1.0 SYSTEM DESCRIPTION

The XR- T7234 E3 UNI IC for ATM consists of the following functional sections/ blocks.

- Transmit Section
  - Transmit Utopia Interface Block
  - Transmit Cell Processor Block
  - Transmit E3 Framer Block
- Receive Section
  - Receive Utopia Interface Block
  - Receive Cell Processor Block
  - Receive E3 Framer Block
- Mcroprocessor Interface Section
- Performance Monitor Section
- Test and Diagnostic Section
- Line Interface Drive and Scan Section

Each of these functional sections (and the blocks, within these sections) combine to make a single chip device that is capable of transmitting and receiving ATM cell data via a E3 Transport Medium.

1.1 SYSTEM LEVEL INTERFACING OF THE XR- T7234 E3 UNI

1.2

- The system designer, when using the XR- T7234 E3 UNI IC for ATM, must (at a minimum) interface this chip to the following entities.
- The ATM Switch (or ATM Layer Processor)
- A local (housekeeping) microprocessor
- The E3 line

Figures 1 and 2 presents two illustrations of the UNI being interfaced to these three entities. A brief discussion on how to interface the UNI to these entities follows.

Interfacing to the ATM Switch (ATM Layer Processor)

Whenever an ATM switch needs to transmit and receive ATM cells to and from the UNI, it will typically use some sort of "ATM Layer" processing entity to accomplish this processing of cell data. This "ATM Switch Processing" entity will be referred as the "ATM Layer Processor" throughout this data sheet. The ATM Layer processor interfaces with the XR- T7234 E3 UNI via the "Utopia Bus" and will write in ATM cell data (in an 8- bit or 16- bit wide parallel format) into the Transmit Utopia Interface block (of the UNI). Additionally, the ATM Layer processor will also receive ATM cells (in this same 8- bit or 16- bit wide parallel format) from the Receive Utopia Interface block (within the UNI IC).

Interfacing to the Local Mcroprocessor

In contrast to the ATM Layer Processor, the "local" microprocessor (µP) interfaces with the UNI via the Microprocessor Interface. This local "housekeeping" microprocessor will typically read and write "configuration information" from or into the on- chip registers within the UNI IC. Further, the local microprocessor will respond to UNI- generated interrupts, read and write PMDL (Path Maintenance Data Link) Messages and OAM cell data into and from the UNI IC. Finally, the local microprocessor will "monitor" the performance of the overall system by periodically reading the contents of the "Performance Monitor" registers.

Note: The local µ should not be confused with the ATM Layer processor. The terms "local µP" and "ATM Layer Processor" will be used throughout this data sheet in order to make the distinction between these two "entities".

Interfacing the UNI to the E3 Line

The UNI can be interfaced to a E3 line that is operating over a copper or optical medium. If the user intends to interface the UNI to a copper E3 line, (e.g., over coaxial cable), then he/ she must connect the dual rail inputs (RxPOS and RxNEG) and the dual rail outputs (TxPOS and TxNEG) to a E3 Line Interface Unit (LIU) IC, (which is transformer- coupled to the E3 line) in order to reliably transmit and receive this data over the copper medium. An example of such an LIU are the XR- T7295E (E3 Line

Receiver IC) and the XR- T7296 (E3 Line Transmitter IC). Figure 1 presents an illustration of the "System- Level" interfacing of the XR- T7234 E3 UNI, when the E3 line signal is transmitted over a copper medium.

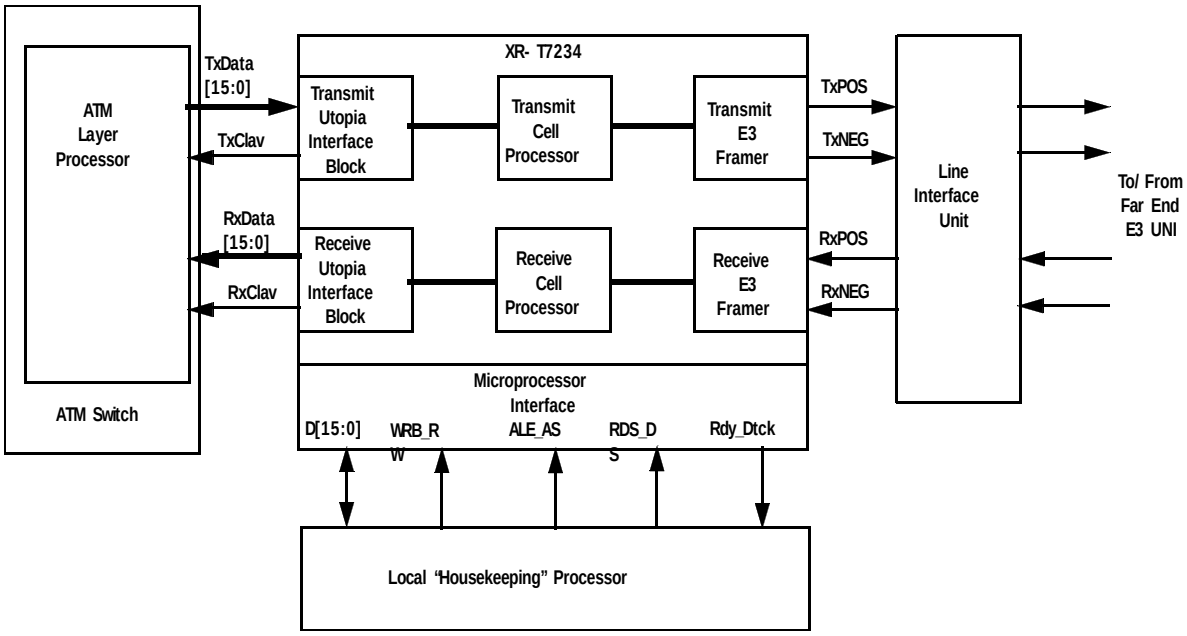

Figure 1. System Level Interfacing of the XR- T7234 E3 UNI (E3 Data is transmitted over Copper Medium).

Additionally, the user would connect the single- rail output pin of the UNI (TxPOS) to the "Electrical" input of an "Electrical to Optical" converter; and connect the single- rail input pin of the UNI (RxPOS) to the "Electrical" output of an "Optical to Electrical" converter. The "Electrical to Optical" and "Optical to Electrical" converters are "entities" that handle the translation between the electronic and photonic modes. Figure 2 presents an illustration of the "System Level" interfacing of the XR- T7234 E3 UNI, when the E3 line signal is transmitted over an optical medium.

The remainder of this text will frequently refer to each of these "entities" as:

- The ATM Layer Processor
- The Local Microprocessor
- The Line Interface Unit (LIU) IC

1.2 Internal Operation of the XR- T7234 E3 UNI device

Whenever an ATM switch, that has access to an E3 line, needs to transmit ATM cell data to a "Far- End" Terminal over the E3 line, it will write the ATM cell data into the Transmit Utopia Interface block of the XR- T7234 E3 UNI device. Afterwards, the Transmit Utopia Interface block will ultimately write this cell data to an internal FIFO (referred to as Tx FIFO throughout this document); where it can be read and further processed by the Transmit Cell Processor. The Transmit Utopia Interface block will also perform some parity checking on the data that it receives from the ATM Layer processor. Finally, the Transmit Utopia Interface block will provide signaling to support data- flow control

between the ATM Layer Processor and the UNI IC.

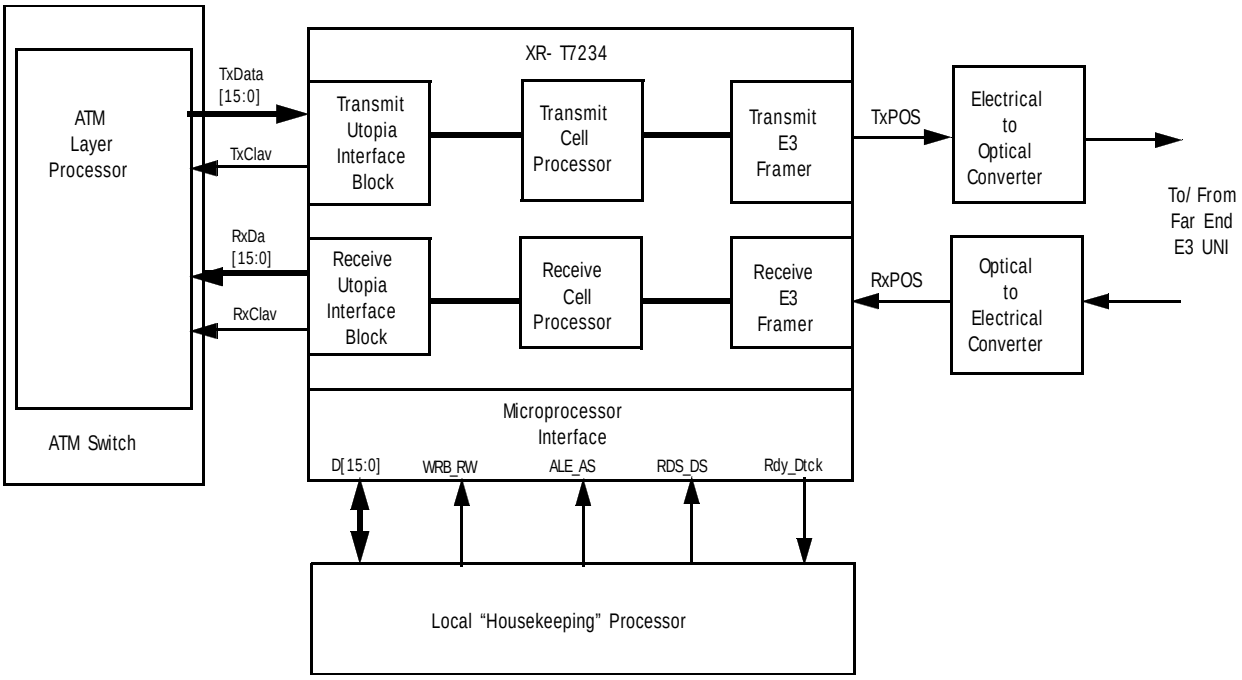

Figure 2. System Level Interfacing of the XR- T7234 E3 UNI (E3 data is transmitted over Optical Fiber).

The Transmit Cell Processor block will read in the ATM cell from the Tx FIFO. It will then (optionally) proceed to take the first four octets of this cell and compute the HEC byte from these bytes. Afterwards the Transmit Cell Processor will insert this HEC byte into the 5th octet position within the cell. The Transmit Cell Processor will also (optionally) scramble the payload portion of the cell (bytes 6 through 53) in order to prevent the user data from mimicking framing or control bits/ bytes. Once the cell has gone through this process it will then be transferred to the Transmit E3 Framer.

If the Tx FIFO (within the Transmit Utopia Interface block) is depleted and has no (user) cells available, then the Transmit Cell Processor will automatically generate Idle cells. These Idle cells will be processed in the exact same manner as are the user cells, prior to transmission to the Transmit E3 Framer block. The Transmit Cell Processor generates these Idle Cells for "Cell- Rate" decoupling purposes. The Transmit Cell Processor also has provisions to allow the user to generate an OAM cell via software control. Note: the OAM cells will be subjected to the same processing (e.g., HEC Byte Calculation/ Insertion and Cell Payload Scrambling) as are user and Idle cells.

The Transmit E3 Framer block will take these ATM cells (from the Transmit Cell Processor), and insert this data into the payload portions of each outbound E3 frame. The Transmit E3 Framer will also generate overhead (OH) bytes that support framing, performance monitoring (the BIP byte), path maintenance data link as well as alarm and status information originating from the "Near- End" Receiver section of the UNI. The purpose of these alarm and status information bits is to alert the "Far- End" Terminal Equipment that the "Near- End" UNI Receiver has detected some problems in receiving data from it. The Transmit E3 Framer will output this E3 data stream to an off- chip LIU (Line Interface Unit) chip via the TxPOS, TxNEG, and TxLineClk output pins. The LIU chip will take on the responsibility of driving the E3 data out on the E3 Transport Medium to the "Far- End" Terminal.

Likewise, whenever ATM cell data arrives to the UNI, over the E3 line, the Receive E3 Framer block will synchronize itself to this incoming E3 Data Stream (containing ATM cells) via the RxPOS, RxNEG, and RxLineClk input pins, and proceed to "strip off" and process the OH bytes of the E3 frame. Once all of the OH bytes have been removed, the payload portion of the received E3 Frame should consist of ATM cells.

The Receive Cell Processor takes this unframed data- stream of ATM cells from the Receive E3 Framer and performs the following operations:

- Cell Delineation
- HEC Byte Verification

The Receive Cell Processor takes the first four octets of the cell (the header) and computes a HEC byte. The Receive Cell Processor will then compare this computed HEC value with that of the fifth octet, within the cell. If the two HEC values are equal, then the cell is then retained for further processing. If the two HEC values are not equal, then the cells with single- bit errors are typically corrected. However, the cell is optionally discarded if multiple- bit errors are detected.

