the Transmit Path Trace Message Buffers within the XRT94L33

TRANSMIT SONET POH PROCESSOR BLOCK - CHANNEL	ADDRESS LOCATION OF THE TRANSMIT PATH TRACE MESSAGE BUFFER
0	0x1D00 – 0x1D3F
1	0x2D00 – 0x2D3F
2	0x3D00 – 0x3D3F
3	0x4D00 – 0x4D3F

If the user wishes to use the "Path Trace Message" buffer as the means to load and transmit the "Path Trace Message" to the remote PTE, then the following steps must be executed.

STEP 1 – Write the contents of this outbound "Path Trace Message" into the "Transmit Path Trace Message" buffer.

As the user writes the contents of the "outbound" Path Trace Message" into the "Transmit Path Trace" Message buffer they must make sure that the first byte of the message is written into Address location 0xNC00 (e.g., the very first address location of the "Transmit Path Trace Message" buffer).

STEP 2 – Specify the length of this "Path Trace Message" by writing the appropriate value into Bits 3 and 2 (J1 Message Length[1:0]) within the "Transmit SONET Path – Transmit J1 Control" Register; as depicted below.

Transmit SONET Path – Transmit J1 Control" Register (Address = 0xN9BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	X

The relationship between the contents of "J1_Message_Length[1:0]" and the corresponding length of the "Path Trace Message" is presented below in Table _.

Table 16 The Relationship Between the contents of "J1_Message_Length[1:0]" and the corresponding "Path Trace Message" Length

J1_MESSAGE_LENGTH[1:0]	PATH TRACE MESSAGE LENGTH (BYTES)
00	1
01	16
10	64
11	64

STEP 3 – Write the value "[0, 1]" into Bits 1 and 0 (J1_Type[1:0]) within the Transmit SONET Path – Transmit J1 Control Register; as depicted below.

Transmit SONET Path – Transmit J1 Control" Register (Address = 0xN9BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	1

This step configures the Transmit SONET POH Processor block to use the contents of the "Transmit Path Trace Message" buffer as the source of the J1 byte, within each "outbound" STS-1 SPE.

2.2.8.3.4.3 Setting and Controlling the "Outbound" J1 Byte via On-Chip Register

The Transmit SONET POH Processor block permits the user to specify the contents of the J1 byte within the "outbound" STS-1 SPE, via software command. The user can configure the Transmit SONET POH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 0]" into Bits 1 and 0 (J1 Type[1:0]) within the "Transmit SONET Path – Transmit J1 Control" Register, as depicted below.

Transmit SONET Path – Transmit J1 Control" Register (Address = 0xN9BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	1	0

This step configures the "Transmit SONET POH Processor" block to read out the contents of the "Transmit Path – Transmit J1 Byte Value" register; and load this value into the "J1 byte-field" within each outbound STS-1 SPE.

STEP 2 – Write the desired byte value (for the "outbound" J1 byte) into the "Transmit SONET Path – Transmit J1 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmitter J1 Value Register (Address = 0xN993)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_J1_Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

2.2.8.3.4.4 Setting and Controlling the "Outbound" J1 Byte via External Input pin

The Transmit SONET POH Processor block permits the user to source the contents of the J1 byte via the "TxPOH_n" input port. The user can configure the Transmit SONET POH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 1]" into Bits 1 and 0 (J1 Type[1:0]) within the "Transmit SONET Path – Transmit J1 Control" Register; as depicted below.

Transmit SONET Path – Transmit J1 Control" Register (Address = 0xN9BB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J1 Message Length[1:0]		J1 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

This step configures the "Transmit SONET POH Processor" block to accept the value of the J1 byte, via the "TxPOH_n" input port and load this value into the J1 byte position within each outbound STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" J1 byte message to the "TxPOH_n" input port.

The procedure for applying the J1 byte to the "TxPOH_n" input port is presented below.

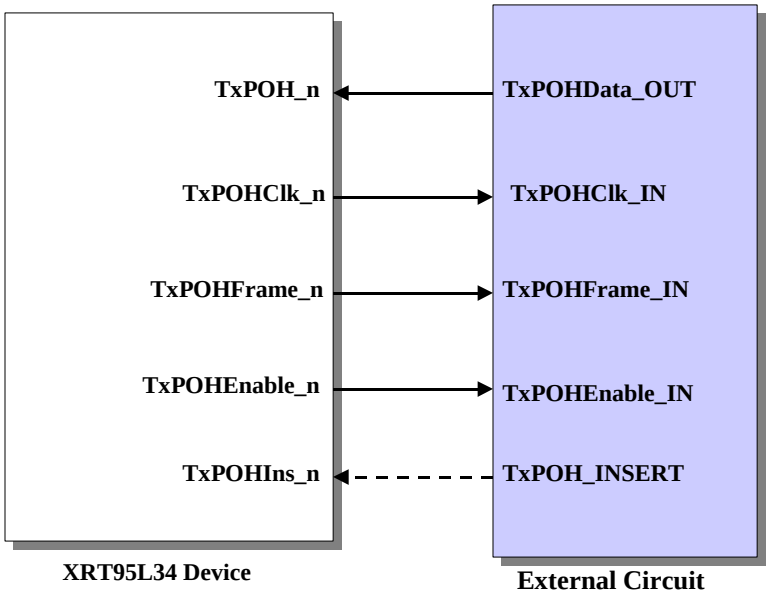
Using the "TxPOH" Input Port to insert the J1 byte value into the outbound STS-1 SPE data-stream

If the user intends to externally insert the J1 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 49.

Figure 49: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 49) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should place the very first bit (e.g., the most significant bit) of the "outbound" J1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the J1 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.5 SUPPORT/HANDLING OF THE C2 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the C2 byte by either of the following options.

- Setting and controlling the "outbound" C2 Byte via Software
- Setting and controlling the "outbound" C2 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.5.1 Setting and Controlling the Outbound C2 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the C2 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 2 (C2 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit C2 Byte Value" register; and load this value into the C2 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound C2 byte) into the "Transmit SONET Path – Transmit C2 Byte Value" register.

Transmit SONET Path – Transmit C2 Byte Value Register (Address = 0xN99F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_C2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.5.2 Setting and Controlling the Outbound C2 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the C2 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 2 (C2 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	1	0	0

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the C2 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the C2 byte (via the "TxPOH_n" input port) and it will write this data into the C2 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" C2 byte to the "TxPOH_n" input port.

The procedure for applying the C2 byte to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the C2 byte value into the outbound STS-1 SPE data-stream

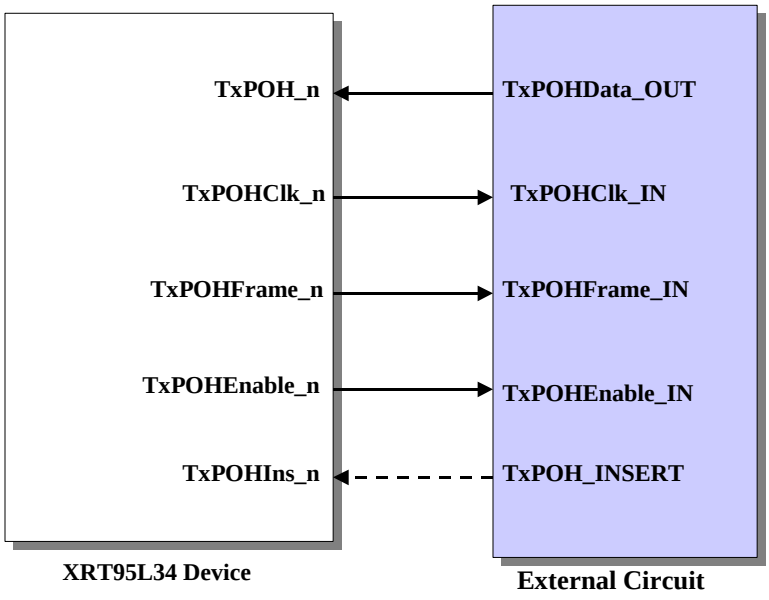
If the user intends to externally insert the C2 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in

Figure 50.

Figure 50: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in*

Figure 50) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 16 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" C2 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the C2 byte is the third byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the C2 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 51 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the C2 byte into the "TxPOH Input Port".

Figure 51 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "C2 byte" into the "TxPOH Input Port".

2.2.8.3.6 SUPPORT/HANDLING OF THE F2 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the F2 byte by either of the following options.

- Setting and controlling the "outbound" F2 Byte via Software
- Setting and controlling the "outbound" F2 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.6.1 Setting and Controlling the Outbound F2 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the F2 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 7 (F2 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit F2 Byte Value" register; and load this value into the F2 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound F2 byte) into the "Transmit SONET Path – Transmit F2 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmit F2 Byte Value Register (Address = 0xN9A3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_F2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.6.2 Setting and Controlling the Outbound F2 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the F2 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 7 (F2 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 0", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 0 (Address = 0xN983)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
F2 Insertion Type	REI-P Insertion Type[1:0]		RDI-P Insertion Type[1:0]		C2 Byte Insertion Type	Unused	Transmit AIS-P Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W
1	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the F2 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the F2 byte (via the "TxPOH_n" input port) and it will write this data into the F2 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" F2 byte to the "TxPOH_n" input port.

The procedure for applying the F2 byte to the "TxPOH_n" input port is presented below.

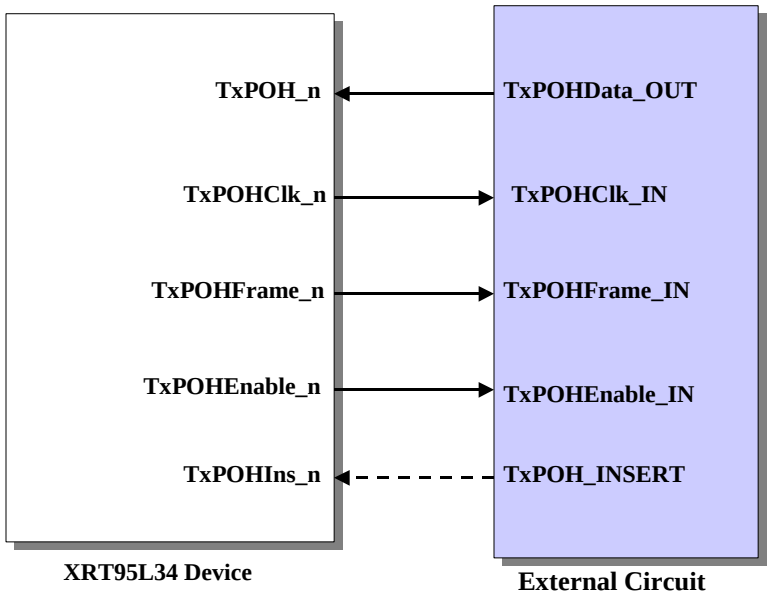
Using the "TxPOH" Input Port to insert the F2 byte value into the outbound STS-1 SPE data-stream

If the user intends to externally insert the F2 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 52.

Figure 52: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 52) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 32 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" F2 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the F2 byte is the fifth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the F2 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.7 SUPPORT/HANDLING OF THE H4 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" H4 Byte via Software
- Setting and controlling the "outbound" H4 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.7.1 Setting and Controlling the Outbound H4 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the H4 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (H4 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit H4 Byte Value" register; and load this value into the C2 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound H4 byte) into the "Transmit SONET Path – Transmit H4 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmit H4 Byte Value Register (Address = 0xN9A7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_H4_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.7.2 Setting and Controlling the Outbound H4 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the H4 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (F2 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the H4 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the H4 byte (via the "TxPOH_n" input port) and it will write this data into the H4 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" H4 byte to the "TxPOH_n" input port.

The procedure for applying the H4 byte to the "TxPOH_n" input port is presented below.

Using the "TxPOH" Input Port to insert the H4 byte value into the outbound STS-1 SPE data-stream

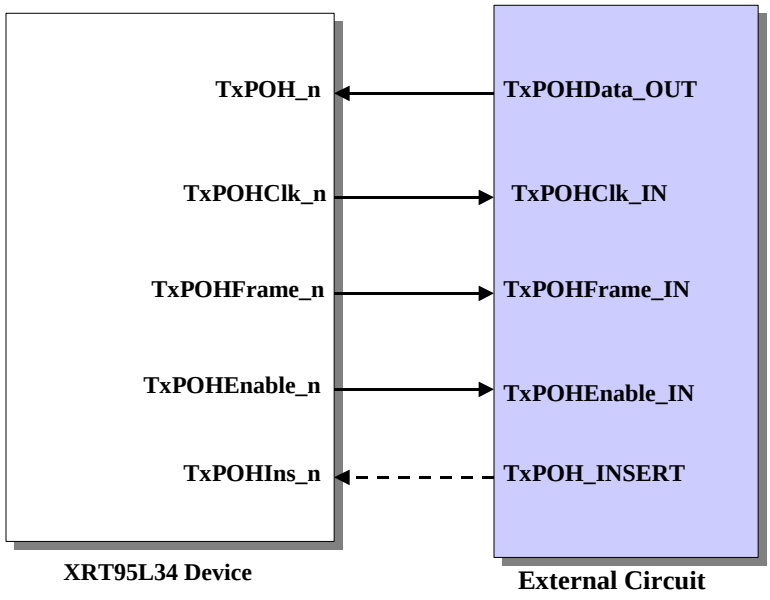
If the user intends to externally insert the H4 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in

Figure 53.

Figure 53: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in*

Figure 53) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 40 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" H4 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the H4 byte is the sixth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the H4 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.8 SUPPORT/HANDLING OF THE Z3 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" Z3 Byte via Software
- Setting and controlling the "outbound" Z3 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.8.1 Setting and Controlling the Outbound Z3 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the Z3 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 1 (Z3 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit Z3 Byte Value" register; and load this value into the Z3 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound Z3 byte) into the "Transmit SONET Path – Transmit Z3 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmit Z3 Byte Value Register (Address = 0xN9AB)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z3_Byte_Value[7:0]							

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.8.2 Setting and Controlling the Outbound Z3 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the Z3 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 1 (Z3 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the Z3 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the Z3 byte (via the "TxPOH_n" input port) and it will write this data into the Z3 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" Z3 byte to the "TxPOH_n" input port.

The procedure for applying the Z3 byte to the "TxPOH_n" input port is presented below.

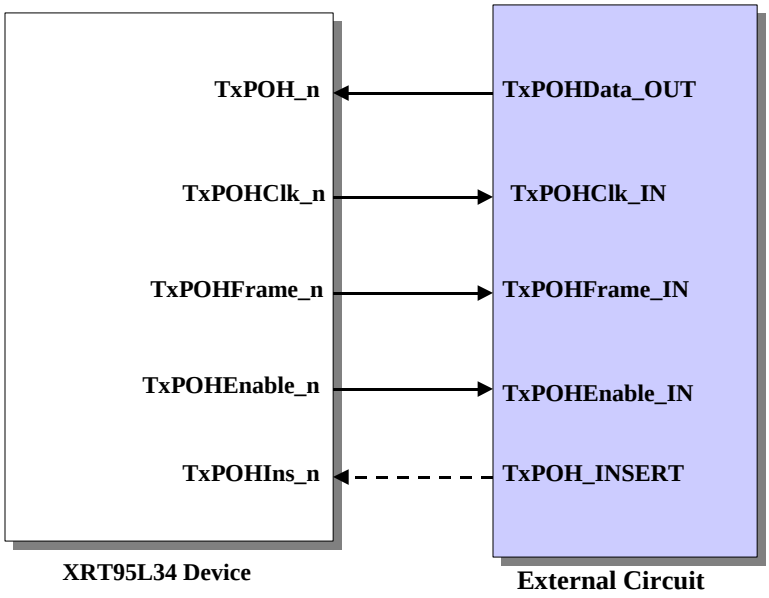
Using the "TxPOH" Input Port to insert the Z3 byte value into the outbound STS-1 SPE data-stream

If the user intends to externally insert the Z3 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 54.

Figure 54 : A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (inFigure 54) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for 48 periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z3 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z3 byte is the seventh byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the Z3 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.9 SUPPORT/HANDLING OF THE Z4 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the Z4 byte by either of the following options.

- Setting and controlling the "outbound" Z4 Byte via Software
- Setting and controlling the "outbound" Z4 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.9.1 Setting and Controlling the Outbound Z4 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the Z4 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (Z4 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit Z4 Byte Value" register; and load this value into the Z4 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound H4 byte) into the "Transmit SONET Path – Transmit Z4 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmit Z4 Byte Value Register (Address = 0xN9AF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z4_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.9.2 Setting and Controlling the Outbound Z4 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the Z4 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (Z4 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	0

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the Z4 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the Z4 byte (via the "TxPOH_n" input port) and it will write this data into the Z4 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" Z4 byte to the "TxPOH_n" input port.

The procedure for applying the Z4 byte to the "TxPOH_n" input port is presented below.

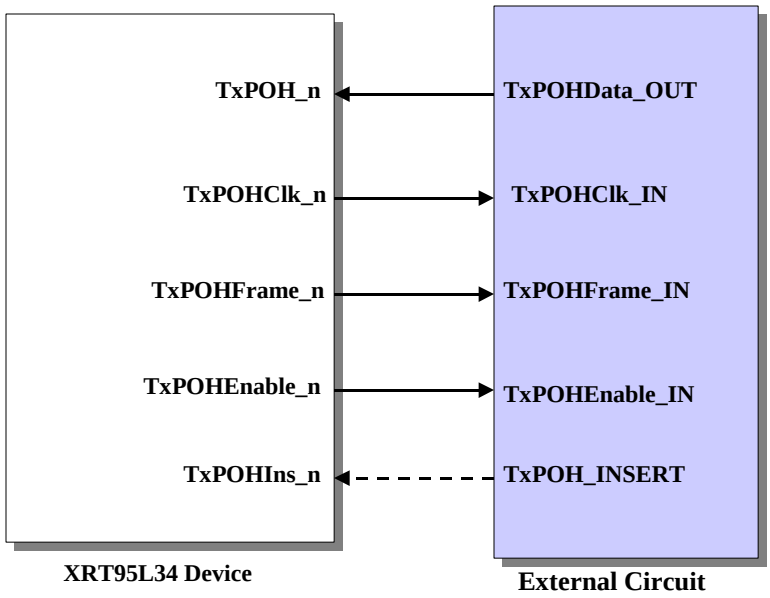
Using the "TxPOH" Input Port to insert the Z4 byte value into the outbound STS-1 SPE data-stream

If the user intends to externally insert the Z4 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 55.

Figure 55: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (inFigure 55) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for _ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z4 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z4 byte is the 8th byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the H4 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.10 SUPPORT/HANDLING OF THE Z5 BYTE

The Transmit SONET POH Processor block permits the user to control the value of the H4 byte by either of the following options.

- Setting and controlling the "outbound" Z5 Byte via Software
- Setting and controlling the "outbound" Z5 Byte via the "TxPOH Input Port"

The details and instructions for using either or these features are presented below.

2.2.8.3.10.1 Setting and Controlling the Outbound Z5 Byte via Software

The Transmit SONET POH Processor block permits the user to specify the contents of the Z5 byte, within the "outbound" STS-1 SPE via software command.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "0" into Bit 0 (Z5 Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	X

This step configures the Transmit SONET POH Processor block to read out the contents of the "Transmit SONET Path – Transmit Z5 Byte Value" register; and load this value into the Z5 byte position within each "outbound" STS-1 SPE.

STEP 2 – Write the desired byte value (for the outbound Z5 byte) into the "Transmit SONET Path – Transmit Z5 Byte Value" register.

The bit-format of this register is presented below.

Transmit SONET Path – Transmit Z5 Byte Value Register (Address = 0xN9B3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_Z5_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.8.3.10.2 Setting and Controlling the Outbound Z5 Byte via the "TxPOH_n Input Port"

The Transmit SONET POH Processor block permits the user to specify the contents of the Z5 byte, within the "outbound" STS-1 SPE, via data applied to the "TxPOH_n" input pin.

The user can configure the Transmit SONET POH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 0 (Z5 Byte Insertion Type) within the "Transmit SONET Path – SONET Control Register – Byte 1", as depicted below.

Transmit SONET Path – SONET Control Register – Byte 1 (Address = 0xN982)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Z5 Insertion Type	Z4 Insertion Type	Z3 Insertion Type	H4 Insertion Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

This step configures the Transmit SONET POH Processor block to use the "TxPOH_n" input port as the source for the Z5 byte, within each "outbound" STS-1 SPE. In this mode, the Transmit SONET POH Processor block will accept the value, corresponding to the Z5 byte (via the "TxPOH_n" input port) and it will write this data into the Z5 byte position, within the "outbound" STS-1 SPE.

STEP 2 – Begin providing the values of the "outbound" Z5 byte to the "TxPOH_n" input port.

The procedure for applying the Z5 byte to the "TxPOH_n" input port is presented below.

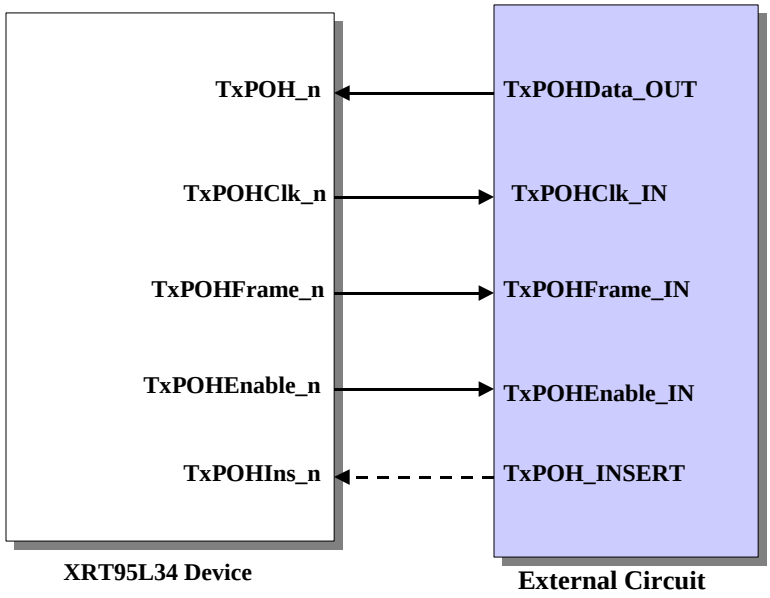
Using the "TxPOH" Input Port to insert the Z5 byte value into the outbound STS-1 SPE data-stream

If the user intends to externally insert the Z5 byte into the outbound STS-1 SPE, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 56.

Figure 56: A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 56) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxPOHEnable_n" and "TxPOHFrame_n" output pins "high", then it should enter a "WAIT STATE" (e.g., where it will wait for _ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" Z5 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit SONET POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the Z5 byte is the ninth byte within the POH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the Z5 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

2.2.8.3.11 POINTER-ADJUSTMENT/NDF OPTIONS

Forcing Pointer Adjustments and NDF Events via Software

The "Transmit SONET POH Processor" block permits the user to insert pointer adjustments or NDF events into the "outbound" STS-1 data stream. Specifically, the Transmit SONET POH Processor block permits the user to implement the following "pointer-related" features.

- To force the pointer to shift to an "arbitrary value"
- To configure the Transmit SONET POH Processor block to only insert pointer-adjustment or NDF events, if no pointer-adjustment (NDF or otherwise) events have occurred within the last three (3) STS-1 framing periods.
- To force a "Positive-Stuff" pointer-adjustment event
- To force a "Negative-Stuff" pointer-adjustment event
- To force a Single NDF Event
- To force a Continuous NDF Event
- To read out and determine the current "pointer-value" within the outbound STS-1 data-stream.

The procedure for implementing each of these "pointer-related" events is presented below.

2.2.8.3.11.1 Forcing the Pointer to Shift to an Arbitrary Value

The Transmit SONET POH Processor block permits the user to assign the arbitrary value to the pointer bytes, within the outbound STS-1 SPE data-stream. The user can invoke this feature by executing the following steps.

STEP 1 – Write the appropriate "NDF", "SS" and H1, H2 Pointer values into the "Transmit SONET Path – Transmit Arbitrary H1 Pointer" register, and the "Transmit SONET Path – Transmit Arbitrary H2 Pointer" register.

The bit-format for these two registers is presented below.

Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer Register (Address = 0xN9BF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NDF Bits				SS Bits		H1 Pointer Value	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit SONET Path – Transmit Arbitrary H2 Byte Pointer Register (Address = 0xN9C3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
H2 Pointer Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Induce a "0 to 1 transition" in Bit 5 (Pointer Force) within the "Transmit SONET Path – Transmit Path Control Register, as depicted below.

Transmit SONET Path – Transmit Path Control Register (Address = 0xN9B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0 ->1	0	0	0	0	0

Once the user induces this "0 to 1 transition" in Bit 5, then the following events will occur, within the very next "outbound" STS-1 frame.

- The NDF bits, within the H1 byte, will be set to the value written into Bits 4 through 7 (NDF Bits) within the "Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer Register.
- The "SS" bits, within the H1 byte, will be set to the value written into Bits 2 and 3 (SS Bits) within the "Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer Register.
- The 10-bit pointer value (within bits 7 and 8, within the H1 byte, and all eight bits within the H2) will be set to the values written into Bits 1 and 0 (H1 Pointer) within the "Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer" register, and Bits 7 through 0 (H2 Pointer Value) within the "Transmit SONET Path – Transmit Arbitrary H2 Byte Pointer" Register.

2.2.8.3.11.2 Forcing Positive-Stuff Pointer Adjustment Events via Software

The Transmit SONET POH Processor block permits the user to insert a "positive-stuff" pointer adjustment event into the outbound STS-1 data-stream. This can be accomplished by inducing a "0 to 1" transition in Bit 2 (Insert Positive Stuff) within the "Transmit SONET Path – Transmit Path Control" Register, as depicted below.

Transmit SONET Path – Transmit Path Control Register (Address = 0xN9B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0	0 -> 1	0	0

Once the user induces this "0 to 1 transition" in Bit 2, then the following events will occur.

- A "positive-stuff" will occur (e.g., a single stuff byte will be inserted into the STS-1 data-stream, immediately after the H3 byte position).
- The "I" bits, within the H1 and H2 bytes will be inverted (to denote an "Incrementing" Pointer Adjustment) event.
- After the "positive-stuff" event, the pointer (consisting of the H1 and H2 bytes) will be incremented by "1" and will be used as the new pointer from this point on.

Note: *The contents of Bit 2 (Insert Positive Stuff) will be automatically cleared after the user has written a "1" into this bit-field. Hence, there is no need for the user to go back and write a "0" into this bit-field.*

2.2.8.3.11.3 Forcing Negative-Stuff Pointer Adjustment Events via Software

The Transmit SONET POH Processor block permits the user to insert a "Positive-Stuff" pointer adjustment event into the outbound STS-1 data-stream. This can be accomplished by inducing a "0 to 1 transition" within Bit 3 (Insert Negative Stuff) within the "Transmit SONET Path – Transmit Path Control" register; as depicted below.

Transmit SONET Path – Transmit Path Control Register (Address = 0xN9B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0 ->1	0	0	0

Once the user induces a "0 to 1" transition in Bit 3, then the following events will occur.

- A "negative-stuff" will occur (e.g., a single payload byte will be inserted into the "H3 byte" position, within the outbound STS-1 data-stream).
- The "D" bits, within the H1 and H2 bytes will be inverted (to denote a "Decrementing" pointer-adjustment event).
- The contents of the H1 and H2 bytes will be decremented by "1" and will be used as the new pointer from this point on.

Note: *The contents of Bit 3 (Insert Negative Stuff) will be automatically cleared after the user has written a "1" into this bit-field. Hence, there is no need for the user to go back and write a "0" into this bit-field.*

2.2.8.3.11.4 Forcing a Single NDF Event via Software

The Transmit SONET POH Processor block permits the user to force a single NDF event into the outbound STS-1 data-stream. This can be accomplished by executing the following steps.

STEP 1- Write the new "desired" pointer value into Bits 1 and 0 (H1 Pointer Value) within the "Transmit SONET Path – Transmit Arbitrary H1 Pointer Register; and Bits 7 through 0 (H2 Pointer Value) within the "Transmit SONET Path – Transmit Arbitrary H2 Pointer Register.

The bit-format of these two registers (with the relevant bit-fields shaded) is presented below.

Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer Register (Address = 0xN9BF)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NDF Bits				SS Bits		H1 Pointer Value	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit SONET Path – Transmit Arbitrary H2 Byte Pointer Register (Address = 0xN9C3)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
H2 Pointer Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Induce a "0 to 1 transition" in Bit 0 (Insert Single NDF Event) within the "Transmit SONET Path – Transmit Path Control" Register; as depicted below.

Transmit SONET Path – Transmit Path Control Register (Address = 0xN9B7)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Pointer Force	Check Stuff	Insert Negative Stuff	Insert Positive Stuff	Insert Continuous NDF Events	Insert Single NDF Event
R/O	R/O	R/W	R/W	W	W	R/W	R/W
0	0	0	0	0	0	0	0->1

Once the user induces this "0 to 1 transition" in Bit 0, then the following events will occur.

- The "N" bits, in the H1 byte (within the very next outbound STS-1 frame) will be set to the value "1001".
- The ten pointer value bits (within the H1 and H2 bytes) will be set to the value that was written into the "Transmit SONET Path – Transmit Arbitrary H1 Byte Pointer" and "Transmit SONET Path – Transmit Arbitrary H2 Byte Pointer" registers; during STEP 1.

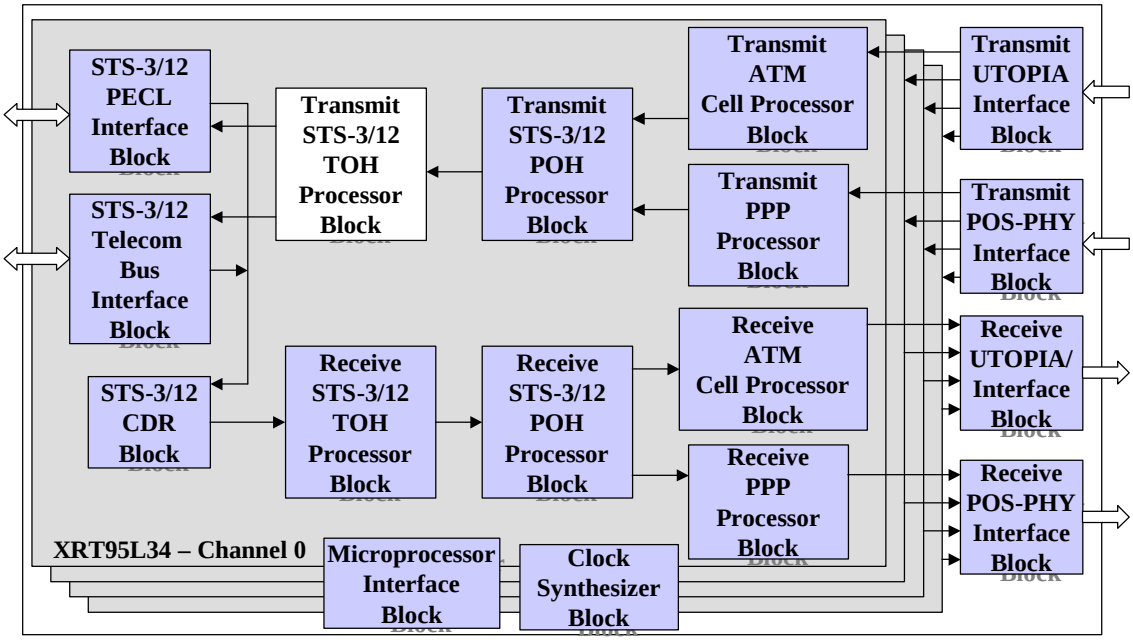
2.2.9 TRANSMIT STS-3 TOH PROCESSOR BLOCK

The purpose of the Transmit STS-3 TOH Processor Block is to accomplish the following.

- To accept STS-3c SPE data from the Transmit STS-3c POH Processor Block (for STS-3c Applications)
- To accept 3 channels of STS-1 SPE data from each of the three (3) Transmit SONET POH Processor blocks (for STS-3 Applications)
- To compute and insert the B1 (Section BIP-8) and B2 (Line BIP-8) bytes.
- To source the K1/K2, E1, M0, J0, F1 and S1 bytes.
- To transmit the RDI-L (Line – Remote Defect Indicator) Indicator.
- To transmit the REI-L (Line – Remote Error Indicator) Indicator.
- To transmit the AIS-L (Line – Alarm Indication Signal) Indicator
- To deliberately transmit an erred B1 or B2 byte (for testing purposes)
- To optionally scramble the outbound STS-3 data.
- To output this data to either the Transmit PECL Interface or the Transmit STS-3 Telecom Bus Interface, for transmission to the remote terminal equipment.

Figure 57 presents an illustration of the Functional Block Diagram of the XRT94L33 ATM UNI/PPP/Mapper IC with the Transmit STS-3 TOH Processor block highlighted.

Figure 57: Illustration of the Functional Block Diagram of the XRT94L33 ATM UNI/PPP/Mapper IC with the Transmit STS-3 TOH Processor Block highlighted



SOME BACKGROUND INFORMATION ON THE TRANSPORT OVERHEAD (TOH) BYTES

As mentioned earlier, the XRT94L33 will handle either STS-1 frame, STS-3c frames, or STS-3. The size of an STS-3 frame is 9 rows by 270 byte columns.

Figure 58 presents a simple illustration of the STS-3 Frame.

Figure 58: Simple Illustration of the STS-3 Frame

In the case of an STS-N frame, the first "N x 3" column are referred to as the "Transport Overhead" (TOH). Therefore, in all, the TOH consists of a total of 27 x N bytes. Each of these POH bytes is briefly described below.

A1, A2 – Framing Alignment Bytes

J0 – The Section Trace (Growth) Byte

B1 – The Section BIP-8 Byte

E1 – The Orderwire

The operation of the "Transmit STS-3 TOH Processor" block is discussed in some detail below.

2.2.9.1 RECEIVING STS-3c SPE DATA FROM THE TRANSMIT STS-3c POH PROCESSOR BLOCK

The Transmit STS-3 TOH Processor block receives the STS-3c SPE data (with newly derived POH bytes) from the Transmit STS-3c POH Processor block.

As the Transmit STS-3 TOH Processor block receives the STS-3c SPE data from the Transmit STS-3c POH Processor block, it will perform the following functions.

- Compute and insert the B1 byte
- Compute and insert the B2 byte
- To source (per user configuration) the K1, K2, E1, M0, J0, F1 and S1 bytes.
- To (automatically or upon software command) transmit the RDI-L indicator
- To (automatically or upon software command) transmit the REI-L indicator
- To (upon software command) transmit the AIS-L indicator
- To (optionally) scramble the data, prior to transmission to the Transmit PECL Interface or the Transmit STS-3 Telecom Bus Interface blocks.

Each of these functions is described in considerable detail below.

2.2.9.2 COMPUTATION AND INSERTION OF THE SECTION BIP-8 (B1) BYTE

The Transmit STS-3 TOH Processor block computes a BIP-8 value over an entire STS-3 frame. The results of this calculation are inserted into the B1 byte-position within the very next STS-3 frame. The Remote LTE (Line Terminating Equipment) will use this byte, in order to perform error-checking/detection on the incoming STS-3 frames that it receives.

TRANSMISSION OF ERRED B1 BYTES IN THE OUTBOUND STS-3 DATA-STREAM

The Transmit STS-3 TOH Processor block permits the user to insert errors into the B1 bytes, within the outbound STS-3 data-stream.

The user can accomplish this by writing a "non-zero" value into the "Transmit STS-3 Transport – Transmit B1 Byte Error Mask" register. The Transmit STS-3 TOH Processor block will perform an XOR operation with the contents of "outbound" B1 byte value and the contents of this register. The results of this calculation are loaded back into the B1 byte position, within the "outbound" STS-3 data-stream.

The bit-format of the "Transmit STS-3 Transport – Transmit B1 Byte Error Mask" Register is presented below.

Transmit STS-3 Transport – Transmit B1 Byte Error Mask Register (Address = 0x1923)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
B1_Byte_Error_Mask[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *For normal (e.g., un-erred) operations, the user should ensure that this register is set to "0x00" (the default value).*

2.2.9.3 COMPUTATION AND INSERTION OF THE LINE BIP-8 (B2) BYTE

The Transmit STS-3 TOH Processor block also computes the BIP-8 value over the Line Overhead bytes, as well as the Envelope Capacity. The results of this calculation are inserted into the B2 byte-position within the very next STS-3 frame. The Remote Terminal Equipment will use to this byte, in order to perform error-checking/detection in the incoming LOH and Envelope Capacity data that it receives.

TRANSMISSION OF ERRED B2 BYTES IN THE OUTBOUND STS-3c DATA-STREAM

The Transmit STS-3 TOH Processor block permits the user to deliberately insert errors into the B2 bytes, within the outbound STS-3 data-stream.

The user can accomplish this by writing a "non-zero" value into the "Transmit STS-3 Transport – Transmit B2 Bit Error Mask" register. The Transmit STS-3 TOH Processor block will perform an XOR operation with the contents of the "outbound" B2 byte value and the contents of this register. The results of this calculation are loaded back into the B2 byte position(s), within the "outbound" STS-3 data-stream.

The bit-format of the "Transmit STS-3 Transport – Transmit B2 Bit Error Mask" Register is presented below.

Transmit STS-3 Transport – Transmit B2 Bit Error Mask Register (Address = 0x192B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_B2_Error_Mask[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *For normal (e.g., un-erred) operations, the user should ensure that this register is set to "0x00" (the default value).*

2.2.9.4 TRANSMISSION OF ERRED FRAMING ALIGNMENT BYTES IN THE OUTBOUND STS-3 DATA-STREAM

To be provided in the next revision of the Data Sheet.

2.2.9.5 TRANSMISSION OF ALARM CONDITIONS

The Transmit STS-3 TOH Processor Block supports the following functions.

- Transmitting the LOS (Loss of Signal) pattern (under Software control)
- Transmitting the AIS-L Pattern (under Software control)
- Transmitting the RDI-L Indicator (automatically and under Software control)
- Transmitting the REI-L Indicator (automatically and under Software control)

Each of these functions is discussed in some detail below.

2.2.9.5.1 TRANSMISSION OF THE LOS CONDITION

The Transmit STS-3 TOH Processor block permits the user to transmit the LOS pattern to the remote terminal equipment. The user can accomplish this by setting Bit 3 (LOS Force) within the Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0, as illustrated below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

Once the user executes this step, then the Transmit STS-3 TOH block will override all of the outbound data, with an “All Zeros” pattern.

Note: *When this bit-field is set, it overrides all of the other bits in this register.*

2.2.9.5.2 TRANSMISSION OF AIS-L INDICATOR

The Transmit STS-3 TOH Processor block permits the user to transmit the AIS-L (Line – Alarm Indication Signal) indicator to the remote terminal equipment, under software control.

Forced Transmission of the AIS-L Indicator

The user can command the Transmit STS-3 TOH Processor block to transmit the AIS-L indicator by setting Bit 4 (AIS-L Force) within the Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0, to “1” as illustrated below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once the user executes this step, then the Transmit STS-3 TOH Processor Block will overwrite all of the Line Overhead, and STS-3 Envelope Capacity bytes, with an “All Ones” pattern. Only the Section Overhead bytes will be unaffected by this register bit setting.

Note: *This bit-field is ignored when the “LOS Force” bit-field is set to “1”.*

2.2.9.5.3 TRANSMISSION OF THE RDI-L INDICATOR

The Transmit STS-3 TOH Processor block provides the user with two options associated with transmitting the Line-Remote Defect Indicator (RDI-L).

- Forced transmission of the RDI-L indicator (e.g., under Software control).
- Automatic transmission of the RDI-L indicator

Each of these options is discussed below.

2.2.9.5.3.1 Forced Transmission of the RDI-L Indicator (under Software Control)

The XRT94L33 permits the user to force the Transmit STS-3 TOH Processor block to transmit the RDI-L indicator to the remote terminal equipment. The user can accomplish this by setting Bit 5 (RDI-L Force), within the Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0, to "1" as illustrated below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

If the user opts to transmit an RDI-L indicator, then bits 6, 7 and 8 (of the K2 byte, within each outbound STS-3c frame) will be set to the value [1, 1, 0].

Note: *This bit-field is ignored when either the "LOS Force" or the "AIS-L Force" bit-fields are set to "1".*

In this case, if the user wishes to terminate the Transmit STS-3c TOH Processor Block's transmission of the RDI-L indicator, then they only need to set Bit 5 back to "0".

2.2.9.5.3.2 Automatic Transmission of the RDI-L (Line – Remote Defect Indicator)

The Transmit STS-3 TOH Processor block permits the user to configure it to automatically transmit the RDI-L indicator, in response to the following conditions.

- The corresponding Receive STS-3 TOH Processor block declares an LOS (Loss of Signal) condition.
- The corresponding Receive STS-3 TOH Processor block is declaring an LOF (Loss of Frame) condition.
- The corresponding Receive STS-3 TOH Processor block is declaring the AIS-L condition.

Bits 0 through 2, within the Transmit STS-3 Transport – RDI-L Control Register, permits the user configure the Transmit STS-3 TOH Processor block to support the "Automatic Transmission of an RDI-L indicator upon detection of these conditions. These bit-fields are highlighted and defined below.

Transmit STS-3 Transport – RDI-L Control Register (Address = 0x1933)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				External RDI-L Enable	Transmit RDI-L upon AIS	Transmit RDI-L upon LOF	Transmit RDI-L upon LOS
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	1	1	1

BIT NUMBER	NAME	TYPE	DESCRIPTION
2	Transmit RDI-L upon AIS	R/W	**Transmit Line Remote Defect Indicator (RDI-L) upon Detection of AIS-L:** This READ/WRITE bit-field permits the user to configure the STS-3 Transmitter to automatically transmit a RDI-L indicator to the remote terminal anytime (and for the duration) that the corresponding STS-3 Receiver is detecting a Line AIS (AIS-L) indicator. 0 – Configures the STS-3 Transmitter to NOT automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver detecting the AIS-L indicator. 1 – Configures the STS-3 Transmitter to automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver detecting the AIS-L indicator.
1	Transmit RDI-L upon LOF	R/W	**Transmit Line Remote Defect Indicator (RDI-L) upon Detection of LOF:** This READ/WRITE bit-field permits the user to configure the STS-3 Transmitter to automatically transmit a RDI-L indicator to the remote terminal anytime (and for the duration) that the corresponding STS-3 Receiver is declaring the LOF defect. 0 – Configures the STS-3 Transmitter to NOT automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver declaring the LOF defect. 1 – Configures the STS-3 Transmitter to automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver declaring the LOF defect.
0	Transmit RDI-L upon LOS	R/W	**Transmit Line Remote Defect Indicator (RDI-L) upon Detection of LOS:** This READ/WRITE bit-field permits the user to configure the STS-3 Transmitter to automatically transmit a RDI-L indicator to the remote terminal anytime (and for the duration) that the corresponding STS-3 Receiver is declaring the LOS defect. 0 – Configures the STS-3 Transmitter to NOT automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver declaring the LOS defect. 1 – Configures the STS-3 Transmitter to automatically transmit the RDI-L indicator, upon the corresponding STS-3 Receiver declaring the LOS defect.

How Automatic Transmission of the RDI-L Indicator Works

Consider that the Transmit STS-3 TOH Processor block (within a given channel) has been configured to automatically transmit the RDI-L indicator whenever the corresponding Receive STS-3 TOH Processor block detects an LOS condition. Figure 59 presents a simple illustration of the Transmit STS-3 TOH Processor and Receive STS-3 TOH Processor block pair. Further, this figure indicates that the Receive STS-3 TOH Processor block is not declaring the LOS condition.

Figure 59 Illustration of a Transmit STS-3 TOH Processor and Receive STS-3 TOH Processor block pair, when LOS is not being declared.

As a consequence, the Transmit STS-3 TOH Processor block will, in-turn, not transmit the RDI-L indicator (by setting bits 6, 7 and 8, of the K2 byte, to a value other than [1, 1, 0]).

Figure 57 presents a different illustration of the Transmit STS-3 TOH Processor and Receive STS-3 TOH Processor blocks. In this case, the Receive STS-3 TOH Processor block is now declaring the LOS condition. As a consequence, the Transmit STS-3 TOH Processor block will, in-turn, transmit the RDI-L indicator, by setting bits 6, 7 and 8 (of the K2 byte), within the outbound STS-3 signal to the value [1, 1, 0].

Figure 60 Illustration of the Transmit STS-3 TOH Processor block transmitting the RDI-L indicator to the Remote Terminal coincident with the Receive STS-3 TOH Processor block declaring an LOS condition.

2.2.9.5.4 TRANSMISSION OF THE REI-L INDICATOR

The Transmit STS-3 TOH Processor block can be configured to transmit the REI-L indicator either (1) automatically, (2) upon software command, or (3) via the TxPOH External Input Port, as described below.

2.2.9.5.4.1 Configuring the Transmit STS-3 TOH Processor Block to automatically transmit the REI-L Indicator

The user can configure the Transmit STS-3 TOH Processor block to automatically transmit the REI-L indicator whenever the corresponding Receive STS-3c TOH Processor block detects at least one B2 byte error.

The user can configure the Transmit STS-3 TOH Processor block to automatically transmit the REI-L indicator, in response to detection of B2 byte errors, by executing the following steps.

STEP 1 – Write the value "0" into Bit 0 (M0M1 Insert Method[1]) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 1"; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Write the value "0" into Bit 7 (M0M1 Insert Method[0]) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0"; as depicted below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Steps 1 and 2 configure the Transmit STS-3 TOH Processor block to automatically set bits 5 through 8 (of the M0 Byte) within the "outbound" STS-3 data-stream, to the appropriate REI-L value based upon receive conditions as detected by the corresponding "Receive STS-3 TOH Processor" block.

STEP 3 – Indicate whether or not the REI-L value (transmitted to the remote LTE) reflects the number of bits (within the B2 byte) that are in error, or the number of erred STS-3 frames that have been detected by the corresponding Receive STS-3 TOH Processor block.

The XRT94L33 permits the user to (1) flag B2 byte errors, and (2) to transmit the resulting REI-L values (back to the remote LTE) by the following means.

- By flagging and reporting the number of bits (within the B2 byte(s)) that have been determined to be in error, within a given STS-3 frame.
- By flagging and reporting whether or not the corresponding Receive STS-3 TOH Processor block is currently receiving an erred STS-3 frame.

The user can choose between these two options by writing the appropriate value into Bit 1 (B2 Error Type) within the "Receive STS-3 Transport – Control Register – Byte 0", as depicted below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	X	0

Setting this bit-field to "0" configures the "Transmit STS-3 TOH Processor" block to set the REI-L bit-fields (within the M0 byte) of the "outbound" STS-3 frame to a value that reflects the number of bits (within the B2 byte) that were determined to be in error (within the most recently received STS-3 frame) by the corresponding Receive STS-3 TOH Processor block. In this case, the REI-L bit-fields can range in value from "0" (for no B2 byte errors) to "8" (for all B2 bits being in error).

Setting this bit-field to "1" configures the "Transmit STS-3 TOH Processor" block to set the REI-L bit-fields (within the M0 byte) of the "outbound" STS-3 frame; to a value that indicates whether or not at least one B2 bit error was detected within the most recently received STS-3 frame; by the corresponding Receive STS-3 TOH Processor block. In this mode, the Transmit STS-3 TOH Processor block will set the "REI-L" bit-fields (within the "outbound" STS-3 frame) to "0" if the corresponding Receive STS-3 TOH Processor block is not detecting any B2 byte errors, in its incoming STS-3 data-stream.

Conversely, the Transmit STS-3 TOH Processor block will set the "REI-L" bit-fields (within the "outbound" STS-3 frame) to "1" if the corresponding Receive STS-3 TOH Processor block receives an STS-3 frame that contains a B2 byte error.

2.2.9.5.4.2 Configuring the Transmit STS-3 TOH Processor block to transmit the REI-L indicator, upon Software Command

The user can configure the Transmit STS-3 TOH Processor block to transmit the REI-L indicator (per software command) by executing the following steps.

STEP 1 – Write the value "0" into Bit 0 (M0M1 Insert Method[1]) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 1" as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 – Write the value "1" into Bit 7 (M0M1 Insert Method[0]) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0" as depicted below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	0	0	0

Steps 1 and 2 configure the Transmit STS-3 TOH Processor block to automatically read out the contents of Bit 0 through 3 (of the Transmit STS-3 Transport – M0M1 Byte Value Register) and write the value of these bits into Bits 5 through 8 (of the M0M1 byte) within the "outbound" STS-3 frame. The bit-format of the "Transmit STS-3 Transport – Transmit M0M1 Byte Value" Register (with the appropriate bits "shaded") is presented below.

Transmit STS-3 Transport – M0M1 Value Register (Address = 0x1937)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_M0M1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

In this mode, the user can transmit an "un-erred" REI-L value by setting Bits 0 through 3 (of the Transmit STS-3 Transport – Transmit M0M1 Byte Value Register) to "[0, 0, 0, 0]". Conversely, the user can transmit an "erred" REI-L value by setting bits 0 through 3 (within the Transmit STS-3 Transport – Transmit M0M1 Byte Value Register) to some value between 1 and 8.

AUTOMATIC TRANSMISSION OF THE REI-L INDICATOR

The Transmit STS-3 TOH Processor block will transmit the Line – Remote Error Indicator to the remote terminal equipment. In an STS-3 frame, the REI-L indicator will be transmitted within the M0 byte. The bit-format of the M0 byte is presented below in

Figure 61.

Figure 61 Bit format of the M0 Byte

BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8
Undefined				B2 Error Count (REI-L)			

Figure 61 indicates that Bits 5 through 8, within the M0 byte are allocated for the REI-L function.

The purpose of the REI-L function, within the M0 is two-fold.

1. It permits a piece of Line Terminating Equipment (LTE), which is transmitting STS-3 data to a remote Line Terminating Equipment, to determine whether or not the remote LTE is receiving its Line Overhead and STS-3c Envelope capacity bytes, in an error-free manner.
2. It permits an LTE (which is receiving STS-3c data from a remote LTE) to inform this remote terminal when it is receiving "erred" Line Overhead and STS-3 Envelope capacity bytes; within the incoming STS-3 frames.

The role of REI-L is best presented in the practical example below.

Example:

Consider a "Near-End" LTE that is communicating with a remote LTE. This "Near-End" LTE consists of the Receive STS-3 TOH Processor block and the Transmit STS-3 TOH Processor block, within a given channel in the XRT94L33 ATM UNI/PPP/Mapper IC; as depicted below in

Figure 62 and Figure 63.

The “Transmit STS-3 TOH Processor” block will transmit STS-3 frames to the remote LTE. Likewise, the “Receive STS-3 TOH Processor” block will receive and process STS-3 frames, originating from the remote LTE. The “Near-End” Receive STS-3 TOH Processor” block is going to verify the values of the B2 bytes, within the incoming STS-3 frames (from the remote LTE). If the “Near-End” Receive STS-3 TOH Processor” block detects no B2 byte errors, in the incoming STS-3 frame, then it will notify the remote LTE of this fact, by having the “Near-End” Transmit STS-3 TOH Processor” block set the “REI-L” bits, within the M0 byte (in the “outbound” STS-3 frame) to the value “0”. This phenomenon is illustrated in

Figure 62 and Figure 63 below.

Figure 62 illustrates the “Near-End” Receive STS-3 TOH Processor block receiving an “error-free” STS-3 frame. In this figure, the locally computed B2 value of “0x5A” matches that received from the Remote LTE.

Figure 63 illustrates the subsequent action of the “Near-End” Transmit STS-3 TOH Processor block, which will transmit an STS-3 frame, with the REI-L bit-fields (within the M0 byte) set to “0”; to the remote LTE. This signaling indicates that the “Near-End” LTE has received an error-free Line Overhead and STS-3 Envelope capacity data in the most recently received STS-3 frame, from the remote LTE.

Figure 62 Illustration of the "Near-End" Receive STS-3 TOH Processor block receiving a STS-3 frame (from the remote LTE) with the correct B2 byte value

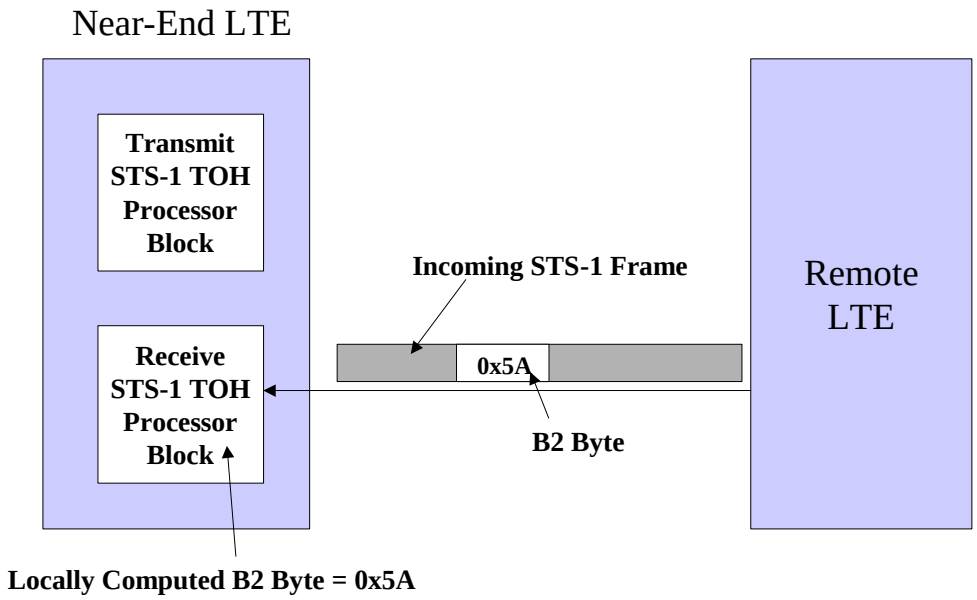
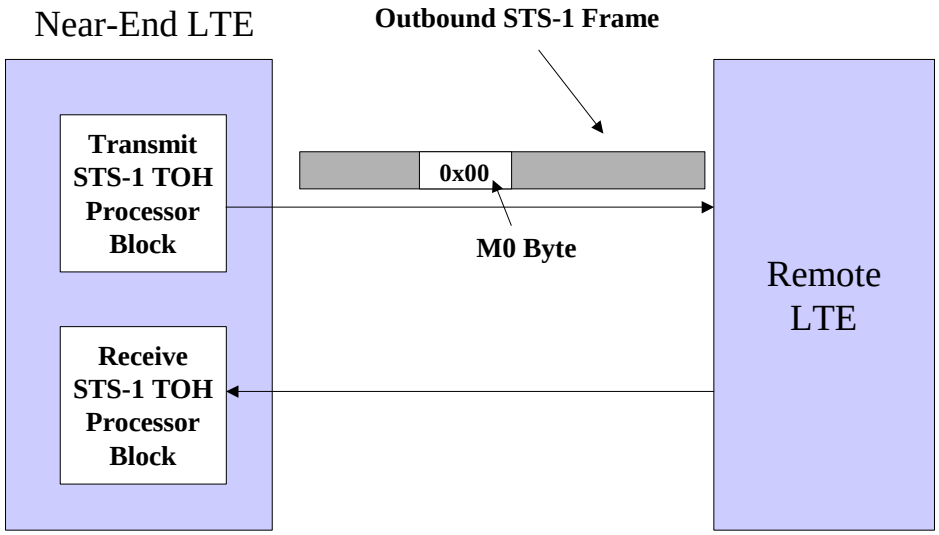


Figure 63 Illustration of the "Near-End" Transmit STS-3 TOH Processor block transmitting an STS-3 frame (to the remote LTE) with the REI-L value set to "0"



Conversely, if the "Near-End" Receive STS-3 TOH Processor block detects an error in the incoming B2 byte, then it will notify the remote LTE of this fact, by having the "Near-End" Transmit STS-3 TOH Processor block set the "REI-L" bits, within the "outbound" STS-3 frame (which is destined for the remote LTE) to the value, reflecting the total number of bit-errors that have been detected within the B2 byte.

Figure 64 illustrates the "Near-End" Receive STS-3 TOH Processor block receiving an erred STS-3 frame from the remote LTE. In this figure, the "Near-End" Receive STS-3 TOH Processor block is receiving an STS-3 frame, with a B2 byte value of "0x5F". This value does not match the "locally computed" B2 byte value of "0x5A". Consequently, there is an error in the Line Overhead and STS-3c Envelope Capacity bytes, within this STS-3 frame. Now, since this B2-byte mismatch is due to two bits being in error, the REI-L bits will be set to the value "2".

Figure 64 Illustration of the "Near-End" Receive STS-3 TOH Processor block receiving an STS-3 frame (from the remote LTE) with an incorrect B2 byte

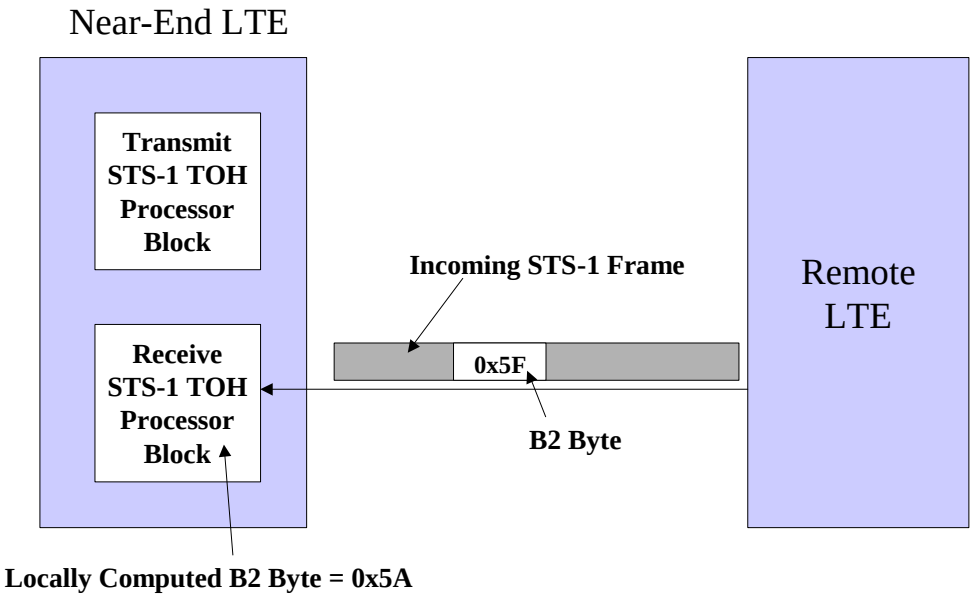
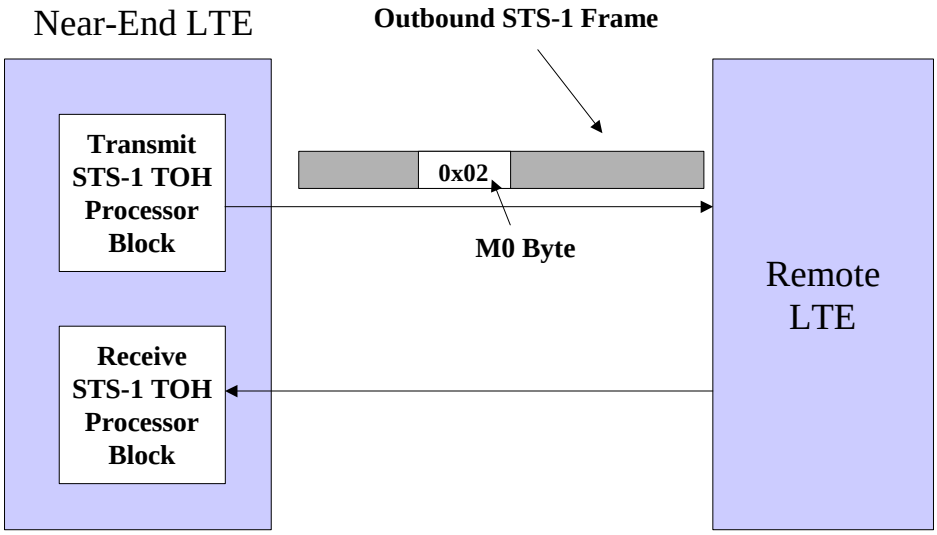


Figure 65 Illustration of the "Near-End" Transmit STS-3 TOH Processor block, transmitting an STS-3 frame (to the remote LTE) with the REI-L bits being set to "2"



For information on how the Receive STS-3 TOH Processor block processes the REI-L bit-fields, within each incoming STS-3 frame, please see Section _.

2.2.9.5.5 TRANSMISSION OF SECTION TRACE MESSAGES VIA THE J0 BYTE

The Transmit STS-3 TOH Processor block permits the user to transmit Section Trace Messages to the remote LTE via the J0 byte. The Transmit STS-3 TOH Processor block permits the user to accomplish this by either of the following options.

- Automatically set the J0 byte (within each outbound STS-3 Frame) to "0x01"
- Set and control the outbound J0 byte via on-chip register
- Set and control the outbound J0 byte via external input pin
- Use the "Transmit Section Trace Message" buffer

The details and instructions for using either of these features are presented below.

2.2.9.5.5.1 Automatically setting the J0 byte (within each outbound STS-3 frame) to "0x01"

The XRT94L33 permits the user to configure each of the four (4) Transmit STS-3 TOH Processor blocks to automatically set the contents of the J0 byte (within each outbound STS-3 frame) to "0x01".

The user can accomplish this by writing the value "[0, 0]" into Bits 1 and 0 (J0 Type[1:0]) within the appropriate "Transmit STS-3 Transport – Transmit J0 Byte Control" Register; as depicted below.

Transmit STS-3 Transport – Transmit J0 Byte Control Register (Address = 0x194F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J0 Message Length[1:0]		J0 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	0

Once the user executes this step, then the corresponding Transmit STS-3 TOH Processor block(s) will be configured to automatically set the J0 byte (within each outbound STS-3 frame) to the value "0x01".

2.2.9.5.5.2 Transmitting 16-byte/64-byte Section Trace Messages using the "Transmit Section Trace Message" Buffer

The XRT94L33 contains a total of four (4) "Transmit Section Trace Message" Buffers (one for each "Transmit STS-3 POH Processor block, within the device). The address location of the "Transmit Section Trace Message" buffer for each of the four Transmit STS-3 TOH Processor blocks is presented below.

Table 17: Address Locations of the Transmit Section Trace Message Buffers within the XRT94L33

TRANSMIT STS-3 TOH PROCESSOR BLOCK - CHANNEL	ADDRESS LOCATION OF THE TRANSMIT SECTION TRACE MESSAGE BUFFER
0	0x1B00 – 0x1B3F
1	0x2B00 – 0x2B3F
2	0x3B00 – 0x3B3F
3	0x4B00 – 0x4B3F

If the user wishes to use the "Section Trace Message" buffer as the means to load and transmit the "Section Trace Message" to the remote STE, then they must execute the following steps.

STEP 1 – Write the contents of this outbound "Section Trace Message" into the "Transmit Section Trace Message" Buffer.

As the user writes the contents of the "outbound" Section Trace Message" into the "Transmit Section Trace" Message buffer they must make sure that the first byte of the message is written into Address location 0xNB00 (e.g., the very first address location of the "Transmit Section Trace Message" buffer).

STEP 2 – Specify the length of this "Section Trace Message" by writing the appropriate value into Bits 3 and 2 (J0 Message Length[1:0]) within the "Transmit STS-3 Transport – Transmit J0 Byte Control" Register; as depicted below.

Transmit STS-3 Transport – Transmit J0 Byte Control Register (Address = 0x194F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J0 Message Length[1:0]		J0 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	X

The relationship between the contents of "J0_Message_Length[1:0]" and the corresponding length of the "Section Trace Message" is presented below in Table _.

Table 18: The Relationship Between the contents of the "J0_Message_Length[1:0]" and the corresponding "Section Trace Message" Length

J0_MESSAGE_LENGTH[1:0]	SECTION TRACE MESSAGE LENGTH (BYTES)
00	1
01	16
10	64
11	64

STEP 3 – Write the value "[0, 1]" into Bits 1 and 0 (J0_Type[1:0]) within the Transmit STS-3 Transport – Transmit J0 Control Register; as depicted below.

Transmit STS-3 Transport – Transmit J0 Byte Control Register (Address = 0x194F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J0 Message Length[1:0]		J0 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	0	1

This step configures the Transmit STS-3 TOH Processor block to use the contents of the "Transmit Section Trace Message" buffer as the source of the J0 byte, within each "outbound" STS-3 Frame.

2.2.9.5.5.3 Setting and Controlling the "Outbound" J0 Byte via On-Chip Register

The Transmit STS-3 TOH Processor block permits the user to specify the contents of the J0 byte within the "outbound" STS-3 frame, via software command. The user can configure the Transmit STS-3 TOH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 0]" into Bits 1 and 0 (J0 Type[1:0]) within the "Transmit STS-3 Transport – Transmit J0 Byte Control" Register, as depicted below.

Transmit STS-3 Transport – Transmit J0 Byte Control Register (Direct Address = 0xN84F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J0 Message Length[1:0]		J0 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	1	0

This step configures the "Transmit STS-3 TOH Processor" block to read out the contents of the "Transmit Transport – Transmit J0 Byte Value" register; and load this value into the "J0 byte-field" within each outbound STS-3 Frame.

STEP 2 – Write the desired byte value (for the "outbound" J0 byte) into the "Transmit STS-3 Transport – Transmit J0 Byte Value" register.

The bit-format of this register is presented below.

Transmit STS-3 Transport – Transmitter J0 Value Register (Address = 0x194B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_J0_Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

2.2.9.5.5.4 Setting and Controlling the "Outbound" J0 Byte via External Input pin

The Transmit STS-3 TOH Processor block permits the user to source the contents of the J0 byte via the "TxTOH_n" input port. The user can configure the Transmit STS-3 TOH Processor block to support this feature by performing the following steps.

STEP 1 – Write the value "[1, 1]" into Bits 1 and 0 (J0 Type[1:0]) within the "Transmit STS-3 Transport – Transmit J0 Byte Control" Register; as depicted below.

Transmit STS-3 Transport – Transmit J0 Control Register (Address = 0x194F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				J0 Message Length[1:0]		J0 Type[1:0]	
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	1	1

This step configures the "Transmit STS-3 TOH Processor" block to accept the value of the J0 byte, via the "TxTOH" input port and load this value into the J0 byte position within each outbound STS-3 frame.

STEP 2 – Begin providing the values of the "outbound" J0 Byte message to the "TxTOH" input port.

The procedure for applying the J0 byte to the "TxTOH" input port is presented below.

Using the "TxTOH" Input Port to insert the J0 byte value into the outbound STS-3 frame

If the user intends to externally insert the J0 byte into the outbound STS-3 frame, the "TxTOH" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do the following.

- Continuously sample the "TxTOHEnable_n" and the "TxTOHFrame_n" output pins upon the rising edge of the "TxTOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxTOH Input Port" is presented below in Figure 66.

Figure 66 A Simple Illustration of the "External Circuit" being interfaced to the "TxTOH Input Port"

Note: *The "TxTOHIns_n" line (in Figure 66) is "dashed" because controlling this signal is not necessary if the user has executed "STEP 1" above.*

- Whenever the "external circuit" samples both the "TxTOHEnable_n" and the "TxTOHFrame_n" output pins "high", then it should enter a WAIT STATE (e.g., where it will wait for 16 periods of "TxTOHClk_n" to elapse). Afterwards, the external circuitry should exit this "WAIT STATE" and then place the very first byte (e.g., the most significant bit) of the "outbound" J0 byte onto the "TxTOH_n" input pin, upon the very next falling edge of "TxTOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3 TOH Processor" block circuitry, upon the very next rising edge of "TxTOHClk_n.

Note: *This "WAIT STATE" period is necessary because the J0 byte is the third byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the J0 byte) onto the "TxTOH_n" input pin, upon each of the next seven falling edge of "TxTOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxTOHEnable_n" and "TxTOHFrame_n" output pins and repeat the above-mentioned process.

Figure 67 presents an illustration of the "TxTOH Input Interface" waveforms, when the "external circuit" is writing the J0 byte into the "TxTOH Input Port".

Figure 67 Illustration of the "TxTOH Input Interface" waveforms, when the "External Circuit" is writing the J0 byte into the "TxTOH Input Port"

2.2.9.5.6 SUPPORT/HANDLING OF THE E1 BYTE

The Transmit STS-3c TOH Processor block permits the user to control the value of the E1 byte which is to be transmitted via the "outbound" STS-3c data-stream by either or the following options.

- Setting and controlling the E1 byte via software
- Setting and controlling the E1 byte via the "TxPOH Input Port"

The details and instructions for using either of these features are presented below.

2.2.9.5.6.1 Setting and Controlling the outbound E1 byte via Software

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the E1 byte within the outbound STS-3c data-stream via software command.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (E1 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1, as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	1	X	X	X	X

The step configures the Transmit STS-3c TOH Processor block to read out the contents of the "Transmit STS-3c Transport – E2 Byte Value Register, and load this value into the E1 byte position within each outbound STS-3c data-stream.

STEP 2 – Write the desired byte values (for the outbound E1 byte) into the "Transmit STS-3c Transport – E1 Byte Value Register.

The bit-format for this register is presented below.

Transmit STS-3 Transport – Transmit E1 Byte Value Register (Address = 0x1943)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_E1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.9.5.6.2 Setting and Controlling the outbound E1 byte via "TxPOH Input Port"

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the E1 byte within the "outbound" STS-3c data-stream via the data applied to the "TxPOH_n" input port.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (STS-3c TOH Insert) within the "Mapper Control Register – Byte 2", as depicted below.

Mapper Control Register – Byte 2 (Direct Address = 0xN601)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
STS-3c OH Pass Thru	STS-3c Remote Loop-back	STS-3c Local Loop-back	STS-3c TOH Insert	Loop-Timing	POH Pass Thru	Receive (Ingress) STS-3c Enable	Transmit (Egress) STS-3c Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit STS-3c TOH Processor" block (associated with Channel "N") to accept its TOH bytes via the "TxPOH Input" port.

STEP 2 – Write the value "0" into Bit 4 (E1 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	0	X	X	X	X

This step configures the Transmit STS-3c TOH Processor block to use the "TxPOH Input" port as the source for the E1 byte, within each outbound STS-3c frame. In this mode, the Transmit STS-3c TOH Processor block will accept the value, corresponding to the E1 byte (via the "TxPOH_n" input port) and it will write this data into the E1 byte position, within the "outbound" STS-3c frame.

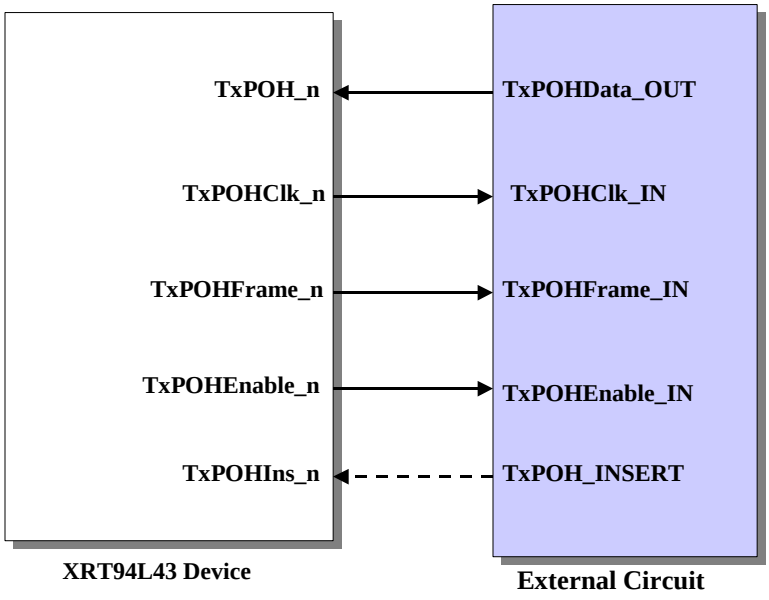
Using the "TxPOH_n Input" port to insert the E1 byte value into the "outbound" STS-3c data-stream

If the user intends to externally insert the E1 byte into the outbound STS-3c data-stream, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 68.

Figure 68 A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 68) is "dashed" because this controlling this signal is not necessary if the user has executed "STEP 2" above.*

- Whenever the "external circuit" samples the "TxPOHFrame_n" output pin "high" and also the "TxPOHEnable_n" pin "low", then it should enter a "WAIT STATE" (e.g., where it waits for __ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" E1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the E1 byte is the _ byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the E1 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".
- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 69 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the E1 byte into the "TxPOH Input Port".

Figure 69 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "E1 byte" into the "TxPOH Input Port"

2.2.9.5.7 SUPPORT/HANDLING OF THE F1 BYTE

The Transmit STS-3c TOH Processor block permits the user to control the value of the F1 byte, which is to be transmitted via the "outbound" STS-3c data-stream by either or the following options.

- Setting and controlling the F1 byte via software
- Setting and controlling the F1 byte via the "TxPOH Input Port"

The details and instructions for using either of these features are presented below.

2.2.9.5.7.1 Setting and Controlling the outbound F1 byte via Software

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the E1 byte within the outbound STS-3c data-stream via software command.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 3 (F1 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1, as depicted below.

Transmit STS-3cTransport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	0	1	X	X	X

The step configures the Transmit STS-3c TOH Processor block to read out the contents of the "Transmit STS-3c Transport – F1 Byte Value Register, and load this value into the F1 byte position within each outbound STS-3c data-stream.

STEP 2 – Write the desired byte values (for the outbound F1 byte) into the "Transmit STS-3c Transport – F1 Byte Value Register.

The bit-format for this register is presented below.

Transmit STS-3c Transport – Transmit F1 Byte Value Register (Address = 0x1943)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_F1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.9.5.7.2 Setting and Controlling the outbound F1 byte via "TxPOH Input Port"

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the E1 byte within the "outbound" STS-3c data-stream via the data applied to the "TxPOH_n" input port.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (STS-3c TOH Insert) within the "Mapper Control Register – Byte 2", as depicted below.

Mapper Control Register – Byte 2 (Direct Address = 0xN601)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
STS-3 OH Pass Thru	STS-3 Remote Loop-back	STS-3 Local Loop-back	STS-3 TOH Insert	Loop-Timing	POH Pass Thru	Unused	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit STS-3c TOH Processor" block (associated with Channel "N") to accept its TOH bytes via the "TxPOH Input" port.

STEP 2 – Write the value "0" into Bit 4 (F1 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	X	0	X	X	X

This step configures the Transmit STS-3c TOH Processor block to use the "TxPOH Input" port as the source for the F1 byte, within each outbound STS-3c frame. In this mode, the Transmit STS-3 TOH Processor block will accept the value, corresponding to the F1 byte (via the "TxPOH_n" input port) and it will write this data into the F1 byte position, within the "outbound" STS-3c frame.

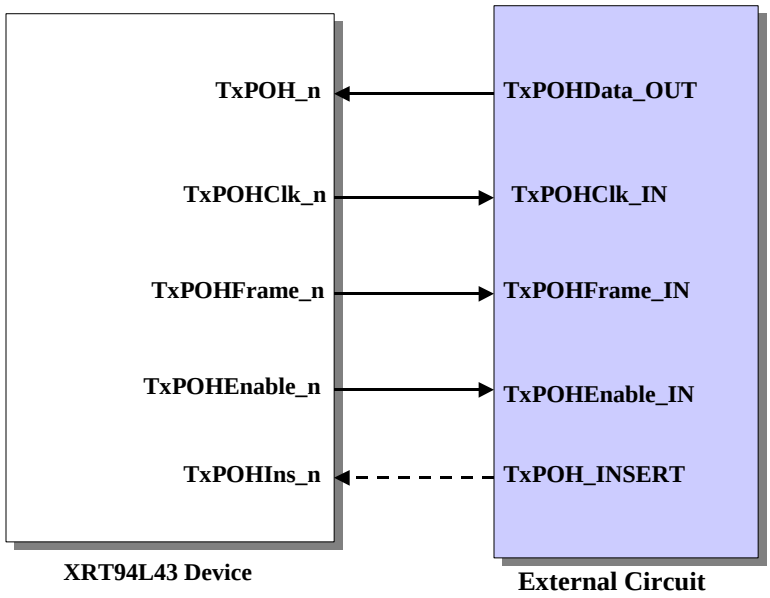
Using the "TxPOH_n Input" port to insert the F1 byte value into the "outbound" STS-3c data-stream

If the user intends to externally insert the F1 byte into the outbound STS-3c data-stream, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 70.

Figure 70 A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in Figure 70) is "dashed" because this controlling this signal is not necessary if the user has executed "STEP 2" above*

- Whenever the "external circuit" samples the "TxPOHFrame_n" output pin "high" and also the "TxPOHEnable_n" pin "low", then it should enter a "WAIT STATE" (e.g., where it waits for __ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" F1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the F1 byte is the _ byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the F1 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 71 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the F1 byte into the "TxPOH Input Port".

Figure 71 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "F1 byte" into the "TxPOH Input Port"

2.2.9.5.8 SUPPORT/HANDLING OF THE SECTION DCC (D1, D2 AND D3) BYTES

2.2.9.5.9 SUPPORT/HANDLING OF THE K1/K2 BYTES

The Transmit STS-3c TOH Processor block permits the user to control the value of the K1 and K2 bytes, which are to be transmitted via the "outbound" STS-3c data-stream, by either or the following options.

- Setting and controlling the K1, K2 bytes via software
- Setting and controlling the K1, K2 bytes via the "TxPOH Input Port"

The details and instructions for using either of these features are presented below.

2.2.9.5.9.1 Setting and Controlling the outbound K1, K2 bytes via Software

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the K1 and K2 bytes within the outbound STS-3c data-stream via software command.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 1 (K1K2 Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1, as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	X	X	X	1	X

The step configures the Transmit STS-3c TOH Processor block to read out the contents of the "Transmit STS-3c Transport – K1K2 (APS) Value Registers (Bytes 0 and 1), and load these values into the K1 and K2 byte positions within each outbound STS-3c data-stream.

STEP 2 – Write the desired byte values (for the outbound K1 and K2 bytes) into the "Transmit STS-3c Transport – K1K2 (APS) Value Registers – (Bytes 0 and 1).

The bit-format for these two registers is presented below.

Transmit STS-3 Transport – K1K2 (APS) Value Register – Byte 1 (Address = 0x192E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_K2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Transmit STS-3 Transport – K1K2 (APS) Value Register – Byte 1 (Address = 0x192F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_K1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.9.5.9.2 Setting and Controlling the outbound K1, K2 bytes via "TxPOH Input Port"

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the K1, K2 bytes within the "outbound" STS-3c data-stream via the data applied to the "TxPOH_n" input port.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (STS-3c TOH Insert) within the "Mapper Control Register – Byte 2", as depicted below.

Mapper Control Register – Byte 2 (Direct Address = 0xN601)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
STS-3c OH Pass Thru	STS-3c Remote Loop-back	STS-3c Local Loop-back	STS-3c TOH Insert	Loop-Timing	POH Pass Thru	Receive (Ingress) STS-3c Enable	Transmit (Egress) STS-3c Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit STS-3c TOH Processor" block (associated with Channel "N") to accept its TOH bytes via the "TxPOH Input" port.

STEP 2 – Write the value "0" into Bit 1 (K1K2 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	X	X	X	0	X

This step configures the Transmit STS-3c TOH Processor block to use the "TxPOH Input" port as the source for the K1, K2 bytes, within each outbound STS-3c frame. In this mode, the Transmit STS-3c TOH Processor block will accept the value, corresponding to the K1 and K2 bytes (via the "TxPOH_n" input port) and it will write this data into the K1 and K2 byte positions, within the "outbound" STS-3c frame.

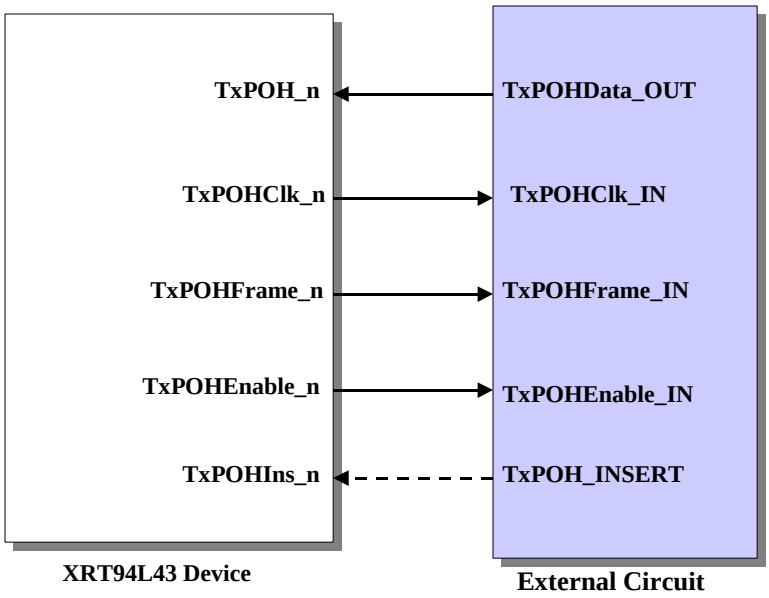
Using the "TxPOH_n Input" port to insert the K1 and K2 byte values into the "outbound" STS-3c data-stream

If the user intends to externally insert the K1 and K2 into the outbound STS-3c data-stream, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 72

Figure 72 A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (in **Error! Reference source not found.**) is "dashed" because this controlling this signal is not necessary if the user has executed "STEP 2" above.*

- Whenever the "external circuit" samples the "TxPOHFrame_n" output pin "high" and also the "TxPOHEnable_n" pin "low", then it should enter a "WAIT STATE" (e.g., where it waits for __ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" K1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the K1 and K2 are the _ and byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining fifteen bits (of the K1 and then K2 bytes) onto the "TxPOH_n" input pin, upon each of the next fifteen falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 73 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the K1 and K2 bytes into the "TxPOH Input Port".

Figure 73 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "K1 and K2 bytes" into the "TxPOH Input Port"

2.2.9.5.10 SUPPORT/HANDLING OF THE LINE DCC (D4 THROUGH D12) BYTE

2.2.9.5.11 SUPPORT/HANDLING OF THE S1 BYTE

The Transmit STS-3c TOH Processor block permits the user to control the value of the S1 byte, which is to be transmitted via the "outbound" STS-3c data-stream, by either or the following options.

- Setting and controlling the S1 byte via software
- Setting and controlling the S1 byte via the "TxPOH Input Port"

The details and instructions for using either of these features are presented below.

2.2.9.5.11.1 Setting and Controlling the outbound S1 byte via Software

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the E1 byte within the outbound STS-3c data-stream via software command.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 2 (S1 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1, as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	X	X	1	X	X

The step configures the Transmit STS-3c TOH Processor block to read out the contents of the "Transmit STS-3c Transport – S1 Byte Value Register, and load this value into the S1 byte position within each outbound STS-3c data-stream.

STEP 2 – Write the desired byte values (for the outbound S1 byte) into the "Transmit STS-3c Transport – S1 Byte Value Register.

The bit-format for this register is presented below.

Transmit STS-3 Transport – Transmit S1 Byte Value Register (Address = 0x193B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_S1_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.9.5.11.2 Setting and Controlling the outbound S1 byte via "TxPOH Input Port"

The Transmit STS-3c TOH Processor block permits the user to specify the contents of the S1 byte within the "outbound" STS-3c data-stream via the data applied to the "TxPOH_n" input port.

The user can configure the Transmit STS-3c TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (STS-3c TOH Insert) within the "Mapper Control Register – Byte 2", as depicted below.

Mapper Control Register – Byte 2 (Direct Address = 0xN601)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
STS-3c OH Pass Thru	STS-3c Remote Loop-back	STS-3c Local Loop-back	STS-3c TOH Insert	Loop-Timing	POH Pass Thru	Receive (Ingress) STS-3c Enable	Transmit (Egress) STS-3c Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit STS-3c TOH Processor" block (associated with Channel "N") to accept its TOH bytes via the "TxPOH Input" port.

STEP 2 – Write the value "0" into Bit 2 (S1 Insert Method) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 1; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Direct Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	X	X	X	0	X	X

This step configures the Transmit STS-3c TOH Processor block to use the "TxPOH Input" port as the source for the S1 byte, within each outbound STS-3c frame. In this mode, the Transmit STS-3c TOH Processor block will accept the value, corresponding to the S1 byte (via the "TxPOH_n" input port) and it will write this data into the S1 byte position, within the "outbound" STS-3c frame.

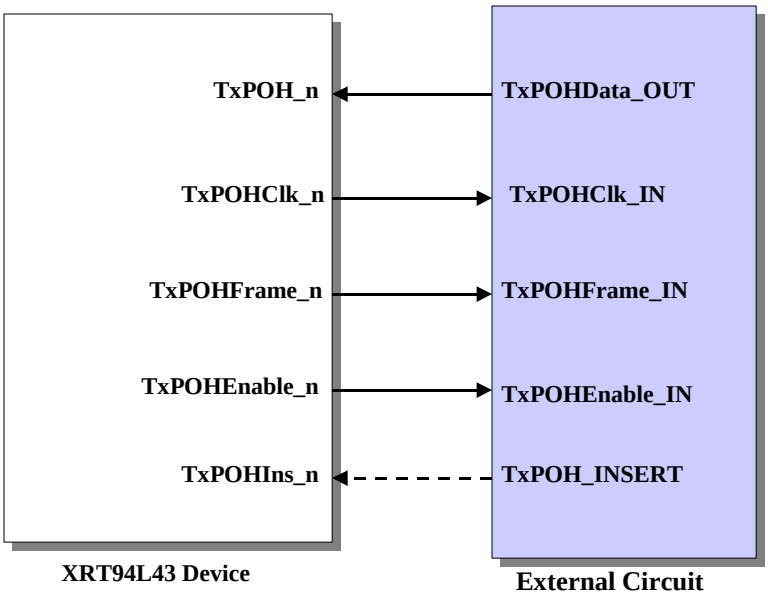
Using the "TxPOH_n Input" port to insert the S1 byte value into the "outbound" STS-3c data-stream

If the user intends to externally insert the S1 byte into the outbound STS-3c data-stream, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 74

Figure 74 A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



The "TxPOHIns_n" line (in

Note: *) is "dashed" because this controlling this signal is not necessary if the user has executed "STEP 2" above.*

- Whenever the "external circuit" samples the "TxPOHFrame_n" output pin "high" and also the "TxPOHEnable_n" pin "low", then it should enter a "WAIT STATE" (e.g., where it waits for __ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" S1 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the S1 byte is the _ byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the S1 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 75 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the S1 byte into the "TxPOH Input Port".

Figure 75 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "S1 byte" into the "TxPOH Input Port"

2.2.9.5.12 SUPPORT/HANDLING OF THE M1 BYTE

2.2.9.5.13 SUPPORT/HANDLING OF THE E2 BYTE

The Transmit STS-3 TOH Processor block permits the user to control the value of the E2 byte, which is to be transmitted via the "outbound" STS-3 data-stream, by either or the following options.

- Setting and controlling the E2 byte via software
- Setting and controlling the E2 byte via the "TxPOH Input Port"

The details and instructions for using either of these features are presented below.

2.2.9.5.13.1 Setting and Controlling the outbound E2 byte via Software

The Transmit STS-3 TOH Processor block permits the user to specify the contents of the E2 byte within the outbound STS-3 data-stream via software command.

The user can configure the Transmit STS-3 TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 5 (E2 Insert Method) within the "Transmit STS-3 Transport – SONET Transmit Control Register – Byte 1, as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	X	X	X	X	X

The step configures the Transmit STS-3 TOH Processor block to read out the contents of the "Transmit STS-3 Transport – E2 Byte Value Register, and load this value into the E2 byte position within each outbound STS-3 data-stream.

STEP 2 – Write the desired byte values (for the outbound E2 byte) into the "Transmit STS-3 Transport – E2 Byte Value Register.

The bit-format for this register is presented below.

Transmit STS-3 Transport – Transmit E2 Byte Value Register (Address = 0x1947)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit_E2_Byte_Value[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.2.9.5.13.2 Setting and Controlling the outbound E2 byte via "TxPOH Input Port"

The Transmit STS-3 TOH Processor block permits the user to specify the contents of the E2 bytes within the "outbound" STS-3 data-stream via the data applied to the "TxPOH_n" input port.

The user can configure the Transmit STS-3 TOH Processor block to accomplish this by performing the following steps.

STEP 1 – Write the value "1" into Bit 4 (STS-3 TOH Insert) within the "Mapper Control Register – Byte 2", as depicted below.

Mapper Control Register – Byte 2 (Direct Address = 0xN601)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
STS-3 OH Pass Thru	STS-3 Remote Loop-back	STS-3 Local Loop-back	STS-3 TOH Insert	Loop-Timing	POH Pass Thru	Receive (Ingress) STS-1 Enable	Transmit (Egress) STS-1 Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This step enables the "Transmit STS-3 TOH Processor" block (associated with Channel "N") to accept its TOH bytes via the "TxPOH Input" port.

STEP 2 – Write the value "0" into Bit 1 (E2 Insert Method) within the "Transmit STS-3c Transport – SONET Transmit Control Register – Byte 1; as depicted below.

Transmit STS-3Transport – SONET Transmit Control Register – Byte 1 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		E2 Insert Method	E1 Insert Method	F1 Insert Method	S1 Insert Method	K1K2 Insert Method	M0M1 Insert Method[1]
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

This step configures the Transmit STS-3 TOH Processor block to use the "TxPOH Input" port as the source for the E2 byte, within each outbound STS-3c frame. In this mode, the Transmit STS-3c TOH Processor block will accept the value, corresponding to the E2 byte (via the "TxPOH_n" input port) and it will write this data into the E2 byte position, within the "outbound" STS-3c frame.

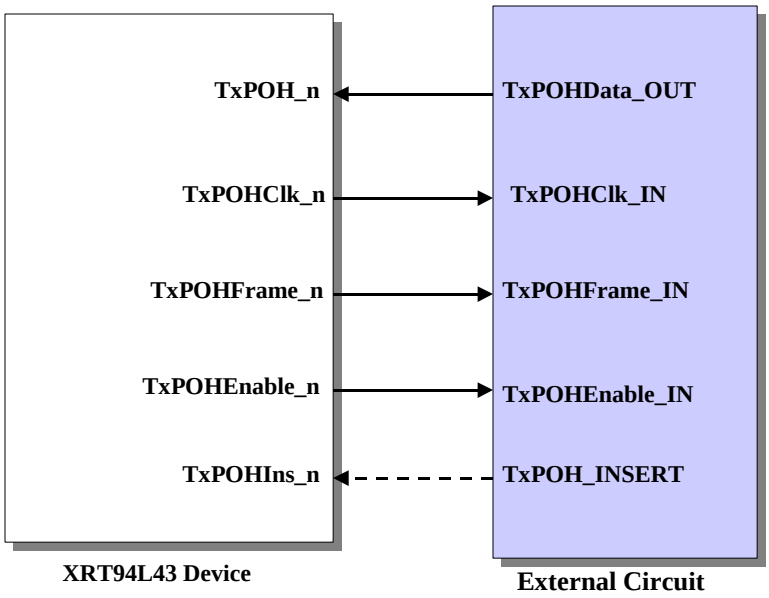
Using the "TxPOH_n Input" port to insert the E2 byte value into the "outbound" STS-3c data-stream

If the user intends to externally insert the E2 byte into the outbound STS-3c data-stream, via the "TxPOH_n" input port, then they must design some external circuitry (which can be realized in an ASIC, FPGA or CPLD solution) to do to the following.

- Continuously sample the "TxPOHEnable_n" and the "TxPOHFrame_n" output pins upon the rising edge of the "TxPOHClk_n" output clock signal.

A simple illustration of this "external circuit" being interfaced to the "TxPOH Input Port" is presented below in Figure 76.

Figure 76 A Simple Illustration of the "External Circuit" being interfaced to the "TxPOH Input Port"



Note: *The "TxPOHIns_n" line (inFigure 76) is "dashed" because this controlling this signal is not necessary if the user has executed "STEP 2" above.*

- Whenever the "external circuit" samples the "TxPOHFrame_n" output pin "high" and also the "TxPOHEnable_n" pin "low", then it should enter a "WAIT STATE" (e.g., where it waits for __ periods of "TxPOHClk_n" to elapse). Afterwards, the external circuit should exit this "WAIT STATE" and then place the very first bit (e.g., the most significant bit) of the "outbound" E2 byte onto the "TxPOH_n" input pin, upon the very next falling edge of "TxPOHClk_n". This data bit will be sampled and latched into the "Transmit STS-3c POH Processor" block circuitry, upon the very next rising edge of "TxPOHClk_n".

Note: *This "WAIT STATE" period is necessary because the E2 byte is the _ byte within the TOH.*

- Afterwards, the "external circuit" should serially place the remaining seven bits (of the E2 byte) onto the "TxPOH_n" input pin, upon each of the next seven falling edges of "TxPOHClk_n".

- The "external circuit" should then revert back to continuously sampling the states of the "TxPOHEnable_n" and "TxPOHFrame_n" output pins and repeat the above-mentioned process.

Figure 77 presents an illustration of the "TxPOH Input Interface" waveforms, when the "external circuit" is writing the E2 byte into the "TxPOH Input Port".

Figure 77 Illustration of the "TxPOH Input Interface" waveforms, when the "External Circuit" is writing the "E2 byte" into the "TxPOH Input Port".

2.2.9.5.14 SCRAMBLING DATA

The Transmit STS-3 TOH Processor block permits the user to either enable or disable scrambling of the STS-3 data, prior to it being transmitted to the remote terminal equipment. The customer can accomplish this by writing the appropriate value into Bit 2 (Scramble Enable), within the Transmit STS-3 Transport – SONET Transmit Control Register, as depicted below.

Transmit STS-3 Transport – SONET Transmit Control Register – Byte 0 (Address = 0x1902)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
M0M1 Insert Method[0]	Unused	RDI-L Force	AIS-L Force	LOS Force	Scramble Enable	B2 Error Insert	A1A2 Error Insert
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	X	0	0

Setting this bit-field to "1" enables the Scrambler. Conversely, setting this bit-field to "0" disables the Scrambler.

If the Scrambler is enabled, then it will scramble the "outbound" STS-3 data with a generating polynomial of $x^7 + x^6 + 1$ and a sequence length of 127.