Cell Filtering

The Receive Cell Processor will optionally detect and remove Idle Cells. And can be configured to filter User and OAM cells based upon their header byte patterns.

Cell De- Scrambling

The Receive Cell Processor will de- scramble the payload portion of the cell (the 6th through the 53rd octet), and pack these octets in with the cell header bytes, and the HEC byte for transmission to the Receive Utopia Interface block.

Once the ATM cells have gone through the cell delineation, HEC Byte Verification, cell payload de- scrambling, and cell filtering processes, then they will be written into the Rx FIFO, within the Receive Utopia Interface Block.

The Receive Utopia Interface block (like its Transmit counterpart) provides the industry standard ATM/ PHY interface functions. The Receive Utopia Interface Block will inform the ATM Layer Processor when it is holding ATM cell data within the RxFIFO that needs to be read. The ATM Layer Processor can then read out this cell data, from the Receive Utopia Interface block, and route it to the remainder of the ATM switch for further processing.

DC ELECTRICAL CHARACTERISTICS

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
ICC	Power Supply Current				mA	
ILL	Data Bus Tri- State Bus Leakage Current				mA	
VIL	Input Low Voltage				V	
VIH	Input High Voltage			VCC	V	

DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
VOL	Output Low Voltage				V	
VOH	Output High Voltage			VCC	V	
IOC	Open Drain Output Leakage Current				mA	

AC ELECTRICAL CHARACTERISTICS

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Transmit Utopia Interface Block (See Figure 91)						
t1	TxData[15:0] to rising edge of TxClk Setup Time				ns	
t2	TxData[15:0] Hold Time from rising edge of TxClk				ns	
t3	TxUtopia Write Enable Setup Time to rising edge of TxClk				ns	
t4	TxUtopia Write Enable Hold Time from rising edge of TxClk				ns	
t5	TxPrty Setup Time to rising edge of TxClk				ns	
t6	TxPrty Hold Time from rising edge of TxClk				ns	
t7	TxSoC Setup Time to rising edge of TxClk				ns	
t8	TxSoC Hold Time from rising edge of TxClk				ns	
t9	TxAddr[4:0] Setup Time to rising edge of TxClk				ns	
t10	TxAddr[4:0] Hold Time from rising edge of TxClk				ns	
t11	TxClav signal valid (not Hi- Z) from first TxClk rising edge of valid and correct TxAddr[4:0]				ns	
t12	TxClav signal Hi- Z from first TxClk rising edge of different TxAddr[4:0]				ns	
Transmit Cell Processor (GFC Serial Input Port) -See Figure 92						
t13	Clock Period of TxGFCClk				ns	
fGFCClk	Frequency of TxGFCClk				Hz	
t14	Delay from rising edge of TxGFCClk to rising edge of TxGFCMSB pin				ns	

AC ELECTRICAL CHARACTERISTICS (CONTINUED)

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
t15	Pulsewidth of TxGFCMSB signal				ns	
t16	TxGFC Data Setup time to rising edge of TxGFCClk				ns	
t17	TxGFC Data Hold time from rising edge of TxGFCClk				ns	
Transmit E3 Framer (Serial Input Port) -See Figure 93						
fTxCHClk	Frequency of TxCHClk signal				Hz	
t24	Period of TxCHClk clock signal				ns	
t25	Delay from rising edge of TxCHFrame signal to rising edge of TxCHClk signal				ns	
t26	TxCH Data Setup time to rising edge of TxCHClk signal				ns	
t27	TxCH Data Hold time from rising edge of TxCHClk signal				ns	
t28	TxCHIns signal setup time to rising edge of TxCHClk				ns	
t29	TxCHIns signal hold time from rising edge of TxCHClk				ns	
Transmit E3 Framer (LIU Interface Port) -See Figures 94 and 95						
t30	Delay time of data on TxPOS or TxNEG, following the rising edge of the TxLineClk				ns	Transmit E3 Framer is configured to update TxPOS and TxNEG on the rising edge of TxLineClk.
t31	Delay time of data on TxPOS or TxNEG following the falling edge of the TxLineClk				ns	Transmit E3 Framer is configured to update TxPOS and TxNEG on the falling edge of TxLineClk.
fTxLineClk	Clock frequency of TxLineClk				Hz	
t32	Period of TxLineClk clock signal				ns	
t33	Bit Period of data on TxPOS or TxNEG pins				ns	
Receive E3 Framer (Serial Output Port) -See Figure 96						
fRxCHClk	Frequency of RxCHClk signal				Hz	
t34	Period of RxCHClk clock signal				ns	
t35	Delay Time from rising edge of RxCHClk to RxCHFrame signal				ns	

AC ELECTRICAL CHARACTERISTICS (CONTINUED)

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
t36	Delay Time from rising edge of RxCHClk to valid data at RxCH				ns	
t37	Bit Period of data at RxCH				ns	
Symbol	Parameter	Min.	Typ	Max.	Units	Conditions
Receive E3 Framer (LIU Interface Port) -See Figures 97 and 98						
t38	RxPOS/ RxNEG data Setup Time to rising edge of RxLineClk				ns	Receive E3 Framer is configured to sample RxPOS and RxNEG on the rising edge of RxLineClk.
t39	RxPOS/ RxNEG data Hold Time from rising edge of RxLineClk				ns	Receive E3 Framer is configured to sample RxPOS and RxNEG on the rising edge of RxLineClk.
t40	RxPOS/ RxNEG data Setup Time to falling edge of RxLineClk				ns	Receive E3 Framer is configured to sample RxPOS and RxNEG on the falling edge of RxLineClk.
t41	RxPOS/ RxNEG data Hold Time from falling edge of RxLineClk				ns	Receive E3 Framer is configured to sample RxPOS and RxNEG on the falling edge of RxLineClk.
fRxLineClk	Clock frequency of RxLineClk				Hz	
t42	Period of RxLineClk clock signal				ns	
Receive Cell Processor (GFC Serial Output Port) -See Figure 99						
t47	Clock Period of RxGFCClk				ns	
t48	Delay from rising edge of RxGFCClk to rising edge of RxGFCMSB pin.				ns	
t49	Pulsewidth of RxGFCMSB signal				ns	
t50	Delay from rising edge of RxGFCMSB signal to first valid bit at RxGFC.				ns	
t51	Delay from rising edge of RxGFCClk to valid bit at RxGFC.				ns	
t52	Pulsewidth of Bit at RxGFC output.				ns	
Receive Utopia Interface Block (See Figure 100)						
t53	Delay time from rising edge of RxClk to Data Valid at RxData[15:0]				ns	

AC ELECTRICAL CHARACTERISTICS (CONTINUED)

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
t54	Rx Utopia Read Enable setup time to rising edge of RxClk				ns	
t55	Delay time from rising edge of RxClk to valid RxPrty bit				ns	
t56	Delay time from rising edge of RxClk to valid RxSoC bit				ns	
t57	Delay time from Read Enable false to Data Bus being tri- stated				ns	
t58	Delay time from Read Enable false to RxPrty bit being tri- stated				ns	
t59	Delay time from Read Enable false to RxSoC bit being tri- stated				ns	
t60	RxAddr[4:0] Setup Time to rising edge of RxClk				ns	
t61	RxAddr[4:0] Hold Time from rising edge of RxClk				ns	
t62	RxClav signal valid (not Hi- Z) from first RxClk rising edge of valid and correct TxAddr[4:0]				ns	
t63	RxClav signal Hi- Z from first RxClk rising edge of different RxAddr[4:0].				ns	
Microprocessor Interface-Intel (See Figure 101)						
t64	A8 - A0 Setup Time to ALE_AS Low				ns	
t65	A8 - A0 Hold Time from ALE_AS Low				ns	
t66	CS* Setup Time to ALE_AS Low				ns	
t67	CS* Hold Time from RDS_DS*, WRB_RW* High					
t68	ALE_AS Hold Time to RDS_DS* Low.				ns	
t69	RDS_DS*, WRB_RW* Pulse Width				ns	
t70	Data Valid from RDS_DS* Low.				ns	
t71	Data Bus Floating from RDS_DS* High				ns	
t72	Data Setup Time to WRB_RW* High				ns	
t73	Data Hold Time from WRB_RW* High				ns	
t74	High Time between Reads and/ or Writes				ns	
Microprocessor Interface-Motorola Read Operations (See Figure 102)						
t75	A8 - A0 Setup Time to ALE_AS High				ns	

AC ELECTRICAL CHARACTERISTICS (CONTINUED)

Test Conditions: TA = 25°C, VCC = 5.0V ± 5% unless otherwise specified						
Symbol	**Parameter**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
t76	A8 - A0 Hold Time from ALE_AS Low				ns	
t77	CS* Setup Time to ALE_AS Low				ns	
t78	WRB_RW Setup Time (for Read) to RDB_DS (Data Strobe)				ns	
t79	ALE_AS Setup Time to RDB_DS (Data Strobe) Low				ns	
t80	Data Valid from RDB_DS Low				ns	
t81	DTACK Low from RDB_DS Low				ns	
t82	CS* High Pulse Width				ns	
t83	Data Bus Floating from CS* High				ns	
t84	Data Bus Floating from RDB_DS High				ns	
t85	DTACK High from RDB_DS High				ns	
Microprocessor Interface - Motorola Write Operations (See Figure 103)						
t86	WRB_RW Setup Time (for Write) to RDB_DS (Data Strobe)				ns	
t87	Data Setup Time to falling edge of RDB_DS (Data Strobe) for Write				ns	
t88	Data Hold Time from falling edge of RDB_DS (Data Strobe) for Write				ns	

PIN DESCRIPTION

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
3	1	D15	I/O	**MSB of Bi-Directional Data Bus (Microprocessor Interface Section):** This pin, along with pins D0-D14, function as the Microprocessor Interface bi-directional data bus, and is intended to be interfaced to the "local" microprocessor. This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus.
	2	TAOS	O	**"Transmit All Ones Signal" (TAOS) Command (for the XR-T7296 E3 Line Transmitter IC).** This output pin is intended to be connected to the TAOS input pin of the XR-T7296 E3 Line Transmitter IC. The user can control the state of this output pin by writing a '0' or '1' to Bit 4 (TAOS) of the Line Interface Drive Register (Address = 84h). If the user commands this signal to toggle "high" then itwill force the XR-T7296 E3 Line Transmitter IC to transmit an "All Ones" pattern onto the line. Conversely, if the user commands this output signal to toggle "low" then the XR-T7296 E3 Line Transmitter IC will proceed to transmit data based upon the pattern that it receives via the TxPOS and TxNEG output pins. Writing a "1" to Bit 4 of the Line Interface Drive Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit-field will cause this output pin to toggle "low". ***Note:*** *If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this output pin for a variety of other purposes.*
4	3	D14	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
5	4	D13	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
6	5	D12	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
	6	DMO	I	**"Drive Monitor Output" Input (from the XR-T7296 E3 Line Transmitter IC).** This input pin is intended to be tied to the DMO output pin of the XR-T7296 E3 Line Transmitter IC. The user can determine the state of this input pin by reading Bit 2 (DMO) within the Line Interface Scan Register (Address = 85h). If this input signal is "high", then it means that the drive monitor circuitry (within the XR-T7296 E3 Line Transmitter IC) has not detected any bipolar signals at the MTIP and MRING inputs within the last 128 ± 32 bit-periods. If this input signal is "low", then it means that bipolar signals are being detected at the MTIP and MRING input pins of the XR-T7296 device. ***Note:*** *If this customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this input pin for a variety of other purposes.*