2.3 RECEIVE DIRECTION

If a given channel (or the entire device) is configured to operate in the ATM Mode, then the purpose of the Receive Section within the XRT94L33 1-Channel STS-3c/STS-3 ATM UNI device is to permit a local ATM Layer (or ATM Adaptation Layer) processor to receive ATM cell data from a remote piece of equipment via an STS-3, STS-3c or DS3/E3 transport medium.

For ATM UNI Applications, the Receive Section of the XRT94L33 chip consists of the following blocks.

- Receive STS-3 TOH Processor Block
- Receive STS-3c POH Processor Block (for STS-3c Applications)
- Receive SONET POH Processor Block (for STS-3 Applications)
- Receive ATM Cell Processor Block
- Receive UTOPIA Interface Block

The Receive STS-3 TOH Processor block will receive an STS-3 signal, either over the PECL interface or via the Receive STS-3 Telecom Bus Interface. As the Receive STS-3 TOH Processor block receives this signal, it will do the following.

- It will locate the boundaries of the incoming STS-3 frames
- It will compute and verify the B1 and B2 bytes
- It will detect and clear the LOS, SEF, LOF, RDI-L and AIS-L defect condition
- It will detect and flag REI-L events
- It will detect and clear the SD and SF conditions
- It will route the STS-3c SPE data to the Receive STS-3c POH Processor block for further processing.

The Receive STS-3c POH Processor block will receive the STS-3c SPE data from the Receive STS-3 TOH Processor block. As the Receive STS-3c POH Processor block receives this signal, it will do the following.

- It will compute and verify the B3 bytes
- It will detect and clear the LOP-P, RDI-P and AIS-P defect conditions

- It will detect and flag REI-P events
- It will detect and clear the UNEQ-P, PLM-P and TIM-P conditions
- It will declare and clear the LOP-C and AIS-C conditions.

Finally, the Receive STS-3c POH Processor block will extract out the payload data from each incoming STS-3c SPE, and will route this data to either the Receive ATM Cell Processor block (for ATM UNI applications) or the Receive PPP Packet Processor block (for PPP applications).

For ATM Applications

The Receive ATM Cell Processor block will receive a continuous stream of un-framed ATM cells, and will perform the following functions on this incoming data-stream.

- Cell Delineation
- HEC Byte Verification
- Idle Cell Filtering
- User Cell Filtering
- Parity Calculation and Insertion

Afterwards, the Receive ATM Cell Processor block will route all cells (which have satisfactorily survived the above-mentioned processing) to the Rx FIFO, within the Receive UTOPIA Interface block.

The ATM Layer Processor will read out the contents of these ATM cells via the Receive UTOPIA Interface block of the XRT94L33. The Receive UTOPIA Interface block provides the industry-standard ATM/PHY interface functions. The Receive UTOPIA Interface block will also provide signaling to support data-flow control between the ATM Layer Processor and the Receive UTOPIA Interface block.

The Receive Section of the XRT94L33 is discussed in considerable detail below.

For PPP Applications

The Receive PPP Packet Processor block will receive a continuous stream of un-framed PPP packets, and will perform the following functions on this incoming data-stream.

- CRC-16/32 Checking
- Byte-De-Stuffing

2.3.1 RECEIVE STS-3 TOH PROCESSOR BLOCK

The purpose of the Receive STS-3 TOH Processor block is to accomplish the following.

- To receive an STS-3 data-stream from the remote LTE, via an Optical Transceiver or the System Back-plane (through the Receive PECL Interface) or via the Receive STS-3 Telecom Bus Interface.
- To acquire and maintain frame synchronization with the incoming STS-3 data-stream
- To optionally de-scramble the incoming STS-3 data-stream.
- To compute and verify the B1 and B2 bytes and increment performance monitor registers anytime it detects B1 and B2 byte errors.
- To declare and clear the following defect conditions
    - LOS (Loss of Signal)
    - SEF (Severely Errored Frame)
    - LOF (Loss of Frame)
    - AIS-L (Line AIS)

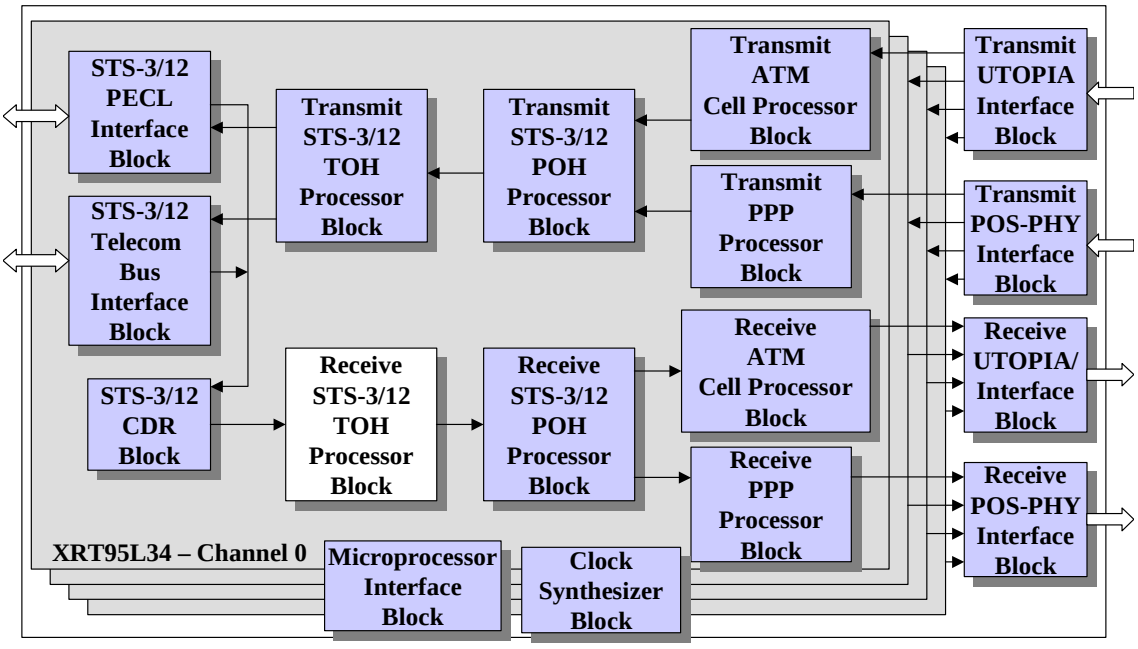
- RDI-L (Line – Remote Defect Indicator)
- SD (Signal Degrade)
- SF (Signal Fail)

- To increment performance monitor registers anytime it detects an REI-L event.
- To (optionally) automatically transmit the AIS-P (Path AIS) indicator, downstream (e.g., towards the Receive STS-3c POH Processor block) upon declaration of the following defect conditions.
  - LOS
  - LOF
  - SD
  - SF
  - AIS-L

Finally, the Receive STS-3 TOH Processor block will route STS-3c SPEs to its corresponding Receive STS-3c POH Processor block.

Figure 78 presents an illustration of the Functional Block Diagram of the XRT94L33 ATM UNI/PPP IC; with the "Receive STS-3 TOH Processor" block highlighted.

Figure 78 Illustration of the Functional Block Diagram of the XRT94L33 ATM UNI/PPP IC; with the Receive STS-3 TOH Processor block highlighted



The operation of the "Receive STS-3 TOH Processor" block is discussed in some detail below.

The Receive STS-3 TOH Processor block can be configured to receive the incoming STS-3 data via the Receive STS-3 PECL Interface block or via the Receive STS-3 Telecom Bus Interface block. A detailed discussion on the Receive STS-3 PECL Interface block can be found in Section 8.0. Likewise, a detailed discussion on the Receive STS-3 Telecom Bus Interface can be found in Section 7.0.

2.3.1.1 DESCRAMBLING OF DATA

The Receive STS-3 TOH Processor block permits the user to optionally descramble the incoming STS-3 data-stream.

2.3.1.2 LOS DECLARATION AND CLEARANCE CRITERIA SONET REQUIREMENTS FOR DECLARING THE LOS DEFECT

According to Telecordia GR-253-CORE, a SONET Network Element must monitor all incoming SONET signals (before de-scrambling) for an "All Zeros" pattern. The Network Element must declare an LOS condition whenever it continues to receives an "All Zeros" pattern for 100μs or longer. Further, the Network Element must not declare an LOS condition, if it receives the "All Zeros" pattern for 2.3μs or less.

2.3.1.2.1 How the Receive STS-3c TOH Processor Block Declares the LOS Defect

The Receive STS-3c TOH Processor block is capable of declaring and clearing the LOS condition. Further, the Receive STS-3c TOH Processor block register set permits the user to define the LOS declaration criteria, by writing the appropriate data into the "Receive STS-3c Transport – LOS Threshold Value – MSB" and "LSB" registers, as illustrated below.

Receive STS-3c Transport – LOS Threshold Value - MSB (Address = 0x112E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOS_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3c Transport – LOS Threshold Value - LSB (Address = 0x112F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOS_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

The contents of these two registers, combined, reflects the number of consecutive "All Zeros" bytes (prior to de-scrambling) that the Receive STS-3 TOH Processor block must detect before it declares the LOS defect condition.

For STS-3c applications, if the user wishes to comply with the LOS Declaration Requirements, per Telecordia GR-253-CORE, then they must write a value that ranges between 0x000F and 0x0288 into the "Receive STS-3 Transport – LOS Threshold Value – MSB/LSB" Register.

If the Receive STS-3c TOH Processor block detects the appropriate number of consecutive "All Zeros", then it will declare the LOS defect condition. The Receive STS-3c TOH Processor block will indicate that it is declaring the LOS defect condition, by doing all of the following.

- It will set Bit 0 (LOS Defect Declared) within the Receive STS-3 Transport Status Register – Byte 0, to "1" as illustrated below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

- It will generate the "Change of LOS Defect Condition" Interrupt, by toggling the "INT*" output pin "LOW", and by setting Bit 0 (Change of LOS Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

2.3.1.2.2 How the Receive STS-3 TOH Processor Block Clears the LOS Defect

The Receive STS-3 TOH Processor block will clear the LOS defect once both of the following conditions have been met.

That the Receive STS-3 TOH Processor block detects proper A1 and A2 bytes in two consecutive STS-3 frames, and

That, in between the detection of the two sets of A1/A2 bytes, the Receive STS-3 TOH Processor block does not detect the "LOS_THRESHOLD[15:0]" number of "All Zero" bytes, within the incoming STS-3 data-stream.

Once the Receive STS-3 TOH Processor block clears the LOS defect, it will notify the system of this fact by doing the following.

- It will set Bit 1 (LOS Detected) within the Receive STS-3c Transport Status Register – Byte 0" to "0" as illustrated below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the "Change of LOS Defect Condition" Interrupt by toggling the "INT*" output pin "LOW", and by setting Bit 0 (Change of LOS Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

2.3.1.2.3 Configuring the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L indicator whenever the Receive STS-3 TOH Processor block declares the LOS Defect Condition

The user can configure the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L indicator whenever (and for the duration that) the corresponding Receive STS-3 TOH Processor block declares the LOS defect condition.

Figure 79 presents an illustration of the Transmit STS-3 TOH Processor block transmitting the RDI-L indicator (up-stream) towards the remote LTE whenever the corresponding Receive STS-3 TOH Processor block is declaring the LOS defect condition.

Figure 79 Illustration of the Transmit STS-3 TOH Processor block transmitting the RDI-L indicator to the remote LTE whenever the corresponding Receive STS-3 TOH Processor block is declaring the LOS defect condition

The user can implement this configuration by setting Bit 0 (Transmit RDI-L upon LOS) within the "Transmit STS-3 Transport – RDI-L Control Register, as depicted below.

Transmit STS-3 Transport – RDI-L Control Register (Address = 0x1933)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				External RDI-L Enable	Transmit RDI-L upon AIS	Transmit RDI-L upon LOF	Transmit RDI-L upon LOS
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

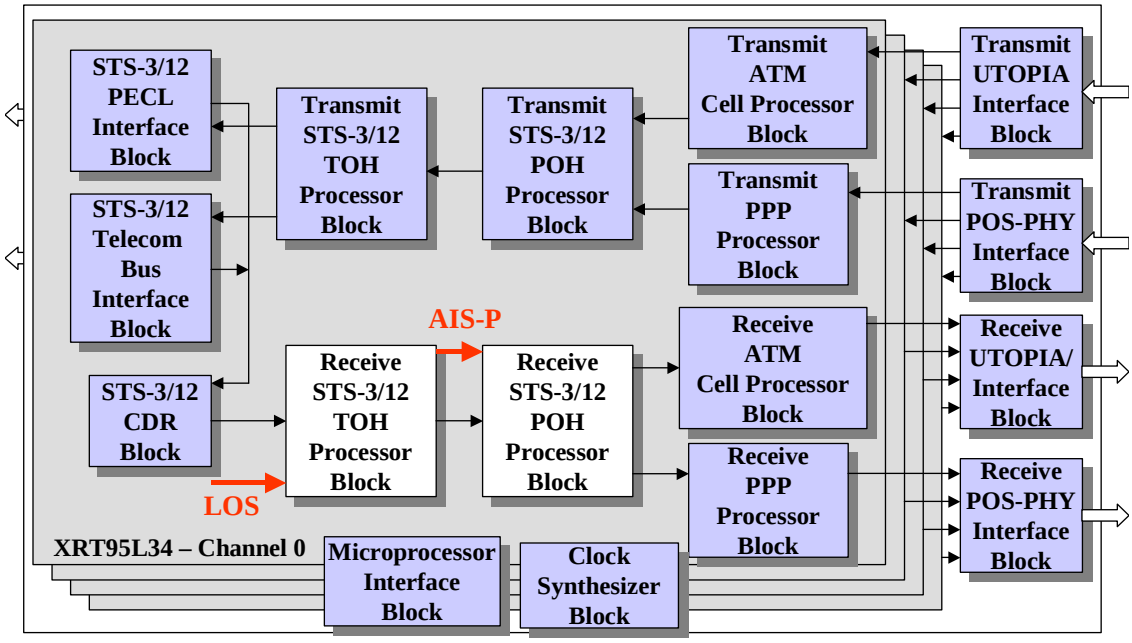
2.3.1.2.4 Configuring the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator, in the Downstream Direction (towards the Receive STS-3c POH Processor blocks) whenever it declares the LOS Defect Condition

The user can configure the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator, in the down-stream direction (towards the corresponding Receive STS-3c POH Processor block) whenever (and for the duration that) it declares the LOS defect condition.

Note: *If the XRT94L33 is configured to operate in either the "1-Channel STS-3 ATM UNI/PPP" Mode, then the Receive STS-3 TOH Processor block will transmit the AIS-P indicator to all three Receive SONET POH Processor blocks, in parallel, whenever it declares the LOS defect condition.*

Figure 80 presents an illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor block) whenever it declares the LOS defect condition.

Figure 80 An Illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor blocks) whenever it declares the LOS defect condition



The user can implement this configuration by setting Bits 1 (Transmit AIS-P [down-stream] upon LOS) and 0 (AUTO AIS), within the "Receive STS-3 Transport – Auto AIS Control" Register, to "1" as depicted below.

Receive STS-3 Transport – Auto AIS Control Register (Address = 0x1163)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit AIS-P (down-stream) upon Section Trace Message Unstable	Transmit AIS-P (down-stream) upon Section Trace Message Mismatch	Transmit AIS-P (down-stream) upon SF	Transmit AIS-P (down-stream) upon SD	Transmit AIS-P (down-stream) upon Loss of Optical Carrier	Transmit AIS-P (down-stream) upon LOF	Transmit AIS-P (down-stream) upon LOS	AUTO AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

Of course, if the Receive STS-3 TOH Processor block transmits the AIS-P indicator towards the Receive STS-3c POH Processor block, then all of the following will happen.

- The Receive STS-3c POH Processor block will declare the AIS-P defect condition.
- The corresponding Receive ATM Cell Processor block will declare the LCD (Loss of Cell Delineation) defect condition.

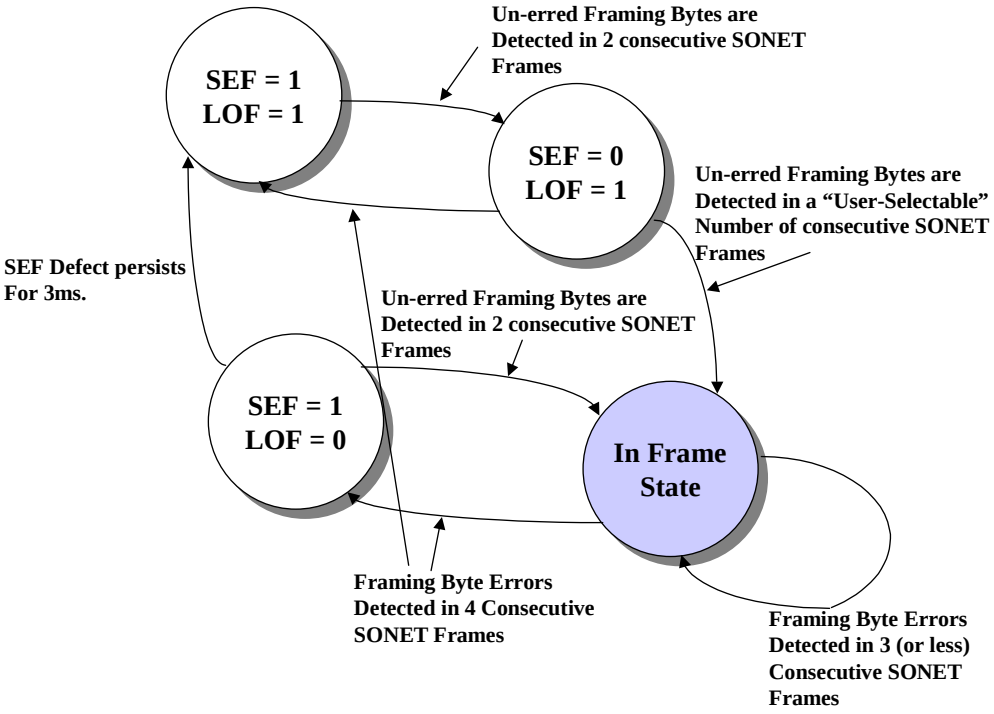
2.3.1.3 RECEIVE STS-3 TOH PROCESSOR BLOCK FRAMING ALIGNMENT

At any given time, the Receive STS-3 TOH Processor block is operating in either one of the following "Framing Alignment" modes.

- In-Frame
- SEF = 1, LOF = 0
- SEF = 1, LOF = 1
- SEF = 0, LOF = 1

The Receive STS-3 TOH Processor block will transition through these various modes depending upon whether or not it has framing alignment with the A1, A2 bytes within the incoming STS-3 frames. Each of these "Framing Alignment" Modes will be discussed in some detail below. The overall framing alignment state-machine (within the Receive STS-3 TOH Processor block) is as presented below in Figure 81.

Figure 81 Illustration of the "Receive STS-3 TOH Processor" Block Framing Alignment State-Machine



2.3.1.3.1 The "SEF = 1, LOF = 1" State

When the XRT94L33 first powers up, the Receive STS-3 TOH Processor block will be operating in this state. In this case, both the LOF (Loss of Frame) and the SEF (Severely Erred Frame) defect conditions are declared. The Receive STS-3 TOH Processor block will indicate this fact by setting both Bits 2 (LOF Defect Declared) and 1 (SEF Defect Declared), within the "Receive STS-3 Transport Status Register – Byte 0" as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	1	0

The Receive STS-3 TOH Processor block will remain in this condition until it receives two consecutive STS-3 frames that contain no framing byte errors. At this point the Receive STS-3 TOH Processor block will clear the SEF defect and will transition into the "SEF = 0, LOF = 1" State.

As the Receive STS-3 TOH Processor block transitions from the "SEF = 1, LOF = 1" state to the "SEF = 0, LOF = 1" state, it will do all of the following.

It will clear Bit 1 (SEF Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "0", as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	0	0

5. It will generate the "Change in SEF Defect Condition" interrupt. The XRT94L33 will indicate that it is generating this interrupt by doing the following.

   Toggling the "INT*" input pin "low" and

   Setting Bit 1 (Change of SEF Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 0" as depicted below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.3.2 The SEF = 0, LOF = 1 State

Once the Receive STS-3 TOH Processor block reaches this state, then it has already cleared the "SEF Defect Condition". For the duration that the Receive STS-3 TOH Processor block is operating in the "SEF = 0, LOF = 1" state, the Receive STS-3 TOH Processor block will be testing the Framing Alignment bytes (A1 and A2) within the incoming STS-3 signal, in order to determine if it can clear the "LOF" defect condition.

In this case, the Receive STS-3 TOH Processor will test the Framing Alignment bytes (within the incoming STS-3 data-stream) a "User-Selectable" number of SONET frame periods. If the Receive STS-3 TOH Processor does not detect any Framing Byte errors (during this "user-selected test" period), then it will clear the LOF defect.

The user can specify the number of consecutive STS-3 frame periods, that the Receive STS-3 TOH Processor block must remain in the "SEF = 0, LOF = 1" state, by writing the appropriate value into Bits 3 through 0 (In-Sync Threshold[3:0]) within the Receive STS-3 Transport – In Sync Threshold Register; as depicted below.

Receive STS-3 Transport – In Sync Threshold Register (Address = 0x112B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				In-Sync Threshold[3:0]			
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	X

The value that the user writes into these bit-fields reflects the number of STS-3 frame periods that the Receive STS-3 TOH Processor block must reside within the "SEF = 0, LOF = 1" state and test the Framing Alignment bytes, within the incoming STS-3 data-stream.

If the Receive STS-3 TOH Processor block does not detect the "User Selectable" number of consecutive "un-erred" STS-3 frames, then it will remain in the "SEF = 0, LOF = 1" state and will continue to test for the "user-selectable" number of consecutive un-erred STS-3 frame. If the Receive STS-3 TOH Processor block were to detect Framing Byte errors in four consecutive STS-3 frames, then it will declare the "SEF" defect and will transition back into the "SEF = 1, LOF = 1" state.

If the Receive STS-3 TOH Processor block receives this "User-Selectable" number of consecutive "unerred" STS-3 frames, then it will clear the "LOF defect" and will transition into the "In-Frame" state.

As the Receive STS-3 TOH Processor block transitions from the "SEF = 0, LOF = 1" state to the "In-Frame" state, it will do all of the following.

It will clear Bit 2 (LOF Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "0" as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

It will generate the "Change in LOF Defect Condition" interrupt. The XRT94L33 will indicate that it is generating this interrupt by doing the following.

a. Toggling the "INT*" input pin "LOW".

b. Setting Bit 2 (Change of LOF Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 0" as depicted below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.3.3 The In-Frame State

Once the Receive STS-3 TOH Processor block reaches this state, it is considered to be operating in a "Normal" manner. In this mode, the Receive STS-3 TOH Processor block will continue to monitor and check the value of the Framing Alignment bytes within the incoming STS-3 data-stream.

In general, the Receive STS-3 TOH Processor block will be tolerant to some occasional Framing Byte errors. However, if the Receive STS-3 TOH Processor block were to detect framing alignment bit errors in four consecutive SONET frames (within the incoming STS-3 data-stream), then it will declare the "SEF" defect. As the Receive STS-3 TOH Processor declares the SEF defect, then it will transition into the "SEF = 1, LOF = 0" state.

As the Receive STS-3 TOH Processor block transitions from the "In-Frame" to the "SEF = 1, LOF = 0" state, it will do all of the following.

5. It will set Bit 1 (SEF Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "1", as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	1	0

6. It will generate the "Change of SEF Defect Condition" interrupt. The XRT94L33 will indicate that it is generating this interrupt by doing the following.

a. Toggling the "INT*" input pin "low" and

b. Setting Bit 1 (Change of SEF Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 0" as depicted below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.3.4 The SEF = 1, LOF = 0 State

Once the Receive STS-3 TOH Processor block reaches this state, then it has already declared the "SEF Defect Condition". For the duration that the Receive STS-3 TOH Processor block is operating in the "SEF = 1, LOF = 0" state, the Receive STS-3 TOH Processor block will be testing the Framing Alignment bytes (A1 and A2) within the incoming STS-3 signal, in order to determine if it should declare the "LOF" or the "In-Frame" condition.

If the Receive STS-3 TOH Processor block receives two consecutive SONET frames that do have contain any Framing Alignment byte errors, then it will clear the SEF Defect and will transition back into the "In-Frame" state.

On the other hand, if the Receive STS-3 TOH Processor continues to detect incoming SONET frames (with Framing Alignment errors) then it will continue to declare the "SEF" defect. If the Receive STS-3 TOH Processor block continues to declare the "SEF Defect" for at least 3ms, then it will declare the "LOF Defect". At this time, the Receive STS-3 TOH Processor block will transition back into the "SEF = 1, LOF = 1" state.

The "SEF" and "LOF" declaration and clearance criteria are summarized, in some detail below.

2.3.1.3.5 THE SEF (SEVERELY ERRED FRAME) DEFECT DECLARATION AND CLEARANCE CRITERIA

The Receive STS-3 TOH Processor block is capable of declaring and clearing the SEF (Severely Erred Frame) defect condition; as described below.

2.3.1.3.5.1 How the Receive STS-3 TOH Processor Block Declares the SEF Defect

The Receive STS-3 TOH Processor block will declare the SEF defect condition anytime it detects Framing Byte (A1 and A2) errors in four consecutive frames.

Whenever the Receive STS-3 TOH Processor block declares the SEF defect condition, then it will do the following.

1. It will indicate that it is declaring the SEF defect condition by setting Bit 1 (SEF Defect Declared), within the Receive STS-3 Transport Status Register – Byte 0" to "1", as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	1	0

2. It will generate the "Change of SEF Defect Condition Interrupt". The Receive STS-3 TOH Processor block will indicate that it is declaring the "Change of SEF Defect Condition" Interrupt by doing the following.
    a. Toggling the "INT*" output pin "LOW".
    b. b. Setting Bit 1 (Change of SEF Defect Condition Interrupt Status), within the Receive STS-3 Transport Interrupt Status Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.3.5.2 How the Receive STS-3 TOH Processor Block clears the SEF Defect Condition

Once the Receive STS-3 TOH Processor block declares the SEF defect condition, then it will clear the SEF Defect Condition if it detects 2 consecutive STS-3 frames with un-erred framing alignment (A1 and A2) bytes.

Once the Receive STS-3 TOH Processor block clears the SEF defect condition, then it will alert the Microprocessor of this fact by doing the following.

7. It will indicate that it is clearing the SEF defect condition by setting Bit 1 (SEF Defect Declared), within the Receive STS-3 Transport Status Register – Byte 0" to "0" as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

8. It will generate the "Change of SEF Defect Condition" Interrupt. The Receive STS-3 TOH Processor block will indicate that it is declaring the "Change of SEF Defect Condition" interrupt by doing the following.
    a. Toggling the "INT*" output pin "LOW".
    b. b. Setting Bit 1 (Change of SEF Defect Condition Interrupt Status), within the Receive STS-3 Transport Interrupt Status Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.3.6 THE LOF (LOSS OF FRAME) DECLARATION AND CLEARANCE CRITERIA

The Receive STS-3 TOH Processor block is capable of declaring and clearing the LOF (Loss of Frame) defect condition, as described below.

2.3.1.3.6.1 How the Receive STS-3 TOH Processor Block declares the LOF Defect Condition

The Receive STS-3 TOH Processor block will declare the LOF defect anytime the Receive STS-3 TOH Processor block continuously declares the SEF defect condition for at least 3ms.

Whenever the Receive STS-3 TOH Processor block declares the LOF Defect condition, then it will do the following.

9. It will indicate that it is declaring the LOF defect condition by setting Bit 2 (LOS Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "1" as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L	S1 Byte	K1, K2 Byte	SF Defect	SD Defect	LOF Defect	SEF Defect	LOS Defect

Defect Declared	Unstable Defect Declared	Unstable Defect Declared	Declared	Declared	Declared	Declared	Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	0	0

10. It will generate the "Change of LOF Defect Condition" Interrupt. The Receive STS-3 TOH Processor block will indicate that it is declaring the "Change of LOF Defect Condition" interrupt by doing the following.
    a. Toggling the "INT*" output pin "low".
    b. Setting Bit 2 (Change of LOF Defect Condition Interrupt Status) within the Receive STS-3 Transport Interrupt Status Register – Byte 0 to "1" as depicted below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

2.3.1.3.6.2 How the Receive STS-3 TOH Processor Block clears the LOF Defect Condition

Once the Receive STS-3 TOH Processor block has declared the LOF defect condition, then it will clear the LOF only after both of the following conditions have been met.

11. That the Receive STS-3 TOH Processor block has cleared the SEF defect.
12. If it detects a "user-specified" number of consecutive STS-3 frames with un-erred framing alignment (e.g., A1 and A2) bytes.

The user can specify the "LOF Clearance Criteria" by writing the appropriate value (in terms of numbers of consecutive STS-3 frames with un-erred A1 and A2 bytes) into Bits 3 through 0 (In-Sync Threshold) within the "Receive STS-3 Transport – In Sync Threshold" Register, as depicted below.

Receive STS-3 Transport – In-Sync Threshold Value (Address =0x112B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FRPATOUT[1:0]		FRPATIN[1:0]		Unused
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/O
0	0	0	0	0	0	0	0

Once the Receive STS-3 TOH Processor block has cleared the SEF defect, then it will proceed to check for the occurrence of "In-Sync Threshold[3:0]" number of incoming STS-3 frames that contain un-erred A1 and A2 bytes. Once all of this has occurred then the Receive STS-3 TOH Processor block will clear the LOF defect condition.

Once the Receive STS-3 TOH Processor block clears the LOF condition, then it will alert the Microprocessor of this fact by doing the following.

13. It will indicate that it is clearing the LOF defect condition by setting Bit 2 (LOF Defect Declared) within the Receive STS-3 Transport Status Register – Byte 0" to "0" as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

14. It will generate the "Change of LOF Defect Condition" Interrupt. The Receive STS-3 TOH Processor block will indicate that it is declaring the "Change of LOF Defect Condition" interrupt by doing the following.

c. Toggling the "INT*" output pin "low".

Setting Bit 2 (Change of LOF Defect Condition Interrupt Status) within the Receive STS-3 Transport Interrupt Status Register – Byte 0 to "1" as depicted below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

2.3.1.3.6.3 Configuring the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L Indicator whenever the Receive STS-3 TOH Processor Block declares the LOF Defect Condition

The user can configure the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L indicator to the remote LTE whenever (and for the duration that) the corresponding Receive STS-3 TOH Processor block declares the LOF defect condition.

Figure 82 presents an illustration of the Transmit STS-3 TOH Processor block transmitting the RDI-L indicator, to the remote LTE, whenever the corresponding Receive STS-3 TOH Processor block declares the LOF defect condition.

Figure 82 Illustration of the Transmit STS-3 TOH Processor block (within a given XRT94L33) transmitting the RDI-L indicator, to the remote LTE whenever the corresponding Receive STS-3 TOH Processor block declares the LOF defect condition.

The user can implement this configuration by setting Bit 1 (Transmit RDI-L upon LOF) within the Transmit STS-3 Transport – RDI-L Control Register, as depicted below.

Transmit STS-3 Transport – RDI-L Control Register (Address = 0x1933)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				External RDI-L Enable	Transmit RDI-L upon AIS-L	Transmit RDI-L upon LOF	Transmit RDI-L upon LOS
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

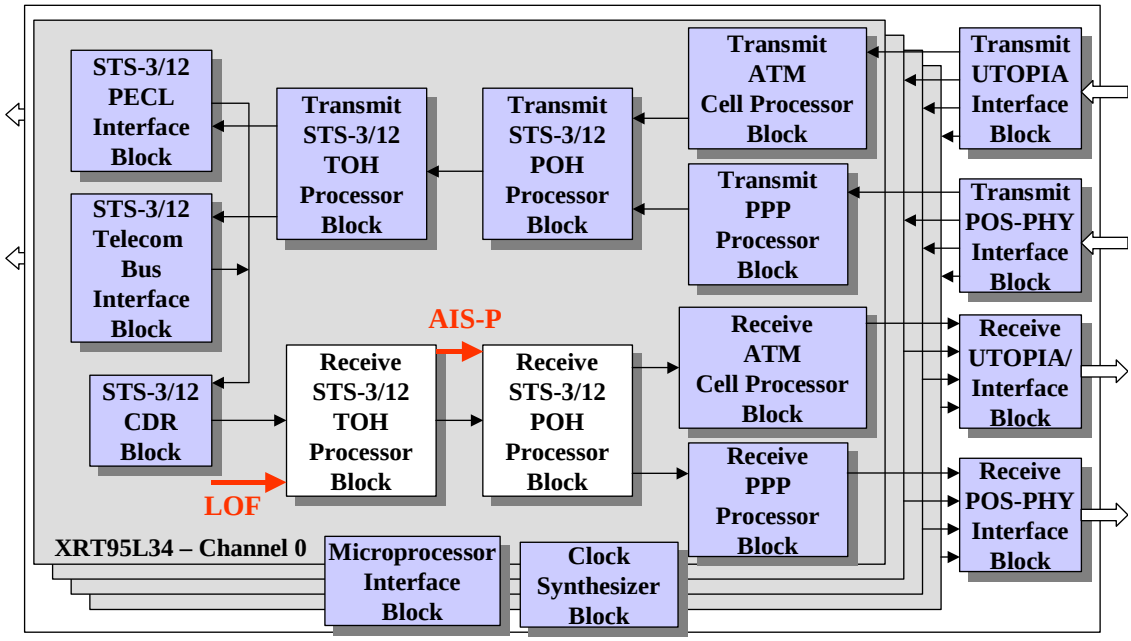
2.3.1.3.6.4 Configuring the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator in the down-stream direction (towards the Receive STS-3c POH Processor Block) whenever it declares the LOF Defect

The user can configure the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator, in the down-stream direction (towards the corresponding Receive STS-3c POH Processor block) whenever (and for the duration that) it declares the LOF defect condition.

Note: *If the XRT94L33 is configured to operate in either the "1-Channel STS-3 ATM UNI/PPP" Mode, then the Receive STS-3 TOH Processor block will transmit the AIS-P indicator to all four Receive SONET POH Processor blocks, in parallel, whenever it declares the LOF defect condition.*

Figure 83 presents an illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor block) whenever it declares the LOF defect condition.

Figure 83 An Illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor blocks) whenever it declares the LOF defect condition



The user can implement this configuration by setting Bits 2 (Transmit AIS-P [down-stream] upon LOF) and 0 (AUTO AIS), within the "Receive STS-3 Transport – Auto AIS Control" Register, to "1" as depicted below.

Receive STS-3 Transport – Auto AIS Control Register (Address = 0x1163)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit AIS-P (down-stream) upon Section Trace Message Unstable	Transmit AIS-P (down-stream) upon Section Trace Message Mismatch	Transmit AIS-P (down-stream) upon SF	Transmit AIS-P (down-stream) upon SD	Transmit AIS-P (down-stream) upon Loss of Optical Carrier	Transmit AIS-P (down-stream) upon LOF	Transmit AIS-P (down-stream) upon LOS	AUTO AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	1

Of course, if the Receive STS-3 TOH Processor block transmits the AIS-P indicator towards the Receive STS-3c POH Processor block, then all of the following will happen.

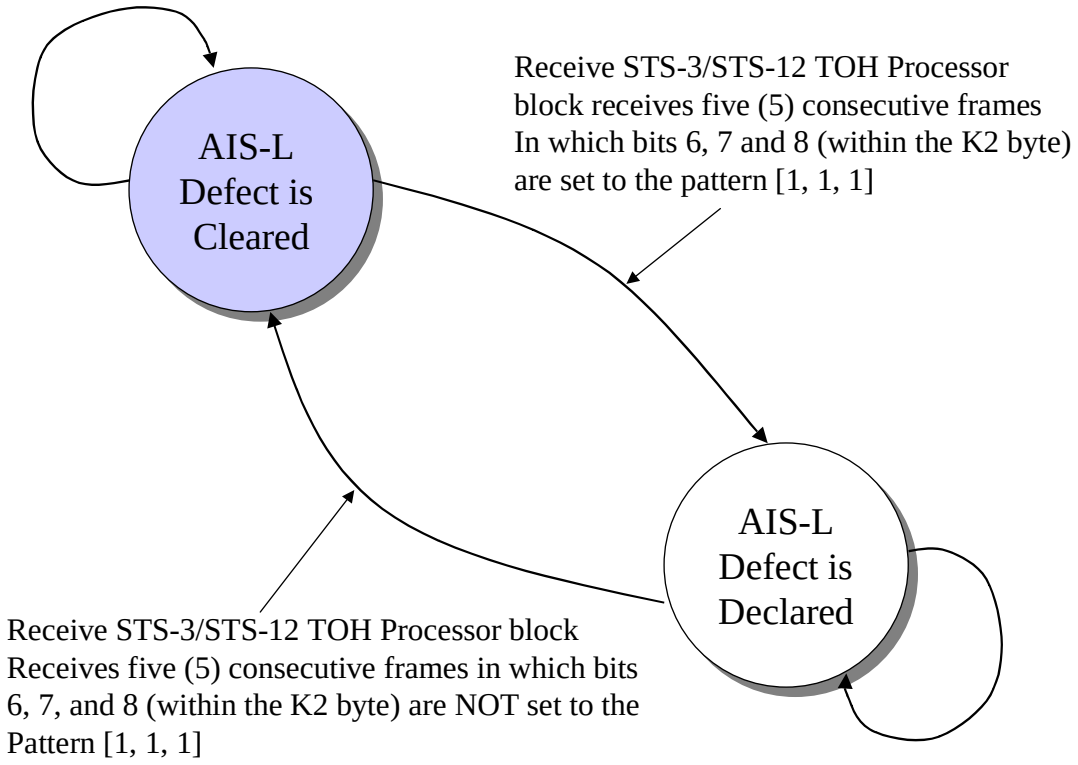
- The Receive STS-3c POH Processor block will declare the AIS-P defect condition.
- The corresponding Receive ATM Cell Processor block will declare the LCD (Loss of Cell Delineation) defect condition.

2.3.1.4 THE AIS-L (LINE AIS) DEFECT DECLARATION AND CLEARANCE CRITERIA

According to Telecordia GR-253-CORE, a Line Terminating Equipment must declare the AIS-L (Line AIS) defect condition whenever it determines that bits 6, 7 and 8 (within the K2 byte) of the incoming STS-3 data-stream, are each set to "1" for five consecutive STS-3 frames.

Figure 84 presents an illustration of the "AIS-L Defect Declaration/Clearance" State Machine diagram that is used by the Receive STS-3 TOH Processor blocks within the XRT94L33.

Figure 84 Illustration of the "AIS-L Defect Declaration/Clearance" State Machine diagram



2.3.1.4.1 How the Receive STS-3 TOH Processor Block Declares the AIS-L Defect Condition

The Receive STS-3 TOH Processor block is capable of declaring and clearing the AIS-L defect condition. If the Receive STS-3 TOH Processor block receives at least five consecutive STS-3 frames, in which bits 6, 7 and 8 (within the K2 byte) are each set to "1", then it will declare the AIS-L defect condition. The Receive STS-3c TOH Processor block will indicate that it is declaring the AIS-L defect condition, by doing all of the following.

- It will set Bit 0 (AIS-L Defect Declared) within the Receive STS-3 Transport Status Register – Byte 1, to "1" as illustrated below.

Receive STS-3 Transport Status Register – Byte 1 (Address = 0x1106)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused					Section Trace Message Mismatch Defect Declared	Section Trace Message Unstable Defect Declared	AIS_L Defect Declared

R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

- It will generate the "Change of AIS-L Defect Condition" Interrupt, by toggling the "INT*" output pin "LOW", and by setting Bit 1 (Change of AIS-L Defect Condition Interrupt Status) within the "Receive STS-3 Transport Interrupt Status Register – Byte 2" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 2 (Address = 0x1109)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						Change of AIS-L Defect Condition Interrupt Status	Change of RDI-L Defect Condition Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	RUR	RUR
0	0	0	0	0	0	1	0

2.3.1.4.2 How the Receive STS-3 TOH Processor Block Clears the AIS-L Defect Condition

The Receive STS-3 TOH Processor block will clear the AIS-L defect once it has received at least 5 consecutive STS-3 frames, in which bits 6, 7 and 8 (of the K2 byte) are set to some value other than "1, 1, 1".

Once the Receive STS-3 TOH Processor block clears the AIS-L defect, it will notify the system of this fact by doing the following.

- It will set Bit 0 (AIS-L Defect Declared) within the Receive STS-3 Transport Status Register – Byte 1" to "0" as illustrated below.

Receive STS-3 Transport Status Register – Byte 1 (Address = 0x1106)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused					Section Trace Message Mismatch Defect Declared	Section Trace Message Unstable Defect Declared	AIS_L Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the "Change of AIS-L Defect Condition" Interrupt by toggling the "INT*" output pin "LOW", and by setting Bit 1 (Change of AIS-L Defect Condition Interrupt Status), within the Receive STS-3 Transport Interrupt Status Register – Byte 2" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 2 (Address = 0x1109)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						Change of AIS-L Defect Condition Interrupt Status	Change of RDI-L Defect Condition Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	RUR	RUR

0	0	0	0	0	0	1	0

2.3.1.4.3 Configuring the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L indicator whenever the Receive STS-3 TOH Processor block declares the AIS-L Defect Condition

The user can configure the Transmit STS-3 TOH Processor block to automatically transmit the RDI-L indicator to the remote LTE whenever (and for the duration that) the corresponding Receive STS-3 TOH Processor block declares the AIS-L defect condition.

Figure 85 presents an illustration of the Transmit STS-3 TOH Processor block transmitting the RDI-L indicator, to the remote LTE, whenever the corresponding Receive STS-3 TOH Processor block declares the AIS-L defect condition.

Figure 85Illustration of the Transmit STS-3 TOH Processor block (within a given XRT94L33) transmitting the RDI-L indicator, to the remote LTE whenever the corresponding Receive STS-3 TOH Processor block declares the AIS-L defect condition.

The user can implement this configuration by setting Bit 2 (Transmit RDI-L upon AIS-L) within the Transmit STS-3 Transport – RDI-L Control Register, as depicted below.

Transmit STS-3 Transport – RDI-L Control Register (Address = 0x1933)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				External RDI-L Enable	Transmit RDI-L upon AIS-L	Transmit RDI-L upon LOF	Transmit RDI-L upon LOS
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	0

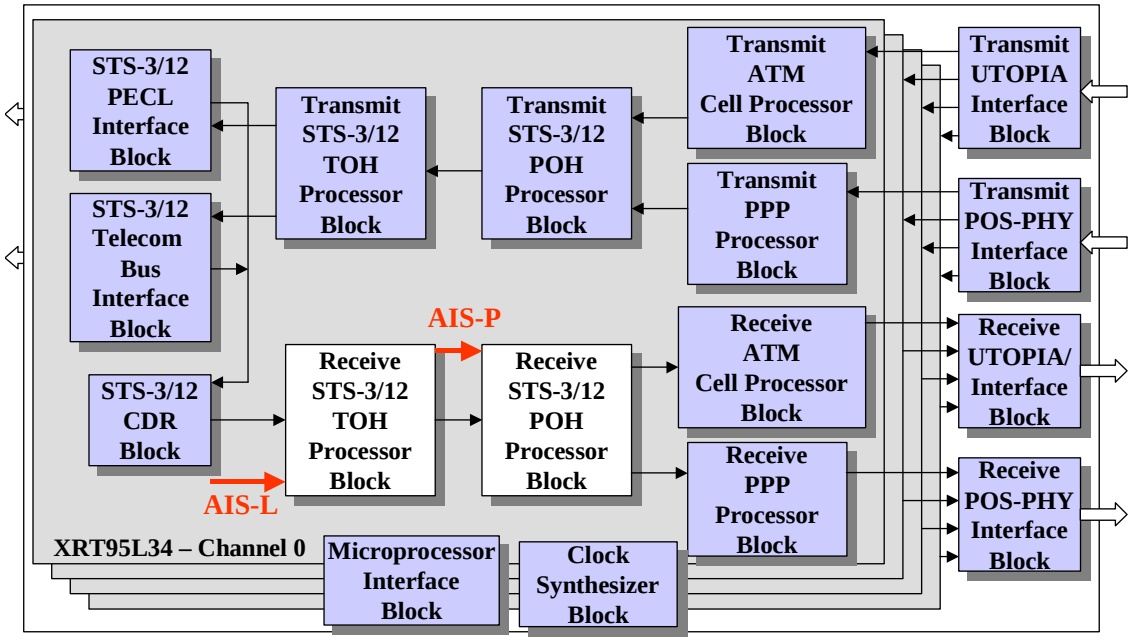
2.3.1.4.4 Configuring the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator in the down-stream direction (towards the Receive STS-3c POH Processor block) whenever it declares the AIS-L Defect Condition

The user can configure the Receive STS-3 TOH Processor block to automatically transmit the AIS-P indicator, in the down-stream direction (towards the corresponding Receive STS-3c POH Processor block) whenever (and for the duration that) it declares the AIS-L defect condition.

Note: *If the XRT94L33 is configured to operate in either the "1-Channel STS-3 ATM UNI/PPP" Mode, then the Receive STS-3 TOH Processor block will transmit the AIS-P indicator to all three Receive SONET POH Processor blocks, in parallel, whenever it declares the AIS-L defect condition.*

Figure 86 presents an illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor block) whenever it declares the AIS-L defect condition.

Figure 86 Illustration of the Receive STS-3 TOH Processor block transmitting the AIS-P indicator, in the down-stream direction (towards the Receive STS-3c POH Processor blocks) whenever it declares the AIS-L defect condition



The user can implement this configuration by setting Bits 0 (AUTO AIS), within the "Receive STS-3 Transport – Auto AIS Control" Register, to "1" as depicted below.

Receive STS-3 Transport – Auto AIS Control Register (Address = 0x1163)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit AIS-P (down-stream) upon Section Trace Message Unstable	Transmit AIS-P (down-stream) upon Section Trace Message Mismatch	Transmit AIS-P (down-stream) upon SF	Transmit AIS-P (down-stream) upon SD	Transmit AIS-P (down-stream) upon Loss of Optical Carrier	Transmit AIS-P (down-stream) upon LOF	Transmit AIS-P (down-stream) upon LOS	AUTO AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Of course, if the Receive STS-3 TOH Processor block transmits the AIS-P indicator towards the Receive STS-3c POH Processor block, then all of the following will happen.

- The Receive STS-3c POH Processor block will declare the AIS-P defect condition.

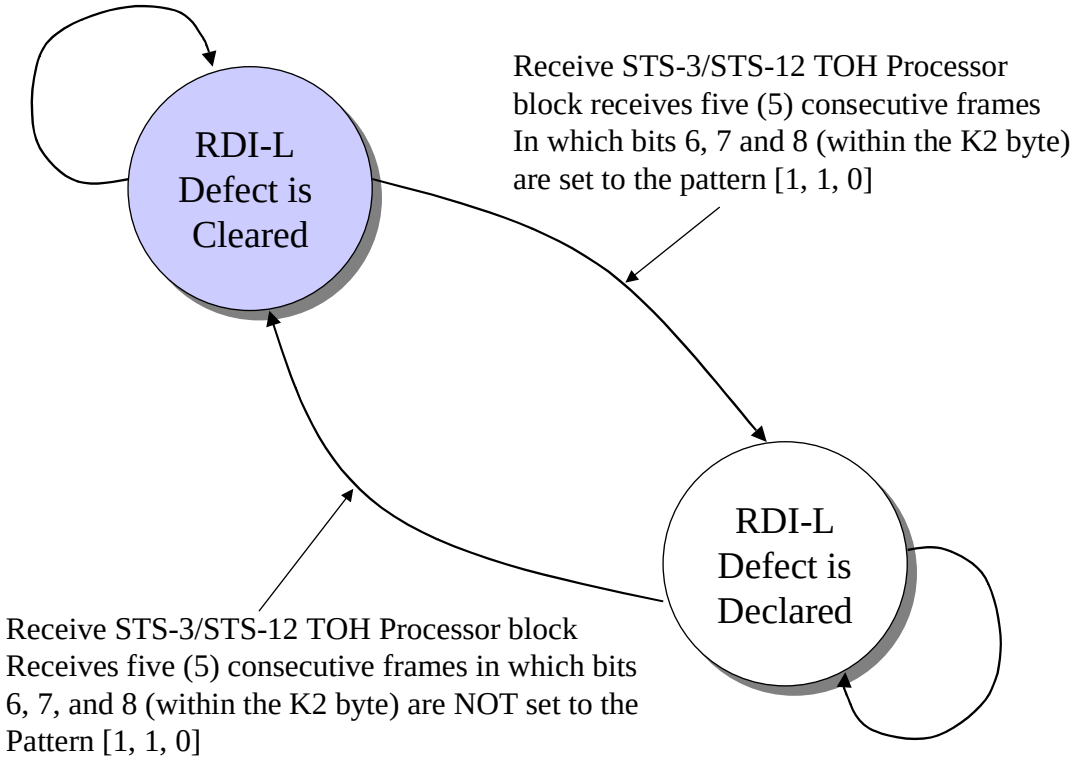
- The corresponding Receive ATM Cell Processor block will declare the LCD (Loss of Cell Delineation) defect condition.

2.3.1.5 RDI-L DECLARATION AND CLEARANCE CRITERIA

According to Telecordia GR-253-CORE, a Line Terminating Equipment must declare the RDI-L (Line – Remote Defect Indicator) condition whenever it determines that bits 6, 7 and 8 (within the K2 byte) of the incoming STS-3c data-stream, are set the pattern [1, 1, 0] for five consecutive STS-3c frames.

Figure 87 presents the illustration of the "RDI-L Declaration/Clearance" State Machine Diagram that is used by the Receive STS-3 TOH Processor block within the XRT94L33.

Figure 87 Illustration of the "RDI-L Declaration/Clearance" State Machine Diagram



2.3.1.5.1 How the Receive STS-3 TOH Processor Block Declares the RDI-L Defect Condition

The Receive STS-3c TOH Processor block is capable of declaring and clearing the RDI-L defect condition. If the Receive STS-3 TOH Processor block receives at least five consecutive STS-3 frames, in which bits 6, 7 and 8 (within the K2 byte) are set to the "[1, 1, 0]" pattern, then it will declare the RDI-L defect condition. The Receive STS-3 TOH Processor block will indicate that it is declaring the RDI-L defect condition by doing all of the following.

- It will set Bit 7 (RDI-L Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "1" as illustrated below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect	K1, K2 Byte Unstable Defect	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared

	Declared	Declared					
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
1	0	0	0	0	0	0	0

- It will generate the "Change in RDI-L Defect Condition" Interrupt, by toggling the "INT*" output pin "LOW", and by setting Bit 0 (Change of RDI-L Defect Condition Interrupt Status) within the "Receive STS-3 Transport Interrupt Status Register – Byte 2" to " 1" as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 2 (Address = 0x1109)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						Change of AIS-L Defect Condition Interrupt Status	Change of RDI-L Defect Condition Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	RUR	RUR
0	0	0	0	0	0	0	1

2.3.1.5.2 How the Receive STS-3 TOH Processor Block Clears the RDI-L Defect Condition

The Receive STS-3 TOH Processor block will clear the RDI-L defect once it has received at least 5 consecutive STS-3 frames, in which bits 6, 7 and 8 (of the K2 byte) are set to some value other than [1, 1, 0].

Once the Receive STS-3 TOH Processor block clears the RDI-L defect, it will notify the system of this fact by doing the following.

- It will set Bit 7 (RDI-L Defect Declared) within the Receive STS-3 Transport Status Register – Byte 0" to "0" as illustrated below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the "Change in RDI-L Defect Condition" Interrupt by toggling the "INT*" output pin "LOW", and by setting Bit 0 (Change of RDI-L Defect Condition Interrupt Status), within the "Receive STS-3 Transport Interrupt Status Register – Byte 2" to "1", as illustrated below.

Receive STS-3 Transport Interrupt Status Register – Byte 2 (Address = 0x1109)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						Change of AIS-L Defect Condition Interrupt Status	Change of RDI-L Defect Condition Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	RUR	RUR
0	0	0	0	0	0	0	1

2.3.1.6 DETECTING AND FLAGGING REI-L (LINE – REMOTE ERROR INDICATOR) EVENTS

The Receive STS-3 TOH Processor block is capable of detecting the REI-L indicator, within the incoming STS-3 data-stream. As the Receive STS-3 TOH Processor block receives a given STS-3 data-stream, it will monitor the contents within the M1 byte. The bit-format of the M1 byte is presented below in Figure 88.

Figure 88 Bit format of the M1 Byte

BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8
B2 Error Count (REI-L)							

The role of the REI-L bit-fields was described in some detail, in Section _. This section indicates that the remote terminal equipment will set the "REI-L" value (within the M1 byte) to "0" during "un-erred" conditions. However, the remote terminal equipment will typically set the "REI-L" value to a value (ranging from "1" to "24") during "erred" conditions.

If the Receive STS-3 TOH Processor block receives an STS-3 frame, that contains a "non-zero" value of REI-L, then it will do the following.

15. It will generate the "Detection of REI-L Error" Interrupt.

Note: *The Receive STS-3 TOH Processor block will indicate this by, pulling the "INT*" output pin "LOW" and by setting Bit 5 (Detection of REI-L Error Interrupt Status), within the "Receive STS-3 Transport – Interrupt Status Register – Byte 0" to "1" as depicted below.*

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

16. It will increment the "Receive STS-3 Transport – REI-L Error Count" Registers

Note: *These registers are actually 32-bit registers, which are located at Direct Address locations 0xNA19 through 0xNA1C. The bit-format of these registers is presented below.*

Receive STS-3 Transport – REI-L Error Count Register – Byte 3 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_L_Error_Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3 Transport – REI-L Error Count Register – Byte 2 (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_L_Error_Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3 Transport – REI-L Error Count Register – Byte 1 (Address = 0x111A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_L_Error_Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3 Transport – REI-L Error Count Register – Byte 0 (Address = 0x111B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_L_Error_Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Note: *The Receive STS-3 TOH Processor block will increment these registers either by the number of erred STS-3 Frames (e.g., STS-3 frames with a non-zero REI-L value) detected, or by the value of the REI-L value, within the M1 byte of a given STS-3 frame; depending upon the user's selection, as described below.*

2.3.1.6.1 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport – REI-L Error Count" Register on an "REI-L Value" basis

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport – REI-L Error Count" Register by the contents within the M1 byte, within each incoming STS-3 frame. Therefore, in this mode, it is possible for the Receive STS-3 TOH Processor block to increment this register by as much as the value "24" for each STS-3 frame.

The user can implement this configuration by setting Bit 2 (REI-L Error Type) within the "Receive STS-3 Transport – Control Register – Byte 0" to "0", as illustrated below.

Receive STS-3c Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.3.1.6.2 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport – REI-L Error Count" Register on a "per STS-3 Frame" basis.

The user can also configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport – REI-L Error Count" Register, by the value "1" for each STS-3 frame that contains a "non-zero" REI-L value.

The user can implement this configuration by setting Bit 2 (REI-L Error Type), within the "Receive STS-3 Transport – Control Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/ SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	0

2.3.1.6.3 Reading out the contents of the "Receive STS-3 Transport – REI-L Error Count Registers, during Performance Monitoring

2.3.1.7 RECEIVE SECTION TRACE MESSAGES VIA THE INCOMING J0 BYTE

2.3.1.8 HANDLING/SUPPORT OF THE INCOMING E1 BYTE

2.3.1.9 HANDLING/SUPPORT OF THE INCOMING F1 BYTE

2.3.1.10 HANDLING/SUPPORT OF THE INCOMING SECTION DCC (D1, D2 AND D3) BYTES

2.3.1.11 SECTION BIP-8 (B1) BYTE VERIFICATION

The Receive STS-3 TOH Processor Block has the responsibility for computing and verifying the Section BIP-8 (e.g., B1) byte within each incoming STS-3 frame. When the Receive STS-3 TOH Processor block executes this function, it will do the following.

- It will read in the contents of a given "newly received" STS-3 frame.
- It will compute the BIP-8 value of this entire STS-3 frame.
- This resulting BIP-8 value will be compared with the contents of the B1 byte, within the very next "newly received" STS-3 frame.

If the Receive STS-3 TOH Processor block detects any B1 byte errors, then it will do the following.

- o It will generate the "Detection of B1 Byte Error" Interrupt, by toggling the "INT*" output pin "LOW" and by setting Bit 3 (Detection of B1 Byte Error Interrupt Status) within the "Receive STS-3 Transport Interrupt Status" Register to "1", as indicated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

o It will increment the "Receive STS-3 Transport B1 Byte Error Count" registers. The "Receive STS-3 Transport B1 Error Count" register is actually a 32 bit register that resides at Address Locations = 0x1110 through 0x1113.

The Receive STS-3 TOH Processor block will increment these registers either by the number of erred STS-3 frames detected, or by the number of B1 bits that are detected to be in error (within a given STS-3 frame), depending upon user selection, as described below.

2.3.1.11.1 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B1 Byte Error Count" Registers on a "per-Frame" basis.

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B1 Byte Error Count" Register, by the value "1" for each STS-3 frame that it determines to have at least one bit-error within the B1 byte.

The user can accomplish this by setting Bit 0 (B1 Byte Error Type), within the "Receive STS-3 Transport Control Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Byte Error Type	B1 Byte Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

2.3.1.11.2 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B1 Error Count" register on a "per B1 bit error" basis.

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B1 Error Count" register by the number of B1 bits, which are determined to be in error. Therefore, in this mode, it is possible for the Receive STS-3 TOH Processor block to increment this register by as much as the value of "8" per STS-3 frame.

The user can accomplish this by setting Bit 0 (B1 Byte Error Type) within the "Receive STS-3 Transport Control Register – Byte 0" to "0", as illustrated below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Byte Error Type	B1 Byte Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.3.1.11.3 B1 Byte Performance Monitoring

2.3.1.12 LINE BIP-8 (B2) BYTE VERIFICATION

The Receive STS-3 TOH Processor Block has the responsibility for computing and verifying the Line BIP-8 (e.g., B2) byte within each incoming STS-3 frame. When the Receive STS-3 TOH Processor block executes this function, it will do the following.

- It will read in the contents of a given "newly received" STS-3 frame.
- It will compute the BIP-8 value over the LOH (Line Overhead) and the Envelope Capacity, within this STS-3 frame.

- This resulting BIP-8 value will be compared with the contents of the B2 byte, within the very next "newly received" STS-3 frame.

If the Receive STS-3 TOH Processor block detects any B2 byte errors, then it will do the following.

o It will generate the "Detection of B2 Byte Error" Interrupt, by toggling the "INT*" output pin "LOW" and by setting Bit 4 (Detection of B2 Byte Error Interrupt Status) within the "Receive STS-3 Transport Interrupt Status" Register to "1", as indicated below.

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

o It will increment the "Receive STS-3 Transport B2 Byte Error Count" registers. The "Receive STS-3 Transport B2 Byte Error Count" register is actually a 32 bit register that resides at Address Locations 0x1114 through 0x1117.

Note: *The Receive STS-3 TOH Processor block will increment these registers either by the number of erred STS-3 frames detected, or by the number of B2 bits that are detected to be in error (within a given STS-3 frame), depending upon user selection, as described below.*

2.3.1.12.1 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B2 Error Count" register on a "per-Frame" basis.

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B2 Byte Error Count" Register, by the value "1" for each STS-3 frame that it determined to have at least one bit-error within the B2 bytes.

The user can accomplish this by setting Bit 1 (B2 Error Type), within the "Receive STS-3 Transport Control Register – Byte 0" to "1", as illustrated below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

2.3.1.12.2 Configuring the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B2 Error Count" register on a "per B2 bit error" basis

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive STS-3 Transport B2 Error Count" Register by the number of "B2 bits, which are determined to be in error. Therefore, in this mode, it is possible for the Receive STS-3 TOH Processor block to increment this register by as much as the value of "24" for each STS-3 frame.

The user can accomplish this by setting Bit 1 (B2 Error Type) within the "Receive STS-3 Transport Control Register – Byte 0" to "0", as illustrated below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The detection of B2 byte errors also plays a role in some of the other following functions.

- The Transmission of the REI-L (Line – Remote Error Indicator) in the "upstream" direction back out the to the Remote Terminal Equipment.
- The Declaration and Clearance of the SD (Signal Degrade) and SF (Signal Fail) defect conditions.

Each of these items will be discussed in some detail in the next few sections.

2.3.1.12.3 B2 Byte Performance Monitoring

2.3.1.12.4 Transmission of REI-L in response to Detection of B2 Byte Errors

2.3.1.13 THE SD (SIGNAL DEGRADE) DECLARATION AND CLEARANCE CRITERIA

The Receive STS-3 TOH Processor block is capable of declaring and clearing the SD defect condition. Further, the Receive STS-3 TOH Processor block register set permits the user to define both the "SD Defect Declaration" and "Clearance" criteria.

Each Receive STS-3 TOH Processor block contains an SD Detector. The SD Detector accumulates B2 byte errors over a "user-defined" monitoring period of time. If the number of B2 errors (accumulated over this "user-defined" period of time) exceeds a user-defined "SD Declaration B2 Byte Error threshold", then the SD Detector will declare the "SD" Defect Condition.

Similarly, if the SD Defect is currently being declared, then the "SD Detector" will continue to accumulate B2 byte errors over another "user-defined" monitor period of time. If the number of B2 byte errors (accumulated over this "user-defined" period of time) is less than a "user-defined SD Defect Clearance B2 Byte Error" threshold, then the SD Detector will clear the SD Defect condition.

It should be noted that the Receive STS-3 TOH Processor block (within the XRT94L33) permits the user to independently specify the "SD Defect Declaration B2 Byte Error" Threshold and the "SD Defect Clearance B2 Byte Error" Threshold. As a consequence, the user can implement some sort of hysteresis within the SD defect declaration or clearance thresholds. In order to further enhance the user's ability to specify the SD defect declaration and clearance thresholds, the user can also independently specify monitoring times that are to be used in order to declare or clear the SD defect condition.

2.3.1.13.1 The Types of B2 Byte Errors

Prior to describing the behavior of the SD Detector, it is imperative to review the types of B2 byte errors that can be accumulated for the sake of SD defect declaration or clearance. The Receive STS-3 TOH Processor block can be configured to tally B2 byte errors in one of two-different manners.

- On a "per-bit" basis
- On a "per-frame" basis

If the user configures the Receive STS-3 TOH Processor block to "tally" B2 byte errors on a "per-bit" basis, then it will declare an error for each bit (within the incoming B2 byte) that is determined to be in error. In this case, the Receive STS-3 TOH Processor block can declare as many as 24 B2 byte errors, per STS-3 frame (e.g., when all 8 bit, within each of the 3 incoming B2 bytes, are in error).

Conversely, if the user configures the Receive STS-3 TOH Processor block to "tally" B2 errors on a "per-frame" basis, then it will declare a single B2 byte error, anytime at least one bit, within the 3 or 12 B2 bytes (within a given incoming STS-3 frame) is determined to be in error. In this case, the Receive STS-3 TOH Processor block will at most declare one B2 byte error, per STS-3 frame.

The user can configure the Receive STS-3 TOH Processor block to "tally" B2 errors on a "per-bit" basis, by setting Bit 1 (B2 Byte Error Type), within the "Receive STS-3 Transport Control Register – Byte 0" to "0", as depicted below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Defect Detect Enable	SD Defect Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Byte Error Type	B1 Byte Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Likewise, the user can configure the Receive STS-3 TOH Processor block to "tally" B2 Byte errors on a "per-frame" basis, by setting Bit 1 (B2 Byte Error Type) to "1", as depicted below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Defect Detect Enable	SD Defect Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Byte Error Type	B1 Byte Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This configuration option, on how the Receive STS-3 TOH Processor block "tallies" B2 byte errors impacts the following functions.

- The SD Defect Declaration and Clearance Criteria
- The SF Defect Declaration and Clearance Criteria
- The amount by which the Receive STS-3 TOH Processor block will increment the "Receive STS-3 Transport – B2 Byte Error Count" Registers.

2.3.1.13.2 The SD (Signal Degrade) Defect Declaration Criteria

The XRT94L33 permits the user to specify three parameters to define the SD Defect Declaration criteria.

- The minimum number of B2 byte errors (e.g., a B2 byte error-threshold) accumulated over a given "SD Set Interval" time period. From this point on, this particular "B2 Byte Error" threshold will be referred to as the "SD Defect Declare B2 Byte Error" Threshold.
- The length (in terms of STS-3 frame periods) of this "SD Set Interval" monitoring time period for the SD Detector to tally B2 byte errors. From this point on, this particular "user-defined" monitoring time period will be referred to as the "SD Defect Declare Monitor" time.
- The maximum number of B2 byte errors that will be counted within a sub-internal period towards the declaration of the SD Defect condition. From this point on, this maximum number of B2 byte errors will be referred to as the "SD Detect B2 Error Burst Limit".

Once the user defines these parameters, then the Receive STS-3 TOH Processor block will begin to count the cumulative number of B2 errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the "SD Defect Declare Monitor" time period.

As long as the Receive STS-3 TOH Processor block does not detect the "SD Defect Declare B2 byte error-threshold" number of B2 byte errors, within this "SD Defect Declare Monitor" time period, then it will not declare the SD Defect Condition. Conversely, if the Receive STS-3 TOH Processor block detects at least the "SD Defect Declare B2 error threshold" number of B2 byte errors, within the "SD Defect Declare Monitor time period, then it will declare the SD Defect Condition.

Figure 89 presents an illustration of the SD Detector accumulating B2 Byte errors over a "monitoring" time period.

Figure 89 A Conceptual Illustration of the SD Detector accumulating B2 Byte errors over a "Monitoring" Time period

SPECIFYING THE "SD DEFECT DECLARE B2 BYTE ERROR" THRESHOLD

The user can specify the "SD Defect Declare B2 Byte Error Threshold" by writing the appropriate value into the "Receive STS-3 Transport – Receive SD Set Threshold – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SD SET Threshold – Byte 1 (Address = 0x1142)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SD SET Threshold – Byte 0 (Address = 0x1143)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SD SET Threshold – Byte 1 and Byte 0" Registers permits the user to write in a 16-bit expression for the "SD Defect Declare B2 Byte Error Threshold".

The "default" value for the "B2 Byte Error Threshold" is 0xFFFF.

SPECIFYING THE LENGTH OF THE "SD DEFECT DECLARE MONITOR TIME" PERIOD

Likewise, the user can specify the length of the "SD Defect Declare Monitor Time" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 2, 1 and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 2 (Address = 0x113D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 1 (Address = 0x113E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 0 (Address = 0x113F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *The "Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the "SD Defect Declare Monitor Time" period. The number that is written into these registers represents the duration of the "SD Defect Declare Monitor Time" period, in terms of 1 millisecond units.*

Example of Defining the SD Declaration Criteria

Suppose that the user writes in the value "0x000F" into the "Receive STS-3 Transport – Receive SD Set Threshold" Registers; as depicted below.

Receive STS-3 Transport – Receive SD SET Threshold – Byte 1 (Address = 0x1142)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD SET Threshold – Byte 0 (Address = 0x1143)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	1	1

Additionally, now suppose that the user writes in the value "0x000010" into the "Receive STS-3 Transport – Receive SD Set Monitor Interval" Registers; as depicted below.

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 2 (Address = 0x113D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 1 (Address = 0x113E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 0 (Address = 0x113F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once the user has executed these two steps, then the "SD Declaration Criteria" will be as summarized below.

- The "SD Defect Declare B2 Byte Error" Threshold = 0x0F (or 15) B2 Errors.
- The SD Defect Declare Monitor Time Period = 0x10 (or 16ms)

Hence, the Receive STS-3 TOH Processor block will accumulate B2 byte errors over a 16ms period.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 byte errors. Based upon the "above-mentioned" configuration selections, if the Receive STS-3 TOH Processor block detects 15 or more B2 byte errors, within a given 16ms period then it will declare the SD defect condition. Conversely, if the SD Detector detects less than 15 B2 byte errors within this 16ms period, then it will NOT declare the SD Defect condition.

Error Burst Filtering of B2 Byte Errors for Declaration of the SD Defect Condition

In some applications it may be necessary to insure that the SD defect condition is declared based upon the long-term performance characteristics of a given incoming STS-3 signal, and is NOT induced by a single "burst of errors" occurring within this STS-3 signal. An example of where this feature might be useful is whenever one implements APS (Automatic Protection Switching) in response to the declaration of the SD Defect Condition. In this situation, it is undesirable to permit a single error-burst (which occurs in an "otherwise" error-free STS-3 signal) to result in APS switching to a redundant STS-3 signal, and thereby temporarily (and needlessly) disrupting ATM or PPP traffic across this APS event.

As a consequence, the SD Detector within the Receive STS-3 TOH Processor block contains an "error-burst" filter which is set by selecting the appropriate "SD Defect B2 Byte Error Burst" limit.

Setting the "SD Defect B2 Byte Error Burst" Limit

As mentioned above, the user can specify the "SD Defect Declare Monitor" Time Interval (e.g., the amount of time that the SD Detector will accumulate B2 byte errors, when determining whether to declare the SD defect or not). This particular "SD Defect Declare Monitor Time" interval is sub-divided into eight (8) "sub-interval" periods. The "Error Burst filter" permits the user to specify an upper limit to the number of B2 byte errors that the SD detector can accumulate during a given "sub-interval" period; when determining whether or not to declare the SD defect condition. The user can set the "Error Burst" filter by writing the appropriate value into the "Receive STS-3 Transport – Receive SD Burst Error Tolerance – Byte 1 & Byte 0" registers as depicted below.

Receive STS-3 Transport – Receive SD Burst Error Tolerance – Byte 1 (Address = 0x1152)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_BURST_TOLERANCE[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SD Burst Error Tolerance – Byte 0 (Address = 0x1153)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_BURST_TOLERANCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Note: *The default value for the contents within the "Receive STS-3 Transport – Receive SD Burst Error Tolerance" Register is 0xFFFF. This means that the SD Detector can receive (and tally) up to 65,535 B2 Byte errors within a given sub-interval period. Beyond this point, the SD Detector will cease to tally anymore B2 byte errors during this "sub-interval" period.*

The value that is written into these registers dictates the maximum number of B2 byte errors that the SD Detector will accumulate over a "sub-interval" period, when checking to determine if it should declare the SD defect condition.

Example of using the SD Error Burst Filter

Let's suppose that a customer wishes to configure the SD Detector to declare the SD defect condition whenever the Bit Error Rate (BER) within an incoming STS-12 signal exceeds 1x10-9. Further, let us assume that the customer wants to continuously check for a BER of 1x10-9 over a "sliding monitor" window of 167 seconds. The user must then execute the following steps in order to implement this configuration setting.

STEP 1 – Setting the "SD Defect Declare Monitor" Interval to 167 seconds

Since 167 seconds corresponds to 167,000ms, then we need to write in the value 167,000 or 0x028C58 into the "Receive STS-3 Transport – Receive SD Set Monitor Interval" Register, as depicted below.

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 2 (Address = 0x113D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 1 (Address = 0x113E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	0	0

Receive STS-3 Transport – Receive SD Set Monitor Interval – Byte 0 (Address = 0x113F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	1	1	0	0	0

STEP 2 – Setting the "SD Defect Declare B2 Error Threshold" to 1x10-9

Within 167 seconds, there will be [167 seconds • 622.08Mbits/second] = 103.9Gb of data that will have been transmitted via an STS-12 data-stream.

As a consequence, if the BER is 1x10-9, then there will be [167 seconds • 622.08Mbits/second • 1x10-9 errors/bits] = 103.9 erred bits within a 167 second period.

As a consequence, we need to write the value 104 (or 0x0068) into the "Receive STS-3 Transport – Receive SD Set Threshold" Registers as depicted below.

Receive STS-3 Transport – Receive SD SET Threshold – Byte 1 (Address = 0x1142)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD SET Threshold – Byte 0 (Address = 0x1143)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_SET_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	1	0	1	0	0	0

Based upon the configuration settings that were implemented within STEPS 1 and 2, the SD Detector will declare the SD Defect condition anytime it detects 104 B2 byte errors within a given 167 second "SD Defect Declare Monitor time" period.

Now, let's suppose that this incoming STS-12 signal is "error-free", and that the SD Detector is currently not declaring the SD defect condition; and that it has currently talled 0 B2 byte errors during the current "SD Defect Declaring Monitor time" period.

Now, let's further suppose that a burst of errors (lasting 1us) occurs within this STS-12 data-stream. We will also presume that the "timing" of this "error-burst" event is such that it straddles two STS-12 frames. If the Receive STS-12 TOH Processor block is configured to accumulate B2 byte errors in a "per-bit" manner (please see Section _ for a definition of "per-bit manner") then the SD Detector can accumulate and tally as much as 96 B2 byte errors/STS-12 x 2 STS-12 frames =) 192 B2 byte errors. In this example, this single burst of errors within the incoming STS-12 signal will cause the SD Detector to declare the SD Defect (since it has accumulated 192 B2 byte errors within a given 167 second "SD Defect Declare Monitor Time" period).

Occurrence whenever the Receive STS-3 TOH Processor block declares the SD Defect Condition

Anytime the Receive STS-3 TOH Processor block declares the SD Defect Condition, then it will do the following.

- It will generate the "Change of SD Defect Condition" Interrupt

Note: *The Receive STS-3 TOH Processor block will indicate that it is generating this interrupt by toggling the "INT*" output pin "low" and by setting the "Change of SD Defect Condition Interrupt Status" bit to "1", as depicted below.*

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

- It will set Bit 3 (SD Defect Declared), within the "Receive STS-3 Transport Status Register – Byte 0" to "1", as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

2.3.1.13.3 The SD (Signal Degrade) Defect Clearance Criteria

The XRT94L33 permits the user to specify the following two parameters to define the SD Defect Clearance criteria.

- The maximum number of B2 errors (e.g., a B2 byte error-threshold) accumulated over a given "SD Defect Clear Monitor" time period.
- The length of this "SD Defect Clear Monitor" time period.

Once the user defines these parameters, then the Receive STS-3 TOH Processor block will begin to count the cumulative number of B2 byte errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the user-defined "SD Defect Clear Monitor" time period.

If the Receive STS-3 TOH Processor block is currently declaring the SD Defect condition, and if continues to detects more than the "SD Defect Clear B2 error threshold" number of B2 errors; within the "SD Defect Clear Monitor" of time, then it will NOT clear the SD defect condition. Conversely, if the Receive STS-3 TOH Processor block detects less than the "SD Defect Clear B2 byte error threshold" number of B2 byte errors, within the "SD Defect Clear Monitor" period of time, then it will clear the SD defect condition.

Specifying the "B2 Byte Error Threshold" for Clearing the SD Defect Condition

The user can specify the "SD Defect Clear B2 Byte Error Threshold" by writing the appropriate value into the "Receive STS-3 Transport – Receive SD Clear Threshold – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SD CLEAR Threshold – Byte 1 (Address = 0x1146)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SD CLEAR Threshold – Byte 0 (Address = 0x1147)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SD Clear Threshold – Byte 1 and Byte 0" registers permits the user to write in a 16-bit expression for the "SD Defect Clear B2 Error Threshold".

The "default" value for the "SD Defect Clear B2 Error Threshold" is "0xFFFF".

Specifying the length of the "SD Defect Clear Monitor" period of Time

Likewise, the user can also specify the "SD Defect Clear Interval" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 2, 1, and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 2 (Address = 0x1159)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 1 (Address = 0x115A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 0 (Address = 0x115B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Note: *The "Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the length of the "SD Defect Clear Monitor period". The number that is written into these registers represents the duration of the "SD Clear Interval" period, in terms one millisecond units.*

Example of Defining the SD Defect Clearance Criteria

Suppose that the user writes in the value "0x0008" into the "Receive STS-3 Transport – Receive SD Defect Clear Threshold" Registers; as depicted below.

Receive STS-3 Transport – Receive SD CLEAR Threshold – Byte 1 (Address = 0x1146)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD CLEAR Threshold – Byte 0 (Address = 0x1147)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

Additionally, now suppose that the user writes in the value "0x000100" into the "Receive STS-3 Transport – Receive SD Clear Monitor Interval" Registers; as depicted below.

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 2 (Address = 0x1159)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 1 (Address = 0x115A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Receive STS-3 Transport – Receive SD Clear Monitor Interval – Byte 0 (Address = 0x115B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SD_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user has executed these two steps, then the "SD Defect Clearance Criteria" will now be as summarized below.

- The SD Defect Clear B2 Byte Error Threshold = 0x08 (or 8) B2 Byte Errors
- The SD Defect Clear Monitor period = 0x100 (or 256ms)

Hence, in order to determine whether to clear the SD defect condition or not, the SD Detector (within the Receive STS-3 TOH Processor block) will tally B2 byte errors over this 256ms period.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 byte errors. If the Receive STS-3 TOH Processor block is currently declaring the SD defect condition; it will now clear the SD defect condition if it detects less than 8 B2 byte errors, within a given 256ms period.

Occurrences whenever the Receive STS-3 TOH Processor block clears the SD Defect Condition

Anytime the Receive STS-3 TOH Processor block clears the SD Defect Condition, then it will do the following.

- It will generate the "Change of SD Defect Condition" Interrupt

Note: *The Receive STS-3 TOH Processor block will indicate that it is generating this interrupt by toggling the "INT*" output pin "low" and be setting the "Change of SD Defect Condition Interrupt Status" bit to "1", as depicted below.*

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Defect Condition Interrupt Status	Change of SD Defect Condition Interrupt Status	Detection of REI-L Error Interrupt Status	Detection of B2 Byte Error Interrupt Status	Detection of B1 Byte Error Interrupt Status	Change of LOF Defect Condition Interrupt Status	Change of SEF Defect Condition Interrupt Status	Change of LOS Defect Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

- It will set Bit 3 (SD Defect Declared) within the "Receive STS-3 Transport Status Register – Byte 0" to "0", as depicted below.

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Defect Declared	S1 Byte Unstable Defect Declared	K1, K2 Byte Unstable Defect Declared	SF Defect Declared	SD Defect Declared	LOF Defect Declared	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

2.3.1.14 SF DECLARATION AND CLEARANCE CRITERIA

The Receive STS-3 TOH Processor block is capable of declaring and clearing the SF condition. Further, the Receive STS-3c TOH Processor block register set permits the user to define the "SF Declaration" and "Clearance" criteria.

The Receive STS-3 TOH Processor block actually consists of two different "SF" Detectors.

- The "Interval" SF Detector
- The "Burst" SF Detector

The "Interval" SF Detector accumulates B2 errors over a long "user-defined" period of time. If the number of B2 errors (accumulated over this "user-defined" period of time) exceeds a user-defined "threshold", then the "Interval" SF Detector will declare an "SF" Condition.

The "Burst" SD Detector functions similarly to that of the "Interval" SF Detector, in that it also accumulates B2 errors over a "user-defined" period of time. Further, the "Burst" SF Detector will declare the SF condition if the number of B2 errors (accumulated over this "user-defined" period of time) exceeds a "user-defined" threshold, then the "Burst" SD Detector will declare an "SF Condition".

There are two main differences between the "Interval" SF Detector and the "Burst" SF Detectors.

1. The “Interval” SF Detector will accumulate B2 errors over a much greater amount of time (e.g., 8 times greater) than that of the “Burst” SF Detector; when determine whether to declare/clear the SF condition, or not.

2. The “Interval” SF Detector permits a “user-defined” hysteresis between the conditions required to declare and clear the SF condition. The “Burst” SF Detector permits no hysteresis between the conditions required to declare and clear the SF condition.

2.3.1.14.1 The Definition of B2 Errors

The Receive STS-3 TOH Processor block can be configured to tally B2 errors in one of two-different manners.

- On a “per-bit” basis
- On a “per-frame” basis

If the user configures the Receive STS-3 TOH Processor block to “tally” B2 errors on a “per-bit” basis, then it will declare an error for each bit (within the incoming B2 byte) that is determined to be in error. In this case, the Receive STS-3 TOH Processor block can declare as many as 8 B2 errors, per STS-3 frame (e.g., when all 8 bit, within the incoming B2 byte, are in error).

Conversely, if the user configures the Receive STS-3 TOH Processor block to “tally” B2 errors on a “per-frame” basis, then it will declare a single B2 error, anytime an incoming B2 byte is determined to be in error. In this case, the Receive STS-3 TOH Processor block will only declare as many as 1 B2 error, per STS-3 frame.

The user can configure the Receive STS-3 TOH Processor block to “tally” B2 errors on a “per-bit” basis, by setting Bit 1 (B2 Error Type), within the “Receive STS-3 Transport Control Register – Byte 0” to “0”, as depicted below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Likewise, the user can configure the Receive STS-3 TOH Processor block to “tally” B2 errors on a “per-frame” basis, by setting Bit 1 (B2 Error Type) to “1”, as depicted below.

Receive STS-3 Transport Control Register – Byte 0 (Address = 0x1103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SF Detect Enable	SD Detect Enable	Descramble Disable	SDH/ SONET*	REI-L Error Type	B2 Error Type	B1 Error Type
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This configuration option, on how the Receive STS-3 TOH Processor block “tallies” B2 errors impacts the following functions.

- The SF Declaration and Clearance Criteria
- The SD Declaration and Clearance Criteria
- The values that the corresponding Transmit STS-3 TOH Processor block, will transmit to the remote terminal, via the REI-L bit-fields (within the M0 byte).

- The amount by which the Receive STS-3 TOH Processor block will increment the "Receive STS-3 Transport – B2 Error Count" Registers.

The Relationship between the "Overall" SF Condition, and the States of the "Interval" and "Burst" Detectors

The "SF Condition" (as declared and cleared by the Receive STS-3 TOH Processor block) is the "logical OR" of the "SF Declaration" state of the "Interval" and "Burst" SF Detector.

In other words, the Receive STS-3 TOH Processor block will declare the SF condition, if either the "Interval" or the "Burst" SF Detector are currently declaring the "SF Condition".

The operation of the "Interval" and "Burst" SF Detectors are both described in detail below.

2.3.1.14.2 The SF (Signal Fail) Defect Declaration Criteria

In this case, the user specifies three parameters to define the SF Declaration criteria.

- The minimum number of B2 errors (e.g., a B2 error-threshold) accumulated over a given "SF Set Interval" time period.
- The length (in terms of SONET frame periods) of this "SF Set Interval" time period.

Once the user defines these parameters, then the Receive STS-3 TOH Processor block will begin to count the cumulative number of B2 errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the user-defined "SF Set Interval" time period.

As long as the Receive STS-3 TOH Processor block does not detect the "B2 error-threshold" number of B2 errors, within this "SF Set Interval" of time, then it will not declare the SF Condition. Conversely, if the Receive STS-3c TOH Processor block detects at least the "B2 error threshold" number of B2 errors, within the "SF Set Interval" of time, then it will declare the SF Condition.

SPECIFYING THE "B2 ERROR THRESHOLD" FOR DECLARING SF

The user can specify the "B2 Error Threshold" by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Set Threshold – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF SET Threshold – Byte 1 (Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF SET Threshold – Byte 0 (Address = 0x1137)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SF SET Threshold – Byte 1 and Byte 0" Registers permits the user to write in a 16-bit expression for the "B2 Error Threshold".

The "default" value for the "B2 Error Threshold" is 0xFFFF.

SPECIFYING THE "SF SET INTERVAL" OF TIME

Likewise, the user can specify the "SF Set Interval" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2, 1 and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2 (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 1 (Address = 0x1132)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 0 (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Notes:

The "Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the "SF Set Interval". The number that is written into these registers represents the duration of the "SF Set Interval" period, in terms of SONET frame periods.

The actual length of time that the Receive STS-3 TOH Processor block will use (to accumulate B2 errors) to declare the SF condition is eight times the value written into these registers.

EXAMPLE OF DEFINING THE SF DECLARATION CRITERIA – SF INTERVAL DETECTOR

Suppose that the user writes in the value "0x000F" into the "Receive STS-3 Transport – Receive SF Set Threshold" Registers; as depicted below.

Receive STS-3 Transport – Receive SF SET Threshold – Byte 1 (Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF SET Threshold – Byte 0 (Address = 0x1137)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	1	1

Additionally, now suppose that the user writes in the value "0x000010" into the "Receive STS-3 Transport – Receive SF Set Monitor Interval" Registers; as depicted below.

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2 (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 1 (Address = 0x1132)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 0 (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once the user has executed these two steps, then the "SF Declaration Criteria" will be as summarized below.

- B2 Error Threshold = 0x0F (or 15) B2 Errors.
- SF Set Interval = 0x10 (or 16 SONET frame periods)

Hence, the Receive STS-3c TOH Processor block will accumulate B2 errors over 128 (e.g., 8 * 16) SONET frame periods.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 errors. Anytime the Receive STS-3 TOH Processor block detects 15 or more B2 errors, within a 16ms period (e.g., 128 SONET frame periods); then it will declare the SF condition.

Occurrence whenever the Receive STS-3 TOH Processor block declares the SF Condition

Anytime the Receive STS-3 TOH Processor block declares the SF Condition, then it will do the following.

- ***It will generate the "Change of SF Condition" Interrupt***

Note: *The Receive STS-3 TOH Processor block will indicate that it is generating this interrupt by toggling the "INT*" output pin "low" and by setting the "Change of SF Condition Interrupt Status" bit to "1", as depicted below.*

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Condition Interrupt Status	Change of SD Condition Interrupt Status	REI-L Error Interrupt Status	B2 Error Interrupt Status	B1 Error Interrupt Status	Change of LOF Condition Interrupt Status	SEF Interrupt Status	Change of LOS Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	0

- ***It will set Bit 4 (SF Detected), within the "Receive STS-3c Transport Status Register – Byte 0" to "1", as depicted below.***

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Declared	S1 Unstable	APS Unstable	SF Detected	SD Detected	LOF Defect Detected	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	1	0	0	0	0

2.3.1.14.3 The SF (Signal Fail) Defect Clearance Criteria

In this case, the user specifies two parameters to define the SF Clearance criteria.

- The maximum number of B2 errors (e.g., a B2 error-threshold) accumulated over a given "SF Clear Interval" time period.
- The length (in terms of SONET frame periods) of this "SF Clear Interval" time period.

Once the user defines these parameters, then the Receive STS-3c TOH Processor block will begin to count the cumulative number of B2 errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the user-defined "SF Clear Interval" time period.

If the Receive STS-3c TOH Processor block is currently declaring the SF condition, and if continues to detects more than the "B2 error threshold" number of B2 errors; within the "SF Clear Interval" of time, then it will NOT clear the SF condition. Conversely, if the Receive STS-3c TOH Processor block detects less than the "B2 error threshold" number of B2 errors, within the "SF Clear Interval" of time, then it will clear the SF condition.

Specifying the "B2 Error Threshold" for Clearing SF

The user can specify the "B2 Error Threshold" by writing the appropriate value into the "Receive STS-3c Transport – Receive SF Clear Threshold – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF CLEAR Threshold – Byte 1 (Address = 0x113A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF CLEAR Threshold – Byte 0 (Address = 0x113B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SF Clear Threshold – Byte 1 and Byte 0" registers permits the user to write in a 16-bit expression for the "B2 Error Threshold".

The "default" value for the "B2 Error Threshold" is "0xFFFF".

Specifying the "SF Clear Interval" of Time

Likewise, the user can specify the "SF Clear Interval" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2, 1, and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2 (Address = 0x115D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 1 (Address = 0x115E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 0 (Address = 0x115F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the "SF Clear Interval". The number that is written into these registers represents the duration of the "SF Clear Interval" period, in terms of SONET frame periods.

The actual length of time that the Receive STS-3 TOH Processor block will use (to accumulate B2 errors) to declare the SF condition is eight times the value written into these registers.

Example of Defining the SF Clearance Criteria

Suppose that the user writes in the value "0x0008" into the "Receive STS-3c Transport – Receive SF Clear Threshold" Registers; as depicted below.

Receive STS-3 Transport – Receive SF CLEAR Threshold – Byte 1 (Address = 0x113A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_THRESHOLD[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF CLEAR Threshold – Byte 0 (Address = 0x113B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_THRESHOLD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

Additionally, now suppose that the user writes in the value "0x000100" into the "Receive STS-3 Transport – Receive SF Clear Monitor Interval" Registers; as depicted below.

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2 (Address = 0x115D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 1 (Address = 0x115E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 0 (Address = 0x115F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user has executed these two steps, then the "SF Clearance Criteria" will be as summarized below.

- B2 Error Threshold = 0x08 (or 8) B2 Errors
- SF Clear Interval = 0x100 (or 256 SONET frame periods)

Hence, the Receive STS-3 TOH Processor block will accumulate B2 errors over 2048 (e.g. 8 * 256) SONET frame periods.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 errors. If the Receive STS-3 TOH Processor block is currently declaring the SF condition; it will now clear the SF condition if it detects less than 8 B2 errors, within a 256ms period (e.g., 2048 SONET frame periods); then it will clear the SF condition.

Occurrences whenever the Receive STS-3 TOH Processor block clears the SF Condition

Anytime the Receive STS-3 TOH Processor block clears the SF Condition, then it will do the following.

It will generate the "Change of SF Condition" Interrupt

Note: *The Receive STS-3 TOH Processor block will indicate that it is generating this interrupt by toggling the "INT*" output pin "low" and be setting the "Change of SF Condition Interrupt Status" bit to "1", as depicted below.*

Receive STS-3 Transport Interrupt Status Register – Byte 0 (Address = 0x110B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Change of SF Condition Interrupt Status	Change of SD Condition Interrupt Status	REI-L Error Interrupt Status	B2 Error Interrupt Status	B1 Error Interrupt Status	Change of LOF Condition Interrupt Status	SEF Interrupt Status	Change of LOS Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	0

- ***It will set Bit 4 (SF Detected) within the "Receive STS-3 Transport Status Register – Byte 0" to "0", as depicted below.***

Receive STS-3 Transport Status Register – Byte 0 (Address = 0x1107)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RDI-L Declared	S1 Unstable	APS Unstable	SF Detected	SD Detected	LOF Defect Detected	SEF Defect Declared	LOS Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

SF DECLARATION CRITERIA – per the "Burst" SF Detector

In this case, the user specifies two parameters to define the SF Declaration criteria.

- The minimum number of B2 errors (e.g., a B2 error-threshold) accumulated over a given "SF Set Interval" time period.
- The length (in terms of SONET frame periods) of this "SF Set Interval" time period.

Once the user defines these parameters, then the Receive STS-3 TOH Processor block will begin to count the cumulative number of B2 errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the user-defined "SF Set Interval" time period.

As long as the Receive STS-3 TOH Processor block does not detects the "B2 error threshold" number of B2 errors, within this "SF Set Interval" of time, then it will not declare the SF Condition. Conversely, if the Receive STS-3 TOH Processor block detects at least the "B2 error threshold" number of B2 errors, within the "SF Set Interval" of time, then it will declare the SF Condition.

Specifying the "B2 Error Threshold" for Declaring SF

The user can specify the "B2 Error Threshold" by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 0 (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 and Byte 0" registers permit the user to write in a 16-bit expression for the "B2 Error Threshold" for the "SF Burst Detector".

The "default" value for the "B2 Error Threshold" is 0xFFFF.

SPECIFYING THE "SF SET INTERVAL" OF TIME

Likewise, the user can specify the "SF Set Interval" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2, 1, and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2 (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 1 (Address = 0x1132)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 0 (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Notes:

The "Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the "SF Set Interval". The number that is written into these registers represents the duration of the "SF Set Interval" period, in terms of SONET frame periods.

The actual length of time that the Receive STS-3 TOH Processor block will use (to accumulate B2 errors) to declare the SF condition is value written into these registers.

EXAMPLE OF DEFINING THE SF DECLARATION CRITERIA – SF BURST DETECTOR

Suppose that the user writes in the value "0x000F" into the "Receive STS-3 Transport – Receive SF Burst Error Tolerance" Registers; as depicted below.

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 0 (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	1	1

Additionally, now suppose that the user writes in the value “0x000010” into the “Receive STS-3 Transport – Receive SF Set Monitor Interval” Register, as depicted below.

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 2 (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_SET_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 1 (Address = 0x1132)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SF_SET_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Set Monitor Interval – Byte 0 (Address = 0x1133)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SF_SET_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once the user has executed these two steps, then the “SF Declaration Criteria” (for the SF Burst Detector) will be as summarized below.

- B2 Error Threshold = 0x0F (or 15) B2 Errors.
- SF Set Interval = 0x10 (or 16 SONET frame periods)

Hence, the Receive STS-3 TOH Processor block will accumulate B2 errors over 16 SONET frame periods.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 errors. Anytime the Receive STS-3 TOH Processor block detects 15 or more B2 errors, within a 2ms period (e.g., 16 SONET frame periods); then it will declare the SF condition.

SF CLEARANCE CRITERIA – PER THE "BURST" SF DETECTOR

In this case, the user specifies two parameters to define the SF Clearance criteria.

- The maximum number of B2 errors (e.g., a B2 error-threshold) accumulated over a given "SF Clear Interval" time period.
- The length (in terms of SONET frame periods) of this "SF Clear Interval" time period.

Once the user defines these parameters, then the Receive STS-3 TOH Processor block will begin to count the cumulative number of B2 errors that it detects within a "sliding window" of time. The length of this "sliding window of time" is dictated by the user-defined "SF Clear Interval" time period.

If the Receive STS-3 TOH Processor block is currently declaring the SF condition, and if it continues to detect more than the "B2 error threshold" number of B2 errors; within the "SF Clear Interval" of time, then it will NOT clear the SF condition. Conversely, if the Receive STS-3 TOH Processor block detects less than the "B2 error threshold" number of B2 errors, within the "SF Clear Interval" of time, then it will clear the SF condition.

SPECIFYING THE "B2 ERROR THRESHOLD" FOR CLEARING SF – THE SF "BURST" DETECTOR

The user can specify the "B2 Error Threshold" by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 and Byte 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 0 (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport- Receive SF Burst Error Tolerance – Byte 1 and Byte 0" registers permit the user to write in a 16-bit expression for the "B2 Error Threshold".

The "default" value for the "B2 Error Threshold" is "0xFFFF".

The "Receive STS-3 Transport – Receive SF Burst Error Tolerance" registers are used to set both the "SF Declaration" and "SF Clearance" criteria. Therefore, any value that the user writes into this register (to set the "SF Clearance" criteria – per the "SF Burst" Detector) will also effect the "SF Declaration" criteria (per the "SF Burst" Detector).

SPECIFYING THE "SF CLEAR INTERVAL" OF TIME

Likewise, the user can specify the "SF Clear Interval" period by writing the appropriate value into the "Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2, 1 and 0" registers, as depicted below.