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
7	7	MOTO	I	**Motorola/ Intel Processor Interface Select Mode:** This input pin allows the user to configure the Microprocessor Interface to interface with either a "Motorola- type" or "Intel- type" microprocessor/ microcontroller. Tying this input pin to VCC, configures the microprocessor interface to operate in the Motorola mode (e.g., the UNI device can be readily interfaced to a "Motorola type" local microprocessor). Tying this input pin to GND configures the Microprocessor Interface to operate in the Intel Mode (e.g., the UNI device can be readily interfaced to a Intel type" local microprocessor).
	8	RLOL	I	**Receive Loss of Lock Indicator-from the XR- T7295E E3 Line Receiver IC.** This input pin is intended to be connected to the RLOL (Receive Loss of Lock) output pin of the XR- T7295E E3 Line Receiver IC. The user can monitor the state of this pin by reading the state of Bit 1 (RLOL) within the Line Interface Scan Register (Address = 85h). If this input pin is "low", then it means that the phase- locked- loop circuitry, within the XR- T7295E device is properly locked onto the incoming E3 data-stream; and is properly recovering clock and data from this E3 data- stream. However, if this input pin is "high", then it means that the phase- locked-loop circuitry, within the XR- T7295E device has lost lock with the incoming E3 data- stream, and is not properly recovering clock and data. For more information on the operation of the XR- T7295E E3 Line Receiver IC, please consult the "XR- T7295E E3 Integrated Line Receiver" data sheet. **Note:** *If the customer is not using the XR- T7295E E3 Line Receiver, he/ she can use this input pin for other purposes.*
8	9	D11	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
	10	TxFrame	O	**Transmit End of E3 Frame Indicator:** This output pin indicates that the last bit of an outbound E3 frame, is being transmitted from the TxPOS and TxNEG output pins. This pin marks the end of E3 frame by pulsing "high" for one bit period at the end of each frame.
9	11	D10	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	12	RECB	O	**Receive Equalization Bypass Control output pin-(to be connected to the XR- T7295E E3 Line Receiver IC).** This output pin is intended to be connected to the RECB input pin of the XR- T7295E E3 Line Receiver IC. The user can control the state of this output pin by writing a '0' or '1' to Bit 5 (RECB) within the Line Interface Driver Register (Address = 84h). If the user commands this signal to toggle "high" then it will cause the incoming E3 line signal to "by- pass" equalization circuitry, within the XR- T7295E Device. Conversely, if the user commands this output signal to toggle "low", then the incoming E3 line signal with be routed through the equalization circuitry. For information on the criteria that should be used when deciding whether to bypass the equalization circuitry or not, please consult the "XR- T7295E E3 Integrated Line Receiver" data sheet. Writing a "1" to Bit 5 of the Line Interface Drive Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit- field will cause this output pin to toggle "low". ***Note:*** *If the customer is not using the XR- T7295E E3 Line Receiver IC, then he/ she can use this output pin for a variety of other purposes.*
10	13	D9	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
11	14	D8	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** This pin is inactive if the Microprocessor Interface block is configured to operate over an 8 bit data bus. (Please see description for D15)
12	15	VDD	***	Power Supply Pin
13	16	D7	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
14	17	D6	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
15	18	D5	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
16	19	D4	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
17	20	Width16	I	**Microprocessor Interface Block Data Bus Width Selector:** This input pin allows the user configure the microprocessor interface, of the UNI, to operate over either an 8 or 16 bit wide data bus. Tying this pin to VCC configures the Microprocessor Interface Data Bus width to be 16 bits. Tying this pin to GND configures the Microprocessor Interface Data Bus width to be 8 bits.
18	21	D3	I/ O	**Bi- directional Data bus (Microprocessor Interface Section):** (Please see description for D15)

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	22	Encodis	O	**Encoder (HDB3) Disable Output pin (intended to be connected to the XR-T7296 E3 Line Transmitter IC).** This output pin is intended to be connected to the Encodis input pin of the XR-T7296 E3 Line Transmitter IC. The user can control the state of this output pin by writing a "0" or "1" to Bit 3 (Encodis) within the Line Interface Driver Register (Address = 84h). If the user commands this signal to toggle "high" then it will disable the HDB3 encoder circuitry within the XR-T7296 IC. Conversely, if the user commands this output signal to toggle "low", then the HDB3 Encoder circuitry, within the XR-T7296 IC will be enabled. Writing a "1" to Bit 3 of the Line Interface Driver Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit-field will cause this output pin to toggle "low". **Notes:** *(1) The user is advised to disable the HDB3 encoder (within the XR-T7296 IC) if the Transmit and Receive E3 Framers (within the UNI) are configured to operate in the HDB3 line code.* *(2) If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this output pin for a variety of other purposes.*
19	23	D2	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
	24	TxLev	O	**Transmit Level Select Output (to be connected to the XR-T7296 E3 Line Transmit IC).** This output pin is intended to be connected to the TxLev input pin of the XR-T7296 E3 Line Transmitter IC. The user can control the state of this output pin by writing a "0" or a "1" to Bit 2 (TxLev) within the Line Interface Driver Register (Address = 84h). If the user commands this signal to toggle "high" then it will cause the XR-T7296 E3 Line Transmitter to increase the amplitude of its output signal on the line, in order to drive the signal over cable lengths of greater than 225 ft. Therefore, the user is recommended to set this output "high", if he/she is driving E3 line signals over 225 ft (or more) of cable. Conversely, if the user is driving E3 line signals over less than 225 ft of cable, then he/she is recommended to set this output pin "low". Writing a "1" to Bit 2 of the Line Interface Drive Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit-field will cause this output pin to toggle "low". ***Note:*** *If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this output pin for a variety of other purposes.*
20	25	D1	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** (Please see description for D15)

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	26	RLOOP	O	**Remote Loopback Output Pin (to the XR-T7296 E3 Line Transmitter IC).** This output pin is intended to be connected to the RLOOP input pin of the XR-T7296 E3 Line Transmitter IC. The user can command this signal to toggle "high" and, in turn, force the XR-T7296 E3 Line Transmitter into the "Remote Loopback" mode. Conversely, the user can command this signal to toggle "low" and allow the XR-T7296 device to operate in the normal mode. (For a detailed description of the XR-T7296 E3 Line Transmitter's operation during Remote Loopback, please see the XR-T7296 E3/STS-1, E3 Integrated Line Transmitter's Data Sheet). Writing a "1" to bit 1 of the "Line Interface Drive Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit-field will cause the RLOOP output to toggle "low". ***Note:*** *If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this output pin for a variety of other purposes.*
21	27	D0	I/O	**Bi-directional Data bus (Microprocessor Interface Section):** (Please see description for D15)
	28	LLOOP	O	**Local Loopback Output Pin (to the XR-T7296 E3 Line Transmitter IC).** This output pin is intended to be connected to the LLOOP input pin of the XR-T7296 LIU IC. The user can command this signal to toggle "high" and, in turn, force the LIU into the "Local Loopback" mode. (For a detailed description of the XR-T7296 E3 Line Transmitter's operation during Local Loopback, please see the XR-T7296 E3/STS-1, E3 Integrated Line Transmitter's Data Sheet). Writing a "1" to bit 1 of the "Line Interface Drive Register (Address = 84h) will cause this output pin to toggle "high". Writing a "0" to this bit-field will cause the RLOOP output to toggle "low". ***Note:*** *If the user is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this output pin for a variety of other purposes.*
22	29	IntB*	O	**Interrupt Request Output:** This open-drain, active-low output signal will be asserted when the UNI device is requesting interrupt service from the local microprocessor. This output pin should typically be connected to the "Interrupt Request" input of the local microprocessor.
	30	RxLCD	O	**Loss of Cell Delineation Indicator:** This active-high output pin will be asserted whenever the Receive Cell Processor has experienced a "Loss of Cell Delineation". This pin will return "low" once the Receive Cell Processor has regained Cell Delineation.
23	31	GND	***	Ground Pin Signal

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
24	32	CSB*	I	**Chip Select Input:** This active- low input signal selects the Microprocessor Interface Section of the UNI device and enables Read/ Write operations between the "local" microprocessor and the UNI on- chip registers and RAM locations.
25	33	RDB_DS	I	**Read Data Strobe (Intel Mode):** If the microprocessor interface is operating in the Intel Mode, then this input will function as the RD* (READ STROBE) input signal from the local µP. Once this active- low signal is asserted, then the UNI will place the contents of the addressed registers (within the UNI) on the Microprocessor Data Bus (D[15:0]). When this signal is negated, the Data Bus will be tri- stated. **Data Strobe (Motorola Mode):** If the microprocessor interface is operating in the Motorola mode, then this pin will function as the active- low Data Strobe signal.
	34	RxGFC	O	**Receive GFC Nibble Field Serial Output pin:** This pin, along with the RxGFCClk and the RxGFCMSB pins form the "Receive GFC Nibble- Field" serial output port. This pin will serially output the contents of the GFC Nibble field of each valid cell that is processed through the Receive Cell Processor. This data is serially clocked out of this pin on the rising edge of the RxGFCClk signal. The Most Significant Bit (MSB) of each GFC value is designated by a pulse at the RxGFCMSB output pin. ***Notes:*** *1. The GFC Nibble Field Serial Output port is only available for the 160 pin packaged device.* *2. The "Receive GFC Nibble- Field" serial output port will only output the GFC Nibble- field values of "valid" cells; not Idle cells.*
26	35	WRB_RW	I	**Write Data Strobe (Intel Mode):** If the microprocessor interface is operating in the Intel Mode, then this active- low input pin functions as the WR* (Write Strobe) input signal from the µP. Once this active- low signal is asserted, then the UNI will latch the contents of the µP Data Bus, into the addressed register (or RAM location) within the UNI IC. **R/ W Input Pin (Motorola Mode):** When the Microprocessor Interface Section is operating in the "Motorola Mode", then this pin is functionally equivalent to the "R/ W" pin. In the Motorola Mode, a "READ" operation occurs if this pin is at a logic "1". Similarly, a WRITE operation occurs if this pin is at a logic "0".
	36	NC	****	Not Bonded Out
27	37	A8	I	**Address Bus Input (Microprocessor Interface-MSB (Most Significant Bit):** This input pin, along with inputs A0-A7 are used to select the on- chip UNI register and RAM space for READ/ WRITE operations with the "local" microprocessor.
	38	NC	****	Not Bonded Out

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
28	39	A7	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
	40	NC	****	Not Bonded Out
29	41	A6	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
30	42	A5	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
31	43	A4	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
32	44	A3	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
33	45	A2	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
34	46	A1	I	**Address Bus Input (Microprocessor Interface):** (Please see description for A8)
	47	RxGFCMSB	O	**Received GFC Nibble Field-MSB Indicator:** This output pin functions as a part of the "Receive GFC- Nibble Field" Serial Output port; which also consists of the RxGFC and RxGFCClk pins. This pin pulses "high" the instant that the MSB (Most Significant Bit) of a GFC Nibble is being output on the RxGFC pin. ***Notes:*** *1. The "Receive GFC Nibble Field" serial output port is only available for the 160 pin packaged devices.* *2. The "Receive GFC Nibble- field" serial output port will only output the GFC Nibble- field values for valid cells. Therefore, this output pin will only pulse "high" while the Receive Cell Processor is processing a valid cell.*
35	48	A0	I	**Address Bus Input (Microprocessor Interface)-LSB (Least Significant Bit):** (Please see description for A8)
	49	RxGFCIk	O	**Received GFC Nibble Serial Output Port Clock Signal:** This output pin functions as a part of the "Receive GFC Nibble- Field" Serial Output Port; also consisting of the RxGFC and RxGFCMSB pins. This pin provides a clock pulse which allows external circuitry to latch in the GFC Nibble- Data via the RxGFC output pin.
36	50	RxClk	I	**Receive Utopia Interface Clock Input:** The byte (or word) data, on the Receive Utopia Data bus is updated on the rising edge of this signal. The Receive Utopia Interface block can be clocked at 25 MHz, 33 MHz, or 50 MHz.
	51	RxCellRxed	O	**Receive Cell Processor-Cell Received Indicator:** This output pin pulses "high" each time the Receive Cell Processor receives a new cell from the Receive E3 Framer.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
37	52	GND	***	Ground Pin Signal
	53	RxData15	O	**Receive Utopia Data Bus Input (MSB):** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
38	54	RxData7	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	55	RxData14	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
39	56	RxData6	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	57	RxData13	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
40	58	RxData5	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	59	VDD	***	Power Supply Pin
41	60	RxData4	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	61	RxData12	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
42	62	RxData3	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	63	RxData11	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
43	64	RxData2	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	65	RxData10	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
44	66	VDD	***	Power Supply Pin
	67	RxData9	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
45	68	RxData1	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	69	RxData8	O	**Receive Utopia Data Bus Output:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor. ***Note:*** *This pin is only available for the 160 pin package version.*
46	70	RxData0	O	**Receive Utopia Data Bus Output-LSB:** This output pin, along with RxData14 through RxData0 functions as the Receive Utopia Data Bus. ATM cell data that has been received from the "Far- End" UNI is output on the Receive Utopia Data Bus, where it can be read and processed by the ATM Layer Processor.
	71	GND	***	Ground Signal Pin
47	72	RxSoC	O	**Receive Utopia Interface-Start of Cell Indicator:** This output pin allows the ATM Layer Processor to determine the boundaries of the ATM cells that are output via the Receive Utopia Data bus. The Receive Utopia Interface block will assert this signal when the first byte (or word) of a new cell is present on the Receive Utopia Data Bus; RxData[15:0].
	73	RxAddr4	I	**Receive Utopia Address Bus input (MSB):** This input pin, along with RxAddr3 through RxAddr0 functions as the Receive Utopia Address bus inputs. These input pins are only active when the UNI device is operating in the Multi- PHY Mode. The Receive Utopia Address Bus input is sampled on the rising edge of the RxClk signal. The contents of this address bus are compared with the value stored in the "Rx UT Address Register (Address = 7Eh). If these two values match, then the UNI will inform the ATM Layer Processor on whether or not it has any new ATM cells to be read from the RxFIFO, by driving the RxClav output to the appropriate level. If these two address values do not match, then the UNI will not respond to the ATM Layer Processor; and will keep its RxClav output signal tri- stated.
48	74	RxPrty	O	**Receive Utopia Interface-Parity Output pin:** The Receive Utopia Interface block will compute the odd- parity of each byte (or word) that it will place on the Receive Utopia Data Bus. This odd- parity value will be output on this pin, while the corresponding byte (or word) is present on the Receive Utopia Data Bus.
	75	RxAddr3	I	**Receive Utopia Address Bus input:** (See Description for RxAddr4)