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2 (Address = 0x115D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 1 (Address = 0x115E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 0 (Address = 0x115F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Notes:

The "Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2, Byte 1 and Byte 0" registers permit the user to write in a 24-bit expression for the "SF Clear Interval". The number that is written into these registers represents the duration of the "SD Clear Interval" periods, in terms of SONET frame periods.

The actual length of time that the Receive STS-3 TOH Processor block will use (to accumulate B2 errors) to declare the SF condition is the value written into these registers.

Examples of Defining the SF Clearance Criteria – per the "SF Burst" Detector

Suppose that the user writes in the value "0x0008" into the "Receive STS-3c Transport – Receive SF Burst Error Tolerance" Registers; as depicted below.

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 1 (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Burst Error Tolerance – Byte 0 (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_BURST_TOLERANCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

Additionally, now suppose that the user writes in the value "0x000100" into the "Receive STS-3 Transport – Receive SF Clear Monitor Interval" Registers; as depicted below.

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 2 (Address = 0x115D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 1 (Address = 0x115E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Receive STS-3 Transport – Receive SF Clear Monitor Interval – Byte 0 (Address = 0x115F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SF_CLEAR_MONITOR_WINDOW[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user has executed these two steps, then the "SF Clearance Criteria" will be as summarized below.

- B2 Error Threshold = 0x08 (or 8) B2 Errors
- SF Clear Interval = 0x100 (or 256 SONET frame periods)

Hence, the Receive STS-3 TOH Processor block will accumulate B2 errors over 256 SONET frame periods.

At this point, the Receive STS-3 TOH Processor block will proceed to count B2 errors. If the Receive STS-3 TOH Processor block is currently declaring the SF condition; it will now clear the SF condition if it detects less than 8 B2 errors, within a 32ms period (e.g., 256 SONET frame periods); then it will clear the SF condition.

2.3.1.15 HANDLING/SUPPORT OF THE INCOMING K1, K2 BYTES

2.3.1.16 HANDLING/SUPPORT OF THE INCOMING LINE DCC (D4 – D12) BYTES

2.3.1.17 HANDLING/SUPPORT OF THE INCOMING S1 BYTE

2.3.1.18 HANDLING/SUPPORT OF THE INCOMING M1 BYTE

2.3.1.19 HANDLING/SUPPORT OF THE INCOMING E2 BYTE

2.3.1.20 RECEIVE STS-3 TOH PROCESSOR BLOCK INTERRUPTS

As described throughout much of this section the Receive STS-3 TOH Processor block will generate an interrupt in response to any of the following conditions.

- Change in the LOS Defect Condition
- Change in the SEF Defect Condition
- Change in the LOF Defect Condition
- Change in the AIS-L Defect Condition
- Change in the SD Defect Condition
- Change in the SF Defect Condition
- Change in the Section-Trace Message Mismatch Defect Condition

2.3.1.20.1 CHANGE OF LOS DEFECT CONDITION INTERRUPT

2.3.1.20.2 CHANGE OF SEF DEFECT CONDITION INTERRUPT

2.3.1.20.3 CHANGE OF LOF DEFECT CONDITION INTERRUPT

2.3.1.20.4 CHANGE OF AIS-L DEFECT CONDITION INTERRUPT

2.3.1.20.5 DETECTION OF B1 BYTE ERROR INTERRUPT

2.3.1.20.6 DETECTION OF B2 BYTE ERROR INTERRUPT

2.3.1.20.7 CHANGE OF SD (SIGNAL DEGRADE) DEFECT CONDITION INTERRUPT

2.3.1.20.8 CHANGE OF SF (SIGNAL FAIL) DEFECT CONDITION

2.3.1.20.9 DETECTION OF REI-L INTERRUPT

2.3.1.20.10 CHANGE OF SECTION TRACE MESSAGE MISMATCH DEFECT CONDITION

2.3.1.20.11 CHANGE OF SECTION TRACE MESSAGE UNSTABLE DEFECT CONDITION

2.3.1.21 RECEIVE STS-3 TOH PROCESSOR BLOCK PERFORMANCE MONITORING PARAMETERS

The XRT94L33 supports Performance Monitoring by providing 32-bit Reset-upon-Read Registers, that track the following parameters.

- B1 Byte Errors
- B2 Byte Errors
- Number of REI-L Events

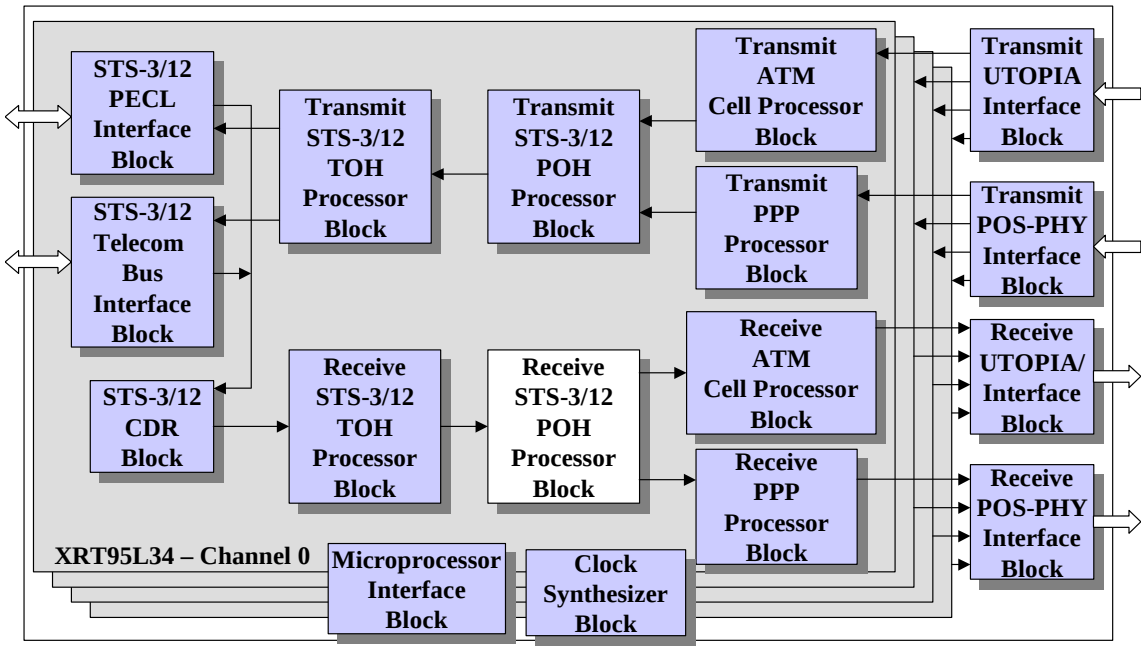
2.3.2 RECEIVE STS-3C POH PROCESSOR BLOCK (FOR STS-3C APPLICATIONS)

The purpose of the Receive STS-3c POH Processor block is to accomplish the following.

- To receive an STS-3c SPE from the Receive STS-3 TOH Processor block
- To acquire and maintain the location of the STS-3c SPE, within the incoming STS-3c data-stream
- To compute and verify the B3 bytes and increment performance monitor registers anytime it detects B3 byte errors.
- To declare and clear the following defect conditions.
    - LOP-P (Loss of Pointer)
    - AIS-P (Path AIS)
    - RDI-P (Path – Remote Defect Indicator)
    - PLM-P (Path – Payload Label Mismatch)
    - UNEQ-P (Path – Unequipped)
    - TIM-P (Path – Trace Identification Mismatch)
- To increment performance monitor registers anytime it detects an REI-P event.
- To receive either 1-byte, 16-byte or 64-byte Path Trace Identification Messages via the J1 byte within each incoming STS-3c SPE; and to detect and declare the TIM-P defect condition when appropriate.

Figure 90 presents an illustration of the Functional Block Diagram of the XRT94L33 Mapper IC; with the “Receive STS-3c POH Processor” block highlighted.

Figure 90 Illustration of the Functional Block Diagram of the XRT94L33 Mapper IC; with the Receive STS-3c POH Processor block highlighted



The operation of the "Receive STS-3c POH Processor" block is discussed in some detail below.

2.3.2.1 POINTER PROCESSING

As the XRT94L33 receives an incoming STS-3 data-stream, the Receive STS-3 TOH Processor block has the responsibility of de-scrambling this incoming data-stream, and acquiring and maintaining frame synchronization with the incoming STS-3 frames. Once these incoming STS-3 frames have been located; the Receive STS-3c POH Processor block has the responsibility of locating and keeping track of the STS-3c SPEs within this STS-3 data-stream.

As required by the SONET/SDH standards, the Receive STS-3c POH Processor blocks accomplish this by monitoring the contents of each set of H1 and H2 bytes within the incoming STS-3 data-stream.

If a Given Channel is Receiving an STS-3c Signal

If a given channel is configured to receive an STS-3c signal, then each incoming STS-3 frame consists of a single STS-3c SPE. The Receive STS-3c POH Processor block determines the location of this one STS-3c SPE by the contents of the H1 and H2 byte the TOH of the incoming STS-3c data-stream.

The H1 and H2 Bytes within an STS-3c Signal

2.3.2.1.1 Handling Incrementing Pointer Adjustment Events

2.3.2.1.2 Handling Decrementing Pointer Adjustment Events

2.3.2.1.3 Handling NDF (New Data Flag) Events

2.3.2.1.4 LOP-P DECLARATION AND CLEARANCE CRITERIA

2.3.2.1.5 AIS-P DECLARATION AND CLEARANCE CRITERIA

2.3.2.2 PROCESSING/HANDLING THE G1 BYTE

2.3.2.2.1 RDI-P DETECTION AND CLEARANCE CRITERIA

2.3.2.2.2 DETECTING/FLAGGING REI-P EVENTS

The Receive STS-3c POH Processor block is capable of detecting the REI-P indicator, within the incoming STS-3c SPE data-stream. As the Receive STS-3c POH Processor block receives a given STS-3c SPE data-stream, it will monitor the contents within Bits _ through _ in the G1 byte. The bit-format of the G1 byte is presented below in Figure 91.

Figure 91 Bit format of the G1 Byte

Figure 91 indicates that Bits _ through _, within the G1 byte are allocated for the REI-P function.

The role of the REI-P bit-fields was described in some detail, in Section _. This section indicates that the remote PTE will set the "REI-P" value (within the G1 byte) to "0" during "un-erred" conditions. However, the remote PTE will typically set the "REI-P" value to a value (ranging from "1" to "8") during "erred" conditions.

If the Receive STS-3c POH Processor block receives an STS-3c SPE, that contains a "non-zero" value of REI-P, then it will do the following.

1. It will generate the "Detection of REI-P Event" Interrupt.

Note: *The Receive STS-3c POH Processor block will indicate this by, pulling the "INT*" output pin "LOW" and by setting Bit 6 (Detection of REI-P Event Interrupt Status), within the "Receive STS-3c Path – SONET Receive Path Interrupt Status – Byte 1" to "1" as depicted below.*

Receive STS-3c Path – SONET Receive Path Interrupt Status – Byte 1 (Address = 0x118A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Detection of REI-P Event Interrupt Status	Change in UNEQ-P Defect Condition Interrupt Status	Change in PLM-P Defect Condition Interrupt Status	New C2 Byte Interrupt Status	Change in C2 Byte Unstable Defect Condition Interrupt Status	Change in RDI-P Unstable Defect Condition Interrupt Status	New RDI-P Value Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

2. It will increment the "Receive STS-3c Path – REI-P Error Count" Registers

NOTE: These registers are actually 32-bit registers, which are located at Direct Address locations 0xNA9C through 0xNA9F. The bit-format of these registers is presented below.

Receive STS-3c Path – REI-P Error Count Register – Byte 3 (Address = 0x119C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3c Path – REI_P Error Count Register – Byte 2 (Address = 0x119D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3c Path – REI_P Error Count Register – Byte 1 (Address = 0x119E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive STS-3c Path – REI_P Error Count Register – Byte 0 (Address = 0x119F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Note: *The Receive STS-3c POH Processor block will increment these registers either by the number of erred STS-3c SPE (e.g., STS-3c SPEs with a non-zero REI-P value) detected, or by the value of the REI-P value, within a given STS-3c frame; depending upon user selection, as described below.*

Configuring the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – REI-P Error Count" Register on a "per-STS-3c SPE" basis.

The user can configure the Receive STS-3c TOH Processor block to increment the "Receive STS-3c Path – REI-P Error Count" Register, by the value "1" for each STS-3c SPE that contains a "non-zero" REI-P value.

The user can accomplish this by setting Bit 1 (REI-P Error Type), within the "Receive STS-3c Path – Control Register – Byte 0" to "1" as illustrated below.

Receive STS-3c Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

Configuring the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – REI-P Error Count" Register on an "REI-P Value" basis.

The user can configure the Receive STS-3c TOH Processor block to increment the "Receive STS-3c Path – REI-P Error Count" Register by the contents within the "REI-P" nibbles, within each incoming STS-3c SPE. Therefore, in this mode, it is possible for the Receive STS-3c POH Processor block to increment this register by as much as the value "8" per STS-3c SPE.

The user can accomplish this by setting Bit 1 (REI-P Error Type) within the "Receive STS-3c Path – Control Register – Byte 0" to "0", as illustrated below.

Receive STS-3c Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.3.2.3 PATH BIP-8 (B3) BYTE VERIFICATION

Configuring the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – B3 Error Count" Register on a "per-SPE" basis.

The Receive STS-3c POH Processor block has the responsibility for computing and verifying the Path BIP-8 (e.g., B3) byte within each incoming STS-3c SPE. When the Receive STS-3c POH Processor block executes this function, it will do the following.

- It will read in the contents of a given "newly received" STS-3c SPE.
- It will compute the BIP-8 value over the SPE.
- This resulting BIP-8 value will be compared with the contents of the B3 byte, within the very next STS-3c SPE.

The user can configure the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – B3 Error Count" Register, by the value "1" for each STS-3c SPE that it determined to have a bit-error.

The user can accomplish this by setting Bit 0 (B3 Error Type), within the "Receive STS-3c Path – Control Register – Byte 0" to "1", as illustrated below.

Receive STS-3c Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Note: *This the user implements this setting, then the corresponding Transmit STS-3c POH Processor block will set the REI-P nibble value (within the G1 byte) to the number of erred SPE that have been detected. In this case, the maximum value that the REI-P nibble (within an STS-3c SPE) will contain will be "1".*

If the Receive STS-3c POH Processor block detects any B3 byte errors, then it will do the following.

a. It will generate the "Detection of B3 Error" Interrupt, by toggling the "INT*" output pin "LOW" and by setting Bit 7 (Detection of B3 Byte Error Interrupt Status) to "1" as indicated below.

Receive STS-3c Path – SONET Receive Path Interrupt Status – Byte 0 (Address = 0x118B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of B3 Byte Error Interrupt Status	Detection of New Pointer Interrupt Status	Detection of Unknown Pointer Interrupt Status	Detection of Pointer Decrement Interrupt Status	Detection of Pointer Increment Interrupt Status	Detection of NDF Pointer Interrupt Status	Change of LOP-P Condition Interrupt Status	Change of AIS-P Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	0

b. It will increment the "Receive STS-3c Path – B3 Error Count" registers. The "Receive STS-3c Path – B3 Error Count" register is actually a 32 bit register that resides at Direct Address 0xNA98 – 0xNA9B.

Note: *The Receive STS-3c POH Processor block will increment these registers either by the number of erred STS-3c SPEs detected, or by the number of B3 bits that are detected to be in error (within a given STS-3c frame), depending upon user selection, as described below.*

Configuring the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – B3 Error Count" register on a "per B3 bit-error" basis

The user can configure the Receive STS-3c POH Processor block to increment the "Receive STS-3c Path – B3 Error Count" Register by the number of "B3 bits" which are determined to be in error. Therefore, in this mode, it is possible for the Receive STS-3c POH Processor block to increment this register by as much as the value of "8" per STS-3c SPE.

The user can accomplish this by setting Bit 0 (B3 Error Type) within the "Receive STS-3c Path – Control Register – Byte 0" to "0", as illustrated below.

Receive STS-3c Path – Control Register – Byte 0 (Address = 0x1183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Note: *If the user implements this setting, then the corresponding Transmit STS-3c POH Processor block will set the REI-P nibble value (within the G1 byte) to the number of B3 bits that have been determined to be in error. In this case, the REI-P nibble value can contain a number as high as "8" for each "outbound" STS-3c frame.*

The detection of B3 byte errors also plays a role in the transmission of the REI-P (Path – Remote Error Indicator) back out to the Remote Terminal Equipment. This item will be discussed in some detail in Section _.

2.3.2.4 HANDLING/PROCESSING THE C2 BYTE

2.3.2.4.1 UNEQ-P DECLARATION AND CLEARANCE CRITERIA

2.3.2.4.2 PLM-P DECLARATION AND CLEARANCE CRITERIA

2.3.2.5 RECEIVING/PROCESSING INCOMING PATH TRACE MESSAGES VIA THE J1 BYTE

2.3.2.5.1 TIM-P DECLARATION AND CLEARANCE CRITERIA

2.3.2.5.2 PATH TRACE UNSTABLE DEFECT DECLARATION AND CLEARANCE CRITERIA

2.3.2.6 RECEIVING/PROCESSING THE F2 BYTE WITHIN THE INCOMING STS-3c SPE DATA STREAM

2.3.2.7 RECEIVING/PROCESSING THE H4 BYTE WITHIN THE INCOMING STS-3c SPE DATA STREAM

2.3.2.8 RECEVING/PROCESSING THE Z3 BYTE WITHIN THE INCOMING STS-3c SPE DATA-STREAM

2.3.2.9 RECEIVING/PROCESSING THE Z4 BYTE WITHIN THE INCOMING STS-3c SPE DATA-STREAM

2.3.2.10 RECEIVING/PROCESSING THE Z5 BYTE WITHIN THE INCOMING STS-3c SPE DATA-STREAM

2.3.2.11 INTERRUPT PROCESSING WITHIN THE RECEIVE STS-3c POH PROCESSOR BLOCK

2.3.2.11.1 CHANGE OF LOP-P DEFECT CONDITION INTERRUPT

2.3.2.11.2 CHANGE OF AIS-P DEFECT CONDITION INTERRUPT

2.3.2.11.3 DETECTION OF B3 BYTE ERROR INTERRUPT

2.3.2.11.4 CHANGE OF UNEQ-P DEFECT CONDITION INTERRUPT

2.3.2.11.5 CHANGE OF PLM-P DEFECT CONDITION INTERRUPT

2.3.2.11.6 DETECTION OF REI-P INTERRUPT

2.3.2.11.7 CHANGE OF TIM-P DEFECT CONDITION INTERRUPT

2.3.2.11.8 CHANGE OF PATH TRACE MESSAGE UNSTABLE CONDITION INTERRUPT

2.3.2.12 PERFORMANCE MONITORING WITH THE RECEIVE STS-3c POH PROCESSOR BLOCK

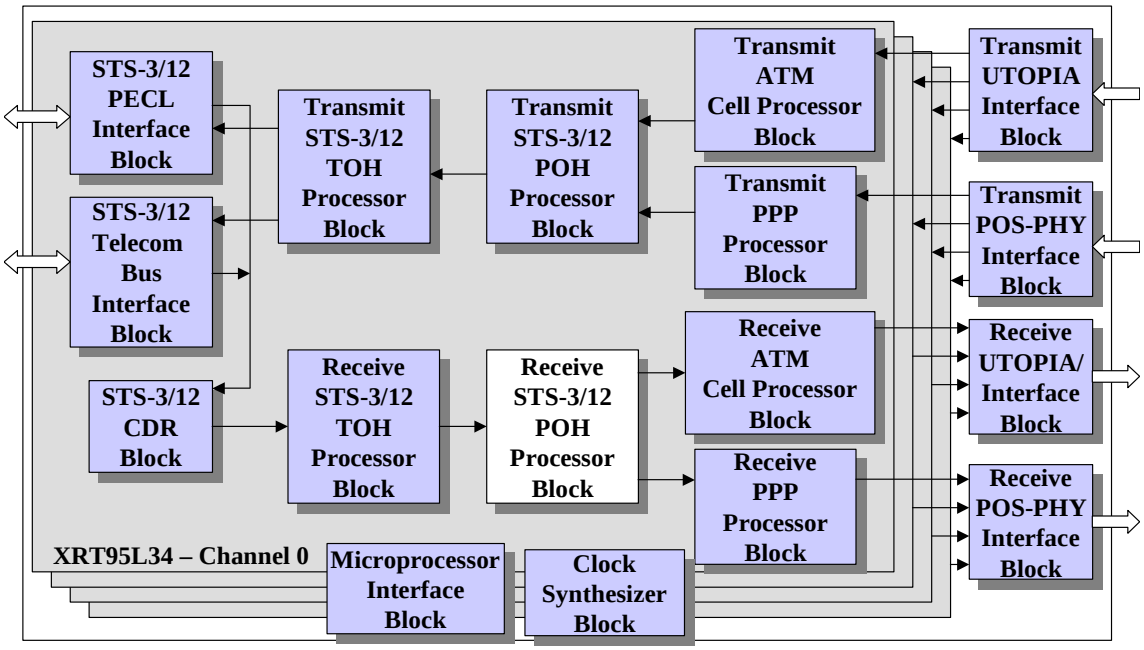
2.3.3 RECEIVE SONET POH PROCESSOR BLOCK

The purpose of the Receive SONET POH Processor block is to accomplish the following.

- To receive an STS-1 SPE from the Receive STS-3 TOH Processor block
- To acquire and maintain the location of the STS-1 SPE, within the incoming STS-1 data-stream
- To compute and verify the B3 bytes and increment performance monitor registers anytime it detects B3 byte errors.
- To declare and clear the following defect conditions.
    - LOP-P (Loss of Pointer)
    - AIS-P (Path AIS)
    - RDI-P (Path – Remote Defect Indicator)
    - PLM-P (Path – Payload Label Mismatch)
    - UNEQ-P (Path – Unequipped)
    - TIM-P (Path – Trace Identification Mismatch)
- To increment performance monitor registers anytime it detects an REI-P event.
- To receive either 1-byte, 16-byte or 64-byte Path Trace Identification Messages via the J1 byte within each incoming STS-1 SPE; and to detect and declare the TIM-P defect condition when appropriate.

Figure 92 presents an illustration of the Functional Block Diagram of the XRT94L33 Mapper IC; with the “Receive SONET POH Processor” block highlighted.

Figure 92 Illustration of the Functional Block Diagram of the XRT94L33 Mapper IC; with the Receive SONET POH Processor block highlighted



2.3.3.1 POINTER PROCESSING

As the XRT94L33 receives an incoming STS-3 data-stream, the Receive STS-3 TOH Processor block has the responsibility of de-scrambling this incoming data-stream, and acquiring and maintaining frame synchronization with the incoming STS-3 frames. Once these incoming STS-3 frames have been located; the Receive SONET POH Processor block has the responsibility of locating and keeping track of the STS-1 SPEs within this STS-3 data-stream.

As required by the SONET/SDH standards, the Receive SONET POH Processor blocks accomplish this by monitoring the contents of each set of H1 and H2 bytes within the incoming STS-3 data-stream.

If a Given Channel is Receiving an STS-1 Signal

If a given channel is configured to receive an STS-1 signal, then each incoming STS-3 frame consists of a single STS-1 SPE. The Receive SONET POH Processor block determines the location of this one STS-1 SPE by the contents of the H1 and H2 byte the TOH of the incoming STS-1 data-stream.

The H1 and H2 Bytes within an STS-1 Signal

2.3.3.1.1 Handling Incrementing Pointer Adjustment Events

2.3.3.1.2 Handling Decrementing Pointer Adjustment Events

2.3.3.1.3 Handling NDF (New Data Flag) Events

2.3.3.1.4 LOP-P DECLARATION AND CLEARANCE CRITERIA

2.3.3.1.5 AIS-P DECLARATION AND CLEARANCE CRITERIA

2.3.3.2 PROCESSING/HANDLING THE G1 BYTE

2.3.3.2.1 RDI-P DETECTION AND CLEARANCE CRITERIA

2.3.3.2.2 DETECTING/FLAGGING REI-P EVENTS

The Receive SONET POH Processor block is capable of detecting the REI-P indicator, within the incoming STS-1 SPE data-stream. As the Receive SONET POH Processor block receives a given STS-1 SPE data-stream, it will monitor the contents within Bits _ through _ in the G1 byte. The bit-format of the G1 byte is presented below in Figure 93

Figure 93 Bit format of the G1 Byte

Figure 93 indicates that Bits _ through _, within the G1 byte are allocated for the REI-P function.

The role of the REI-P bit-fields was described in some detail, in Section _. This section indicates that the remote PTE will set the "REI-P" value (within the G1 byte) to "0" during "un-erred" conditions. However, the remote PTE will typically set the "REI-P" value to a value (ranging from "1" to "8") during "erred" conditions.

If the Receive SONET POH Processor block receives an STS-1 SPE, that contains a "non-zero" value of REI-P, then it will do the following.

1. It will generate the "Detection of REI-P Event" Interrupt.

Note: *The Receive SONET POH Processor block will indicate this by, pulling the "INT*" output pin "LOW" and by setting Bit 6 (Detection of REI-P Event Interrupt Status), within the "Receive SONET Path – SONET Receive Path Interrupt Status – Byte 1" to "1" as depicted below.*

Receive SONET Path – SONET Receive Path Interrupt Status – Byte 1 (Address = 0xN18A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Detection of REI-P Event Interrupt Status	Change in UNEQ-P Defect Condition Interrupt Status	Change in PLM-P Defect Condition Interrupt Status	New C2 Byte Interrupt Status	Change in C2 Byte Unstable Defect Condition Interrupt Status	Change in RDI-P Unstable Defect Condition Interrupt Status	New RDI-P Value Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

2. It will increment the "Receive SONET Path – REI-P Error Count" Registers

Note: *These registers are actually 32-bit registers, which are located at Direct Address locations 0xNA9C through 0xNA9F. The bit-format of these registers is presented below.*

Receive SONET Path – REI-P Error Count Register – Byte 3 (Address = 0xN19C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive SONET Path – REI_P Error Count Register – Byte 2 (Address = 0xN19D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive SONET Path – REI_P Error Count Register – Byte 1 (Address = 0xN19E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive SONET Path – REI_P Error Count Register – Byte 0 (Address = 0xN19F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
REI_P_Error_Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Note: *The Receive SONET POH Processor block will increment these registers either by the number of erred STS-1 SPE (e.g., STS-1 SPEs with a non-zero REI-P value) detected, or by the value of the REI-P value, within a given STS-1 frame; depending upon user selection, as described below.*

Configuring the Receive SONET POH Processor block to increment the "Receive SONET Path – REI-P Error Count" Register on a "per-STS-1 SPE" basis.

The user can configure the Receive STS-1 TOH Processor block to increment the "Receive SONET Path – REI-P Error Count" Register, by the value "1" for each STS-1 SPE that contains a "non-zero" REI-P value.

The user can accomplish this by setting Bit 1 (REI-P Error Type), within the "Receive SONET Path – Control Register – Byte 0" to "1" as illustrated below.

Receive SONET Path – Control Register – Byte 0 (Address = 0xN183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

Configuring the Receive SONET POH Processor block to increment the "Receive SONET Path – REI-P Error Count" Register on an "REI-P Value" basis.

The user can configure the Receive STS-3 TOH Processor block to increment the "Receive SONET Path – REI-P Error Count" Register by the contents within the "REI-P" nibbles, within each incoming STS-1 SPE. Therefore, in this mode, it is possible for the Receive SONET POH Processor block to increment this register by as much as the value "8" per STS-1 SPE.

The user can accomplish this by setting Bit 1 (REI-P Error Type) within the "Receive SONET Path – Control Register – Byte 0" to "0", as illustrated below.

Receive SONET Path – Control Register – Byte 0 (Address = 0xN183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

2.3.3.3 PATH BIP-8 (B3) BYTE VERIFICATION

Configuring the Receive SONET POH Processor block to increment the "Receive SONET Path – B3 Error Count" Register on a "per-SPE" basis.

The user can configure the Receive SONET POH Processor block to increment the "Receive SONET Path – B3 Error Count" Register, by the value "1" for each STS-1 SPE that it determined to have a bit-error.

The user can accomplish this by setting Bit 0 (B3 Error Type), within the "Receive SONET Path – Control Register – Byte 0" to "1", as illustrated below.

Receive SONET Path – Control Register – Byte 0 (Address = 0xN183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Note: *This the user implements this setting, then the corresponding Transmit SONET POH Processor block will set the REI-P nibble value (within the G1 byte) to the number of erred SPE that have been detected. In this case, the maximum value that the REI-P nibble (within an STS-1 SPE) will contain will be "1".*

The Receive SONET POH Processor block has the responsibility for computing and verifying the Path BIP-8 (e.g., B3) byte within each incoming STS-1 SPE. When the Receive SONET POH Processor block executes this function, it will do the following.

- It will read in the contents of a given "newly received" STS-1 SPE.
- It will compute the BIP-8 value over the SPE.
- This resulting BIP-8 value will be compared with the contents of the B3 byte, within the very next STS-1 SPE.

If the Receive SONET POH Processor block detects any B3 byte errors, then it will do the following.

a. It will generate the "Detection of B3 Error" Interrupt, by toggling the "INT*" output pin "LOW" and by setting Bit 7 (Detection of B3 Byte Error Interrupt Status) to "1" as indicated below.

Receive SONET Path – SONET Receive Path Interrupt Status – Byte 0 (Address = 0xN18B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of B3 Byte Error Interrupt Status	Detection of New Pointer Interrupt Status	Detection of Unknown Pointer Interrupt Status	Detection of Pointer Decrement Interrupt Status	Detection of Pointer Increment Interrupt Status	Detection of NDF Pointer Interrupt Status	Change of LOP-P Condition Interrupt Status	Change of AIS-P Condition Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	0

b. It will increment the "Receive SONET Path – B3 Error Count" registers. The "Receive SONET Path – B3 Error Count" register is actually a 32 bit register that resides at Direct Address 0xNA98 – 0xNA9B.

Note: *The Receive SONET POH Processor block will increment these registers either by the number of erred STS-1 SPEs detected, or by the number of B3 bits that are detected to be in error (within a given STS-1 frame), depending upon user selection, as described below.*

Configuring the Receive SONET POH Processor block to increment the "Receive SONET Path – B3 Error Count" register on a "per B3 bit-error" basis

The user can configure the Receive SONET POH Processor block to increment the "Receive SONET Path – B3 Error Count" Register by the number of "B3 bits" which are determined to be in error. Therefore, in this mode, it is possible for the Receive SONET POH Processor block to increment this register by as much as the value of "8" per STS-1 SPE.

The user can accomplish this by setting Bit 0 (B3 Error Type) within the "Receive SONET Path – Control Register – Byte 0" to "0", as illustrated below.

Receive SONET Path – Control Register – Byte 0 (Address = 0xN183)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Check Stuff	RDI-P Type	REI-P Error Type	B3 Error Type
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

Note: *If the user implements this setting, then the corresponding Transmit SONET POH Processor block will set the REI-P nibble value (within the G1 byte) to the number of B3 bits that have been determined to be in error. In this case, the REI-P nibble value can contain a number as high as "8" for each "outbound" STS-1 frame.*

The detection of B3 byte errors also plays a role in the transmission of the REI-P (Path – Remote Error Indicator) back out to the Remote Terminal Equipment. This item will be discussed in some detail in Section _.

2.3.3.4 HANDLING/PROCESSING THE C2 BYTE

2.3.3.4.1 UNEQ-P DECLARATION AND CLEARANCE CRITERIA

2.3.3.4.2 PLM-P DECLARATION AND CLEARANCE CRITERIA

2.3.3.5 RECEIVING/PROCESSING INCOMING PATH TRACE MESSAGES VIA THE J1 BYTE

2.3.3.5.1 TIM-P DECLARATION AND CLEARANCE CRITERIA

2.3.3.5.2 PATH TRACE UNSTABLE DEFECT DECLARATION AND CLEARANCE CRITERIA

2.3.3.6 RECEIVING/PROCESSING THE F2 BYTE WITHIN THE INCOMING STS-1 SPE DATA STREAM

2.3.3.7 RECEIVING/PROCESSING THE H4 BYTE WITHIN THE INCOMING STS-1 SPE DATA STREAM

2.3.3.8 RECEVING/PROCESSING THE Z3 BYTE WITHIN THE INCOMING STS-1 SPE DATA-STREAM

2.3.3.9 RECEIVING/PROCESSING THE Z4 BYTE WITHIN THE INCOMING STS-1 SPE DATA-STREAM

2.3.3.10 RECEIVING/PROCESSING THE Z5 BYTE WITHIN THE INCOMING STS-1 SPE DATA-STREAM

2.3.3.11 INTERRUPT PROCESSING WITHIN THE RECEIVE SONET POH PROCESSOR BLOCK

2.3.3.11.1 CHANGE OF LOP-P DEFECT CONDITION INTERRUPT

2.3.3.11.2 CHANGE OF AIS-P DEFECT CONDITION INTERRUPT

2.3.3.11.3 DETECTION OF B3 BYTE ERROR INTERRUPT

2.3.3.11.4 CHANGE OF UNEQ-P DEFECT CONDITION INTERRUPT

2.3.3.11.5 CHANGE OF PLM-P DEFECT CONDITION INTERRUPT

2.3.3.11.6 DETECTION OF REI-P INTERRUPT

2.3.3.11.7 CHANGE OF TIM-P DEFECT CONDITION INTERRUPT

2.3.3.11.8 CHANGE OF PATH TRACE MESSAGE UNSTABLE CONDITION INTERRUPT

2.3.3.12 PERFORMANCE MONITORING WITH THE RECEIVE SONET POH PROCESSOR BLOCK

2.3.4 RECEIVE ATM CELL PROCESSOR BLOCK

The next functional block, within the Receive Path is the Receive ATM Cell Processor block. Figure 94 presents an illustration of the XRT94L33 Functional Block Diagram, with the "Receive ATM Cell Processor" block highlighted.

Figure 94 Illustration of the XRT94L33 Functional Block Diagram, with the Receive ATM Cell Processor block highlighted

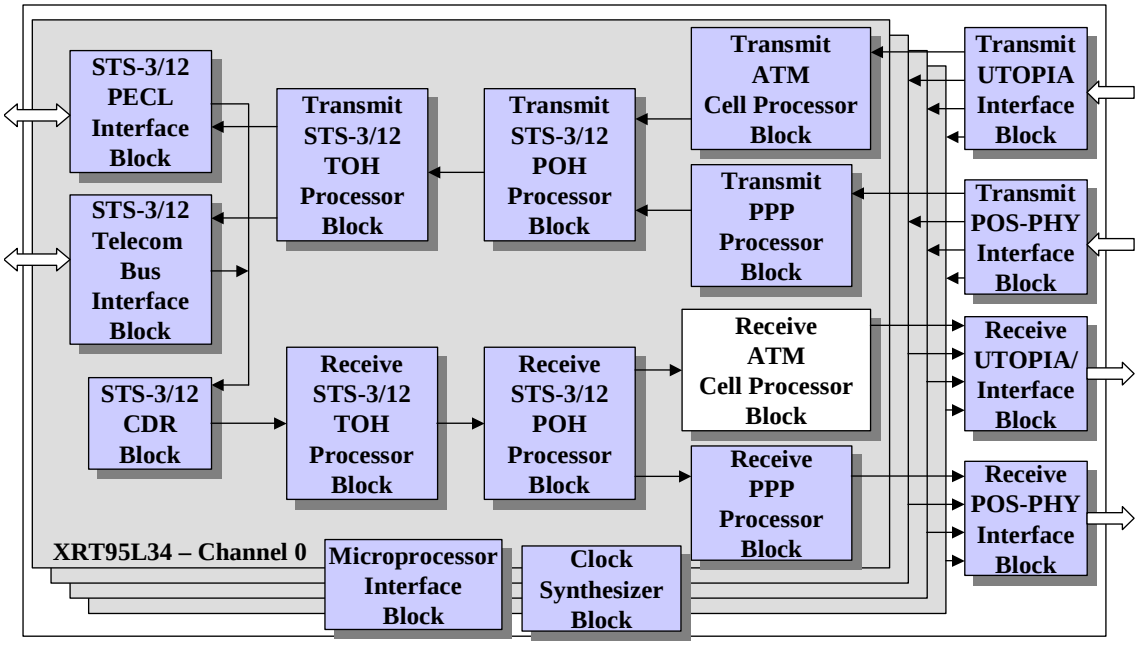


Figure 95 presents a simple block diagram of the Receive ATM Cell Processor block (with the external pins indicated).

Figure 95 Simple Illustration of the Receive ATM Cell Processor block and the Associated External Pins

A Very Brief Description of the Receive ATM Cell Processor Block

The Receive ATM Cell Processor block receives an un-delineated stream of ATM cell data from the Receive STS-3c POH Processor block. As this data-stream passes through the Receive ATM Cell Processor block it will initially pass through the "HEC Byte Verification" Block which is responsible for Cell Delineation and HEC Byte Verification.

The Receive ATM Cell Processor receives a continuous unframed stream of ATM cells from the Receive STS-3c POH Processor block. As the Receive ATM Cell Processor block receives this ATM cell data, it will then perform all of the following functions on this ATM cell data.

- Cell Delineation
- HEC Byte Verification
- Cell Payload De-Scrambling (optional)
- Idle Cell Filtering
- User Cell Filtering

Finally, all ATM cells that successfully make it through the "above-mentioned" processing will be written into the "RxFIFO" within the Receive UTOPIA Interface block, where it can be read out via the Receive UTOPIA Data Bus (by the ATM Layer Processor block).

Functional Description of the Receive ATM Cell Processor

This section presents an in-depth functional description of the Receive ATM Cell Processor block. Additionally, this section presents all of the configuration options associated with the Receive ATM Cell Processor block.

The Receive ATM Cell Processor block consists of the following functional blocks.

- HEC Byte Verification block
- Cell Payload De-Scrambler block
- Idle Cell Filter block
- Receive User Cell Filter block
- Receive Cell Extraction Buffer/Processor
- Receive Cell Insertion Buffer/Processor
- Parity Calculation and Insertion block
- Receive GFC Nibble-Field Serial Output Port

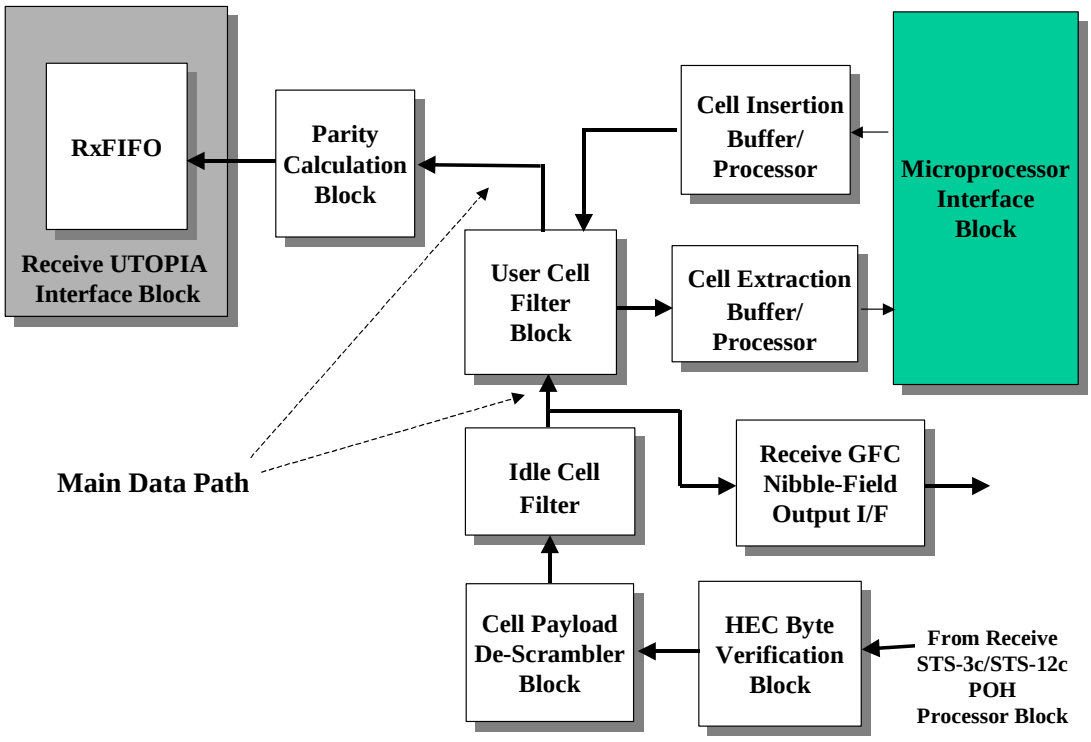
Figure 96 presents an illustration of the functional block diagram of the Receive ATM Cell Processor block with each of these "above-mentioned" functional blocks noted.

Each of these "sub-blocks" will be discussed in some detail below. However, before we get too much into the detailed functional description of the Receive ATM Cell Processor block; the user MUST note that the Receive ATM Cell Processor block will NOT even function unless the user enables the "Receive ATM Cell Processor" block for operation. The user can enable the Receive ATM Cell Processor block by setting Bit 1 (Receive ATM Cell Processor Enable), within the "Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 2" to "1" as depicted below.

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 2 (Address = 0xN701)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						Receive ATM Cell Processor Enable	Test Cell Receiver Mode Enable
R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	1	0

Figure 96 Functional Block Diagram of the Receive ATM Cell Processor block

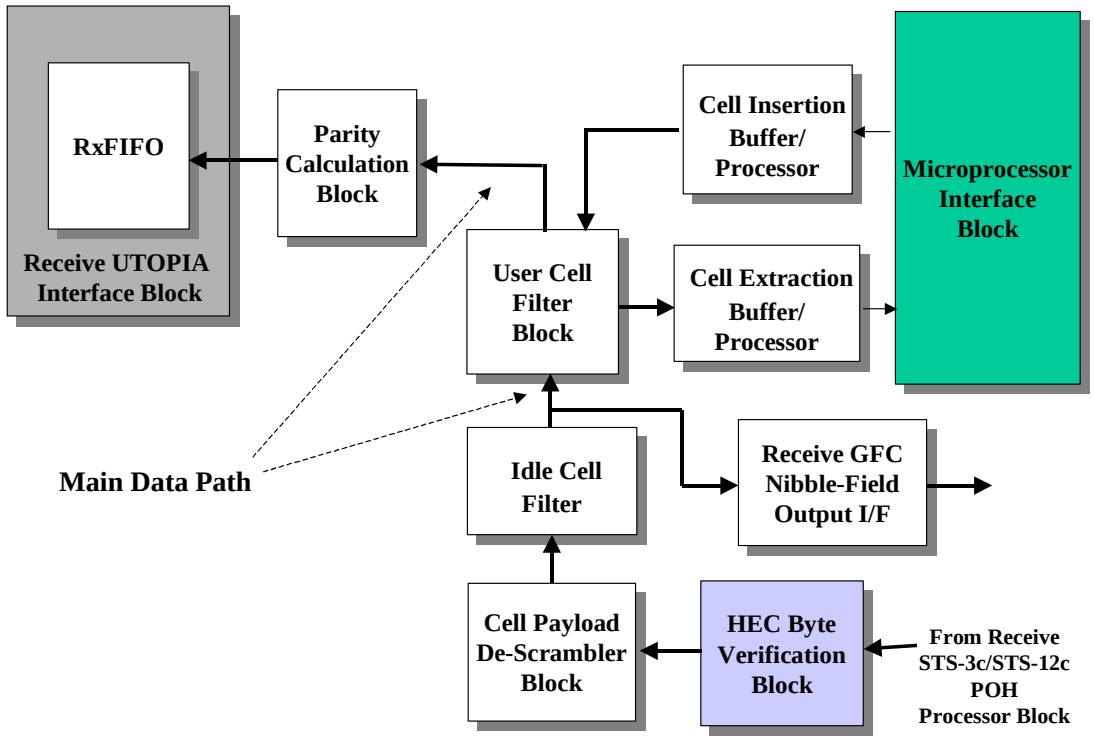


Each of these sub-blocks will be discussed in some detail below.

2.3.4.1 THE HEC BYTE VERIFICATION BLOCK

The HEC Byte Verification Block is the very first block (within the signal path of the Receive ATM Cell Processor block) to receive and process ATM cell data. Figure 97 presents an illustration of the "Functional Block Diagram" of the Receive ATM Cell Processor Block with the "HEC Byte Verification" block highlighted.

Figure 97 Illustration of the "Functional Block Diagram" of the Receive ATM Cell Processor Block with the HEC Byte Verification" block highlighted



The HEC Byte Verification block will receive an un-delineated stream of ATM cells, from the Receive STS-3c POH Processor block. As the HEC Byte Verification block receives this data-stream it is responsible for performing the following two functions, within the Receive ATM Cell Processor block

- Cell Delineation
- HEC Byte Verification

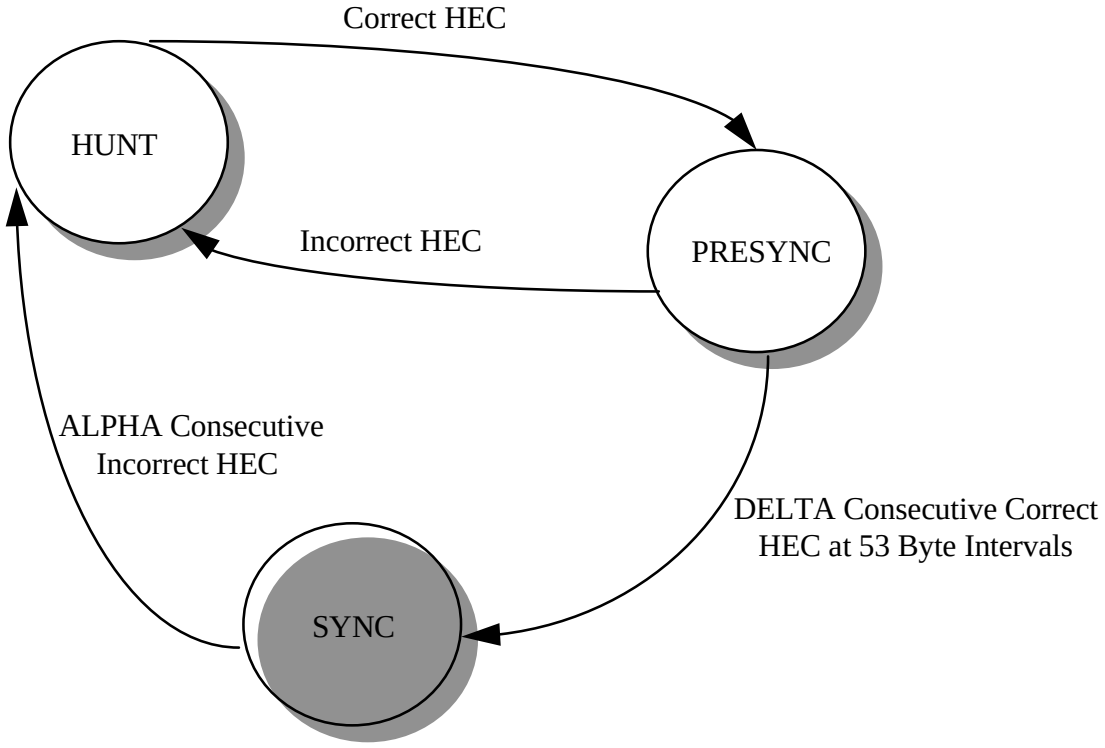
Each of these functions is described in some detail below.

2.3.4.1.1 Acquiring and Maintaining Cell Delineation with the Incoming ATM Cell Data-Stream

As the Receive STS-3c POH Processor block receives the incoming STS-3c SPE data, the contents of these STS-3c SPEs will be extracted out and will be routed to the Receive ATM Cell Processor block for further processing.

In ATM applications, the contents of each of these STS-3c SPEs are ATM cells. As the HEC Byte Verification block receives this stream of ATM cells, this ATM cell data will initially be "unframed". Therefore, the HEC Byte Verification block will have to use the "HEC Byte" Cell-Delineation algorithm in order to locate the boundaries of these cells. The HEC Byte Cell Delineation algorithm contains three states: HUNT, PRESYNC and SYNC, as depicted in the State Machine Diagram in Figure 98. Each of these three states is discussed below.

Figure 98 Cell Delineation Algorithm Employed by the Receive ATM Cell Processor block



The HUNT State

When the XRT94L33 is first powered up and is configured in the ATM Mode, then the HEC Byte Verification block will initially be operating in the "HUNT" state. While the HEC Byte Verification block is operating in the "HUNT" state, it has no knowledge of the location of the boundaries of the incoming cells. In the HUNT state, the HEC Byte Verification block is searching through the incoming ("unframed") cell data-stream for a possible valid cell header pattern (e.g., one that does not produce a HEC byte error). Therefore, while in this state, the HEC Byte Verification will read in five octets of the data that it receives from the Receive STS-3c POH Processor block. The HEC Byte Verification block will then compute a "HEC byte value" based upon the first four of these five octets. The HEC Byte Verification block will then compare this internally computed value with that of the 5th "read-in" octet. If the two values are not the same, then the HEC Byte Verification block will conclude that it has not located the boundaries of the ATM cells within the incoming data-stream and will increment its sampling set (of the 5 bytes, within the incoming data-stream) by one bit, and will repeat the above-mentioned process with this new set of "candidate" header bytes. In other words, the HEC Byte Verification block obtains its next selection of five octets, 53 bytes and 1 bit later.

If the HEC Byte Verification block comes across a set of five octets, that are such that the computed HEC byte value does match the 5th (read in) octet, then the HEC Byte Verification block will transition into the PRESYNC state.

The PRE-SYNC State

The HEC Byte Verification block will transition from the "HUNT" state to the "PRESYNC" state; when it has located an "apparently" valid set of cell header bytes. However, it is possible that the HEC Byte Verification block is being "fooled" by user data that simply mimics the cell header byte pattern. Therefore, further evaluation is required in order to confirm that this set of five octets are truly valid cell header bytes. The purpose of the "PRE-SYNC" state is to facilitate this further evaluation.

When the HEC Byte Verification block is operating in the PRE-SYNC state, it will then begin to sample five (5) "candidate header bytes" from the data within the incoming data-stream, repeatedly at 53 byte intervals. During this sampling process, the HEC Byte Verification block will compute and compare its newly computed "HEC byte value" with that of the fifth (read-in) octet. If the HEC Byte Verification block, while operating in the PRE-SYNC state, comes across a single invalid cell header byte pattern, then the HEC Byte Verification block will transition back to the "HUNT" state. However, if the HEC Byte Verification block detects "DELTA" consecutive valid cell byte headers, then it will transition into the SYNC state.

The SYNC State

Once the HEC Byte Verification block has transitioned into the "SYNC" state, then this means that the Receive ATM Cell Processor block is "officially" delineation ATM cells. The Receive ATM Cell Processor block will notify the Microprocessor (and external circuitry) of this transition into the SYNC state by doing all of the following.

1. It will indicate that it as cleared the LCD defect condition by setting Bit 0 (LCD Defect Declared) and the Bits 2 and 1 (Cell Delineation Status[1:0]) bit-fields, within the "Receive ATM Cell Processor Block – Receive ATM Status Register" to "0", as depicted below.

Receive ATM Cell Processor Block – Receive ATM Status Register (Address = 0xN707)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				PRBS Lock Indicator	Cell Delineation Status[1:0]		LCD Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

2. It will generate the "Clearance of LCD Defect Condition" interrupt. The Receive ATM Cell Processor block will indicate that it is declaring the "Clearance of LCD Defect Condition" Interrupt by doing the following.

- Toggling the "INT*" output pin "low".
- Setting Bit 1 (Clearance of LCD Defect Interrupt Status), within the "Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0" to "1" as depicted below.

Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0 (Address = 0xN70B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive Cell Insertion Interrupt Status	Receive FIFO Overflow Interrupt Status	Receive Cell Extraction Memory Overflow Interrupt Status	Receive Cell Insertion Memory Overflow Interrupt Status	Detection of Correctable HEC Byte Error Interrupt Status	Detection of Uncorrectable HEC Byte Error Interrupt Status	Clearance of LCD Defect Interrupt Status	Declaration of LCD Defect Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

Whenever the HEC Byte Verification block is operating in the SYNC state, it will tolerate a certain number of errors in the header bytes of the incoming cells. Additionally, in some case, the HEC Byte Verification block will even attempt to correct some of these errors. However, the occurrence of "ALPHA" consecutive cells with header byte errors (single or multi-bit) will cause the HEC Byte Verification block to return to the "HUNT" state. If this were to occur, then the Receive ATM Cell Processor block will notify the external circuitry that it is not properly delineating cells by doing the following.

1. It will indicate that it has declared the LCD defect condition by setting Bit 0 (LCD Defect Declared) and Bits 2 and 1 (Cell Delineation Status[1:0]) bit-fields, within the "Receive ATM Cell Processor Block – Receive ATM Status Register to "1" as depicted below.

Receive ATM Cell Processor Block – Receive ATM Status Register (Address = 0xN707)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				PRBS Lock Indicator	Cell Delineation Status[1:0]		LCD Defect Declared
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	1	1

2. It will generate the "Declaration of LCD Defect Condition" interrupt. The Receive ATM Cell Processor block will indicate that it is declaring the "Declaration of LCD Defect Condition" Interrupt by doing the following.
    a. Toggling the "INT*" output pin "low".
    b. Setting Bit 0 (Declaration of LCD Defect Interrupt Status), within the "Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0" to "1" as depicted below.

Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0 (Address = 0xN70B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive Cell Insertion Interrupt Status	Receive FIFO Overflow Interrupt Status	Receive Cell Extraction Memory Overflow Interrupt Status	Receive Cell Insertion Memory Overflow Interrupt Status	Detection of Correctable HEC Byte Error Interrupt Status	Detection of Uncorrectable HEC Byte Error Interrupt Status	Clearance of LCD Defect Interrupt Status	Declaration of LCD Defect Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

The remaining discussion of the Receive ATM Cell Processor block, within this data sheet, presumes that it (the Receive ATM Cell Processor block) is operating in the "SYNC" state and properly delineating cells.

Once the Receive ATM Cell Processor is properly delineating cells then it will proceed to route these cells through a series of "filters"; prior to allowing these cells to be written to the RxFIFO within the Receive UTOPIA Interface block.

Ultimately, the sequence of filtering/processing that each cell must go through is listed below in sequential order.

- HEC Byte Verification
- Idle Cell Filtering
- User Cell Filtering
- Cell Payload De-Scrambling

The next few sections discusses each of these forms of "cell filtering".

2.3.4.1.2 HEC Byte Verification Once the HEC Byte Verification block is properly delineating cells, then the HEC Byte Verification block will (as its name implies) perform "HEC Byte Verification" of incoming cells data from the Receive STS-3c POH Processor block in order to protect against misrouted or mis-inserted cells. In performing HEC Byte Verification the HEC Byte Verification block will take the first four byte of each cells (e.g., the header bytes) and will independently compute its own value for the HEC byte. Afterwards, the HEC Byte Verification block will compare its value of the HEC byte with the fifth octet that it has received from the Receive STS-3c POH Processor block. If the two "HEC byte" values match then the "Receive ATM Cell Processor" block will retain this cell for further processing. However, if the HEC Byte Verification block detects errors in the header bytes of a cell, then the HEC Byte Verification block will call up and employ the "HEC Byte Error Correction/Detection" Algorithm (see below).

The HEC Byte Verification block will compute its version of the HEC byte via the generating polynomial $x^8 + x^2 + x + 1$. The user should be aware that the HEC bytes of the incoming cell might have been modulo-2 added with the Coset polynomial $x^6 + x^4 + x^2 + 1$. If this is the case then the Receive ATM Cell Processor block must be configured to account for this by writing a "1" to Bit 1 (COSET Polynomial Addition) within the "Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1.

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1 (Address = 0xN702)

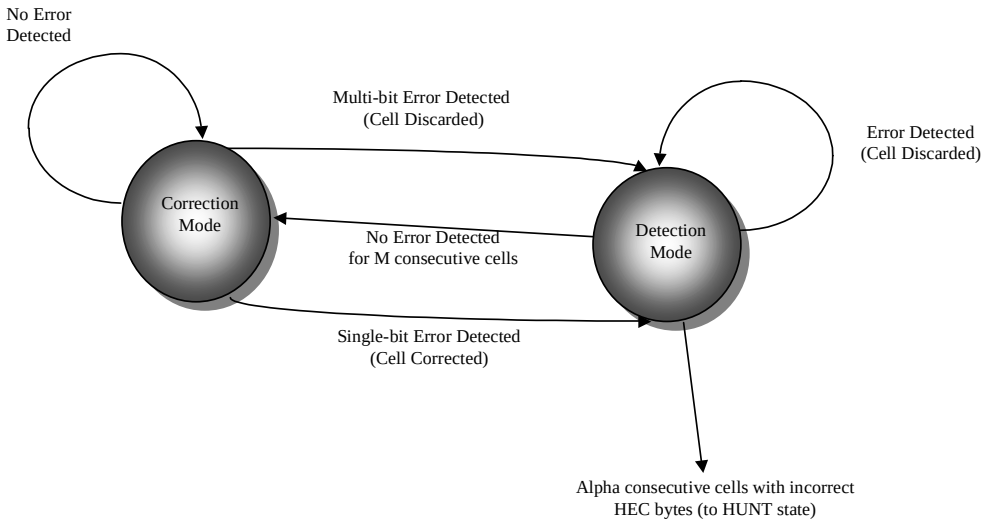
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			GFC Extraction Enable	HEC Byte Correction Enable	Uncorrectable HEC Byte Error Discard	COSET Polynomial Addition	Regenerate HEC Byte Enable
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	0

The "HEC Byte Error Correction/Detection" Algorithm

If the HEC Byte Verification block detects one or more errors in the header bytes of a given incoming ATM cell, then the "HEC Byte Error Correction/Detection" algorithm will be employed. The "HEC Byte Error Correction/Detection" Algorithm has two states: "Detection" Mode and "Correction" Mode.

Figure 99 presents a State Machine Diagram of the "HEC Byte Error Correction/Detection" Algorithm. Each of these states is discussed below.

Figure 99 State Machine Diagram of the HEC Byte Error Correction/Detection Algorithm



The "Correction" State

When the HEC Byte Verification Block is operating within the "Correction" State, within the "HEC Byte Error Correction/Detection" Algorithm, then cells that contain single bit errors (within the Header bytes) will be corrected. However, cells with multiple bit errors are discarded, unless configured by the user.

Note: *The user can configure the Receive ATM Cell Processor block to retain cells that contain multi-bit errors, by setting Bit 2 (Uncorrectable HEC Byte Error Discard), within the "Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1" register, to "0", as depicted below.*

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1 (Address = 0xN702)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			GFC Extraction Enable	HEC Byte Correction Enable	Uncorrectable HEC Byte Error Discard	COSET Polynomial Addition	Regenerate HEC Byte Enable
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	0

If the HEC Byte Verification Block is operating in the "Correction State" then it will do all of the following anytime it detects any incoming ATM cells that contain header byte errors.

If the HEC Byte Verification Block detects a Single-Bit Error within the Header bytes of an ATM Cell:

1. It will generate the "Detection of Correctable HEC Byte Error" Interrupt. The Receive ATM Cell Processor block will indicate that it is declaring the "Detection of Correctable HEC Byte Error" Interrupt, by doing the following.

   a. Toggling the "INT*" output pin "low".

   b. Setting Bit 3 (Detection of Correctable HEC Byte Error Interrupt Status), within the "Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0" to "1" as depicted below.

Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0 (Address = 0xN70B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive Cell Insertion Interrupt Status	Receive FIFO Overflow Interrupt Status	Receive Cell Extraction Memory Overflow Interrupt Status	Receive Cell Insertion Memory Overflow Interrupt Status	Detection of Correctable HEC Byte Error Interrupt Status	Detection of Uncorrectable HEC Byte Error Interrupt Status	Clearance of LCD Defect Interrupt Status	Declaration of LCD Defect Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

2. It will increment the "Receive ATM Cell Processor Block – Receive ATM Cell with Correctable HEC Byte Error Count" Registers. This is 32-bit RESET-upon-READ register that resides at Address Locations 0xN730 through 0xN733. The bit format of these registers is presented below.

Receive ATM Cell Processor Block – Receive ATM Cells with Correctable HEC Byte Error Count Register – Byte 3 (Address = 0xN730)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Correctable HEC Byte Error Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Correctable HEC Byte Error Count Register – Byte 2 (Address = 0xN731)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Correctable HEC Byte Error Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Correctable HEC Byte Error Count Register – Byte 1 (Address = 0xN732)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Correctable HEC Byte Error Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Correctable HEC Byte Error Count Register – Byte 0 (Address = 0xN733)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Correctable HEC Byte Error Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

3. It will correct the “Single-Bit” Error within this particular ATM cell.

4. The “HEC Byte Verification” Block will transition into the “Detection” Mode.

If the HEC Byte Verification Block detects a Multi-Bit Error within the Header bytes of an incoming ATM Cell:

1. It will generate the “Detection of Uncorrectable HEC Byte Error” Interrupt. The Receive ATM Cell Processor block will indicate that it is declaring the “Detection of Uncorrectable HEC Byte Error” Interrupt, by doing the following.

   a. Toggling the “INT*” output pin “low”.

   b. Setting Bit 2 (Detection of Uncorrectable HEC Byte Error Interrupt Status), within the “Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0” to “1” as depicted below.

2. It will increment the “Receive ATM Cell Processor Block – Receive ATM Cell with Uncorrectable HEC Byte Error Count” Registers. This is a 32-bit RESET-upon-READ register that resides at Address Locations 0xN734 through 0xN737. The bit format of these registers is presented below.

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 3 (Address = 0xN734)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 2 (Address = 0xN735)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 1 (Address = 0xN736)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 0 (Address = 0xN737)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

Received Cells with Uncorrectable HEC Byte Error Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

3. It will NOT correct the "Single-Bit" Error within this particular ATM cell. In fact, unless the Receive ATM Cell Processor block is configured otherwise, it will discard this particular ATM cell.

Note: *Whenever the Receive ATM Cell Processor block discards ATM cells for "Uncorrectable HEC Byte" errors it will also increment the "Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count" Registers. These registers are also 32-bit RESET-upon-READ registers that reside at Address Locations 0xN72C through 0xN72F. The bit format of these registers is presented below.*

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 3 (Address = 0xN72C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 2 (Address = 0xN72D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 1 (Address = 0xN72E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 0 (Address = 0xN72F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

4. The HEC Byte Verification block will transition into the "Detection" Mode.

The user can enable or disable entry into, and operation within the "Correction" Mode by setting Bit 3 (HEC Byte Correction Enable) within the "Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1", to the appropriate value as depicted below.

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 1 (Address = 0xN702)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			GFC Extraction Enable	HEC Byte Correction Enable	Uncorrectable HEC Byte Error Discard	COSET Polynomial Addition	Regenerate HEC Byte Enable
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	0	1	0

Setting this bit-field to "0" disables "Correction Mode" operation within the HEC Byte Verification Block. Conversely, setting this bit-field to "1" enables "Correction Mode" operation within the HEC Byte Verification Block.

The "Detection" State

Unless "Correction Mode" operation is disabled (per the procedure described above), the HEC Byte Verification Block will transition into the "Detection" State (within the "HEC Byte Error Detection/Correction" algorithm.

Whenever the "HEC Byte Verification" block is operating in the "Detection" State, then ALL errored cells (e.g., those incoming cells that contain single-bit errors and multi-bit errors) will be discarded, unless configured otherwise by the user.

More specifically, whenever the HEC Byte Verification block detects ANY cells with header byte errors, then it will do all of the following.

1. It will generate the "Detection of Uncorrectable HEC Byte Error" Interrupt. The Receive ATM Cell Processor block will indicate that it is declaring the "Detection of Uncorrectable HEC Byte Error" Interrupt, by doing the following.

   a. Toggling the "INT*" output pin "low".

   b. Setting Bit 2 (Detection of Uncorrectable HEC Byte Error Interrupt Status), within the "Receive ATM Cell Processor Block – Receive ATM Interrupt Status Register – Byte 0" to "1" as depicted below.

2. It will increment the "Receive ATM Cell Processor Block – Receive ATM Cell with Uncorrectable HEC Byte Error Count" Registers. This is a 32-bit RESET-upon-READ register that resides at Address Locations 0xN734 through 0xN737. The bit format of these registers is presented below.

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 3 (Address = 0xN734)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 2 (Address = 0xN735)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 1 (Address = 0xN736)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive ATM Cells with Uncorrectable HEC Byte Error Count Register – Byte 0 (Address = 0xN737)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Received Cells with Uncorrectable HEC Byte Error Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

3. It will NOT correct the “Single-Bit” Error within this particular ATM cell. In fact, unless the Receive ATM Cell Processor block is configured otherwise, it will discard this particular ATM cell.

Note: *Whenever the Receive ATM Cell Processor block discards ATM cells for “Uncorrectable HEC Byte” errors it will also increment the “Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count” Registers. These registers are also 32-bit RESET-upon-READ registers that reside at Address Locations 0xN72C through 0xN72F. The bit format of these registers is presented below.*

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 3 (Address = 0xN72C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[31:24]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 2 (Address = 0xN72D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 1 (Address = 0xN72E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Cell Processor Block – Receive Discarded ATM Cell Count – Byte 0 (Address = 0xN72F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive – Discarded ATM Cell Count[7:0]							

RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

4. The HEC Byte Verification block will remain in the "Detection" Mode.

How Does the HEC Byte Verification Block transition back into the "Correction" State?

The HEC Byte Verification block will transition back into the "Correction" state once it has received "M" consecutive cells with correct HEC byte values. This value for "M" is also known as the "Correction Threshold". The user has the option to use the following values for "M": 1, 2, 4 and 8. The user can configure the Receive ATM Cell Processor block to use any of these values for "M" by writing the appropriate value to the "Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 0", as depicted below.

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 0 (Address = 0xN703)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Insert into UDF1 Enable	HEC Status into UDF2 Enable	HEC Byte Correction Threshold[1:0]		Receive UTOPIA Parity - ODD	Unused		Descramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
1	1	X	X	1	0	0	0

Table 19 presents the relationship between the contents of the "HEC Byte Correction Threshold[1:0]" bit-fields and the corresponding "Correction Threshold" configured.

Table 19 The Relationship between the contents of the "HEC Byte Correction Threshold[1:0]" Bit-Fields the Corresponding "Correction Threshold" configured

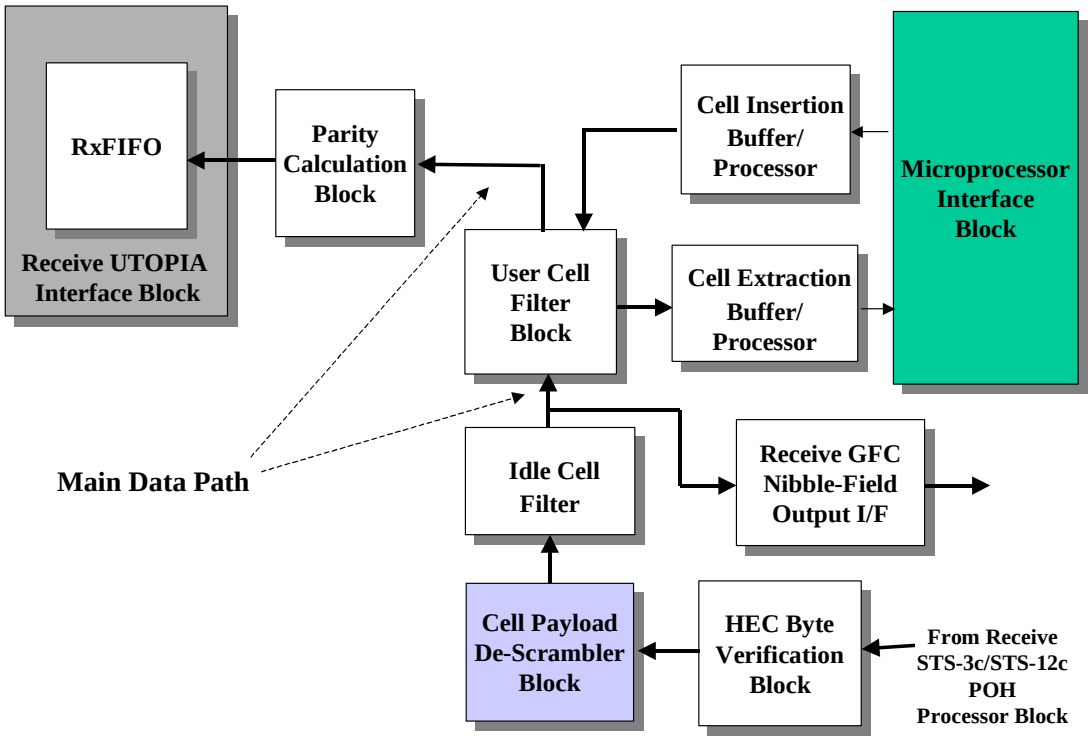
HEC BYTE CORRECTION THRESHOLD[1:0]	RESULTING "CORRECTION THRESHOLD" CONFIGURED (NUMBER OF CELLS)
00	1 Cell with No HEC Byte Errors
01	2 Consecutive Cells with No HEC Byte Errors
10	4 Consecutive Cells with No HEC Byte Errors
11	8 Consecutive Cells with No HEC Byte Errors

2.3.4.2 CELL PAYLOAD DE-SCRAMBLER BLOCK

After the "HEC Byte Verification" block, the very next block within the signal path (within the Receive ATM Cell Processor block) is the "Cell Payload De-Scrambler" Block.

Figure 100 presents an illustration of the "Functional Block Diagram" of the Receive ATM Cell Processor Block, with the "Cell Payload De-Scrambler" Block highlighted.

Figure 100 Illustration of the "Functional Block Diagram" of the Receive ATM Cell Processor Block with the "Cell Payload De-Scrambler" Block highlighted



In numerous applications the payload portion of the incoming cells will scrambled by the remote terminal equipment. These cells are scrambled in order to prevent the user data from mimicking framing or control bytes. Therefore, the Receive ATM Cell Processor provides the user will the option of de-scrambling the payload of these cells in order to restore the original content of the cell payload. (Please note that this cell de-scrambler presumes that the cell payload were scrambled via the scrambling generating polynomial of $x^{43} + 1$.) The user can configure this option by setting Bit 2 (De-Scramble Enable) within the "Receive ATM Cell Processor – Receive ATM Control Register – Byte 0", as depicted below.

Receive ATM Cell Processor Block – Receive ATM Control Register – Byte 0 (Address = 0xN703)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
HEC Byte Insert into UDF1 Enable	HEC Status into UDF2 Enable	HEC Byte Correction Threshold[1:0]		Receive UTOPIA Parity - ODD	Unused		Descramble Enable
R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/W
1	1	X	X	1	0	0	1

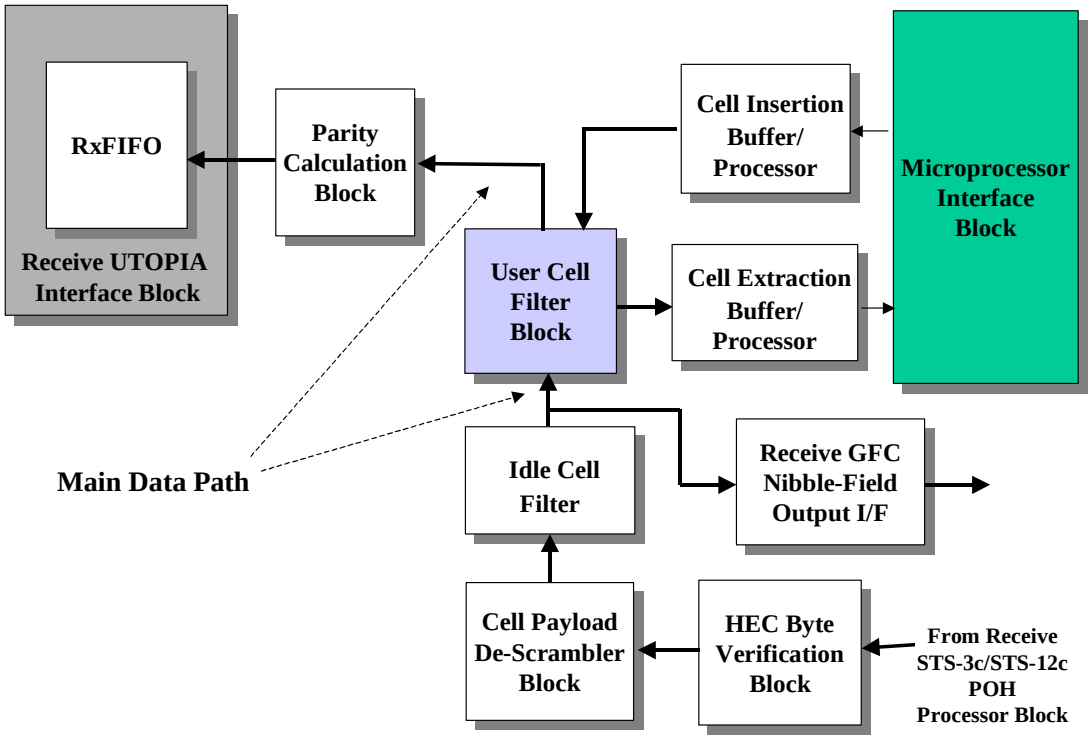
2.3.4.3 IDLE CELL FILTER

Idle Cell Filtering is actually achieved through the user of the Receive User Cell Filter. As a consequence, we will now proceed to discuss the "Receive User Cell Filter".

2.3.4.4 THE RECEIVE USER CELL FILTER BLOCK

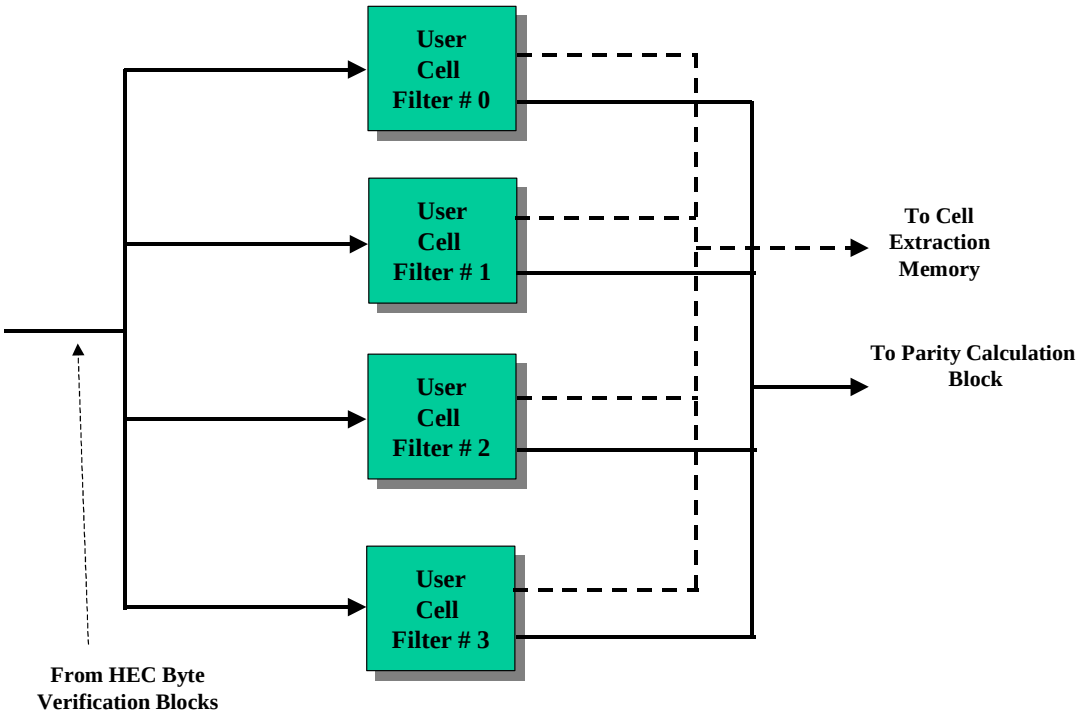
Once a given ATM cell has successfully passed through the "HEC Byte Verification" block, and has been processed by the "Cell De-Scrambler" block, it will now be processed by the "Receive User Cell Filter" block. Figure 101 presents the functional block diagram of the Receive ATM Cell Processor block with the "Receive User Cell Filter" block highlighted.

Figure 101 Illustration of the "Receive ATM Cell Processor" Functional Block Diagram with the "Receive User-Cell Filter" block highlighted



The Receive ATM Cell Processor block consists of four parallel User Cell Filters. These User Cell Filters are connected in parallel, in the sense that the entire "receive" ATM cell traffic will be presented to the inputs of each of the four "user-cell" filters, at the same time. Figure 102 attempts to clarify this phenomenon by presenting an illustration as to how each of these user cell filters is connected to the ATM traffic.

Figure 102 An Illustration of the Configuration of the Four Receive User Cell Filters within the Receive ATM Cell Processor block



Each of these four (4) User Cell filters can be configured to analyze all of the header bytes, within a given ATM cell. Based upon the value of these header bytes (within a given cell), each of the four User Cell Filter blocks can be configured to perform either of the following functions.

- To filter (e.g., discard) the cell
- To NOT filter (e.g., permit the cell to pass through, along the Data Path) the cell
- To replicate (or copy) the cell, and route the replicate cell to the "Receive ATM Cell Processor" block – Extraction Processor block.
- To NOT replicate the cell.

Instructions on how to implement the following User Cell Filter options are presented below.

- Enabling/Disabling the User Cell Filter
- Defining the User Cell Filter Action
- Defining the User Cell Filter Sense
- Specifying the User Cell Filter Values

ENABLING/DISABLING THE RECEIVE USER CELL FILTER

The Receive ATM Cell Processor block permits the user to either enable or disable each of the four Receive User Cell Filters. The user can accomplish this by writing the appropriate data into Bit 3 (User Cell Filter Enable), within the "Receive ATM Filter Control – Byte 0" register; as depicted below.

Receive ATM Filter # 0, # 1, # 2, # 3 Control – Byte 0 (Address = 0xN743, 0xN753, 0xN763, 0xN773)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	0	0	0

Note: *As the bit-format table for the "Receive ATM Filter Control – Byte 0" register implies, each of the four (4) Receive User Cell Filters can be individually enabled or disabled. Further, each of these four Receive User Cell Filters can be individually configured to either "Copy" or "Discard" cells (or both).*

Setting this bit-field to "1" enables the corresponding Receive User Cell Filter. Conversely, setting this bit-field to "0" disables the "Receive User Cell Filter".

If a given Receive User Cell Filter is enabled, then it will perform action on all incoming ATM cells, based upon the settings of the remaining register bits within this register; and that within both the "Receive ATM Filter Control – Pattern" and "Receive ATM Filter Control – Check" registers. Conversely, if the given Receive User Cell Filter is disabled, then all User Cells will pass through the Receive User Cell Filter, and towards the "Parity Calculation & Insertion" block without any such actions performed on these cells.

SELECTING THE RECEIVE USER CELL FILTER ACTION

The Receive ATM Cell Processor block permits the user to specify the action that the User Cell Filter will take on each cell, that meets certain "User-Defined" Filtering requirements. The user can accomplish this by setting Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable), within the Receive ATM Filter Control – Byte 0" Register; to the appropriate values. These two bit-fields are highlighted below.

Receive ATM Filter # 0, # 1, # 2, # 3 Control – Byte 0 (Address = 0xN743, 0xN753, 0xN763, 0xN773)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell Enable	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	X	X	X

Bit 2 – Copy Cell Enable

This bit-field permits the user to configure the "Receive User Cell" filter (within the Receive ATM Cell Processor block) to either replicate (copy) or not replicate a given cell that complies with the "user-defined" filtering requirements. All copied cells are routed to the "Receive ATM Cell Processor Extraction Memory", where their contents can be read out by the user via the Microprocessor Interface.

Setting this bit-field to "1" configures the Receive User Cell Filter to copy all cells that comply with the "user-cell" filter requirements, and to route these cells to the "Receive ATM Cell Processor – Extraction Memory".

Conversely, setting this bit-field to "0" configures the Receive User Cell Filter to NOT copy these cells, and NOT route these cells to the "Receive ATM Cell Processor – Extraction Memory.

Bit 1 – Discard Cell Enable

This bit-field permits the user to configure the "Receive User Cell" filter (within the Receive ATM Cell Processor block) either discard or not discard a given cell that complies with the "user-defined" filtering

requirements. Cells that are discarded will not be routed to the "Parity Calculation and Insertion" block. Cells that are NOT discarded will proceed on through the remainder of the Receive ATM Cell Processor block circuitry (and RxFIFO), for further processing.

Setting this bit-field to "1" configures the Receive User Cell Filter to discard all cells that comply with the "user-cell" filter requirements.

Conversely, setting this bit-field to "0" configures the Receive User Cell Filter to NOT discard the cells that comply with the "user-cell" filter requirements.

DEFINING THE USER CELL FILTER SENSE

The Sense of a given Receive User Cell Filter is defined by the state of Bit 0 (User Cell Filter Sense) within the "Receive ATM Filter Control – Byte 0" Register; as described below.

BIT 0 - USER CELL FILTER SENSE

Receive ATM Filter # 0 , # 1, # 2, # 3 Control – Byte 0 (Address = 0xN743, 0xN753, 0xN763, 0xN773)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				User Cell Filter Enable	Copy Cell Enable	Discard Cell Enable	User Cell Filter Sense
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	1	X	X	X

This bit-field controls the "Filter Mode". If the user sets this bit-field to "1" then the "Receive User-Cell" Filter will act, per the settings within Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable) on cells with header byte patterns MATCHING the "user-cell" filtering criteria. Conversely, setting this bit-field to "0" configures the "Receive User-Cell" Filter to act, per the settings within Bits 2 (Copy Cell Enable) and 1 (Discard Cell Enable) on cells with header byte patterns NOT MATCHING the "user-cell" filtering criteria.

SPECIFYING THE RECEIVE USER CELL FILTERING CRITERIA

As described above, each of the four Receive User Cell Filters (within the Receive ATM Cell Processor block) can be configured to perform a variety of actions (e.g., copy cells, discard cells, etc); based upon whether the Header Byte Patterns of User Cells MATCH; or DO NOT MATCH a particular "User-Defined" Filter criteria.

The "User-Defined" Filter criteria (for each of the four Receive User Cell Filter blocks, within the Receive ATM Cell Processor block) are ultimately defined by the values residing within a total of eight (8) registers. Four of these registers are referred to as "User Cell Filter – Pattern" Registers; and the remaining four registers are referred to as "User Cell Filter – Check" Registers. Each of these register types are defined below.

RECEIVE USER CELL FILTER – PATTERN REGISTERS

The four User Cell Filter – Pattern Registers permit the user to specify the Header Byte Pattern for the Receive User Cell Filter. There are four User Cell Filter – Pattern Registers (one for each of the four (4) header bytes, within an ATM cell). The bit-format of these "User Cell Filter – Pattern Registers" is presented below.

Receive ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 1 (Address = 0xN744, 0xN754, 0xN764, 0xN774)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 2 (Address = 0xN745, 0xN755, 0xN765, 0xN775)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 3 (Address = 0xN746, 0xN756, 0xN766, 0xN776)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 3[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Pattern – Header Byte 4 (Address = 0xN747, 0xN757, 0xN767, 0xN777)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Pattern – Header Byte 4[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *Each of the four Receive User Cell Filters (within the Receive ATM Cell Processor block) contains their own set of "User Cell Filter – Pattern" registers.*

RECEIVE USER CELL FILTER – CHECK REGISTERS

The four User Cell Filter – Check Registers permit the user to specify which bits (within the Header bytes of User Cells) will be checked and compared with the contents of the "User Cell Filter – Pattern Registers". There are four (4) "User Cell Filter-Check" Registers (one for each of the four header bytes within an ATM cell). Setting a particular bit-field to "0" configures the "Receive User Cell Filter" to NOT compare the corresponding bit-field (within the header bytes of the incoming user cell) to the corresponding bit-value, within the User Cell Filter – Pattern Register; during User Cell pattern checking. Conversely, setting a particular bit-field to "1" configures the "Receive User Cell Filter" to compare the corresponding bit-field (within the header bytes of the incoming user cell) to the corresponding bit-value, within the User Cell Filter – Pattern Register; during User Cell pattern checking.

For example, if the user were write the value "0x00" into a given "User Cell Filter – Check Register", then the Receive User Cell Filter will NOT check and compare the value of the corresponding header byte, within an incoming User Cell; to the contents of the corresponding "User Cell Filter – Pattern" register during User Cell pattern checking. Conversely, if the user were to write the value "0xFF" into a given "User Cell Filter – Check Register", then the User Cell Filter will check and compare all eight bits (within the corresponding byte) within an incoming User Cell; to the contents of the corresponding "User Cell Filter – Pattern" register during User Cell pattern checking.

The bit-format of the four "User Cell Filter – Check Registers" is presented below.

Receive ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 1 (Address = 0xN748, 0xN758, 0xN768, 0xN778)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 2 (Address = 0xN749, 0xN759, 0xN769, 0xN779)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 3 (Address = 0xN74A, 0xN75A, 0xN76A, 0xN77A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 3[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 Check – Header Byte 4 (Address = 0xN74B, 0xN75B, 0xN76B, 0xN77B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
User Cell Filter Check – Header Byte 4[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Note: *Each of the four User Cell Filters (within the Receive ATM Cell Processor block) contains their own set of "User Cell Filter – Check" registers.*

COUNTING FILTERED CELLS

The user can keep account of the number of User Cells that have been filtered out by the Receive User Cell Filter by periodically reading out the contents of the "Receive ATM Filter Count" Registers. Each time a user cell is filtered (e.g., either discarded or copied) by the "Receive User Cell" Filter, then this counter will be incremented by the value of "1". This counter will increment until it reaches the value of 0xFFFFFFFF. At this time, the counter will "saturate" and will not increment any further until read. These registers are 32 bits, in width and are "RESET-upon-READ. The bit-format for these registers (for each of the four filters) is presented below.

Receive ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 3 (Address = 0xN74C, 0xN75C, 0xN76C, 0xN77C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[31:24]							

RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 2 (Address = 0xN74D, 0xN75D, 0xN76D, 0xN77D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[23:16]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 1 (Address = 0xN74E, 0xN75E, 0xN76E, 0xN77E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[15:8]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Receive ATM Filter # 0, # 1, # 2, # 3 – Filtered Cell Counter – Byte 0 (Address = 0xN74F, 0xN75F, 0xN76F, 0xN77F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Filtered Cell Count[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Each of the four User Cell Filters (within the Receive ATM Cell Processor block) contains their own set of "Filtered Cell Counter" registers.

The Procedure for Reading Out the Contents of the Receive ATM Filter – Filtered Cell Counter Register

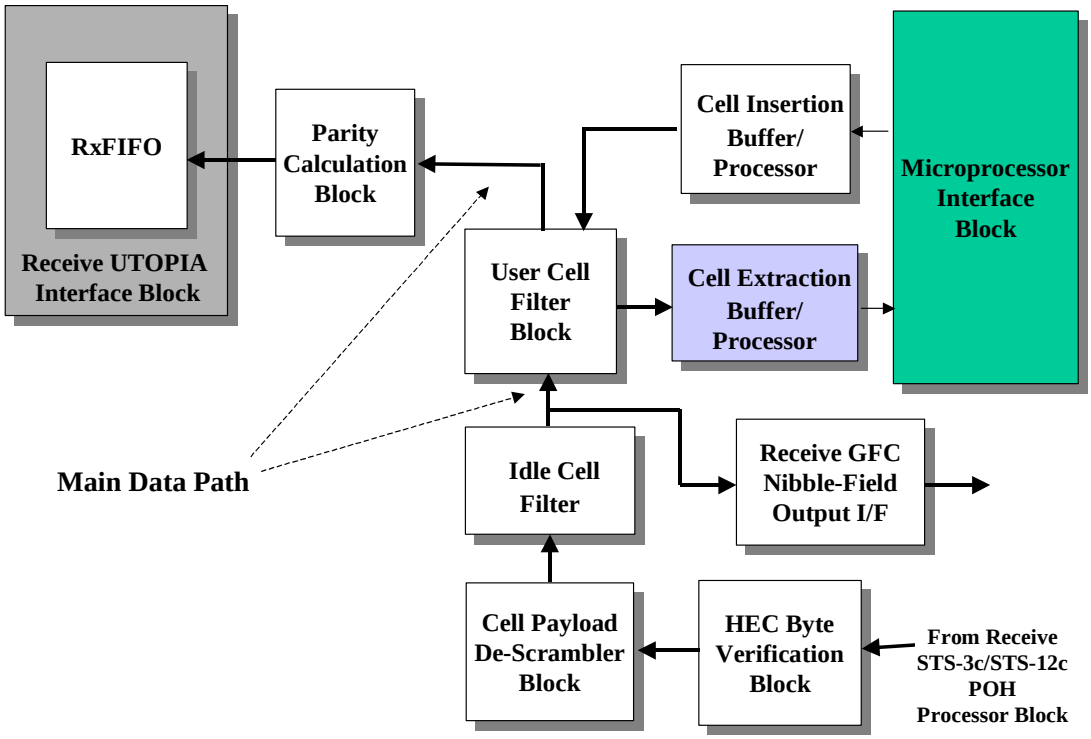
In order to properly read out the contents of these registers, the user must read these registers in the following order.

- Receive ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 3 (Address = 0xN74C, 0xN75C, 0xN76C, 0xN77C)
- Receive ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 2 (Address = 0xN74D, 0xN75D, 0xN76D, 0xN77D)
- Receive ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 1 (Address = 0xN74E, 0xN75E, 0xN76E, 0xN77E)
- Receive ATM Filter #0, #1, #2, #3 – Filtered Cell Counter – Byte 0 (Address = 0xN74F, 0xN75F, 0xN76F, 0xN77F)

2.3.4.5 THE RECEIVE CELL EXTRACTION PROCESSOR

The Receive ATM Cell Processor block consists of a "Receive Cell Extraction Buffer/Processor" block. Figure 103 presents the functional block diagram of the "Receive ATM Cell Processor block with the "Receive Cell Extraction Buffer/Processor" block highlighted.

Figure 103 Illustration of the Receive ATM Cell Processor block Functional Block Diagram, with the "Receive Cell Extraction Buffer/Processor" block highlighted

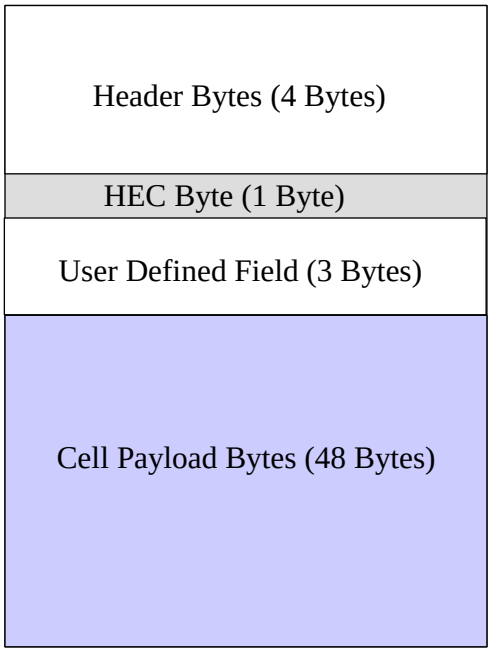


The Receive Cell Extraction Buffer/Processor block permits the user to read out the contents of an "inbound" ATM cell via the Microprocessor Interface. If the user configures the "Receive User Cell Filter" appropriately, the "Receive User Cell Filter" will copy (e.g., replicate) the contents of certain cells (which comply with the user cell filtering requirements). These copied cells will be routed to the "Receive Cell Extraction Buffer/Processor" block where it can be read out and processed by the Microprocessor Interface.

The Format of the ATM Cell that is Read from the "Receive Cell Extraction" Buffer

As the user reads out the contents of an ATM cell from the "Receive Cell Extraction" Buffer (via the Microprocessor Interface), they will be expected to read this ATM cell data via a 32-bit wide register/buffer interface. As a consequence, the user must read out 56 byte size ATM cells from the "Receive Cell Extraction" Buffer. The byte format of this 56-byte ATM cell is as illustrated below.

Figure 104 Byte Format of the ATM Cells that are read out from the "Receive Cell Extraction" Memory



As a consequence, the user must read out a total of 14 "32-bit" words from the "Receive Cell Extraction" buffer for each ATM cell that is read out of the Receive Cell Extraction Buffer.

Procedure for Reading a Cell from the "Receive Cell Extraction Buffer"

The user can read an ATM cell from the "Transmit Cell Extraction Buffer" within the Transmit ATM Cell Processor block, by executing the following steps.

STEP 1 – Flush the contents of the "Receive Cell Extraction Buffer"

This is accomplished by executing a "Transmit Cell Extraction Buffer" RESET. The user can perform this "Cell Extraction Buffer" RESET by doing the following.

STEP 1a – Write a "0" into Bit 4 (Extraction Memory RESET*) within the "Transmit ATM Cell – Memory Control" Register; as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1->0	0	1	1	0

STEP 1b – Write a "1" into Bit 4 (Extraction Memory RESET*), within the "Receive ATM Cell – Memory Control" Register; as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0->1	0	1	1	0

Note: *This step should typically be performed upon power-up, prior to processing any ATM cell traffic through the XRT94L33. This step is not necessary after the first cell has been read from the "Receive Cell Extraction Buffer" following a power cycle to the chip.*

STEP 2 – Check and see if an ATM cell exists in the "Receive Cell Extraction Buffer"

This can be accomplished by one of two approaches.

- Polling approach
- Interrupt-driven approach

Each of these approaches is described below.

Executing STEP 2 using the Polling Approach

The user can determine whether or not a cell is available, within the "Receive Cell Extraction Buffer" by testing the state of Bit 3 (Extraction Memory CLAV) within the Receive ATM Cell – Memory Control Register; as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	X	1	1	0

If Bit 3 is set to "1", then the "Receive Cell Extraction Buffer" contains an ATM cell that needs to be read. At this point, the user should proceed on to STEP 3.

Conversely, if Bit 3 is set to "0", then the "Receive Cell Extraction Buffer" does not contain an ATM cell that needs to be read. At this point, the Microprocessor Interface should continue to poll the state of this bit-field and wait until this bit-field toggles to "1".