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
49	76	RxClav	O	**Receive Utopia-Cell Available:** The Receive Utopia Interface block will assert this output pin in order to indicate that the Rx FIFO has some ATM cell data that needs to be read by the ATM Layer Processor. The exact functionality of this pin depends upon whether the UNI is operating in the "Octet Level" or "Cell Level" handshake mode. **Octet Level Handshaking Mode:** When the Receive Utopia Interface block is operating in the "octet- level handshaking" mode; this signal is asserted (toggles "high") when at least one byte of cell data exists within the RxFIFO (within the Receive Utopia Interface block). This output pin will toggle "low" when the RxFIFO is depleted of ATM cell data. **Cell Level Handshaking Mode:** When the Receive Utopia Interface block is operating in the "cell- level handshaking" mode; this signal is asserted if the RxFIFO contains at least one full cell of data. This signal toggle "low" if the RxFIFO is depleted of data, or if it contains less than one full cell of data. **Multi- PHY Operation:** When the UNI chip is operating in the Multi- PHY mode, this signal will be tri- stated until the RxClk cycle following the assertion of a valid address on the Receive Utopia Address bus input pins (e.g., if the contents on the Receive Utopia Address bus pins match that with the Receive Utopia Address Register). Afterwards, this output pin will behave in accordance with the "cell- level handshaking" mode.
50	-	GND	****	Ground Signal Pin
	77	RxAddr2	I	**Receive Utopia Address Bus input:** (See Description for RxAddr4)
	78	VDD	****	Power Supply Pin
51	79	RxAddr0	I	**Receive Utopia Address Bus input-LSB:** (See Description for RxAddr4)
52	80	RxAddr1	I	**Receive Utopia Address Bus input:** (See Description for RxAddr4)
53	81	RxEnB*	I	**Receive Utopia Interface-Output Enable:** This active- low input signal is used to control the drivers of the Receive Utopia Data Bus. When this signal is "high" (negated) then the Receive Utopia Data Bus is tri- stated. When this signal is asserted, then the contents of the byte or word that is at the "front of the RxFIFO" will be "popped" and placed on the Receive Utopia Data bus on the very next rising edge of RxClk.
54	82	GND	****	Ground Signal Pin
55	83	GND	****	Ground Signal Pin
56	84	TDI	NC	**Boundary Scan Pin:** Not Bonded out.
57	85	VDD	****	Power Supply Pin

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
58	86	RLOS	I	**Receive LOS (Loss of Signal) Indicator Input (from XR-T7295E E3 Line Receiver).** This input pin is intended to be connected to the RLOS (Receive Loss of Signal) output pin of the XR-T7295E E3 Line Receiver IC. The user can monitor the state of this pin by reading the state of Bit 0 (RLOS) within the Line Interface Scan Register (Address = 85h). If this input pin is "low", then it means that the XR-T7295E device is detecting a sufficient amount of signal energy on the line, due to the incoming E3 data-stream. However, if this input pin is "high", then it means that the XR-T7295E device is not detecting a sufficient amount of signal energy on the line, due to the incoming E3 data-stream, and may be experiencing a "Loss of Signal" condition. For more information on the operation of the XR-T7295E E3 Line Receiver IC, please consult the "XR-T7295E E3 Integrated Line Receiver" data sheet. ***Note:*** *Asserting the RLOS input pin will cause the XR-T7234 E3 UNI device to declare an "LOS (Loss of Signal) condition. Therefore, this input pin should not be used as a general purpose input.*
59	87	NC	****	Not Bonded Out
	88	RxLOS	O	**Receive E3 Framer-Loss of Signal Output Indicator:** This pin is asserted when the Receive E3 Framer encounters a string of 32 consecutive 0's via the RxPOS and RxNEG pins. This pin will be negated once the Receive E3 Framer has detected a string of 32 consecutive bits, that does not contain a string of 4 consecutive "0s".
60	89	RxOH	O	**Receive E3 Frame Overhead Bit Serial Output pin:** This output pin, along with RxOHClk and RxOHFrame, combine to form the "Receive E3 Framer OH Byte" Serial output port. The UNI Receive E3 Framer will extract the overhead bytes from the incoming E3 signal, and serially output these bytes on this output pin on the rising edge of the RxOHClk output signal.
	90	RxOOF	O	**Receiver E3 Framer-"Out of Frame" Indicator:** The UNI Receive E3 Framer will assert this output signal whenever it has declared an "Out of Frame" (OOF) condition with the incoming E3 frames. This signal is negated when the framer correctly locates the FA1 and FA2 frame alignment bytes and regains synchronization with the E3 frame.
61	91	GND	***	Ground Signal Pin
	92	RxAIS	O	**Receive "Alarm Indication Signal" Output pin:** The UNI will assert this pin to indicate that the Alarm Indication Signal (AIS) has been identified in the Receive E3 data stream. The Receive E3 Framer will declare an AIS condition, if it detects two consecutive E3 frames, each containing 7 or less "0s".

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
62	93	RxOHClk	O	**Receive E3 Framer Overhead Bytes Serial Output Port Clock.** This pin, along with the RxOH and RxOHFrame pins function as the "Receive E3 Framer Overhead byte" serial output port. This pin functions as a clock signal that can be used by external circuitry to latch and process the serial data from the RxOH pin.
	94	NC	****	Not Bonded Out
63	95	RxOHFrame	O	**Receive E3 Framer Overhead Byte Serial Output Port-Frame Boundary Indicator:** This pin, along with the RxOH and RxOHClk signals comprise the "Receive E3 Framer OH Bit" Serial Output Port. This pin pulses high when the first overhead bit of a E3 frame (e.g., the MSB- bit in the Frame Alignment octet: FA1) is output at the RxOH pin.
	96	RxFrame	O	**Receive Boundary of E3 Frame Output Indicator:** This output pin indicates the boundary of the incoming E3 frame as they appear, at the RxPOS and RxNEG inputs. This pin marks the end of a E3 frame by pulsing high for one bit period at the end of each frame.
64	97	RxPOS	I	**Receive Positive Data Input:** The exact role of this input pin depends upon whether the UNI is operating in the Unipolar or Bipolar Mode. **Unipolar Mode:** This input pin functions as the "Single- Rail" input for the "incoming" E3 data stream. The signal at this input pin will be sampled and latched (into the Receive E3 Framer) on the "user- selected" edge of the RxLineClk signal. **Bipolar Mode:** This input functions as one of the dual rail inputs for the incoming AMI/ HDB3 encoded E3 data that has been received from an external Line Interface Unit (LIU) IC. RxNEG functions as the other dual rail input for the UNI. When this input pin is asserted, it means that the LIU has received a "positive polarity" pulse from the line.
65	98	RxNEG	I	**Receive Negative Data Input:** The exact role of this input pin depends upon whether the UNI is operating in the Unipolar or Bipolar Mode. **Unipolar Mode:** This input pin is inactive, and should be pulled ("low" or "high") when the UNI is operating in the Unipolar Mode. **Bipolar Mode:** This input pin functions as one of the dual rail inputs for the incoming AMI/ HDB3 encoded E3 data that has been received from an external Line Interface Unit (LIU) IC. RxPOS functions as the other dual rail input for the UNI. When this input pin is asserted, it means that the LIU has received a "negative polarity" pulse from the line.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
66	99	RxLineClk	I	**Receiver LIU (Recovered) Clock:** This input signal serves three purposes: 1. The Receive E3 Framer uses it to sample and "latch" the signals at the RxPOS and RxNEG input pins (into the Receive E3 Framer circuitry). 2. This input signal functions as the timing reference for the Receive Framer block. 3. The Transmit E3 Framer block can be configured to use this input signal as its timing reference. ***Note:*** *This signal is the recovered clock from the external E3 LIU (Line Interface Unit) IC, which is derived from the incoming E3 data.*
67	100	NC	****	Not Bonded Out
68	101	RxRed	O	**Receiver Red Alarm Indicator-Receive E3 Framer:** The UNI asserts this output pin to denote that one of the following events has been detected by the Receive E3 Framer: • LOS-Loss of Signal Condition • OFF-Out of Frame Condition • AIS-Alarm Indication Signal Detection
	102	NC	****	Not Bonded Out
69	103	TxInClk	I	**Transmit E3 Framer-Clock Signal:** The Transmit E3 Framer can be configured to use this input signal as its timing reference. If this input pin is chosen to be the timing reference, then the user must supply a high quality 34.368 MHz signal to this input pin. In this configuration, frame generation, by the Transmit E3 Framer, will be asynchronous (with any other timing signals within the UNI). However, frame timing will be based upon this clock signal. ***Note:*** *This input pin should be tied to "GND" if it is not used as the Transmit E3 Framer timing reference.*
	104	NC	****	Not Bounded Out
70	105	GND	***	Ground Signal Pin
	106	NC	****	Not Bonded Out
71	107	TxFrameRef	I	**Transmit E3 Framer-Frame Reference Input Pin:** The Transmit E3 Framer can be configured to use this input signal as the "framing" reference for the Transmit E3 Framer block. If this input pin is chosen to be the framing reference, then any rising edge at this input will cause the Transmit E3 Framer to begin its creation a new E3 M-frame. Consequently, the user must supply a 8kHz clock signal to this input pin. ***Note:*** *This input pin should be tied to "GND" if it is not used as the Transmit E3 Framer frame reference signal.*
	108	NC	****	Not Bonded Out

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
72	109	TxPOS	O	**Transmit Positive Polarity Pulse:** The exact role of this output pin depends upon whether the UNI is operating in the Unipolar or Bipolar Mode. **Unipolar Mode:** This output pin functions as the "Single- Rail" output signal for the "outbound" E3 data stream. The signal, at this output pin, will be updated on the "user- selected" edge of the TxLineClk signal. **Bipolar Mode:** This output pin functions as one of the two dual rail output signals that commands the sequence of pulses to be driven on the line. TxNEG is the other output pin. This input is typically connected to the TPDATA input of the external E3 Line Interface Unit IC. When this output is asserted, it will command the LIU to generate a positive polarity pulse on the line.
	110	NC	****	Not Bonded Out
73	111	TxNEG	O	**Transmit Negative Polarity Pulse:** The exact role of this output pin depends upon whether the UNI is operating in the Unipolar or Bipolar Mode. **Unipolar Mode:** This output signal pulses "high" for one bit period, at the end or each "outbound" E3 frame. This output signal is at a logic "low" for all of the remaining bit- periods of the "outbound" E3 frames **Bipolar Mode:** This output pin functions as one of the two dual- rail output signals that commands the sequence of pulses to be driven on the line. TxPOS is the other output pin. This input is typically connected to the TNDATA input of the external E3 Line Interface Unit IC. When this output is asserted, it will command the LIU to generate a negative polarity pulse on the line.
74	112	TxLineClk	O	**Transmit Line Interface Clock:** This clock signal is output to the Line Interface Unit, along with the TxPOS and TxNEG signals. The purpose of this output clock signal is to provide the LIU with timing information that it can use to generate the AMI pulses and deliver them over the transmission medium to the Far- End Receiver. The user can configure the source of this clock to be either the RxLineClk (from the Receiver portion of the UNI) or the TxInClk input. The nominal frequency of this clock signal is 34.368 MHz.
75	113	VDD	***	Power Supply Pin
	114	NC	****	Not Bonded Out
76	115	TxAISEn	I	**Transmit AIS Pattern input:** When this input pin is set "high" the Transmit E3 Framer will insert the AIS (e.g., an all "1s") pattern into the E3 output data stream.
	116	NC	****	Not Bonded Out

PIN DESCRIPTION (contd.)

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
77	117	TxOHIns	I	**Transmit E3 Overhead Byte Serial Input Port Enable:** When the user wishes to input his/ her value for the overhead bytes into the outbound E3 data stream, he/ she should assert this input pin. When this pin is "high" then the TxOH Serial Input Port will become active, and will began sampling the TxOH input pin upon the rising edge of TxOHClk signal. When this pin is "low", then the TxOH Serial Input Port will be disabled, and the overhead bytes of the outbound E3 data stream will be internally generated.
	118	NC	****	Not Bonded Out
78	119	TxOH	I	**Transmit E3 Framer Overhead Bytes Serial Input Port input:** This pin, along with the TxOHIns, TxOHMSB and TxOHClk pins comprise the "Transmit E3 Framer OH Bytes" Serial Input Port. This input pin is active when the TxOHIns input pin is "high"; and is disabled when TxOHIns is "low". When this input pin is active, it will sample the input signal on the rising edge of the TxOHClk signal. The data that is received via this input will be inserted into the Overhead bytes of the outbound E3 Frame (via the Transmit E3 Framer Block)
	120	TCellTxed	O	**Transmit Cell Processor-Cell Transmitted Indicator:** This output pin pulses "high" each time the Transmit Cell Processor transmits a cell to the Transmit E3 Framer.
79	121	TxOHClk	O	**Transmit E3 Framer Overhead Bit Serial Input Port-clock signal output.** This output clock signal, along with the TxOH, TxOHFrame, and TxOHIns pins, comprise the "Transmit E3 Framer OH Bytes" serial input port. When this serial port is active, then the data applied to the TxOH input pin will be into the serial port on the rising edge of this clock signal. ***Note:*** *The TxOHClk signal is always active whether the "TxOH Serial Input" port is active or not.*
80	122	TxOHFrame	O	**Transmit E3 Framer Overhead Bytes Serial Input Port-framing signal indicator.** This output signal, along with the TxOH, TxOHClk and TxOHIns pins comprise the "Transmit E3 Framer OH Bytes" serial input port. This output pin pulses "high" when the value for the first "X" bit (in F- Frame #1) is expected at the TxOH input pin. ***Note:*** *This output pin is always active whether the "Transmit E3 Framer OH Byte" Serial Input port is active or not.*

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
81	123	TxEnB*	I	**Transmit Utopia Interface Block-Write Enable:** This active- low signal, from the ATM Layer processor enables the data on the Transmit Utopia Data Bus to be written into the TxFIFO on the rising edge of TxClk. When this signal is asserted, then the contents of the byte or word that is present, on the Transmit Utopia Data Bus, will be latched into the Transmit Utopia Interface block, on the rising edge of TxClk. When this signal is negated, then the Transmit Utopia Data bus inputs will be tri- stated.
82	124	TxSoC	I	**Transmitter-Start of Cell (SoC) Indicator Input:** This input pin is driven by the ATM Layer processor and is used to indicate the start of an ATM cell that is being transmitted from the ATM layer device. This input pin must be pulsed "high" when the first byte (or word) of a new cell is present on the Transmit Utopia Data Bus. This input pin must remain 'low' at all other times.
83	125	TxPrty	I	**Transmit Utopia Data Bus-Parity Input:** The ATM Layer processor will apply the parity value of the byte or word which is being applied to the Transmit Utopia Data Bus (e.g., TxData[7:0] or TxData[15:0]) inputs of the UNI, respectively. Note: this parity value should be computed based upon the odd- parity of the data applied at the Transmit Data Bus. The Transmit Utopia Interface block (within the UNI) will independently compute the odd- parity value of each byte (or word) that it receives from the ATM Layer processor and will compare it with the logic level of this input pin.
84	126	TxClav	O	**Transmit Utopia Interface-Cell Available Output Pin:** This output pin supports data flow control between the ATM Layer processor and the Transmit Utopia Interface block. The exact functionality of this pin depends upon whether the UNI is operating in the "Octet Level" or "Cell Level" handshaking mode. **Octet Level Handshaking:** When the Transmit Utopia Interface block is operating in the octet- level handshaking mode, this signal is negated (toggles "low") when the TxFIFO is not capable of handling four more write operations; by the ATM Layer processor to the Transmit Utopia Interface block. This signal will be asserted when the TxFIFO is capable of receiving four or more write operation of ATM cell data. **Cell Level Handshaking:** When the Transmit Utopia Interface block is operating the cell- level handshaking mode, this signal is asserted (toggles "high") when the TxFIFO is capable of receiving at least one more full cell of data from the ATM Layer processor. This signal is negated, if the TxFIFO is not capable of receiving one more full cell of data from the ATM Layer processor. **Multi- PHY Operation:** When the UNI chip is operating in the Multi- PHY mode, this signal will be tri- stated until the TxClk cycle following the assertion of a valid address on the Transmit Utopia Address bus input pins (e.g., when the contents on the Transmit Utopia Address bus pins match that within the Transmit Utopia Address Register). Afterwards, this output pin will behave in accordance with the cell- level handshake mode.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
85	127	GND	***	Ground Signal Pin.
	128	TxData8	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
86	129	TxData0	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	130	TxData9	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
87	131	TxData1	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	132	TxData10	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** This input pin is only available in the 160 pin package version.
88	133	TxData2	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	134	TxData11	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
89	135	TxData3	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	136	VDD	***	Power Supply Pin
90	137	TxData4	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	138	TxData12	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
91	139	TxData5	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	140	TxData13	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
92	141	TxData6	I	**Transmit Utopia Data Bus Input**: Please see description for TxData15
	142	TxData14	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15 ***Note:*** *This input pin is only available in the 160 pin package version.*
93	143	TxData7	I	**Transmit Utopia Data Bus Input:** Please see description for TxData15
	144	TxData15	I	**Transmit Utopia Data Bus Input-MSB:** This input pin, along with TxData14 through TxData0 comprise the Transmit Utopia Data Bus input pins. When the ATM Layer Processor wishes to transmit ATM cell data through the XR- T7234 E3 UNI, it must place this data on these pins. The data, on the Transmit Utopia Data Bus is latched into the Transmit Utopia Interface block the rising edge of TxClk. ***Note:*** *This input pin is only available in the 160 pin package version.*
94	145	VDD	***	Power Supply Pin

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	146	TxAddr4	I	**Transmit Utopia Address Bus-MSB Input:** This input pin, along with TxAddr3 through TxAddr0 comprise the Transmit Utopia Address Bus input pins. The Transmit Utopia Address Bus is only in use when the UNI is operating in the M-PHY mode. When the ATM Layer processor wishes to write data to a particular UNI device, it will provide the address of the "intended UNI" on the Transmit Utopia Address Bus. The contents of the Transmit Utopia Address Bus input pins are sampled on the rising edge of TxClk. The E3 UNI will compare the data on the Transmit Utopia Address Bus input pins with the pre-programmed contents of the TxUT Address Register (Address = 82h). If these two values are identical and if the TxENB pin is asserted, then the TxClav output pin will be driven to the appropriate state (based upon the TxFIFO fill level) for the Cell Level handshake mode of operation.
95	147	TxAddr0	I	**Transmit Utopia Address Bus Input-LSB:** (See Description for TxAddr4)
	148	TxAddr3	I	**Transmit Utopia Address Bus input:** (See Description for TxAddr4)
96	149	TxAddr1	I	**Transmit Utopia Address Bus input:** (See Description for TxAddr4)
	150	TxAddr2	I	**Transmit Utopia Address Bus input:** (See Description for TxAddr4)
97	151	TxClk	I	**Transmit Utopia Interface Clock:** The Transmit Utopia Interface clock is used to latch the data on the Transmit Utopia Data bus, into the Transmit Utopia Interface block. This clock signal is also used as the timing source for circuitry used to process the ATM cell data into and through the TxFIFO. During Multi-PHY operation, the data on the Transmit Utopia Address bus pins is also sampled on the rising edge of TxClk.
	152	GND	***	Ground Signal Pin
98	153	GND	***	Ground Signal Pin
	154	TxGFCMSB	O	**Transmit GFC Nibble-Field Serial Input Port-MSB Indicator:** This signal, along with TxGFC and TxGFCClk combine to function as the "Transmit GFC Nibble Field" serial input port. This output signal will pulse "high" when the MSB (most significant bit) of the GFC Nibble (for a given valid cell) is expected at the TxGFC input pin. ***Note:*** *The "Transmit GFC Nibble-field" serial input port only inserts the GFC value into valid cells. Therefore, this output pin will only pulse "high" when the Transmit Cell Processor is processing a "valid" cell. This output pin will not pulse "high" when the Transmit Cell Processor is processing Idle Cells.*
99	155	ResetB	I	**Reset Input:** When this "active-low" signal is asserted, the UNI device will be asynchronously reset. Additionally, all outputs will be "tri-stated", and all on-chip register will be reset to their default values.

PIN DESCRIPTION (CONTINUED)

Pin No.	100 Pin Package Pin No. 160 Pin Package	Symbol	Type	Description
	156	TxGFCClk	O	**Transmit GFC Nibble Field Serial Input Port Clock:** This signal, along with TxGFC, and TxGFCMSB combine to function as the "Transmit GFC Nibble- field" serial input port. The "Transmit GFC Nibble- field" serial input port uses this output clock signal to sample the values applied to the TxGFC pin, on its rising edge. This pin will provide four rising edges for each valid cell being transmitted.
100	157	ALE_AS	I	**Address Latch Enable/ Address Strobe:** This input is used to latch the address (present at the Mcroprocessor Interface Address Bus, A[8:0]) into the UNI Mcroprocessor Interface circuitry and to indicate the start of a READ/ WRITE cycle. This input is active- high in the Intel Mode (MOTO = 'low') and active- low in the Motorola Mode (MOTO = 'high').
	158	TxGFC	I	**Transmit GFC Nibble- Field Serial Input Port:** This signal, along with TxGFCClk and TxGFCMSB combine to function as the "Transmit GFC Nibble- field" serial input port. The user will specify the value of the GFC field, within a given valid (user or OAM) ATM cell, by serial transmitting its four bit value into this input. Each of these four bits will clocked into the UNI via rising edge of the TxGFCClk clock output signal. ***Note:*** *The "Transmit GFC Nibble- field" Serial input port will only insert the GFC Nibble field value into valid cells. Therefore, this input pin will only read in the GFC Nibble value when a valid cell is being processed.*
1	159	GND	***	Ground Signal Pin
2	160	Rdy_Dtck	O	**READY or DTACK:** This "active- low" output pin will function as the READY output, when the microprocessor interface is running in the "Intel" Mode; and will function as the DTACK output, when the microprocessor interface is running in the "Motorola" Mode. **"Intel" Mode-READY Output:** When the UNI negates this output pin (e.g., toggles it "low"), it indicates (to the µP) that the current READ or WRITE cycle is to be extended until this signal is asserted (e.g., toggled "high"). **"Motorola" Mode-DTACK (Data Transfer Acknowledge) Output:** The UNI device will assert this pin in order to inform the local microprocessor that the present READ or WRITE cycle is nearly complete. If the UNI device requires that the current READ or WRITE cycle be extended, then the UNI will delay its assertion of this signal. The 68000 family of µPs requires this signal from its peripheral devices, in order to quickly and properly complete a READ or WRITE cycle.

PRINCIPLE OF OPERATION

2.0 The User Network Interface (UNI)

The term UNI refers to an interface between an ATM user or end- point (e.g., workstations, bridges, routers, private switches) and an ATM network node (typically a switch). The UNI could conform to any of a number of physical transmission media standards including the synchronous as well as the pleisochronous digital hierarchies. Figure 3 illustrates an example of a possible application of the XR- T7234 UNI.

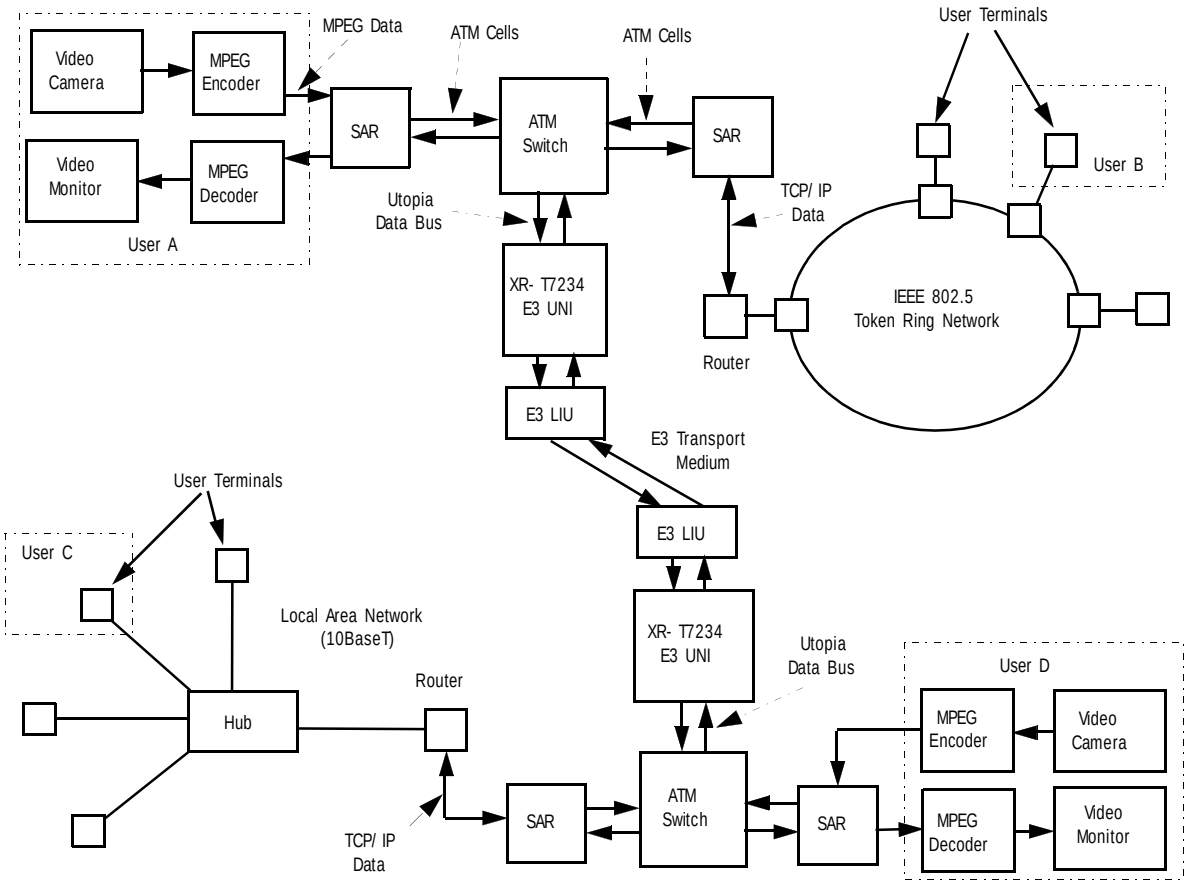

Figure 3. An Example of an Application of the XR- T7234 UNI.