Executing STEP 2 using the Interrupt-Driven Approach

In order to reduce or eliminate the Microprocessor Overhead of continuously polling the state of Bit 3, the user can use the "Receive Cell Extraction" Interrupt feature, within the chip. If the Microprocessor invokes this feature, then the XRT94L33 will generate an interrupt anytime a new cell has been received and loaded into the "Receive Cell Extraction Buffer".

The user can enable the "Cell Extraction" Interrupt by setting Bit 5 (Cell Extraction Interrupt Enable), within the "Receive ATM Cell Processor – Interrupt Enable" Register to "1" as indicated below.

Receive ATM Cell Processor – Interrupt Enable Register (Address = 0xN70F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

Once the "Cell Extraction Buffer" receives a "COPIED" cell from the "Receive User Cell Filter", then the XRT94L33 will do all of the following:

- It will toggle the "INT*" output pin "LOW".
- It will set Bit 5 (Cell Extraction Interrupt Status) within the Receive ATM Cell Processor – Interrupt Status Register, to "1" as depicted below.

Receive ATM Cell Processor – Interrupt Status Register (Address = 0xN70B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

At this point, the user can now proceed onto STEP 3.

STEP 3 – Read out the very first 32-bit word of this new ATM cell from the "Receive Cell Extraction Buffer".

This is accomplished by executing the following four sub-steps.

STEP 3a – Read the contents of the first byte (of this newly received ATM cell) from the "Receive ATM Cell – Insertion/Extraction Memory Register – Byte 3; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 3 (Address = 0xN714)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A7	A6	A5	A4	A3	A2	A1	A0

Note: *In this case, the contents of the first byte (within this ATM cell) is of the value [A7, A6, A5, A4, A3, A2, A1, A0]*

STEP 3b – Read the contents of the second byte (of this new ATM cell) from the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN715)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B7	B6	B5	B4	B3	B2	B1	B0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [B7, B6, B5, B4, B3, B2, B1, B0]*

STEP 3c – Read the contents of the third byte (of this new ATM cell) from the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 1; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN716)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Insertion/Extraction Memory Data[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
C7	C6	C5	C4	C3	C2	C1	C0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [C7, C6, C5, C4, C3, C2, C1, C0]*

STEP 3d – Read the contents of the fourth byte (of this new ATM cell) from the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 0; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN717)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D7	D6	D5	D4	D3	D2	D1	D0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [D7, D6, D5, D4, D3, D2, D1, D0]*

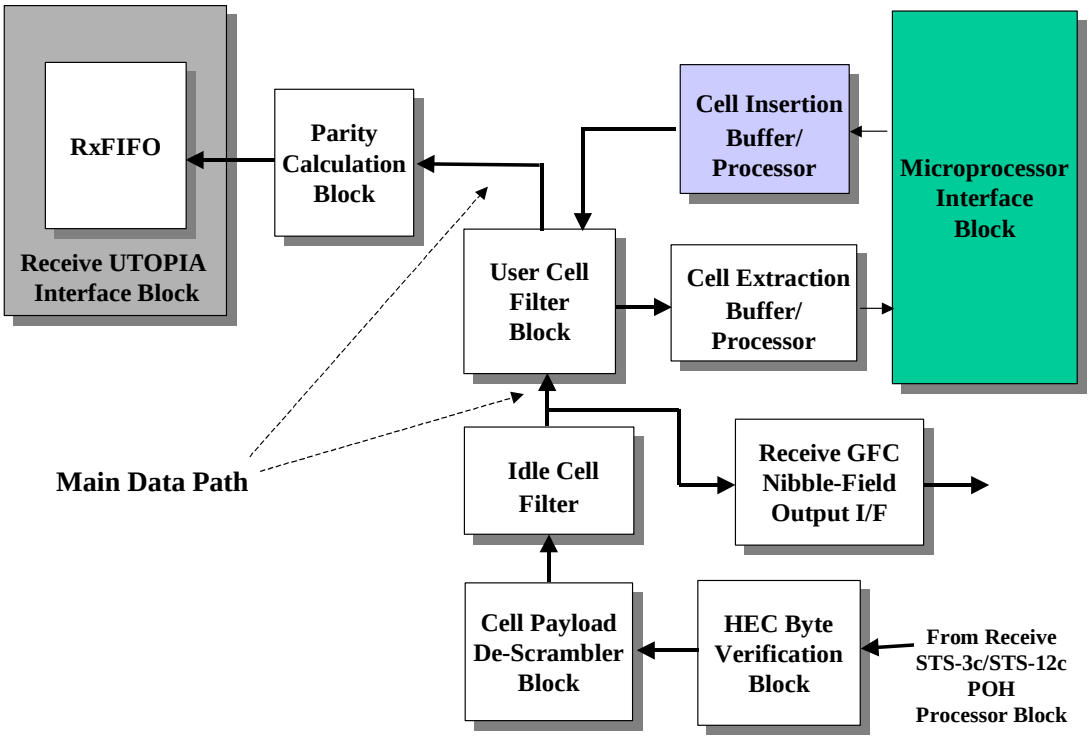
STEP 4 – Continue to read out the contents of the remaining bytes of this ATM cell from the "Receive Cell Extraction Memory".

This is accomplished by executing the procedure, outlined in STEP 3, repeatedly for 13 more times.

2.3.4.6 4THE RECEIVE CELL INSERTION BUFFER/PROCESSOR

The Receive ATM Cell Processor block consist of a "Receive Cell Insertion Buffer/Processor" block. Figure 105_ presents the functional block diagram of the Receive ATM Cell Processor block with the Receive Cell Insertion Buffer/Processor" block highlighted.

Figure 105 Illustration of the Receive ATM Cell Processor block Functional Block Diagram, with the "Receive Cell Insertion Buffer/Processor" block highlighted



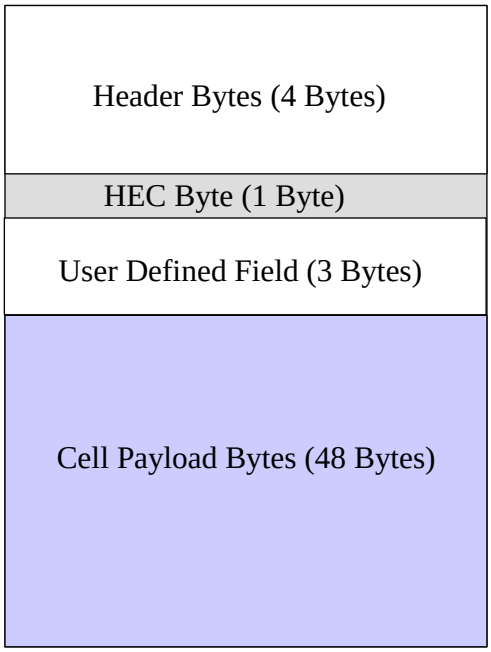
The Receive Cell Insertion Buffer/Processor block permits the user to load the contents of an "outbound" ATM cell into the "Receive Cell Insertion Buffer" via the Microprocessor Interface. Once this cell has been loaded into the "Receive Cell Insertion Buffer", then it will be transmitted to the "RxFIFO" where it will ultimately wait to be read out of the Receive UTOPIA Interface block via the ATM Layer Processor block. This feature can be very useful for debugging and diagnostics on the "UTOPIA" side of the chip.

The Format of ATM Cell Data that is written into the "Receive Cell Insertion" Buffer

As the user loads the contents of an ATM cell into the "Receive Cell Insertion" Buffer (via the Microprocessor Interface), they will be expected to write this ATM cell data into a 32 bit wide register/buffer interface. As a consequence, the user must write in 56-byte size ATM cells into the "Receive Cell Insertion" buffer.

The byte format of this 56 byte ATM cell is as illustrated below in Figure 106.

Figure 106 Byte-Format of the ATM Cell that is to be loaded into the "Receive Cell Insertion" Memory



As a consequence, the user must write in a total of 14 "32-bit words" into the "Receive Cell Insertion" buffer for each ATM cell that is written into the "Receive Cell Insertion" Buffer.

Procedure for Writing a Cell into the "Receive Cell Insertion Buffer"

The user can write an ATM cell into the "Receive Cell Insertion Buffer" within the Receive ATM Cell Processor block, by executing the following steps.

STEP 1 – Flush the contents of the "Receive Cell Insertion Buffer"

This is accomplished by executing a "Receive Cell Insertion Buffer" RESET. The user can perform this "Transmit Cell Insertion Buffer" RESET by doing the following.

STEP 1a – Write a "0" into Bit 2 (Insertion Memory RESET*), within the "Receive ATM Cell – Memory Control" register; as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1->0	1	0

STEP 1b – Write a "1" into Bit 2 (Insertion Memory RESET*), within the "Receive ATM Cell – Memory Control" Register; as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0->1	1	0

Note: *This step should typically be performed upon power-up, prior to writing in any ATM cell data into the "Receive Cell Insertion Buffer". This step is not necessary after the first cell has been written into the "Receive Cell Insertion Buffer" following a power cycle to the chip.*

STEP 2 – Check and Verify that there is sufficient space available (within the Receive Cell Insertion Buffer) to handle this cell.

This can be accomplished by one of two approaches.

- Polling approach
- Interrupt-driven approach.

Each of these approaches is described below.

Executing STEP 2 using the Polling Approach

The user can determine whether or not there is room (to write another ATM cell of data) in to the "Receive Cell Insertion" Buffer" by polling the state of Bit 1 (Insertion Memory ROOM) within the "Receive ATM Cell – Memory Control Register" as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	X	0

If Bit 1 (Insertion Memory ROOM) is set to "1" then the "Receive Cell Insertion Buffer" is NOT too full to accept another cell. At this point, the Microprocessor can now move onto STEP 3.

Conversely, if Bit 1 is set to "0" then the "Receive Cell Insertion Buffer" is too full to accept another cell. The Microprocessor Interface should continue to poll the state of this bit-field and wait until this bit-field toggles to "1".

Executing STEP 2 using the Interrupt-Driven Approach

In order to reduce or eliminate the Microprocessor Overhead of continuously polling the state of Bit 1, the user can use the "Receive Cell Insertion" Interrupt feature, within the chip. If the Microprocessor invokes this feature, then the XRT94L33 will generate an interrupt anytime a cell (residing in the Receive Cell Insertion Buffer) has been inserted into the "Receive Input Data Path" (thereby freeing up some space within the Receive Cell Insertion Buffer).

The user can enable the "Receive Cell Insertion" Interrupt by setting Bit 4 (Cell Insertion Interrupt Enable), within the "Receive ATM Cell Processor – Interrupt Enable" Register to "1" as indicated below.

Receive ATM Cell Processor – Interrupt Enable Register (Address = 0xN70F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Enable	Cell Insertion Interrupt Enable	Cell Extraction Memory Overflow Interrupt Enable	Cell Insertion Memory Overflow Interrupt Enable	Detection of HEC Byte Error Interrupt Enable	Detection of Parity Error Interrupt Enable
R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Once a cell (residing within the Receive Cell Insertion Buffer) has been inserted into the "Receive Input Data Path", then the XRT94L33 will do all of the following:

- It will toggle the "INT*" output pin "LOW".
- It will set Bit 4 (Cell Insertion Interrupt Status), within the Receive ATM Cell Processor – Interrupt Status Register; to "1" as depicted below.

Receive ATM Cell Processor – Interrupt Status Register (Address = 0xN70B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Cell Extraction Interrupt Status	Cell Insertion Interrupt Status	Cell Extraction Memory Overflow Interrupt Status	Cell Insertion Memory Overflow Interrupt Status	Detection of HEC Byte Error Interrupt Status	Detection of Parity Error Interrupt Status
R/O	R/O	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

At this point, the user can now proceed on with STEP 3.

STEP 3 – Inform the "Receive Cell Insertion Processor" that the very next 32-bit word to be written into the "Receive Cell Insertion Buffer" is the first word of a new ATM cell.

This is accomplished by writing the value "1" into Bit 0 (Insertion Memory Write SoC), within the "Receive ATM Cell – Memory Control Register" as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	1

STEP 4 – Write the very first 32-bit word of this new ATM cell into the "Receive Cell Insertion Buffer".

This is accomplished by executing the following four sub-steps.

STEP 4a - Write the contents of first byte (of this new ATM cell) into the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 3; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 3 (Address = 0xN714)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A7	A6	A5	A4	A3	A2	A1	A0

Note: *In this case, the contents of the first byte (within this ATM cell) is of the value [A7, A6, A5, A4, A3, A2, A1, A0]*

STEP 4b – Write the contents of the second byte (of this new ATM cell) into the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN715)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B7	B6	B5	B4	B3	B2	B1	B0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [B7, B6, B5, B4, B3, B2, B1, B0]*

STEP 4c – Write the contents of the third byte (of this new ATM cell) into the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 1; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN716)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
C7	C6	C5	C4	C3	C2	C1	C0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [C7, C6, C5, C4, C3, C2, C1, C0]*

STEP 4d – Write the contents of the fourth byte (of this new ATM cell) into the Receive ATM Cell – Insertion/Extraction Memory Register – Byte 0; as depicted below.

Receive ATM Cell – Insertion/Extraction Memory Register – Byte 2 (Address = 0xN717)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Insertion/Extraction Memory Data[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D7	D6	D5	D4	D3	D2	D1	D0

Note: *In this case, the contents of the second byte (within this ATM cell) is of the value [D7, D6, D5, D4, D3, D2, D1, D0]*

STEP 5 – Inform the "Cell Insertion Processor" that the very next 32-bit word to be written into the "Cell Insertion Buffer" is NOT the first word of a new ATM cell.

This is accomplished by writing the value "0" into Bit 0 (Insertion Memory Write SoC), within the "Receive ATM Cell – Memory Control Register" as depicted below.

Receive ATM Cell – Memory Control Register (Address = 0xN713)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Extraction Memory RESET*	Extraction Memory CLAV	Insertion Memory RESET*	Insertion Memory ROOM	Insertion Memory Write SoC
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	0

STEP 6 – Continue to write the contents of the remaining bytes of this ATM cell into the "Cell Insertion Memory".

This is accomplished by executing the procedure, outlined in STEP 4, repeatedly for 13 times.

Notes:

As the user writes the contents of an ATM cell into the "Receive Cell Insertion Memory" they must follow the mandated ATM cell byte format, as depicted in Figure 106.

If the user performs a READ operation to the "Receive ATM Cell – Insertion/Extraction Memory Register – Bytes 3 through 0" they will NOT be reading out the contents of the Receive Cell Insertion Buffer. Instead, the user will be reading out data from the "Receive Cell Extraction" buffer.

2.3.4.7 THE RECEIVE GFC NIBBLE-FIELD SERIAL OUTPUT PORT

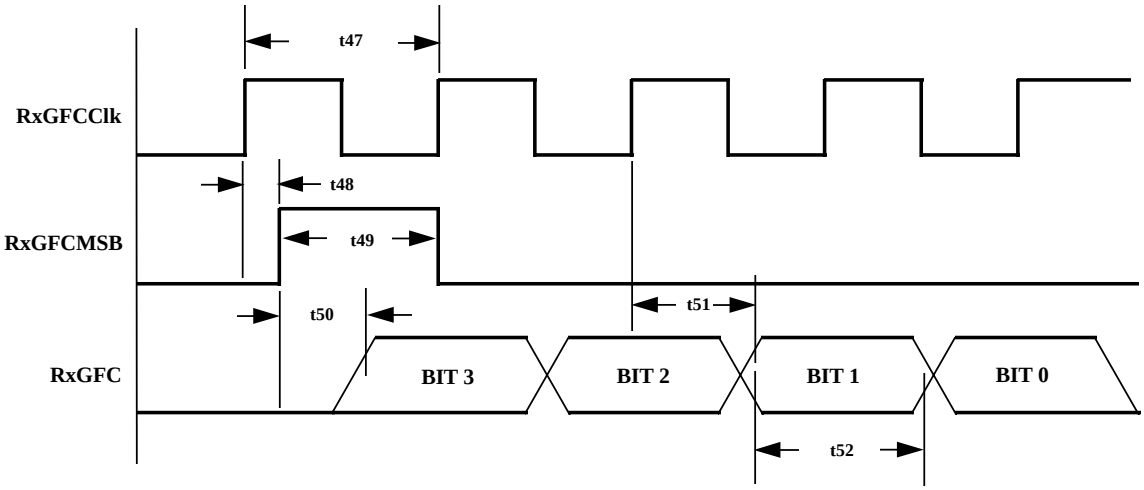
The first four bit-field of each cell header are the GFC bits. The Receive Cell processor will output the contents of the GFC Nibble-field for each cell that it receives, via the "GFC Nibble Field" serial output port.

The "Receive GFC Nibble-Field" serial output port consists of the following pins.

- RxGFC
- RxGFCClk
- RxGFCMSB

The data is output via the RxGFC output pin. The order of transmission, within a given cell, is with the MSB first and in descending order until transmitting the LSB bit. Afterwards, the "GFC Nibble-field" serial output port will output the MSB for the GFC Nibble-field of the next cell. This data is clocked out on the rising edge of the RxGFCClk output signal. The RxGFCMSB output pin will be pulsed "high" each time the MSB of the GFC Nibble field, for a given cell, is present at the RxGFC input. Figure 107 presents an illustration depicting the behavior of the RxGFC Serial Output Port signals.

Figure 107 Illustration of the Behavior of the RxGFC Serial Output Port signals



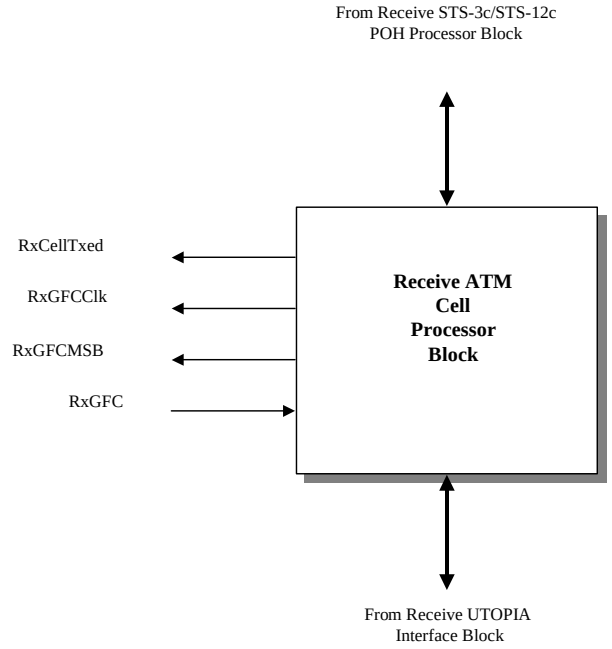
4.3.4.8 RECEIVE ATM CELL PROCESSOR BLOCK INTERRUPTS

- Cell Delineation
- HEC Byte Verification
- Idle Cell Filtering (optional)
- User/OAM Cell Filtering (optional)
- Cell-payload de-scrambling (optional)

The Receive ATM Cell Processor will also output the GFC Nibble value of each incoming cell, via the "Receive GFC Nibble Field" Serial Output port.

Figure 108 presents a simple block diagram of the Receive ATM Cell Processor block along with its external pins.

Figure 108 Simple Illustration of the Receive ATM Cell Processor, with associated Pins



2.3.4.8 7.3.2.8 RECEIVE ATM CELL PROCESSOR INTERRUPTS

The Receive Cell Processor will generate interrupts upon

- HEC Errors
- OAM Cell received
- Loss of Cell Delineation

If one of these conditions occur, and if that particular condition is enabled for interrupt generation, then when the local μC/μP reads the UNI Interrupt Status Register, as shown below, it should read '1xxxxxxxb' (where the -b suffix denotes a binary expression, and

'x' denotes a "don't care" value).

Operation Block Interrupt Status Register – Byte 0 (Address = 0x0113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive ATM Cell Processor Block Interrupt Status	Receive STS-3 TOH Block Interrupt Status	Receive SONET/VC-3 POH Block Interrupt Status	Receive PPP Processor Block Interrupt Status	Transmit ATM Cell Processor Block Interrupt Status	Unused		Transmit PPP Processor Block Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
1	0	0	0	0	0	0	0

At this point, the local μC/μP will have determined that the Receive Cell Processor block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly. In order to accomplish this the local μC/μP should now read the "Rx CP Interrupt Status Register" (Address = 0x0113). The bit format of this register is presented below.

CELL DELINEATION ALGORITHM

HEC BYTE VERIFICATION

COSET POLYNOMIAL

CORRECTION MODE

DETECTION MODE

IDLE CELL FILTERING

USER CELL FILTERING

DESCRAMBLING

GFC (GENERIC FLOW CONTROL) EXTRACTION

OAM CELL SUPPORT AND RECEPTION

INTERRUPT SUPPORT FOR THE RECEIVE CELL PROCESSOR BLOCK

2.3.5 THE RECEIVE UTOPIA INTERFACE BLOCK

The Receive UTOPIA Interface Block complies with "UTOPIA Level 1, 2 and 3" compliant interface to interconnect the UNI chip to ATM layer or ATM Adaptation Layer processors, operating up to 800 Mbps. Additionally, the XRT94L33 provides the user with the option of varying the following features associated with the Receive UTOPIA Bus Interface.

- Operating the UTOPIA Data Bus per the UTOPIA Levels 1, 2 and 3 standards
- Transmit UTOPIA Data Bus width of 8 or 16 bits
- The cell size (e.g., the number of octets being processed per cell via the UTOPIA bus)
- Whether the Receive UTOPIA Clock De-Skewing PLL is enabled or not.

Figure 109 presents a simplistic illustration of the Receive UTOPIA Interface block, along with its external output pins.

Figure 109 A Simple Illustration of the Receive UTOPIA Interface block

A discussion of the operation of the Receive UTOPIA Bus Interface along with each of these options will be presented below.

THE PINS OF THE RECEIVE UTOPIA BUS INTERFACE

The ATM Layer Processor will interface to the Receive UTOPIA Interface block via the following pins.

- RxUData[15:0] - Receive UTOPIA Data Bus Input pins
- RxUAddr[4:0] - Receive UTOPIA Address Bus Input pins
- RxUClk Receive UTOPIA Interface block clock input pin
- RxUClkO Receive UTOPIA Interface block clock output pin
- RxUSoC Receive "Start of Cell" Indicator input pin
- RxUPrty Receive UTOPIA – Odd Parity Output pin
- RxUEnB* Receive UTOPIA Data Bus – Read Enable Input pin

- RxUClav/RFullB* RxFIFO Cell Available

RxUData[15:0] - Receive UTOPIA Data Bus outputs

The ATM Layer Processor will read ATM cell data from the Receive UTOPIA Interface block in a byte-wide (or word-wide) manner, via these output pins. The Receive UTOPIA Data bus can be configured to operate in the "8 bit wide" or "16 bit wide" mode (See Section _). If the "8-bit wide" mode is selected, then only the RxUData[7:0] output pins will be active and capable of transmitting data. If the 16-bit wide mode is selected, then all 16 output pins (e.g., RxUData[15:0]) will be active. The Receive UTOPIA Data bus is tri-stated while the active low RxUEnB* (Receive UTOPIA Bus - Output Enable) input signal is "high". Therefore, the ATM Layer Processor must assert this signal (e.g., toggle RxUEnB* low) in order to read the ATM cell data from the Receive UTOPIA Interface block. The data on the Receive UTOPIA Data Bus output pins are updated on the rising edge of the Receive UTOPIA Interface block clock signal, RxUClk.

RxUAddr[4:0] - Receive UTOPIA Address Bus inputs

These input pins are used only when the UNI is operating in the Multi-PHY mode. Therefore, for more information on the Receive UTOPIA Address Bus, please see Section _.

RxUClk - Receive UTOPIA Interface block - Clock Signal input pin

The Receive UTOPIA Interface block uses this signal to update the data on the Receive UTOPIA Data Bus. The Receive UTOPIA Interface block also uses this signal to sample and latch the data on the Receive UTOPIA Address bus pins (during Multi-PHY operation), into the Receive UTOPIA Interface block circuitry. This clock signal can run at frequencies of 25 MHz, 33 MHz, or 50 MHz.

RXUCLKO – RECEIVE UTOPIA INTERFACE BLOCK CLOCK OUTPUT SIGNAL

If the "Receive UTOPIA Clock De-Skewing" PLL is enabled, then the ATM Layer Processor can use to this signal to sample the contents of the "outbound" ATM cells, as well as to when to assert the UTOPIA Address (during Multi-PHY operation). Similar to the "RxUClk" signal, this signal can run at frequencies up to 50MHz.

RxUEnB* - Receive UTOPIA Data Bus - Output Enable Input

The Receive UTOPIA Data bus is tri-stated while this input signal is negated. Therefore, the ATM Layer Processor must assert this "active-low" signal (toggle it "low") in order to read the byte (or word) from the Receive UTOPIA Interface block via the Receive UTOPIA Data bus.

RxUPrty - Receive UTOPIA - Odd Parity Bit output pin

The Receive UTOPIA Interface Block will compute the odd-parity of each byte (or word) of ATM cell data that it will place on the Receive UTOPIA Data bus. The Receive UTOPIA Data bus will output the value of the computed parity bit at the RxUPrty output pin, while the corresponding byte (or word) is present on the Receive UTOPIA Data Bus. This features allows the ATM Layer Processor to perform parity checking on the data that it receives from the Receive UTOPIA Interface Block.

RxUSoC - Receive UTOPIA - "Start of Cell" Indicator output pin

The Receive UTOPIA Interface block will pulse this output signal "high", for one clock period of RxUClk, when the first byte (or word) of a new ATM cell is present on the Receive UTOPIA Data Bus. This signal will be "low" at all other times.

RxUClav/RxEmptyB* - Rx FIFO Cell Available/RxEmpty*

This output signal is used to alert the ATM Layer Processor that the Rx FIFO contains some ATM cell data that is available for reading. Please see Section _ for more information regarding this signal.

SELECTING THE UTOPIA LEVEL

The XRT94L33 permits the user to configure the Receive UTOPIA Interface block in either of the following "UTOPIA Levels".

- UTOPIA Level 3

- UTOPIA Level 1 or 2

The user can configure the Receive UTOPIA Interface block (within the XRT94L33) to operate in the appropriate UTOPIA Level, by writing the appropriate value into Bit 7 (UTOPIA Level) within the "Receive UTOPIA Control Register", as depicted below.

Receive UTOPIA Control Register – Byte 0, Address = 0x0403

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Receive UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	0	0	0	X	X	1	1

Setting this bit-field to "0" configures the Receive UTOPIA Interface block to support "UTOPIA Level 3" signaling. Conversely, setting this bit-field to "1" configures the Receive UTOPIA Interface block to support the "UTOPIA Levels 1 and 2" form of signaling. A description of the operation of the Receive UTOPIA Interface block, for UTOPIA Level 1, 2 and 3 operation is presented below.

UTOPIA Level 1 and 2 Operation

UTOPIA Level 3 Operation

SELECTING THE UTOPIA DATA BUS WIDTH

The UTOPIA data bus width can be selected to be either 8 or 16 bits by writing the appropriate data into Bits 3 and 2 (Receive UTOPIA Data Bus Width[1:0]) within the "Receive UTOPIA Control" Register, as depicted below.

Receive UTOPIA Control Register – Byte 0, Address = 0x0403

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Receive UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	X	X	1	1

If the user chooses a UTOPIA Data Bus width of 8 bits, then only the Receive UTOPIA Data outputs: RxUData[15:8] will be active. (The output pins: RxUData[7:0] will not be active). If the user chooses a UTOPIA Data bus width of 16 bits, then all of the Receive UTOPIA Data output: RxUData[15:0] will be active. The following table relates the value of Bits 2 and 3 (Receive UTOPIA Data Bus Width[1:0]) within the Receive UTOPIA Control Register, to the corresponding width of the UTOPIA Data bus.

Table 20 The Relationship between the contents of "Receive UTOPIA Data Bus Width[1:0] within the Receive UTOPIA Control Register and the operating width of the UTOPIA Data Bus

RECEIVE UTOPIA DATA BUS WIDTH[1:0]	WIDTH OF UTOPIA DATA BUS
00	In-active:
01	8 bits
10	16 bits
11	Not valid (do not use)

Selecting the Cell Size (Number of Octets per Cell)

The XRT94L33 permits the user to select the number of octets per cell that the Receive UTOPIA Interface block will process. Specifically, the user has the following cell size options.

- If the UTOPIA Data Bus width is set to 8 bits then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  53 bytes (with either a dummy or actual HEC byte in the cell)

- If the UTOPIA Data Bus width is set to 16 bits, then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  54 bytes (with either a dummy or actual HEC byte, and a stuff byte in the cell)

The user makes his/her selection by writing the appropriate data into bits 1 and 0 (Cell_Size_Sel[1:0]) within the Receive UTOPIA Control Register, as depicted below.

Receive UTOPIA Control Register – Byte 0, Address = 0x0403

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level	Multi-PHY Mode	Back-to-Back Polling Enable	Direct Status Access	Receive UTOPIA Data Bus Width[1:0]		Cell_Size_Sel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	X	X

The following table presents the relationship between the value of this bit and the number of octets/cell that the Receive UTOPIA Interface block will process.

Table 21 The Relationship between the contents of Bits 1 and 0 (Cell_Size_Sel[1:0]) within the Transmit UTOPIA Control Register, and the number of octets per cell that will be processed by the Transmit UTOPIA Interface blocks per assertion of TxUSOC

CELL_SIZE_SEL[1:0]	NUMBER OF BYTES/CELLS
00	52 bytes/cell
01	53 bytes/cell (only value if the Transmit UTOPIA Data Bus Width = 8 bits)
10	54 bytes/cell
11	Unused

Once the user has implemented his/her selection for the cell size, then the Receive UTOPIA Interface block will be configured to process the "Cell Size" number of octets per cell.

CELL LEVEL HANDSHAKING

ATM Forum documentation refers to both "Cell Level" and "Octet-Level" handshaking. However, the XRT94L33 only supports the "Cell-Level" Handshaking mode. Octet-level handshaking is NOT supported. In the "Cell-Level" Handshaking mode, when the RxUClav output is at a logic "1", it means that the Rx FIFO contains at least one complete ATM cell of data that is available for reading by the ATM Layer Processor. When RxUClav toggles from "high" to "low", it indicates that Rx FIFO contains less than one complete ATM cell. The ATM Layer processor is expected to monitor the RxUClav output, and quickly respond and read the Rx FIFO, whenever the RxUClav output signal is asserted.

Figure 110 presents a timing diagram that illustrates the behavior of various Receive UTOPIA Interface block signals, when the Receive UTOPIA Interface block is operating in the "Cell-Level" Handshaking Mode.

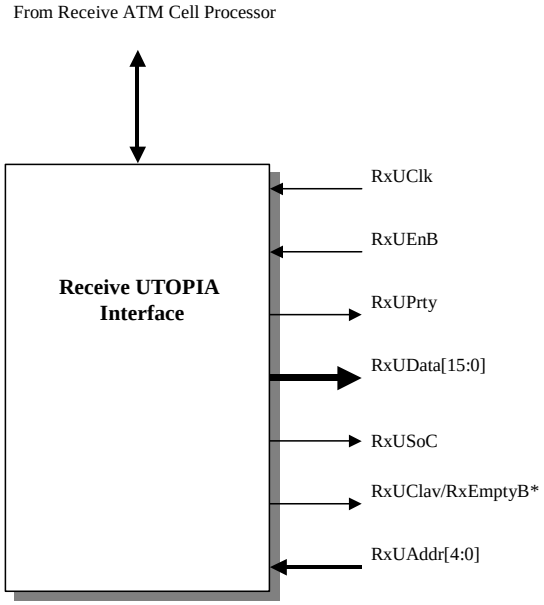
Figure 110

The UNI can operate in either the “Octet-Level” or “Cell-Level” Handshake mode, when operating in the Single-PHY mode. However, only the Cell-Level Handshake Mode is available when the UNI is operating in the Multi-PHY mode. For more information on Single PHY and Multi PHY operation, please see Section _.

The user can configure the UNI to operate in one of these two handshake modes by writing the appropriate data to Bit 5 (Handshake Mode) of the UTOPIA Configuration Register, as depicted below.

This interface supports both an 8 and 16 bit wide data bus. Since data is received at clock rates independent of the ATM layer clock rate, the received cell data is written into an internal FIFO by the Receive Cell Processor block. This FIFO will be referred to as the Rx FIFO throughout this document. The Receive Cell Processor will delineate, check for HEC byte errors, filter and de-scramble ATM Cells. Whatever cells were not discarded, by the Receive Cell Processor, will be written into the Rx FIFO, where it can be read out from the UNI device, by the ATM Layer Processor. The Receive UTOPIA Interface Block will inform the ATM Layer processor that it has cell data available for reading, by asserting the RxUClav pin “high”. Figure 111 presents a simple illustration of the Receive UTOPIA Interface block and the associated pins.

Figure 111 Simple Block Diagram of Receive UTOPIA Block of UNI



DETAILED FUNCTIONAL DESCRIPTION OF RECEIVE UTOPIA

The purposes of the Receive UTOPIA Interface block are to:

- Receive filtered ATM cell data from the Receive Cell Processor and make this data available to the AAL or ATM Layer Processor.
- Inform the ATM Layer Processor whenever the RxFIFO contains cell data that needs to be read.
- Inform the ATM Layer Processor that it has no more cell data to be read.

- Compute and present the odd-parity value of the byte (or word) that is present at the Receive UTOPIA Data Bus.

- Indicate the boundaries of cells, to the ATM Layer processor, by pulsing the RxUSoC (Receive Start of Cell) pin each time the first byte (or word) of a new cell is present on the Receive UTOPIA Data Bus.

The Receive UTOPIA Interface Block consists of the following sub-blocks:

- Receive UTOPIA Output Interface
- Receive UTOPIA Cell FIFO (Rx FIFO)
- Receive UTOPIA FIFO Manager

The Receive UTOPIA Interface block consists of an output interface complying to the "UTOPIA Level 2 Interface Specifications", and the RxFIFO. The width of the Receive UTOPIA Data Bus is user-configurable to be either 8 or 16 bits. The Receive UTOPIA Interface block also allows the ATM Layer processor to perform parity checking on all data that it receives from it (the Receive UTOPIA Interface block), over the Receive UTOPIA Data bus. The Receive UTOPIA Interface block computes the odd-parity of each byte (or word) that it will place on the Receive UTOPIA data bus. The Receive UTOPIA Interface block will then output the value of this computed parity at the RxPrty pin, while the corresponding data byte (word) is present at the RxData[15:0] output pins.

The Receive UTOPIA Interface block can be configured to process 52, 53, and 54 bytes per cell; and will assert the RxUSoC (Receive "Start of Cell") output pin at the cell boundaries. If the Receive UTOPIA Interface block detects a "runt" cell (e.g., a cell that is smaller than what the Receive UTOPIA Interface block has been configured to handle), it will generate an interrupt to the local μP, discard this "runt" cell, and resume normal operation.

The physical size of the Rx FIFO is sixteen cells. The incoming data (from the Receive Cell Processor) is written into the Rx FIFO, where it can be read in and processed by the ATM Layer Processor. A FIFO Manager maintains the Rx FIFO and indicates the FIFO Empty and FIFO Full to the local μP. Additionally the FIFO Manager will indicate that ATM Cell Data is available in the RxFIFO, by asserting the RxUClav output pin.

The following sections discusses each functional sub-block of the Receive UTOPIA Interface block in detail. Additionally, these sections discuss many the of the features associated with the Receive UTOPIA Interface block as well as how the user can optimize these features in order to suit his/her application needs. Detailed discussion of Single-PHY and Multi-PHY operation will be presented in its own section even though it involves the use of all of these functional blocks.

2.3.5.1 RECEIVE UTOPIA BUS OUTPUT INTERFACE

The Receive UTOPIA output interface complies with UTOPIA Level 1, 2 and 3 standard interface (e.g., the Receive UTOPIA can support both Single-PHY and Multi-PHY operations). Additionally, the XRT94L33 provides the user will the option of varying the following features associated with the Receive UTOPIA Bus interface.

- Operating the Receive UTOPIA Data Bus per the UTOPIA Levels 1, 2 or 3 standards
- Receive UTOPIA Data Bus width of 8 or 16 bits.
- The cell size (e.g., the number of octets being processed per cell via the UTOPIA bus)
- Assigning a UTOPIA Address to a given STS-3c port.
- Whether the Receive UTOPIA Clock De-Skewing PLL is enabled or not.

Figure 112 presents a simple illustration of the Receive UTOPIA Interface block, along with its external input pins.

Figure 112 A Simple Illustration of the Receive UTOPIA Interface block

A discussion of the operation of the Receive UTOPIA Bus Interface along with each of these options will be presented below.

2.3.5.1.1 The Pins of the Receive UTOPIA Bus Interface

The ATM Layer processor will interface to the Receive UTOPIA Interface block via the following pins.

- RxUData[15:0] - Receive UTOPIA Data Bus output pins.
- RxUAddr[4:0] - Receive UTOPIA Address Bus input pins.
- RxUClk - Receive UTOPIA Interface Block clock input pin.
- RxUSoC - Receive “Start of Cell” Indicator output pin.
- RxUPrty - Receive UTOPIA - Odd Parity output pin.
- RxUEnB* - Receive UTOPIA Data Bus - Output Enable input pin.
- RxUClav/RxFullB* - RxFIFO Cell Available output pin.

Each of these signals is briefly discussed below.

RxUData[15:0] - Receive UTOPIA Data Bus outputs

The ATM Layer Processor will read ATM cell data from the Receive UTOPIA Interface block in a byte-wide (or 16-bit word-wide) manner, via these output pins. The Receive UTOPIA Data bus can be configured to operate in the “8 bit wide” or “16 bit wide” mode (See Section _). If the “8-bit wide” mode is selected, then only the RxUData[7:0] output pins will be active and capable of transmitting data to the ATM Layer Processor. If the 16-bit wide mode is selected, then all 16 output pins (e.g., RxUData[15:0]) will be active. The Receive UTOPIA Data bus is tri-stated while the active low RxUEnB* (Receive UTOPIA Bus - Output Enable) input signal is “high”. Therefore, the ATM Layer Processor must assert this signal (e.g., toggle RxUEnB* low) in order to read the ATM cell data from the Receive UTOPIA Interface block. The data on the Receive UTOPIA Data Bus output pins are updated on the rising edge of the Receive UTOPIA Interface block clock signal, RxUClk.

RxUAddr[4:0] - Receive UTOPIA Address Bus inputs

These input pins are used only when the XRT94L33 is operating in the Multi-PHY mode. Therefore, for more information on the Receive UTOPIA Address Bus, please see Section _.

RxUClk - Receive UTOPIA Interface Block - Clock Signal input pin

The Receive UTOPIA Interface block uses this signal to update the data on the Receive UTOPIA Data Bus. The Receive UTOPIA Interface block also uses this signal to sample and latch the data on the Receive UTOPIA Address bus pins (during Multi-PHY operation), into the Receive UTOPIA Interface block circuitry. This clock signal can run at frequencies of up to 50 MHz.

RxUEnB* - Receive UTOPIA Data Bus - Output Enable Input

The Receive UTOPIA Data bus is tri-stated while this input signal is negated. Therefore, the user must assert this “active-low” signal (toggle it “low”) in order to read the byte (or word) from the Receive UTOPIA Interface block via the Receive UTOPIA Data bus.

RxUPrty - Receive UTOPIA - Odd Parity Bit output pin

The Receive UTOPIA Interface Block will compute the odd-parity of each byte (or word) of ATM cell data that it will place on the Receive UTOPIA Data bus. The Receive UTOPIA Data bus will output the value of the computed parity bit at the RxUPrty output pin, while the corresponding byte (or word) is present on the Receive UTOPIA Data Bus. This feature allows the ATM Layer Processor to perform parity checking on the data that it receives from the Receive UTOPIA Interface Block.

RxUSoC - Receive UTOPIA - “Start of Cell” Indicator output pin

The Receive UTOPIA Interface block will pulse this output signal “high”, for one clock period of RxUClk, when the first byte (or word) of a new cell is present on the Receive UTOPIA Data Bus. This output signal will be “low” at all other times.

RxUClav/RxEmptyB* - Rx FIFO Cell Available/RxEmpty*

This output signal is used to alert the ATM Layer Processor that the Rx FIFO contains some ATM cell data that is available for reading. Please see Section 7.4.2.2.1 for more information regarding this signal.

CONFIGURATION OPTIONS WITH THE RECEIVE UTOPIA INTERFACE BLOCK

Selecting the UTOPIA Level

The XRT94L33 permits the user to configure the Receive UTOPIA Interface block in either of the following "UTOPIA Levels".

- UTOPIA Level 3
- UTOPIA Level 1 or 2

The user can configure the Receive UTOPIA Interface block (within the XRT94L33) to operate in the appropriate UTOPIA Level, by writing the appropriate value into Bit 7 (UTOPIA Level) within the "Receive UTOPIA Control Register", as depicted below.

Receive UTOPIA/POS-PHY Control Register – Byte 0 (Address = 0x0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	Receive UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	1	0	0	1	1	1	1

Setting this bit-field to "0" configures the Receive UTOPIA Interface block to support "UTOPIA Level 3" signaling. Conversely, setting this bit-field to "1" configures the Receive UTOPIA Interface block to support the "UTOPIA Levels 1 and 2" form of signaling. A description of the operation of the Receive UTOPIA Interface block, for UTOPIA Level 1, 2 and 3 operation is presented below.

2.3.5.1.2 UTOPIA Level 1 and 2 Operation of the Receive UTOPIA Interface Block

This section presents an in-depth write up of the UTOPIA Level 1 and 2 protocols.

When the Receive UTOPIA Interface block has been configured to operate in the "UTOPIA Level 2" Mode, then it will either be configured to operate in the "Single-PHY" or "Multi-PHY" mode, as described below.

2.3.5.1.2.1 Selecting the UTOPIA Data Bus Width

The user can configure the width of the Receive UTOPIA Data Bus to be either 8 or 16 bits by writing the appropriate data into Bits 3 and 2 (Receive UTOPIA Data Bus Width[1:0]) within the "Receive UTOPIA Control" Register, as depicted below.

Receive UTOPIA/POS-PHY Control Register – Byte 0 (Address = 0x0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	Receive UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	0	0	X	X	1	1

If the user chooses a UTOPIA Data Bus width of 8 bits, then only the Receive UTOPIA Data outputs: RxUData[15:8] will be active. (The output pins: RxUData[7:0] will not be active). If the user chooses a UTOPIA Data Bus width of 16 bits, then all of the Receive UTOPIA Data outputs: RxUData[15:0] will be

active. The following table relates the value of Bits 2 and 3 (Receive UTOPIA Data Bus Width[1:0]) within the Receive UTOPIA Control Register, to the corresponding width of the Receive UTOPIA Data bus.

Note: *This configuration setting does not apply to the Transmit UTOPIA Interface block. The user will still need to specify the width of the Transmit UTOPIA Data Bus separately, as described in Section _.*

Table 22 The Relationship between the contents of "Receive UTOPIA Data Bus Width[1:0] within the Receive UTOPIA Control Register and the operating width of the Receive UTOPIA Data bus

RECEIVE UTOPIA DATA BUS WIDTH[1:0]	WIDTH OF RECEIVE UTOPIA DATA BUS
00	In-active:
01	8 bits
10	16 bits
11	Not valid (do not use)

SELECTING THE CELL SIZE (NUMBER OF OCTETS PER CELL)

The XRT94L34 device permits the user to select the number of octets per cell that the Receive UTOPIA Interface block will output, following each assertion of the RxUSoC output pin. Specifically, the user has the following cell size options.

- If the UTOPIA Data Bus width is set to 8 bits then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  53 bytes (with either a dummy or actual HEC byte in the cell)

- If the UTOPIA Data Bus width is set to 16 bits then the user can choose:

  52 bytes (with no HEC byte in the cell), or

  54 bytes (with either a dummy or actual HEC byte, and a stuff byte in the cell)

The user makes their selection by writing the appropriate data into bits 1 and 0 (Cell_Size_Sel[1:0]) within the Receive UTOPIA Control Register, as depicted below.

Receive UTOPIA/POS-PHY Control Register – Byte 0 (Address = 0x0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	Receive UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	X	X

The following table presents the relationship between the value of these bits and the number of octets/cell that the Receive UTOPIA Interface block will process.

Table 23 The Relationship between the contents of Bits 1 and 0 (Cell_Size_Sel[1:0]) within the Receive UTOPIA Control Register, and the number of octets per cell that will be processed by the Receive UTOPIA Interface blocks following each assertion of the RxUSOC output pin

CELL_SIZE_SEL[1:0]	NUMBER OF BYTES/CELLS
00	52 bytes/cell
01	53 bytes/cell (only valid if the Receive UTOPIA Data Bus Width = 8 bits)
10	54 bytes/cell
11	Unused

Once the user has implemented his/her selection for the cell size, then the Receive UTOPIA Interface block will be configured to output "Cell Size" number of octets, per each assertion of "RxUSoC".

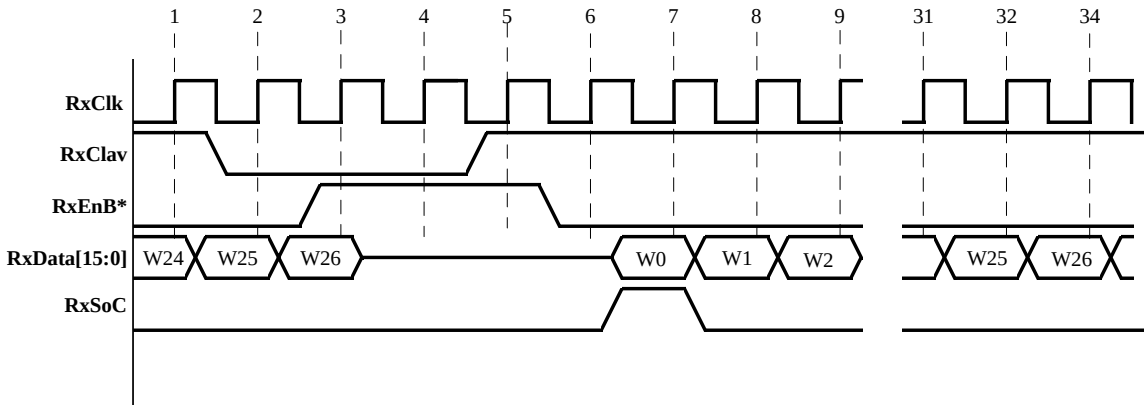
2.3.5.1.2.2 Cell Level Handshaking

ATM Forum documentation refers to both "Cell Level" and "Octet-Level" handshaking. However, the XRT94L33 only supports the "Cell-Level" Handshaking mode. Octet-level handshaking is NOT supported. In the "Cell Level Handshaking mode, when the XRT94L33 sets the RxUClav output pin to a logic "1", it means that the RxFIFO contains at least one ATM cell's worth of data that needs to be read out by the ATM Layer Processor. However, when RxUClav toggles from "high" to "low" it indicates that the RxFIFO does not contain any cells (following the one that is currently being read out of the Receive UTOPIA Interface block). The ATM Layer processor is expected to poll the state of the RxUClav output pin towards the end of the

reception of the cell currently being read out, and to proceed with reading out the next ATM cell from the Receive UTOPIA Interface only if RxUClav is at a logic "high".

Figure 113 presents a timing diagram of that illustrates the behavior of various Receive UTOPIA Interface block signals, when the Receive UTOPIA Interface block is operating the "Cell-Level" Handshaking Mode.