The XR- T7234 UNI can functionally be subdivided into 6 different sections, as shown in Figure 4.

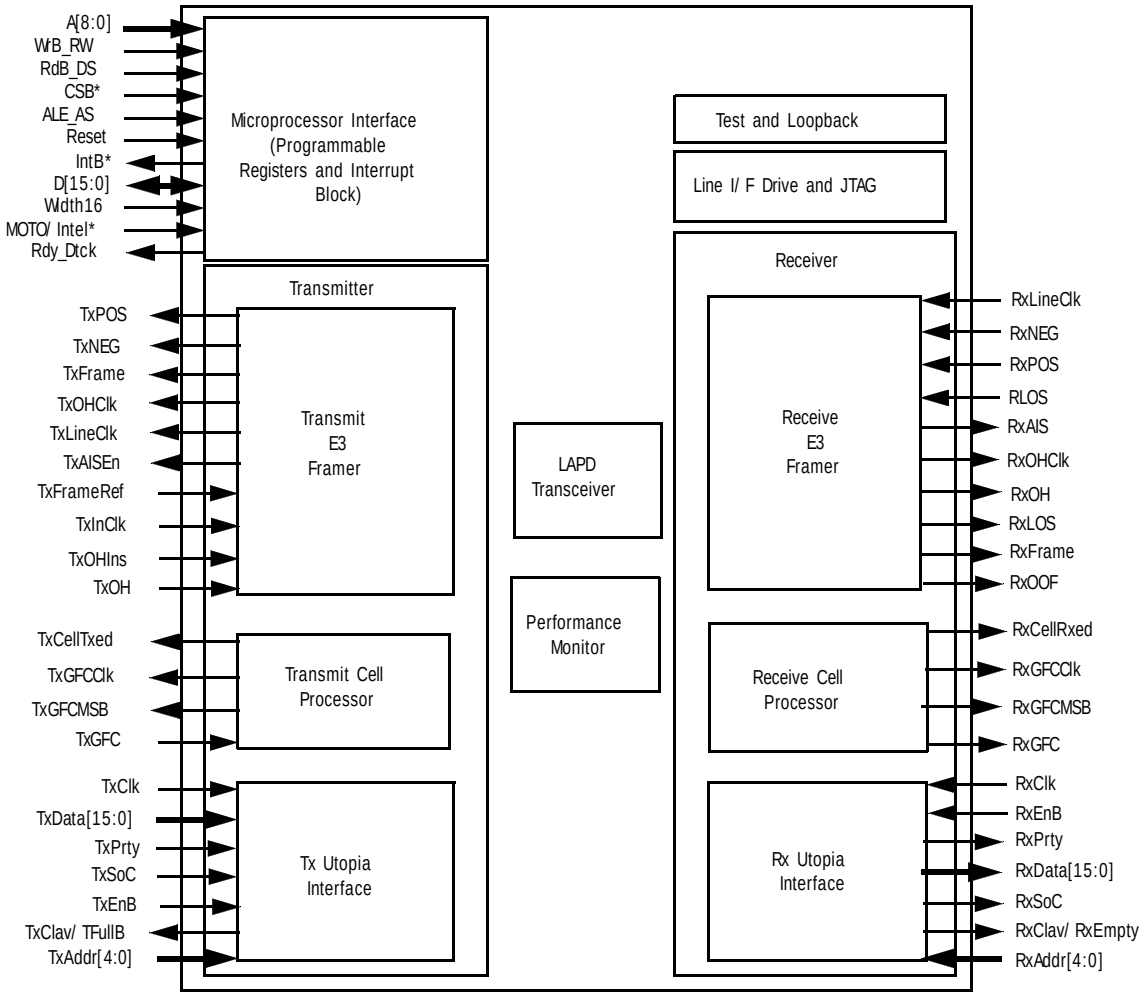

Figure 4. Functional Block Diagram of the XR- T7234 E3 UNI

These Functional Blocks are:

- Microprocessor Interface and Programmable Registers
- Test and Diagnostic Section
- Line Interface Drive and Scan Section
- Transmit Section
- Receive Section
- Performance Monitors

Each of these functional blocks will be discussed in detail below.

3.0 Microprocessor Interface Section and On-Chip Programmable Registers

The Microprocessor Interface section supports communication between the "local" microprocessor (µP) and the UNI device. In particular, the Microprocessor Interface section supports the following operations between the local microprocessor and the UNI.

- The writing of configuration data into the UNI on-chip (addressable) registers.
- The writing of "outbound" OAM cell data into the "Transmit OAM Cell" Buffer (within the UNI).
- The writing of an "outbound" PMDL (Path Maintenance Data Link) message into the "Transmit LAPD Message" buffer (within the UNI).
- The UNI IC's generation of an Interrupt Request to the µP.
- The µP's servicing of the interrupt request from the UNI.
- The monitoring of the UNI system's "health" by periodically reading the on-chip Performance Monitor registers.
- The reading of an "inbound" OAM cell data from the "Receive OAM Cell" buffer (within the UNI).
- The reading of an "inbound" PMDL Message from the "Receive LAPD" Buffer (within the UNI).

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

Figure 5 presents a simple block diagram of the Microprocessor Interface Section, within the UNI device.

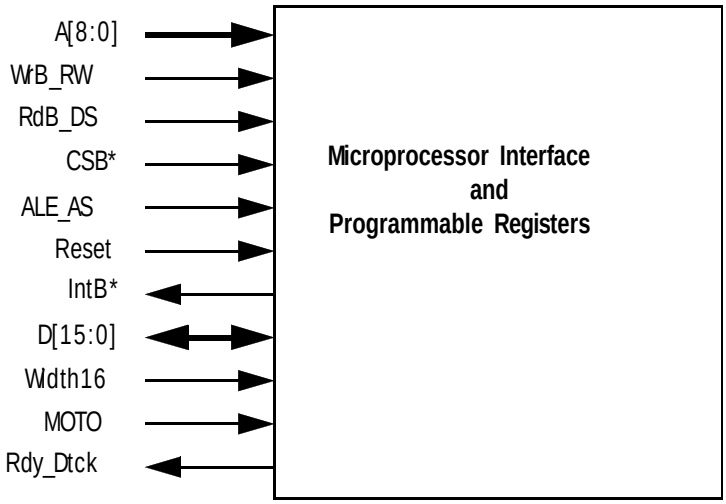

Figure 5. Simple Block Diagram of Microprocessor Interface block of UNI.

3.1 The Microprocessor Interface Signals

The UNI may be configured into a wide variety of different operating modes and have its performance monitored by software through a standard (local "housekeeping") microprocessor, using data, address and control signals.

Note: This local "housekeeping" Microprocessor should not be confused with the ATM Layer Processor that interfaces to the UNI via the Transmit and Receive Utopia Interface Blocks.

The local µ configures the UNI (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 micro-processor interface can be configured to operate in the "Motorola" mode or in the "Intel" mode. 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 microprocessors.

Tables 1 lists and describes those Microprocessor Interface signals whose role is constant across the two modes. Table 2 describes the role of some of these signals when the Microprocessor Interface is operating in the "Intel" Mode. Likewise, Table 3 describes the role of these signals when the Microprocessor Interface is operating in the "Motorola" Mode.

Table 1. Description of the Microprocessor Interface Signals that exhibit constant roles in both the "Intel" and "Motorola" Modes.

Pin Name	Type	Description
MOTO	I	Selection input for Intel/ Motorola µP Interface. Setting this pin to a logic "high" configures the Microprocessor Interface to operate in the "Motorola" mode. Likewise, setting this pin to a logic "low" configures the Microprocessor Interface to operate in the "Intel" Mode.
Width16	I	Select input for the Data Bus width: Setting this pin to a logic "high" configures the width of the Micro-processor Interface data bus width to be 16 bits. Likewise, setting this pin to a logic "low" selects a data bus width of 8 bits.
D[15:0]	I/ O	Bi- directional Data Bus for register read or write operations. Note: If the "Width16" input is "low", then only D[7:0] is active.
A[8:0]	I	Nine 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.
CSB*	I	Chip Select input. This "active- low" signal selects the Microprocessor Interface of the UNI device and enables read/ write operations with the on- chip registers/ on- chip RAM.
IntB*	O	Interrupt Request Output: This "open- drain/ active- low" output signal will inform the local µP that the UNI has an interrupt condition that needs servicing.

Table 2. 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 UNI Microprocessor Interface will place the contents of the addressed register on the Data Bus pins (D[15:0]). The Data Bus pins 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[15:0]) 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 UNI device, 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 3. 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[8:0] into the Microprocessor Interface circuitry. The contents of the Address Bus are latched into the UNI device 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 UNI) during a Write Cycle.
WRB_RW	R/W*	I	Read/ Write* Input: When this pin is "high", it indicate a Read Cycle. When this pin is low, it indicates a Write cycle.
Rdy_Dtck	DTACK*	O	Data Transfer Acknowledge: The UNI device 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.

3.2 Access in the Burst Mode

The UNI device provides the user with the ability to quickly access a series of on- chip registers in consecutive, sequential address order. This feature is known as "burst mode" operation. A burst access is started by the microprocessor asserting the ALE_AS pin, like any normal access. However, the subsequent register accesses are completed without asserting the ALE_AS pin. The UNI device will automatically, internally increment the address that is being accessed, within its address space. The Rdy_Dtck pin is used to lengthen the individual register accesses if needed.

3.3 On- Chip Register Organization

The Microprocessor Interface section, within the UNI device allows the user to do the following.

- Configure the UNI into a wide variety of operating modes.
- Employ various features of the UNI device
- Perform status monitoring
- Enable/ Disable and service Interrupt Conditions

All of these things are accomplished by reading from or writing to the many on- chip registers, within the UNI device. Table 4 lists each of these registers and their corresponding address location within the UNI address space.

3.3.1 UNI Register Addressing

The array of on- chip registers consists of a variety of register types. These registers are denoted in Table 4, as follows:

- R/O-Read Only Registers
- R/W-Read/ Write Registers
- RUR-Reset upon Read Registers
- Sem-Semaphore Bit- field

Additionally some of these registers consists of both R/O and R/W bit- fields. These registers are denoted in Table 4 as "Combination of R/W and R/O".

The bit- format and definitions for each of these registers are presented in Section 3.3.2.

Table 4. Register Addressing of the UNI Programmable Registers

Address	Read Mode Register	Write Mode Register	Register Type
00h	UNI Operating Mode Register	UNI Operating Mode Register	R/W
01h	UNI I/O Control Register	UNI I/O Control Register	R/W
02h	Part Number Register		R/O
03h	Version Number Register		R/O
04h	UNI Interrupt Enable Register	UNI Interrupt Enable Register	R/W
05h	UNI Interrupt Status Register		R/O
06h	Test Cell Control and Status Register	Test Cell Control and Status Register (R/W portion only)	Combination of R/O and R/W
07h	Future Use	Future Use	
08h	Test Cell Header Byte 1	Test Cell Header Byte 1	R/W

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
09h	Test Cell Header Byte 2	Test Cell Header Byte 2	R/W
0Ah	Test Cell Header Byte 3	Test Cell Header Byte 3	R/W
0Bh	Test Cell Header Byte 4	Test Cell Header Byte 4	R/W
0Ch	Test Cell Error Accumulator-Most Significant Byte		R/O
0Dh	Test Cell Error Accumulator-Least Significant Byte		R/O
0Eh	Rx E3 Framer Configuration and Status Register-1	Rx E3 Configuration and Status Register-1 (R/W portion only)	Combination of R/O and R/W
0Fh	Rx E3 Framer Configuration and Status Register -2	Rx E3 Framer Configuration and Status Register-2	Combination of R/O and R/W
10h	Rx E3 Framer Interrupt Enable Register-1	Rx E3 Framer Interrupt Enable Register-1 (R/W portion only)	Combination of R/O and R/W
11h	Rx E3 Framer Interrupt Enable Register -2	Rx E3 Framer Interrupt Enable Register-2 (R/W portions only)	Combination of R/O and R/W
12h	Rx E3 Framer Interrupt Status Register-1		Combination of R/O and RUR
13h	Rx E3 Framer Interrupt Status Register-2		Combination of R/O and RUR
14h	Rx NR Register		R/O
15h	Rx GC Register		R/O
16h	Rx TTB- 0 Register		R/O
17h	Rx TTB- 1 Register		R/O
18h	Rx TTB- 2 Register		R/O
19h	Rx TTB- 3 Register		R/O
1Ah	Rx TTB- 4 Register		R/O
1Bh	Rx TTB- 5 Register		R/O
1Ch	Rx TTB- 6 Register		R/O
1Dh	Rx TTB- 7 Register		R/O
1Eh	Rx TTB- 8 Register		R/O
1Fh	Rx TTB- 9 Register		R/O
20h	Rx TTB- 10 Register		R/O

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
21h	Rx TIB- 11 Register		R/O
22h	Rx TIB- 12 Register		R/O
23h	Rx TIB- 13 Register		R/O
24h	Rx TIB- 14 Register		R/O
25h	Rx TIB- 15 Register		R/O
26h	Rx E3 LAPD Control Register	Rx E3 LAPD Control Register	R/W
27h	Rx E3 LAPD Status Register		R/O
28h	Tx E3 Framer Configuration Register	Tx E3 Framer Configuration Register	R/W
29h	Tx GC Byte Register	Tx GC Byte Register	R/W
2Ah	Tx MA Byte Register	Tx MA Byte Register	R/W
2Bh	Tx NR Byte Register	Tx NR Byte Register	R/W
2Ch	Tx TIB- 0 Register	Tx TIB- 0 Register	R/W
2Dh	Tx TIB- 1 Register	Tx TIB- 1 Register	R/W
2Eh	Tx TIB- 2 Register	Tx TIB- 2 Register	R/W
2Fh	Tx TIB- 3 Register	Tx TIB- 3 Register	R/W
30h	Tx TIB- 4 Register	Tx TIB- 4 Register	R/W
31h	Tx TIB- 5 Register	Tx TIB- 5 Register	R/W
32h	Tx TIB- 6 Register	Tx TIB- 6 Register	R/W
33h	Tx TIB- 7 Register	Tx TIB- 7 Register	R/W
34h	Tx TIB- 8 Register	Tx TIB- 8 Register	R/W
35h	Tx TIB- 9 Register	Tx TIB- 9 Register	R/W
36h	Tx TIB- 10 Register	Tx TIB- 10 Register	R/W
37h	Tx TIB- 11 Register	Tx TIB- 11 Register	R/W
38h	Tx TIB- 12 Register	Tx TIB- 12 Register	R/W
39h	Tx TIB- 13 Register	Tx TIB- 13 Register	R/W
3Ah	Tx TIB- 14 Register	Tx TIB- 14 Register	R/W
3Bh	Tx TIB- 15 Register	Tx TIB- 15 Register	R/W
3Ch	TxFA1 Error Mask Register	TxFA1 Error Mask Register	R/W
3Dh	TxFA2 Error Mask Register	TxFA2 Error Mask Register	R/W

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
3Eh	TxBIP-8 Error Mask Register	TxBIP-8 Error Mask Register	R/W
3Fh	Tx E3 LAPD Status/Interrupt Register	Tx E3 LAPD Status/Interrupt Register (R/W portions thereof)	Combination of R/W, R/C, and RUR
40h	PMON LCV Event Count Register-MSB		RUR
41h	PMON LCV Event Count Register-LSB		RUR
42h	PMON Framing Error Event Count-MSB		RUR
43h	PMON Framing Error Event Count-LSB		RUR
44h	PMON Received FEBE Event Count-MSB		RUR
45h	PMON Receive FEBE Event Count-LSB		RUR
46h	PMON Framer Parity Error Event Count-MSB		RUR
47h	PMON Framer Parity Error Event Count-LSB		RUR
48h	PMON Received Single-Bit HEC Error Count-MSB		RUR
49h	PMON Received Single-bit HEC Error Count-LSB		RUR
4Ah	PMON Received Multiple-bit HEC Error Count-MSB		RUR
4Bh	PMON Received Multiple-bit HEC Error Count-LSB		RUR
4Ch	PMON Received Idle Cell Count-Most Significant Byte		RUR
4Dh	PMON Received Idle Cell Count-Least Significant Byte		RUR
4Eh	PMON Received Valid Cell Count-Most Significant Byte		RUR
4Fh	PMON Received Valid Cell Count-Least Significant Byte		RUR
50h	PMON Discarded Cell Count— Most Significant Byte		RUR

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
51h	PMON Discarded Cell Count— Least Significant Byte		RUR
52h	PMON Transmitted Idle Cell Count-Most Significant Byte		RUR
53h	PMON Transmitted Idle Cell Count-Least Significant Byte		RUR
54h	PMON Transmitted Valid Cell Count-Most Significant Byte		RUR
55h	PMON Transmitted Valid Cell Count-Least Significant Byte		RUR
56h	PMON Holding Register		R/O
57h	One Second Error Status Register		R/O
58h	LCV-One Second Accumulator Register-Most Significant Byte		R/O
59h	LCV - One Second Accumulator Register - Least Significant Byte		R/O
5Ah	BIP8-One Second Accumulation Register-MSB		R/O
5Bh	BIP8-One Second Accumulation Register-LSB		R/O
5Ch	HEC Errors-One Second Accumulator Register-Most Significant Byte		R/O
5Dh	HEC Errors-One Second Accumulator Register-Least Significant Byte		R/O
5Eh	Rx CP Configuration Register	Rx CP Configuration Register	R/W
5Fh	Rx CP Additional Configuration Register	Rx CP Additional Configuration Register	R/W
60h	Rx CP Interrupt Enable Register	Rx CP Interrupt Enable Register	R/W
61h	Rx CP Interrupt Status Register	Rx CP Interrupt Status Register	RUR
62h	Rx CP Idle Cell Pattern Header Byte- 1 Register	Rx CP Idle Cell Pattern Header Byte- 1 Register	R/W
63h	Rx CP Idle Cell Pattern Header Byte- 2 Register	Rx CP Idle Cell Pattern Header Byte2 Register	R/W
64h	Rx CP Idle Cell Pattern Header Byte- 3 Register	Rx CP Idle Cell Pattern Header Byte- 3 Register	R/W
65h	Rx CP Idle Cell Pattern Header Byte- 4 Register	Rx CP Idle Cell Pattern Header Byte- 4 Register	R/W
66h	Rx CP Idle Cell Mask Byte- 1 Register	Rx CP Idle Cell Mask Byte- 1 Register	R/W

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
67h	Rx CP Idle Cell Mask Byte- 2 Register	Rx CP Idle Cell Mask Byte- 2 Register	R/W
68h	Rx CP Idle Cell Mask Byte- 3 Register	Rx CP Idle Cell Mask Byte- 3 Register	R/W
69h	Rx CP Idle Cell Mask Byte- 4 Register	Rx CP Idle Cell Mask Byte- 4 Register	R/W
6Ah	Rx CP User Programmable Cell Filter Pattern Header Byte 1 Register	Rx CP User Programmable Cell Filter Pattern Header Byte 1 Register	R/W
6Bh	Rx CP User Programmable Cell Filter Pattern Header Byte 2 Register	Rx CP User Programmable Cell Filter Pattern Header Byte 2 Register	R/W
6Ch	Rx CP User Programmable Cell Filter Pattern Header Byte 3 Register	Rx CP User Programmable Cell Filter Pattern Header Byte 3 Register	R/W
6Dh	Rx CP User Programmable Cell Filter Pattern Header Byte 4 Register	Rx CP User Programmable Cell Filter Pattern Header Byte 4 Register	R/W
6Eh	Rx CP User Programmable Cell Filter Mask Byte 1 Register	Rx CP User Programmable Cell Filter Mask Byte 1 Register	R/W
6Fh	Rx CP User Programmable Cell Filter Mask Byte 2 Register	Rx CP User Programmable Cell Filter Mask Byte 2 Register	R/W
70h	Rx CP User Programmable Cell Filter Mask Byte 3 Register	Rx CP User Programmable Cell Filter Mask Byte 3 Register	R/W
71h	Rx CP User Programmable Cell Filter Mask Byte 4 Register	Rx CP User Programmable Cell Filter Mask Byte 4 Register	R/W
72h	Tx CP Control/ Interrupt Register	Tx CP Control/ Interrupt Register	R/W
73h	Tx CP OAM Register		Sem
74h	Tx CP HEC Error Mask Register	Tx CP HEC Error Mask Register	R/W
75h	Future Use	Future Use	****
76h	Tx CP Idle Cell Pattern Header Byte- 1 Register	Tx CP Idle Cell Pattern Header Byte - 1 Register	R/W
77h	Tx CP Idle Cell Pattern Header Byte- 2 Register	Tx CP Idle Cell Pattern Header Byte - 2 Register	R/W
78h	Tx CP Idle Cell Pattern Header Byte- 3 Register	Tx CP Idle Cell Pattern Header Byte - 3 Register	R/W
79h	Tx CP Idle Cell Pattern Header Byte- 4 Register	Tx CP Idle Cell Pattern Header Byte - 4 Register	R/W
7Ah	Tx CP Idle Cell Pattern Header Byte - 5 Register	Tx CP Idle Cell Pattern Header Byte - 5 Register	R/W
7Bh	Tx CP Idle Cell Payload Register	Tx CP Idle Cell Payload Register	R/W
7Ch	Utopia Configuration Register	Utopia Configuration Register	R/W
7Dh	Rx UT Interrupt Enable/ Status Register	Rx UT Interrupt Enable/ Status Register	Combination of R/W and RUR
7Eh	Rx Utopia Address Register	Rx Utopia Address Register	R/W

Table 4. Register Addressing of the UNI Programmable Registers (Continued)

Address	Read Mode Register	Write Mode Register	Register Type
7Fh	Future Use	Future Use	****
80h	Tx UTOPIA Interrupt/ Status Register	Tx UTOPIA Interrupt/ Status Register	Combination of R/W and RUR
81h	Future Use	Future Use	****
82h	Tx Utopia Address Register	Tx Utopia Address Register	R/W
83h	Tx FIFO Status Register		R/O
84h	Line Interface Drive Register	Line Interface Drive Register	R/W
85h	Line Interface Scan Register		R/O
86h- DDh	Transmit LAPD Message Buffer	Transmit LAPD Message Buffer	R/W
DEh- 135h	Receive LAPD Message Buffer	Receive LAPD Message Buffer	R/O
136h- 16Bh	Transmit OAM Cell Buffer	Transmit OAM Cell Buffer	R/W
16Ch- 1A1	Receive OAM Cell Buffer	Receive OAM Cell Buffer	R/O

3.3.2 UNI Register Description

This section provides a functional description of each bit- field within each of the on- chip UNI Registers.

Note: For all on- chip registers, a table containing the bit- format of the register is presented. Additionally, these tables also contain the default values for each of these register bits. Finally, the functional description, associated with each register bit- field is presented, along with a reference to a Section Number, within this Data Sheet, that provides a more in- depth discussion of the functions associated with this register bit- field.

3.3.2.1 UNI Operating Mode Register

Address = 00h, UNI Operating Mode Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	Disable LOC	Int LOS Enable	Reset By Reg	Cell Loop			
back	Line Loop						
back	TimRef						
Sel(1)	TimRef						

Address = 00h, UNI Operating Mode Register							
Sel(0)							
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	1	0	0	0	0	0

Bit 6-Disable LOC

This "Read/Write" bit-field allows the user to enable or disable the "Loss of Clock signal" circuit. Writing a "1" to this bit-field disables the "Loss of Clock" circuit. Writing a "0" to this bit-field enables the "Loss of Clock" circuit.

Bit 5-Int LOS Enable

This "Read/Write" bit-field allows the user to define the "Loss of Signal" (LOS) declaration criteria that the Receive E3 Framer will use. At most, the Receive E3 Framer can declare an LOS condition if one of the following two conditions are met.

1. The RxPOS and RxNEG input pins are 'stuck' at "0" for 32 or more consecutive bit-periods.
2. The XRT7295 E3 Line Receive IC asserts the RLOS input pin of the UNI (please see Section 5.0).

Writing a "0" to this bit-field configures the Receive E3 Framer to declare an LOS condition only if the RLOS input pin is asserted. In this configuration, the Receive E3 Framer will not declare an LOS condition if it detects a string of 32 (or more) consecutive "0s", in the incoming E3 data stream via the RxPOS and RxNEG pins.