Figure 113 Timing Diagram of various Receive UTOPIA Interface block signals, when the Receive UTOPIA Interface block is operating in the "Cell Level Handshaking" Mode



Notes: *regarding Figure 113:*

The Receive UTOPIA Data bus is configured to be 16 bits wide. Hence, the data, which the Receive UTOPIA places on the Receive UTOPIA Data bus, is expressed in terms of 16 bit words: W0 - W26.

The Receive UTOPIA Interface block is configured to handle 54 bytes/cell. Hence, Figure 86 illustrates the ATM Layer processor reading in 27 words (W0 through W26) for each ATM cell.

In Figure 113 the ATM Layer processor is just finishing up its reading of an ATM cell. Prior to clock edge #2, the RxFIFO does not contain enough ATM cell data to make up at least one cell. Hence, the Receive UTOPIA Interface block negates the RxUClav signal. The ATM Layer processor detects that the RxClav signal has toggled "low"; at clock edge #2. Hence, the ATM Layer processor will finish reading in the current ATM cell; from the Receive UTOPIA Interface block of the XRT94L33 (e.g., words W25 and W26). Afterwards, the ATM Layer processor will negate the RxUEnB* signal and will cease to read in anymore ATM cell data from the Receive UTOPIA Interface block; until RxUClav toggles "high" again.

The RxFIFO accumulates enough cell data to make up a complete ATM cell shortly before clock edge #5. At this point the Receive UTOPIA Interface block reflects this fact by asserting the RxUClav signal. The ATM Layer processor detects that the RxUClav signal has toggled "high" at clock edge #5. Consequently, the ATM Layer processor then asserts the RxUEnB* signal (e.g., toggles it "low") after clock edge #5. The Receive UTOPIA Interface block detects the fact that the RxUEnB* input pin has been asserted at clock edge #6. The Receive UTOPIA Interface block then responds to this signaling by placing the first word of the next cell on the Receive UTOPIA Data bus. Afterwards, the ATM Layer processor continues to read in the remaining words of this cell.

2.3.5.1.2.3 UTOPIA Modes of Operation (Single PHY and Multi-PHY operation)

The XRT94L33 can be configured to support either Single-PHY or Multi-PHY operation. Each of these operating modes is discussed below.

2.3.5.1.2.4 Single PHY Operation

The XRT94L33 permits the user to configure it to operate in either the "Single-PHY" or "Multi-PHY" Mode. The user can configure the chip to operate in the "Single-PHY" Mode by setting Bit 6 (Multi-PHY Mode) to "0"; as illustrated below.

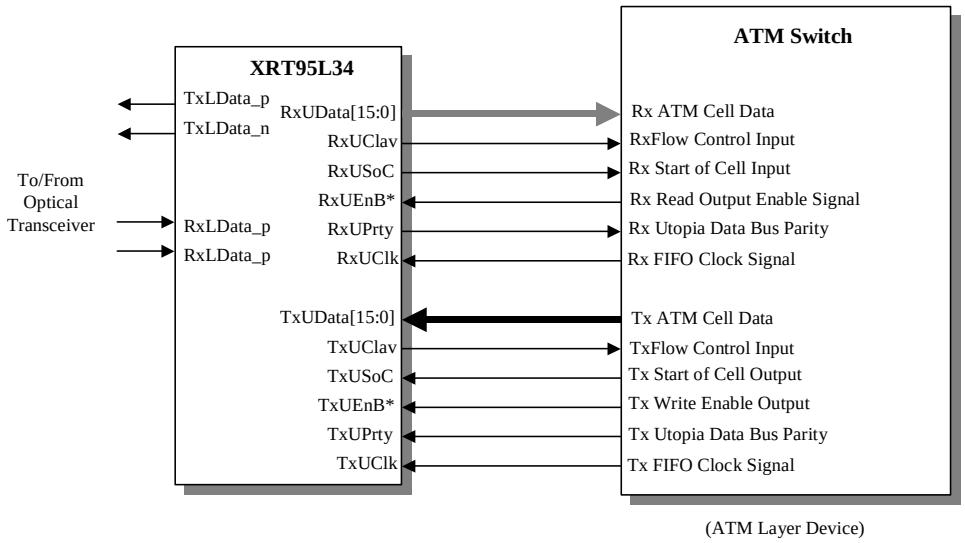
Receive UTOPIA/POS-PHY Control Register – Byte 0 (Address = 0x0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	Receive UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	X	X

Note: *This configuration setting does not apply to the Transmit UTOPIA Interface block. Therefore, the user will also need to configure the Transmit UTOPIA Interface block into the Single-PHY Mode, as described in Section _.*

In Single-PHY Mode operation, the ATM Layer Processor is pumping data into and receiving data from only one PHY-Layer device, as depicted below in Figure 114

Figure 114 Simple Illustration of Single - PHY Mode Operation



This section presents a detailed description of the Receive UTOPIA Interface block operating in the "Single-PHY" Mode. A description of the Transmit UTOPIA Interface block operating in the "Single-PHY" Mode is presented in Section _. Whenever the Receive UTOPIA Interface block has been configured to operate in the Single-PHY Mode, and whenever the ATM Layer Processor wishes to read out one or a series of ATM cells from the Receive UTOPIA Interface block, it must do the following.

1. Check the level of the RxUClav pin

If the RxUClav pin is "high" then the RxFIFO contains some ATM cell data that needs to be read by the ATM Layer processor. In this case, the ATM Layer processor should begin to read the cell data from the Receive UTOPIA Interface block. However, if the RxUClav pin is "low", then the RxFIFO does not contain any cell data, that can be read. In this case, the ATM Layer processor should wait until RxUClav toggles "high" before attempting to read any more cell data from the "Receive UTOPIA Interface block".

2. Assert the RxUEnB* pin and read the first byte (or word) of the new cell from the Receive UTOPIA Data Bus.

Once the ATM Layer processor has detected that RxUClav has toggled "high", then it should assert the RxUEnB* input pin (e.g., toggling it "low"). Once the Receive UTOPIA Interface block has determined that the RxUEnB* input pin is "low", then it will begin to place some cell data onto the Receive UTOPIA Data Bus. If this first byte (or word) is the beginning of a new ATM cell, then the ATM Layer processor should verify that this byte (or word) is indeed the beginning of a new cell, by observing the RxUSoC output pin (of the XRT94L33 IC) pulsing "high" for one clock period of RxUClk.

3. Compute the odd-parity of the byte (or word) that is being read from the Receive UTOPIA Data bus, and compare the value of this parity bit with that of the RxUPrty output pin.

This operation is optional, but should be done concurrently while checking for the assertion of the RxUSoc output pin.

When reading in the subsequent bytes (or words) of the cell, the ATM Layer must do the following.

- Repeat Steps 1 and 2.
- The ATM Layer processor should check the RxUClav signal level just as it (the ATM Layer processor) is reading in the very last byte (or word) of a given cell. If the RxUClav level is "high", then the ATM Layer processor should proceed to read in the next cell from the Receive UTOPIA Interface block. However, if the RxUClav level is "low", then the ATM Layer processor should halt reading in data, when it reaches the end of the cell (that it is currently reading in).
- The ATM Layer processor should keep a count on the total number of bytes that have been read in since the last assertion of the RxUSoC output pin. This will help the ATM Layer processor to determine when it has reached the boundary of a given cell.

The above-mentioned procedure is also depicted in "Flow Chart Form" in Figure 115 and in Timing Diagram form in Figure 116.

Figure 115 Flow Chart depicting the approach that the ATM Layer Processor should take when reading ATM Cell data from the Receive UTOPIA Interface, when the XRT94L33 is operating in the Single-PHY Mode

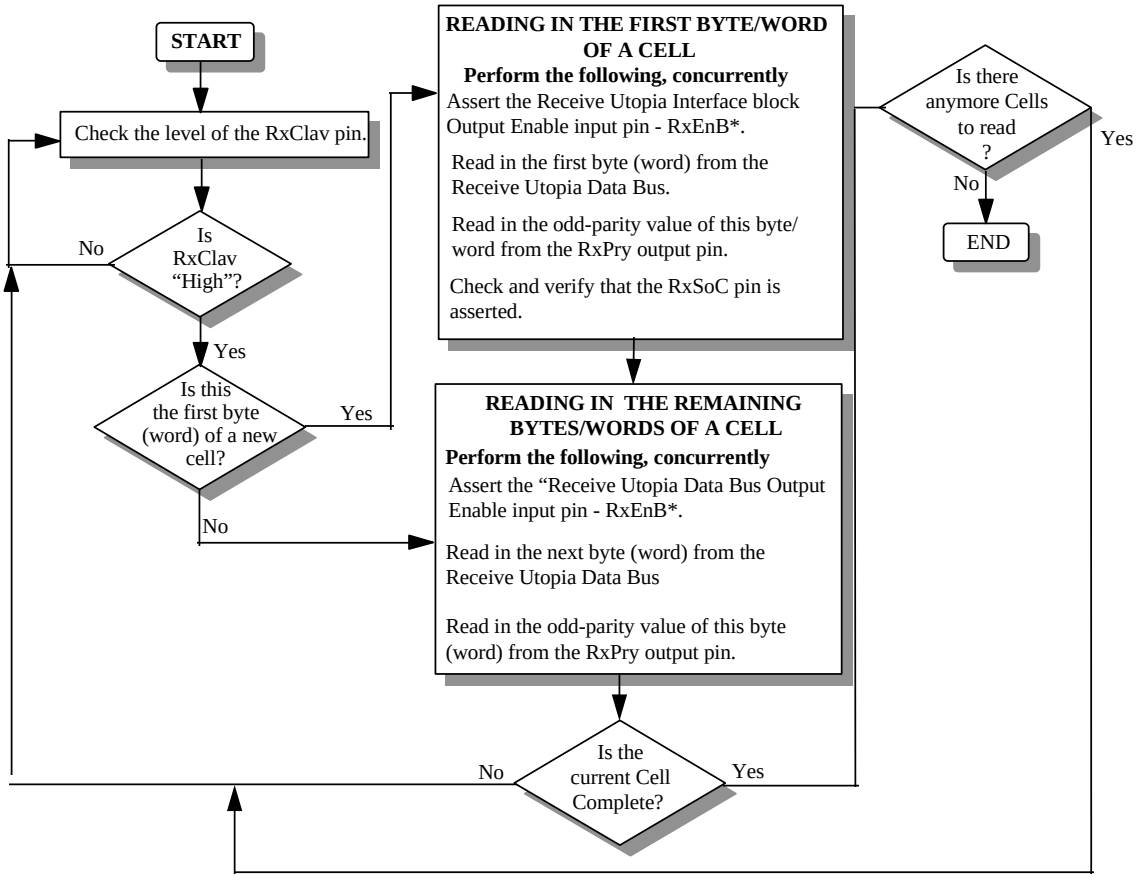
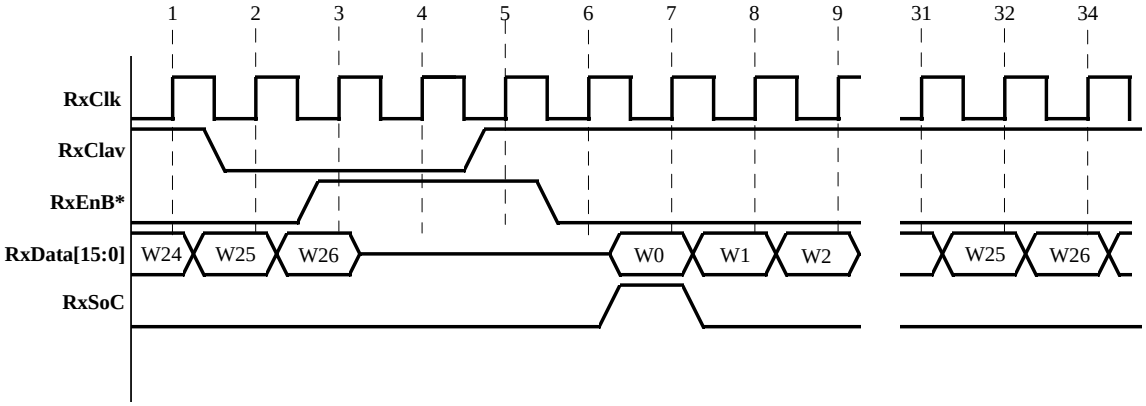


Figure 116 Timing Diagram of various Receive UTOPIA Interface block signals, when the Receive UTOPIA Interface block is operating in the "Cell Level Handshaking" Mode



Notes regarding Figure 116**:**

The Receive UTOPIA Data bus is configured to be 16 bits wide. Hence, the data, which the Receive UTOPIA places on the Receive UTOPIA Data bus, is expressed in terms of 16 bit words: W0 - W26.

The Receive UTOPIA Interface block is configured to handle 54 bytes/cell. Hence, Figure 86 illustrates the ATM Layer processor reading in 27 words (W0 through W26) for each ATM cell.

Figure 116, the ATM Layer processor is just finishing up its reading of an ATM cell. Prior to clock edge #2, the RxFIFO does not contain enough ATM cell data to make up at least one cell. Hence, the Receive

UTOPIA Interface block negates the RxUClav signal. The ATM Layer processor detects that the RxClav signal has toggled "low"; at clock edge #2. Hence, the ATM Layer processor will finish reading in the current ATM cell; from the Receive UTOPIA Interface block of the XRT94L33 (e.g., words W25 and W26). Afterwards, the ATM Layer processor will negate the RxUEnB* signal and will cease to read in anymore ATM cell data from the Receive UTOPIA Interface block; until RxUClav toggles "high" again.

The RxFIFO accumulates enough cell data to make up a complete ATM cell shortly before clock edge #5. At this point the Receive UTOPIA Interface block reflects this fact by asserting the RxUClav signal. The ATM Layer processor detects that the RxUClav signal has toggled "high" at clock edge #5. Consequently, the ATM Layer processor then asserts the RxUEnB* signal (e.g., toggles it "low") after clock edge #5. The Receive UTOPIA Interface block detects the fact that the RxUEnB* input pin has been asserted at clock edge #6. The Receive UTOPIA Interface block then responds to this signaling by placing the first word of the next cell on the Receive UTOPIA Data bus. Afterwards, the ATM Layer processor continues to read in the remaining words of this cell.

2.3.5.1.2.5 Multi-PHY Operation

The XRT94L33 permits the user to configure it to operate in the "Multi-PHY" Mode. This can be accomplished by setting Bit 6 (Multi-PHY Mode), within the "Receive UTOPIA Control Register – Byte 0" to "1" as depicted below.

Receive UTOPIA/POS-PHY Control Register – Byte 0 (Address = 0x0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	Receive UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	0	0	1	1	X	X

Note: *This configuration setting does not apply to the Transmit UTOPIA Interface block. Therefore, the user will also need to configure the Transmit UTOPIA Interface block into the "Multi-PHY" Mode, as described in Section _.*

In the "Multi-PHY" operating mode, the ATM Layer Processor may be writing data into and reading data from several UNI (e.g., PHY Layer) devices in parallel. Figure 114 presents an illustration of a simple "Multi-PHY System" consisting of a single ATM Layer Processor being interfaced to two (2) UNI devices. When the XRT94L33 is operating in the Multi-PHY mode, the Receive UTOPIA Interface block will support two kinds of operations with the ATM Layer Processor.

- Polling for UNI (PHY Layer) devices that contain ATM cell waiting to be read.
- Selecting which UNI (out of several possible UNI devices) to read ATM cell data from.

Each of these operations is discussed in the sections below. However, prior to discussing each of these operations, the reader must understand the following.

"Multi-PHY" operation involves the use of one (1) ATM Layer processor and several UNI (or PHY-Layer) devices, within a system. The ATM Layer processor is expected to read/write ATM cell data from/to these UNI devices. Hence, "Multi-PHY" operation requires, at a minimum, some means for the ATM Layer processor to uniquely identify a particular UNI device (among all of the UNI devices within the "Multi-PHY" system) that it wishes to "poll", write ATM cell data to, or read ATM cell data from. Actually, "Multi-PHY" operation provides an addressing scheme allows the ATM Layer processor to uniquely identify "UTOPIA Interface Blocks" (e.g., Transmit and Receive) within all of the UNI devices, operating in the "Multi-PHY" system. In order to uniquely identify a given "UTOPIA Interface block", within a "Multi-PHY" system, each "Transmit and Receive UTOPIA Interface Blocks are assigned a 5-bit "UTOPIA address" value. The user assigns this address value to a particular "Receive UTOPIA Interface block" by writing this address value into

the “Receive UTOPIA Address Register” (Address = 0x0513) and appropriate data into the “Receive UTOPIA Port Address” Register (Address = 0x0517); within its “host” XRT94L33; per the procedure (as presented below). The bit-format of the “Receive UTOPIA Address Register” and “Receive Port Address” Register is presented below.

Receive UTOPIA Address Register (Address = 0x0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

Receive UTOPIA Port Number Register (Address = 0x0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

2.3.5.1.2.6 Assigning Receive UTOPIA Addresses to Each STS-3c Channel within the XRT94L33

The XRT94L33 can be configured to function as a four channel ATM UNI over either an STS-3 or an STS-3c signal. As a consequence, for Multi-PHY Operation, the XRT94L33 can be assigned as many as three "Receive UTOPIA Addresses" at a given time (one for each STS-1 channel).

The user can assign a "Receive UTOPIA Address" value to a given channel, within the XRT94L33 by executing the following steps.

STEP 1 – Assign a "Receive UTOPIA Address" to Channel 0

The user can accomplish this by doing the following.

STEP 1a – Set Bits 0 through 4 ("Rx_UTOPIA_Port_Number[4:0]") within the "Receive UTOPIA Port Number Register" to "0x00"; as depicted below.

Receive UTOPIA Port Number Register (Address = 0x0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures an internal "Receive UTOPIA Address assignment" pointer to point to "STS-3c Channel 0" within the XRT94L33.

STEP 1b – Set Bits 0 through 4 ("Rx_UTOPIA_Addr[4:0]") within the "Receive UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [a4, a3, a2, a1, a0]; as depicted below.

Receive UTOPIA Address Register (Address = 0x0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	A4	A3	A2	A1	A0

This step configures the Receive UTOPIA Address, of the value [A4, A3, A2, A1, A0] to be assigned to Channel 0.

Notes:

During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Receive UTOPIA Interface Addresses within a "Multi-PHY" system.

Use of the value "0x1F" is NOT permitted.

These steps do not assign a Transmit UTOPIA Address value to the STS-3c Channel 0. This assignment must be performed separately per the "Transmit UTOPIA Address" assignment instructions, presented in Section _.

Once the user has executed STEPS 1a and 1b, then STS-3c Channel 0 (within the XRT94L33) has been assigned the "Transmit UTOPIA Address" of [a4, a3, a2, a1, a0].

STEP 2 – Assign a "Receive UTOPIA Address" to STS-3c Channel 1

The user can accomplish this by doing the following.

STEP 2a – Set Bits 0 through 4 ("Rx_UTOPIA_Port_Number[4:0]") within the "Receive UTOPIA Port Number Register" to "0x01"; as depicted below.

Receive UTOPIA Port Number Register (Address = 0x0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This step configures an internal "Receive UTOPIA Address assignment" pointer to point to "STS-3c Channel 1" within the XRT94L33.

STEP 2b – Set Bits 0 through 4 ("Rx_UTOPIA_Addr[4:0]") within the "Receive UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [b4, b3, b2, b1, b0]; as depicted below.

Receive UTOPIA Address Register (Address = 0x0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	B4	B3	B2	B1	B0

This step configures the Receive UTOPIA Address, of the value [B4, B3, B2, B1, B0] to be assigned to Channel 1.

Notes:

During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Receive UTOPIA Addresses within a "Multi-PHY" system.

Use of the value "0x1F" is NOT permitted.

These steps do not assign a Transmit UTOPIA Address value to STS-3c Channel 1. This assignment must be performed separately per the "Transmit UTOPIA Address Assignment" instructions presented in Section _.

Once the user has executed STEPS 2a and 2b, then STS-3c Channel 1 (within the XRT94L33) has been assigned the "Receive UTOPIA Address" of [b4, b3, b2, b1, b0].

STEP 3 – Assign a "Receive UTOPIA Address" to STS-3c Channel 2

The user can accomplish this by doing the following.

STEP 3a – Set Bits 0 through 4 ("Rx_UTOPIA_Port_Number[4:0]") within the "Receive UTOPIA Port Number Register" to "0x02"; as depicted below.

Receive UTOPIA Port Number Register (Address = 0x0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

This step configures an internal "Receive UTOPIA Address assignment" pointer to point to "STS-3c Channel 2" within the XRT94L33.

STEP 3b – Set Bits 0 through 4 ("Rx_UTOPIA_Addr[4:0]") within the "Receive UTOPIA Address" Register to the desired "Multi-PHY" Address value for this channel [c4, c3, c2, c1, c0]; as depicted below.

Receive UTOPIA Address Register (Address = 0x0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	C4	C3	C2	C1	C0

This step configures the Receive UTOPIA Address, of the value [C4, C3, C2, C1, C0] to be assigned to STS-3c Channel 2.

Notes:

During this step, the user can write in any value, between 0x00 and 0x1E, provided this "Multi-PHY Address" is unique among all of the Receive UTOPIA Interface addresses within a "Multi-PHY" system.

Use of the value "0x1F" is NOT permitted.

These steps do not assign a Transmit UTOPIA Address value to STS-3c Channel 2. This assignment must be performed separately per the "Transmit UTOPIA Address Assignment" instructions presented in Section _.

Once the user has executed STEPS 3a and 3b then STS-3c Channel 2 (within the XRT94L33) has been assigned the "Receive UTOPIA Address" of [c4, c3, c2, c1, c0].

STEP 4 – Assign a "Receive UTOPIA Address" to STS-3c Channel 3

The user can accomplish this by doing the following.

STEP 4a – Set Bits 0 through 4 ("Rx_UTOPIA_Port_Number[4:0]") within the "Receive UTOPIA Port Number Register" to "0x03"; as depicted below.

Receive UTOPIA Port Number Register (Address = 0x0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Port_Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	1

This step configures an internal "Receive UTOPIA Address assignment" pointer to point to "STS-3c Channel 3" within the XRT94L33.

STEP 4b – Set Bits 0 through 4 ("Rx_UTOPIA_Addr[4:0]") within the "Receive UTOPIA Address" Register to the "Multi-PHY" Address value for this channel [d4, d3, d2, d1, d0]; as depicted below.

Receive UTOPIA Address Register (Address = 0x0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Rx_UTOPIA_Addr[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	D4	D3	D2	D1	D0

This step configures the Receive UTOPIA Address, of the value [D4, D3, D2, D1, D0] to be assigned to STS-3c Channel 3.

Notes:

During this step, the user can write in any value, between 0x00 and 0x1E provided this "Multi-PHY Address" is unique among all of the Receive UTOPIA Interface Addresses within a "Multi-PHY" system.

Use of the value "0x1F" is NOT permitted.

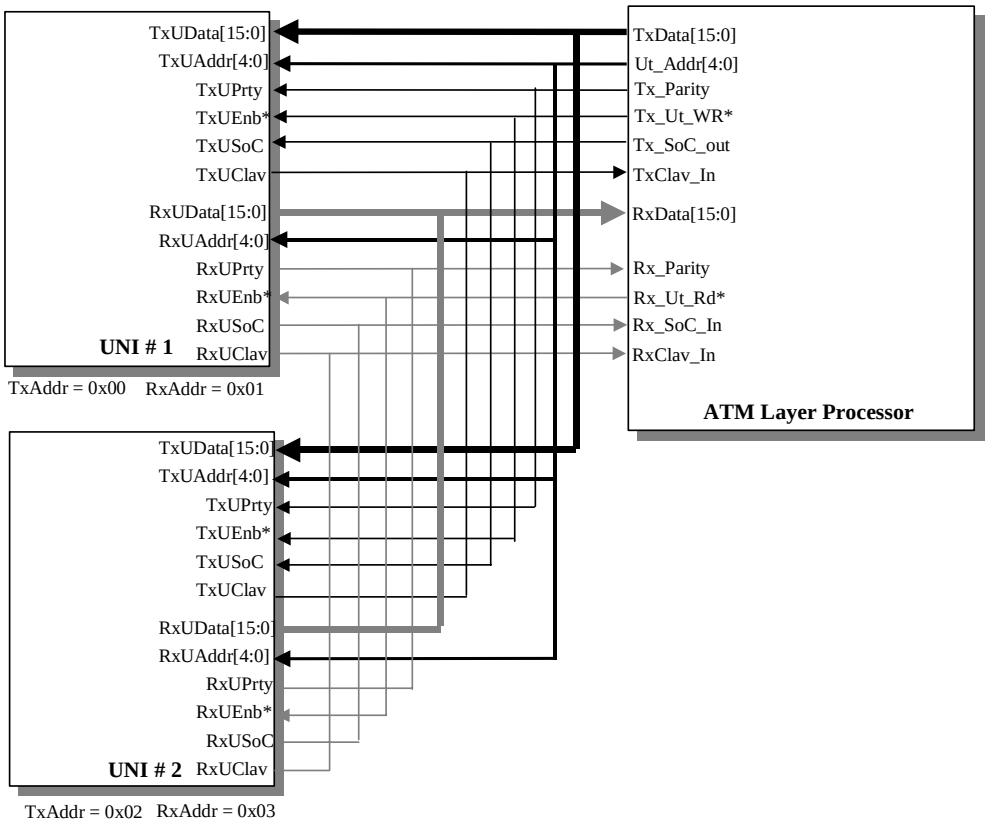
These steps do not assign a Transmit UTOPIA Address value to STS-3c Channel 3. This assignment must be performed separately per the "Transmit UTOPIA Address Assignment" instructions presented in Section _.

Once the user has executed STEPS 4a and 4b then STS-3c Channel 3 (within the XRT94L33) has been assigned the "Receive UTOPIA Address" of [d4, d3, d2, d1, d0].

2.3.5.1.2.7 ATM Layer Processor "polling" of the UNIs, in the Multi-PHY Mode

In this section, the various Multi-PHY Operations (e.g., polling and selection for writing) will be first discussed for a "Conceptual Multi-PHY" System, and then later, specifically for the XRT94L33. When the XRT94L33 is operating in the "Multi-PHY" mode, then the Receive UTOPIA Interface block will automatically be configured to support "polling". "Polling" allows an ATM Layer processor (which is interfaced to several UNI devices) to determine which UNIs are capable of receiving and handling additional ATM cell data, at any given time. The manner in which the ATM Layer processor "polls" its UNI devices, (per the "Conceptual Multi-PHY" system) follows.

Figure 117 An Illustration of the "Conceptual Multi-PHY System consisting of UNI Devices #1 and #2



2.3.5.1.2.8 ATM Layer Processor "polling" in a Conceptual Multi-PHY System

Figure 117depicts a "Multi-PHY" system consisting of a single ATM Layer processor and two (2) UNI devices, which are designated as "UNI #1" and "UNI #2". In this figure, both of the UNIs are connected to the ATM Layer processor via a common "Transmit UTOPIA" Data Bus, a common "Receive UTOPIA" Data Bus, a common "TxUClav" line, a common "RxUClav" line, as well as common TxUEnB*, RxUEnB*, TxUSoC and RxUSoC lines. The ATM Layer processor will also be addressing both the Transmit and Receive UTOPIA Interface blocks via a common "UTOPIA" address bus (Ut_Addr[4:0]) Therefore, the Transmit and Receive UTOPIA Interface Blocks, within a given UNI might have different addresses; as depicted in Figure 117.

The UTOPIA Address values, that have been assigned to each of the Transmit and Receive UTOPIA Interface blocks, within Figure 26, are listed below in Table 24.

Table 24 UTOPIA Address Values of the UTOPIA Interface blocks illustrated in Figure 117.

BLOCK	UTOPIA ADDRESS VALUE
Transmit UTOPIA Interface block - UNI #1	0x00
Receive UTOPIA Interface block - UNI #1	0x01
Transmit UTOPIA Interface block - UNI #2	0x02
Receive UTOPIA Interface block - UNI #2	0x03

Recall, that the Receive UTOPIA Interface blocks were assigned these addresses by writing these values into registers that are similar to the "Receive UTOPIA Port Number (Address = 0x0517) and the Receive UTOPIA Address Register" (Address = 0x0513) within these UNI devices. The discussion of the Transmit UTOPIA Interface blocks, within UNIs #1 and #2 is presented in Section _.

Polling Operation

Consider that the ATM Layer processor is currently reading a continuous stream of ATM cell data from UNI #1. While reading this ATM cell data from UNI #1, the ATM Layer processor can also "poll" UNI #2 for "availability" (e.g., tries to determine if UNI # 2 contains any ATM cells, within its RxFIFO, that needs to be read out via its Receive UTOPIA Interface block).

The ATM Layer processor's role in the "polling" operation

The ATM Layer processor accomplishes this "polling" operation by executing the following steps.

1. Assert the RxUEnB* input pin (if it is not asserted already).

The UNI device (being "polled") will know that this is only a "polling" operation, if the RxUEnB* input pin is asserted, prior to detecting its UTOPIA Address on the "UTOPIA Address" bus (RxUAddr[4:0]).

2. The ATM Layer processor places the address of the Receive UTOPIA Interface Block of UNI #2 onto the UTOPIA Address Bus, Ut_Addr[4:0],

3. The ATM Layer processor will then check the value of its "RxUClav_in" input pin (see Figure 117***).***

The ATM Layer Processor is suppose to check the state of the "RxUClav" signal, one "RxUClk" period after placing the UTOPIA Address (corresponding to a particular UNI device) on the "RxUAddr[4:0]" input pins. If "RxUClav" is sampled "high" then this means that this particular UNI device contains at least one ATM cell of data (within its RxFIFO) that needs to be read out by the ATM Layer Processor. Conversely, if RxUClav is sampled "low" then this means that this particular UNI device does not contain at least one ATM cell of data within its RxFIFO, that needs to be read out via the ATM Layer Processor.

The UNI devices role in the "polling" operation

UNI #2 will sample the signal levels placed on its Rx UTOPIA Address input pins (RxUAddr[4:0]) on the rising edge of its "Receive UTOPIA Interface block" clock input signal, RxUClk. Afterwards, UNI #2 will compare the

value of these “Receive UTOPIA Address Bus input pin” signals with that of the contents of its “Rx UTOPIA Address Register (Address = 0x0513).

If these values do not match, (e.g., RxUAddr[4:0] ≠ 0x03) then UNI #2 will keep its “RxUClav” output signal “tri-stated”; and will continue to sample its “Receive UTOPIA Address bus input” pins; with each rising edge of RxUClk.

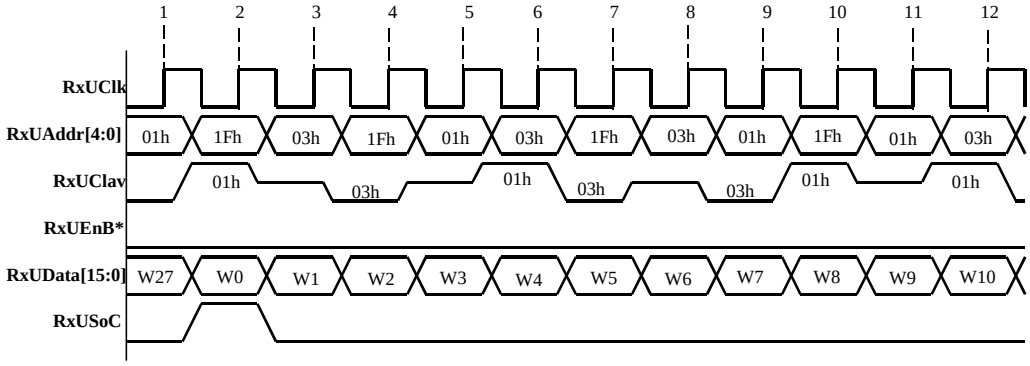
If these two values do match, (e.g., RxUAddr[4:0] = 0x03) then UNI #2 will drive its “RxUClav” output pin to the appropriate level, reflecting its RxFIFO “fill-status”. Since the UNI is only operating in the “Cell Level Handshaking” mode, the UNI will drive the RxUClav output signal “high” if it contains at least one complete ATM cell of data within its RxFIFO, that needs to be read out by the ATM Layer Processor. Conversely, the UNI will drive the “RxUClav” output signal “low” if its RxFIFO contains less than one complete ATM cell of data.

When UNI #2 has been selected for “polling”, UNI #1 will continue to keeps its “RxUClav” output signal “tri-stated”. Therefore, when UNI #2 is driving its “RxUClav” output pin to the appropriate level; it will be driving the entire “RxUClav” line, within the “Multi-PHY” system. Consequently, UNI#2 will also be driving the “RxUClav_in” input pin of the ATM Layer processor (see Figure 117).

If UNI #2 drives the “RxUClav” line “low”, upon the application of its address on the UTOPIA Address Bus, then the ATM Layer processor will “learn” that there are no ATM cells of data (within the RxFIFO of UNI # 2) that need to be read out via the “Receive UTOPIA Interface” block. However, if UNI #2 drives the RxUClav line “high” (during “polling”), then the ATM Layer processor will know that UNI # 2 does contain at least one ATM cell of data (within its RxFIFO) that needs to be read out via the Receive UTOPIA Interface block.

Figure 118presents a timing diagram, that depicts the behavior of the ATM Layer processor’s and the UNI’s signals during polling.

Figure 118 Timing Diagram illustrating the Behavior of various signals from the ATM Layer processor and the UNI, during Polling



Notes regarding Figure 118

The Receive UTOPIA Data bus is configured to be 16 bits wide. Hence, the data, which the ATM Layer processor places on the Receive UTOPIA Data bus, is expressed in terms of 16 bit words: (e.g., W0 - W26).

The Receive UTOPIA Interface block is configured to handle 54 bytes/cell. Hence, Figure 92 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.

The ATM Layer processor is currently reading ATM cell data from the Receive UTOPIA Interface block, within UNI #1 (RxAddr[4:0] = 0x01) during this “polling process”.

The RxFIFO, within UNI#2’s Receive UTOPIA Interface block (RxAddr[4:0] = 0x03) is either depleted or does not contain enough data to constitute a complete ATM cell. Hence, the RxUClav line will be driven “low” whenever this particular Receive UTOPIA Interface block is “polled”.

The Receive UTOPIA Address of 0x1F is not associated with any UNI device, within this “Multi-PHY” system. Hence, the RxUClav line is tri-stated whenever this address is “polled”.

Although Figure 118 depicts connections between the Transmit UTOPIA Interface block pins and the ATM Layer processor; the Transmit UTOPIA Interface operation, in the Multi-PHY mode, will not be discussed in this section. Please see Section _ for a discussion on the Transmit UTOPIA Interface block during Multi-PHY operation.

2.3.5.1.2.9 ATM Layer Processor "polling" with the XRT94L33

In Figure 117, a simple illustration of the "Conceptual Multi-PHY" system consisting of two single-channel UNI devices was presented. In reality, a given Multi-PHY system can or will consist of numerous "multi-channel" UNI devices. The XRT94L33 is an example of this, being a "4-channel" UNI device. Therefore, the XRT94L33 can be thought of as consisting of four ATM UNIs within a single package (one for each STS-3c port within the device).

It is important to note that although the XRT94L33 consists of a single "Receive UTOPIA Interface" block, the XRT94L33 can be assigned four unique (4) Transmit UTOPIA Addresses as presented in Section _.

2.3.5.1.2.10 Reading ATM Cell Data from a Different UNI

After the ATM Layer processor has "polled" each of the UNI devices, within its system, it must now select a UNI, and begin reading ATM cell data from that device. The ATM Layer processor makes its selection and begins the reading process by:

1. Applying the UTOPIA Address of the "target" UNI on the "UTOPIA Address Bus".

2. Negate the RxUEnB* signal. This step causes the "addressed" UNI to recognize that it has been selected to transmit the next set of ATM cell data to the ATM Layer processor.

3. Assert the RxUEnB* signal.

4. Check and insure that the RxUSoC output pin (of the selected UNI) pulses "high" when the first byte or word of ATM cell data has been placed on the Receive UTOPIA Data Bus.

5. Begin reading the ATM Cell data in a byte-wide (or word-wide) manner from the Receive UTOPIA Data bus.

Figure 119 presents a flow-chart that depicts the "UNI Device Selection and Read" process in Multi-PHY operation.

Figure 119 Flow-Chart of the "UNI Device Selection and Read Procedure" for the Multi-PHY Operation

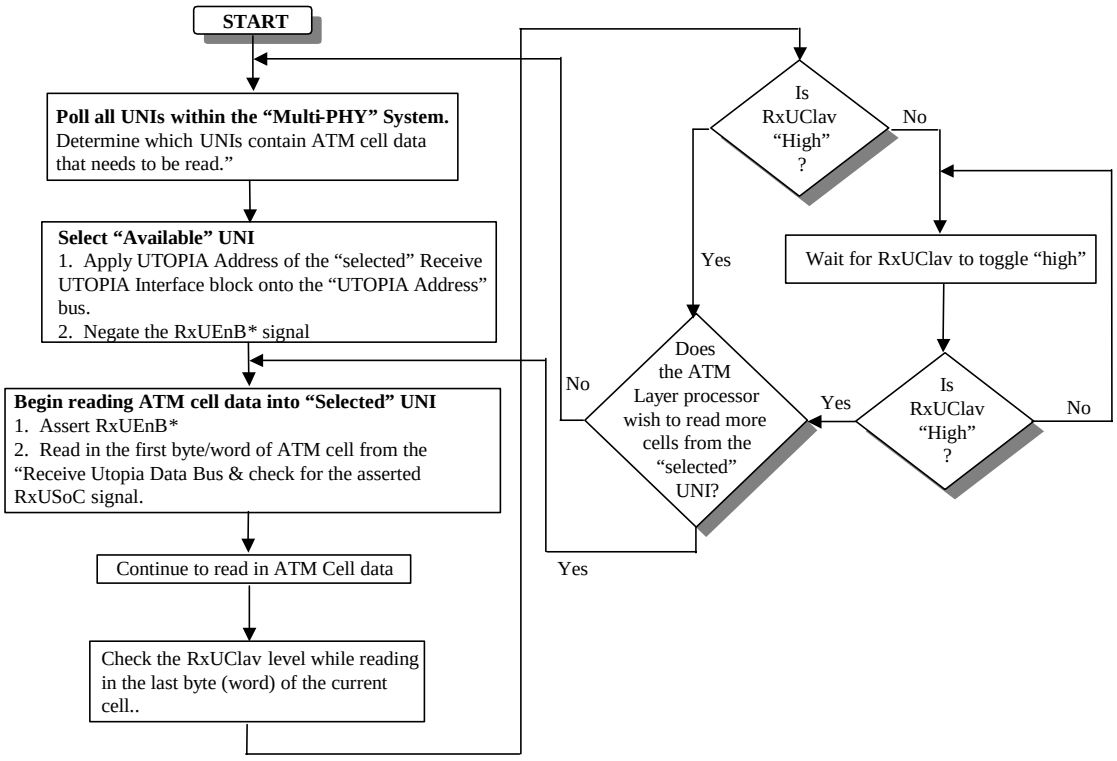
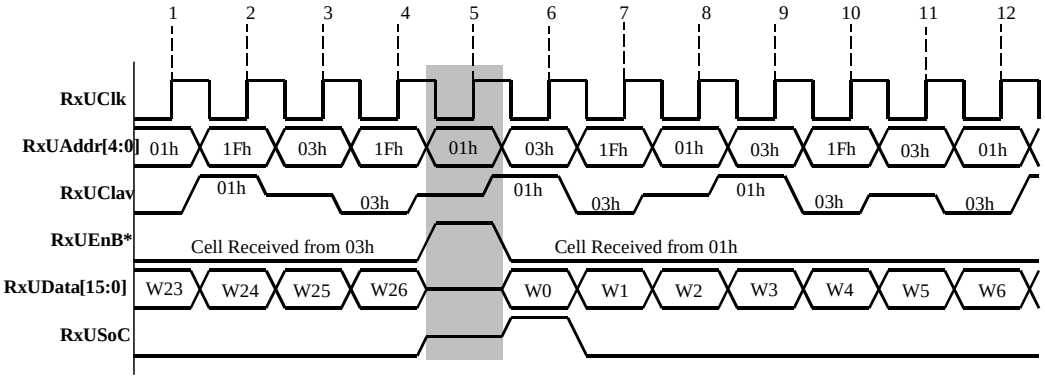


Figure 120 presents a timing diagram that illustrates the behavior of various "Receive UTOPIA Interface block" signals, during the "Multi-PHY" UNI Device Selection and Read operation.

Figure 120 Timing Diagram of the Receive UTOPIA Data and Address Bus signals, during the "Multi-PHY" UNI Device Selection and Write Operations



Notes regarding Figure 120**:**

The Receive UTOPIA Data bus is configured to be 16 bits wide. Hence, the data, which the Receive UTOPIA Interface block places on the Receive UTOPIA Data bus, is expressed in terms of 16-bit words (e.g., W0 - W26).

The Receive UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, Figure 94 illustrates the ATM Layer processor reading 27 words (e.g., W0 through W26) for each ATM cell.

In Figure 120, the ATM Layer processor is initially reading ATM cell data from the Receive UTOPIA Interface within UNI #2 (RxUAddr[4:0] = 0x03). However, the ATM Layer processor is also polling the Receive UTOPIA Interface block within UNI #1 (RxUAddr[4:0] = 0x01) and some "non-existent" device at RxUAddr[4:0] = 0x1F. The ATM Layer processor completes its reading of the cell from UNI #1 at clock edge

#4. Afterwards, the ATM Layer will cease to read any more cell data from UNI #1, and will begin to read some cell data from UNI #2 (RxUAddr[4:0] = 0x03). The ATM Layer processor will indicates its intention to select a new UNI device for reading by negating the RxEnB* signal, at clock edge #5 (see the shaded portion of Figure 120). At this time, UNI #1 will notice two things:

1. The UTOPIA Address for the Receive UTOPIA Interface block, within UNI #1 is on the Receive UTOPIA Address bus (RxUAddr[4:0] = 0x01).

2. The RxUEnB* signal has been negated.

UNI #1 will interpret this signaling as an indication that the ATM Layer processor is going to be performing read operations from it. Afterwards, the ATM Layer processor will begin to read ATM cell data from the Receive UTOPIA Interface block, within UNI #1.

3.0 PPP APPLICATIONS

3.1 TRANSMIT DIRECTION

3.1.1 TRANSMIT POS-PHY INTERFACE BLOCK

3.1.2 TRANSMIT PPP PACKET PROCESSOR BLOCK

3.2 RECEIVE DIRECTION

3.2.1 RECEIVE PPP PACKET PROCESSOR BLOCK

3.2.2 RECEIVE POS-PHY INTERFACE BLOCK

4.0 THE CLOCK SYNTHESIZER BLOCK

5.0 USING THE STS-3 TELECOM BUS INTERFACE

6.0 USING THE STS-3 PECL INTERFACE

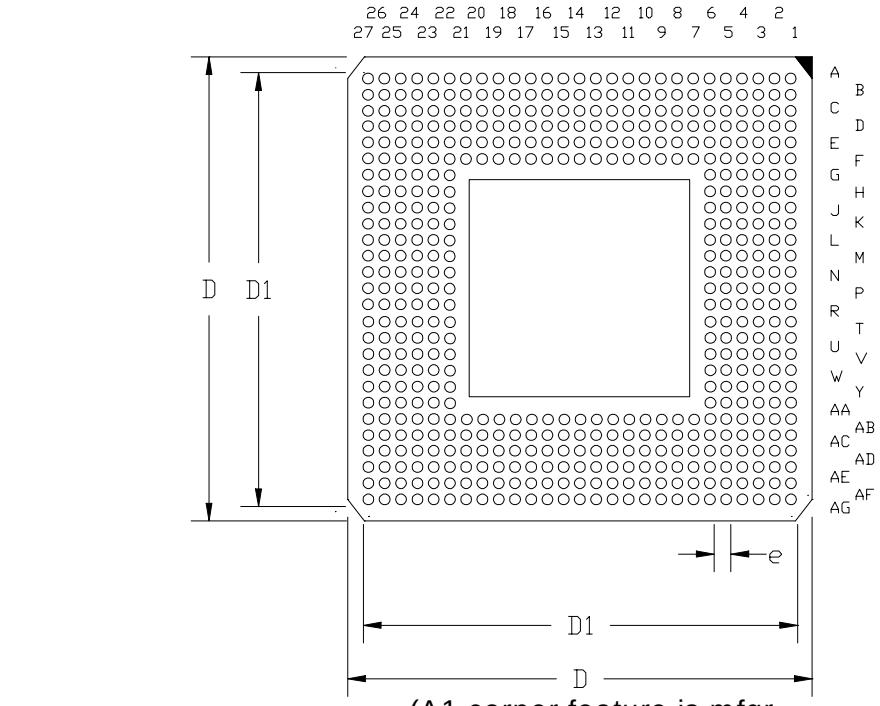
6.1 INTERFACING THE XRT94L33 TO AN OPTICAL TRANSCEIVER

6.2 INTERFACING THE XRT94L33 TO A HIGH-SPEED BACK-PLANE INTERFACE

7.0 AUTOMATIC PROTECTION SWITCHING WITHIN THE XRT94L33 DEVICE

Package Outline Drawing

504 Tape Ball Grid Array
(35 mm x 35 mm - TBGA)
Bottom View



Note: The control dimension is in millimeter.

	INCHES		MILLIMETERS	
SYMBOL	**MIN**	**MAX**	**MIN**	**MAX**
A	0.051	0.067	1.30	1.70
A1	0.020	0.028	0.50	0.70
A2	0.031	0.039	0.80	1.00
D	1.370	1.386	34.80	35.20
D1	1.300 BSC		33.02 BSC	
b	0.024	0.035	0.60	0.90
e	0.050 BSC		1.27 BSC	
P	0.006	0.012	0.15	0.30

NOTES:

NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Datasheet November 2006

Стандарт Электрон Связь

Мы молодая и активно развивающаяся компания в области поставок электронных компонентов. Мы поставляем электронные компоненты отечественного и импортного производства напрямую от производителей и с крупнейших складов мира.

Благодаря сотрудничеству с мировыми поставщиками мы осуществляем комплексные и плановые поставки широчайшего спектра электронных компонентов.

Собственная эффективная логистика и склад в обеспечивает надежную поставку продукции в точно указанные сроки по всей России.

Мы осуществляем техническую поддержку нашим клиентам и предпродажную проверку качества продукции. На все поставляемые продукты мы предоставляем гарантию .

Осуществляем поставки продукции под контролем ВП МО РФ на предприятия военно-промышленного комплекса России , а также работаем в рамках 275 ФЗ с открытием отдельных счетов в уполномоченном банке. Система менеджмента качества компании соответствует требованиям ГОСТ ISO 9001.

Минимальные сроки поставки, гибкие цены, неограниченный ассортимент и индивидуальный подход к клиентам являются основой для выстраивания долгосрочного и эффективного сотрудничества с предприятиями радиоэлектронной промышленности, предприятиями ВПК и научно-исследовательскими институтами России.

С нами вы становитесь еще успешнее!

Наши контакты:

Телефон: +7 812 627 14 35

Электронная почта: sales@st-electron.ru

Адрес: 198099, Санкт-Петербург, Промышленная ул, дом № 19, литера Н, помещение 100-Н Офис 331