Writing a "1" to this bit-field configures to the Receive E3 Framer to declare an LOS condition if either one of the following two conditions occur:

- The RxPOS and RxNEG input pins are stuck at "0" for 32 or more consecutive bit-periods.
- If the RLOS input pin is asserted.

Bit 4 Reset By Reg (Reset by Register Setting)

This "Read/Write" bit-field allows the local µP/µC to command a reset of the entire UNI IC. When the UNI is reset, both the TxFIFO and the RxFIFO are flushed, all on-chip registers are reset to their default values, and all configurations are automatically set to their default conditions.

Writing a "1" to this bit-field will reset the UNI IC. Writing a "0" to this bit-field imposes no change in the UNI IC.

Bit 3-"Cell Loopback" Mode

This "Read/Write" bit-field allows the user to configure the UNI to operate in the "Cell Loopback" mode. This is a operating mode that is available via the UNI Test and Diagnostic Section. When the UNI is operating in this mode, then the ATM Cells that are delineated and pass through the Receive Cell Processor will be routed directly (internally) to the Tx FIFO (within the Transmit Utopia Interface block).

Writing a "1" to this bit-field enables the "Cell Loopback" Mode. Writing a "0" to this bit-field disables the Cell Loopback Mode.

For more information on the Cell Loopback Mode of operation, please see Section 4.1.3.

Bit 2-"Line Loopback" Mode

This "Read/Write" bit-field allows the user to configure the UNI to operate in the "Line Loopback" mode. This is a operating mode that is available via the UNI Test and Diagnostic Section. When the UNI is operating in this mode, then the data stream from the TxPOS and TxNEG pins of the Transmit E3 Framer, are (internally) looped back into the Receive RxPOS and RxNEG input pins of the Receive E3 Framer.

Writing a "1" to this bit-field enables the "Line Loopback" Mode. Writing a "0" to this bit-field disables the "Line Loopback" Mode.

For more information on the Line Loopback Mode of operation, please see Section 4.1.1.

Bits 1, 0–TimRefSel[1, 0] (Timing / Framing Reference Select Bits–Transmit E3 Framer)

These two "Read/ Write" bit- fields allows the user to select the timing reference signal for the Transmit E3 Framer block. The relationship between these two bit- fields and these three parameters are tabulated below.

TimRefSel[1, 0]	Transmit E3 Framer Timing/ Framing Reference
00	**TxInClk** • Framing is asynchronous from Power On. • Framer Timing is derived from the 34.386 MHz clock signal, which is applied at the TxInClk pin. (For more information please see Section 6.3.3.6.2)
01	**TxFrameRef** • Framing is derived from the TxFrameRef input (e.g., a new E3 frame is started, upon the rising edge of the TxFrameRef input signal). • Framer Timing is derived from the 34.386 MHz clock signal at the TxInClk pin. (For more information please see Section 6.3.3.6.3)
10	**RxLineClk** • Framing is derived from the Receive E3 Framer • Framer Timing is derived from the RxLineClk input signal. • (For more information, please see Section 6.3.3.6.1)
11	TxInClk • Framing is asynchronous from Power On. • Framer Timing is derived from the 34.368 MHz clock signal, which is applied at the TxInClk pin. (For more information please see Section 6.3.3.6.2)

3.3.2.2 UNI I/O Control Register

Address = 01h, UNI I/O Control Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit 7-LOC Rx

This "Read- Only" bit- field indicates whether or not the UNI is currently experiencing a "Loss of RxLineClk signal" event. This UNI will set this bit- field to "1" if it is currently experiencing a "Loss of RxLineClk signal" event. Conversely, the UNI will clear this bit- field (to "0") if it detects the presence of the RxLineClk signal.

Bit 6-LOC Tx

This "Read- Only" bit- field indicates whether or not the UNI is currently experiencing a "Loss of TxInClk signal" event. This UNI will set this bit- field to "1" if it is currently experiencing a "Loss of TxInClk signal" event. Conversely, the UNI will clear this bit- field (to "0") if it detects the presence of the TxInClk signal.

Bit 5-Int En Reset (Automatic Reset of Interrupt Enable Bits) Select

This "Read/ Write" bit- field allows the user to configure the UNI to automatically clear the "Interrupt Enable" bit of an 'activated' interrupt, during the Interrupt Service Routine. If the user selects this option, then an interrupt will be automatically disabled following its activation. The user must go back and write to the appropriate register(s) in order to enable the interrupt once again. This option is useful in preventing a recursively occurring interrupt from "tying up the system" and loading down the local µC/ µP.

Writing a '1' to this bit- field configures the UNI to automatically disable interrupts, following their activation. Writing a '0' to this bit- field configures the UNI to leave the Interrupt Enable bits enabled, following interrupt activation.

Bit 4-AMI/ HDB3 (Line Code)*

This "Read/ Write" bit- field allows the user to specify whether the E3 line code should be in the AMI (Alternate Mark Inversion) or HDB3 (Bipolar, with 4 Zero Substitution) format.

Writing a "1" to this bit- field configures the line code (of the Transmit and Receive E3 Framers) to be AMI. Writing a "0" to this bit- field, configures the line code (of the Transmit and Receive E3 Framers) to be HDB3. For more information into the AMI and HDB3 format, please see Sections 6.3.3.7.1.2.1 and 6.3.3.7.1.2.2.

Note: This bit- field is ignored if Bit 3 is "1".

Bit 3-Unipolar/ Bipolar (Line Code)*

This "Read/ Write" bit- field allows the user to configure the Transmit and Receive E3 Framers to transmit/ receive data in a Unipolar (Single- Rail) or in a Bipolar (Dual- Rail) format.

If the user selects the "Bipolar" format, then he/ she can manipulate Bit 4 (of this register) in order to select either the AMI or HDB3 line code.

Writing a "0" to this bit- field configures the Transmit E3 Framer to transmit data in a bipolar (dual- rail) format; and the Receive E3 Framer to receive data from the "line" in a bipolar (dual- rail) format. Writing a "1" to this bit- field configures the Transmit E3 Framer to transmit data to the line, in a unipolar (single- rail) format, and the Receive E3 Framer to receive data from the "line" in a unipolar format.

For more information on Unipolar and Bipolar formats, please see Sections 6.3.3.7.1.1 and 6.3.3.7.1.2.

Bit 2-TxLineClk Inv

This "Read/ Write" bit- field allows the user to configure the Transmit E3 Framer to update the outbound E3 data on the TxPOS and TxNEG output pins, on either the rising or the falling edge of TxLineClk.

Writing a "0" to this bit- field configures the Transmit E3 Framer to update TxPOS and TxNEG on the rising edge of TxLineClk.

Writing a "1" to this bit- field configures the Transmit E3 Framer to update TxPOS and TxNEG on the falling edge of TxLineClk.

For more information on this feature, please see Section 6.3.3.7.2.

Bit 1-RxLineClk Inv

This "Read/Write" bit-field allows the user to configure the Receive E3 Framer to "sample" the incoming E3 data at the RxPOS and RxNEG input pins, on either the rising or falling edge of RxLineClk.

Writing a "0" configures the Receive E3 Framer to sample the RxPOS and RxNEG input pins on the rising edge of the RxLineClk.

Writing a "1" configures the Receive E3 Framer to sample the RxPOS and RxNEG input pins on the falling edge of the RxLineClk.

For more information on this feature, please see Section 7.1.2.1.3.

Bit 0-Reframe (Receive E3 Framer)

When a "0" to "1" transition is detected in this bit-field, then the Receive E3 Framer will be forced to start a frame search.

3.3.2.3 Part Number Register

Address = 02h, Part Number Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Part Number							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	1	1

3.3.2.4 Version Number Register

Address = 03h, Version Number Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Version Number							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	1	1

3.3.2.5 UNI Interrupt Enable Register

Address = 04h, UNI Interrupt Enable Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec Interrupt Enable	Tx E3 Framer Interrupt Enable	Rx E3 Framer Interrupt Enable	Tx CP Interrupt Enable	Rx CP Interrupt Enabl	Tx Utopia Interrupt Enable	Rx Utopia Interrupt Enable
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit 6-One Second Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "One Second" Interrupt, that is automatically generated by the UNI device, once for each second.

Writing a "0" to this bit- field disables this interrupt. Conversely, writing a "1" to this bit- field enables the "One-Second" interrupt.

Bit 5-Tx E3 Framer Interrupt Enable

This "Read/ Write" bit- field allows the user to disable the "Transmit E3 Framer block" related interrupt, or to enable this interrupt, that has been enabled via the "Tx E3 LAPD Status/ Interrupt" Register (Address = 3Fh).

Writing a "0" to this bit- field disables the "Transmit E3 Framer block" related interrupt (independent of the enable/ disable status of this interrupt within the "Tx E3 LAPD Status/ Interrupt" Register). Writing a "1" to this bit- field enables this interrupt provided that it has already been enabled via the "Tx E3 LAPD Status/ Interrupt" register.

Bit 4-Rx E3 Framer Interrupt Enable

This "Read/ Write" bit- field allows the user to globally disable all "Receive E3 Framer" block interrupts; or to enable those "Receive E3 Framer" interrupts that are enabled via "Rx E3 Interrupt Enable Register- 1" (Address = 10h), or the "Rx E3 Interrupt Enable Register - 2" (Address = 11h).

Writing a "0" to this bit- field disables ALL "Receive E3 Framer" block interrupts (independent of the enable/ disable status of these interrupts within these other registers). Writing a "1" to this bit- field enables those "Receive E3 Framer" interrupt that have already been enabled via these other registers.

Bit 3-Tx CP Interrupt Enable

This "Read/ Write" bit- field allows the user to disable the "Transmit Cell Processor block" related interrupt, or to enable this interrupt, that has been enabled via the "Tx CP Control/ Interrupt" Register (Address = 72h).

Writing a "0" to this bit- field disables the "Transmit Cell Processor block" related interrupt (independent of the enable/ disable status of this interrupt within the "Tx CP Control/ Interrupt" Register). Writing a "1" to this bit- field enables this interrupt, provided it has been enabled within the "Tx CP Control/ Interrupt" Register.

Bit 5-Rx CP Interrupt Enable

This "Read/ Write" bit- field allows the user to globally disable all "Receive Cell Processor block" interrupts; or to enables those interrupts that have been enabled via the "Rx CP Interrupt Enable" Register (Address = 60h).

Writing a "0" to this bit- field disables ALL "Receive Cell Processor block" interrupts (independent of the enable/ disable status of these interrupts via the "Rx CP Interrupt Enable" Register). Writing a "1" to this bit- field enables those "Receive Cell Processor block" interrupt that have already been enabled via the "Rx CP Interrupt Enable" Register.

Bit 1-Tx Utopia Interrupt Enable

This "Read/ Write" bit- field allows the user to globally disable all "Transmit Utopia Interface block" interrupts; or to enable those interrupts that have been enabled via the "Tx Utopia Interrupt Enable/ Status" Register (Address = 80h).

Writing a "0" to this bit- field disables ALL "Transmit Utopia Interface block" interrupts (independent of the enable/ disable status of these interrupts within the "Tx Utopia Interrupt Enable/ Status" Register). Writing a "1" to this bit- field enables those "Transmit Utopia Interface block" interrupts that have already been enabled via the "Tx Utopia Interrupt Enable/ Status" Register.

Bit 0-Rx Utopia Interrupt Enable

This "Read/ Write" bit- field allows the user to globally disable all "Receive Utopia Interface block" interrupts; or to enables those interrupts that have been enabled via the "Rx Utopia Interrupt Enable/ Status" Register (Address = 7Dh).

Writing a "0" to this bit- field disables ALL "Receive Utopia Interface block" interrupts (independent of the enable/ disable status of these interrupts within the "Rx Utopia Interrupt Enable/ Status" Register). Writing a "1" to this bit- field enables those "Receive Utopia Interface block" interrupts that have already been enabled via the "Rx Utopia Interrupt Enable/ Status" register.

3.3.2.6 UNI Interrupt Status Register

Address = 05h, UNI Interrupt Status Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec Interrupt Status	Tx E3 Framer Interrupt Enable	Rx E3 Framer Interrupt Enable	Tx CP Interrupt Enable	Rx CP Interrupt Enable	Tx Utopia Interrupt Enable	Rx Utopia Interrupt Enable
RO	RUR	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Bit 6-One Second Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "One Second" interrupt has occurred, since the last read of this register.

If this bit- field is "0", then a "One Second" interrupt has not occurred.

However, if this bit- field is "1", then the "One Second" interrupt has occurred.

Bit 5—Tx E3 (Framer) Interrupt Status

This "Read- Only" bit- field indicates whether or not a "Transmit E3 Framer block" interrupt request is pending.

If this bit- field is "0", then no "Transmit E3 Framer block" interrupt request is pending.

However, if this bit- field is "1", then a "Transmit E3 Framer" block interrupt request is pending and awaiting service. Since the Transmit E3 Framer has one potential interrupt (LAPD Message Transfer Complete), the user should include a read to the Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh), during the Interrupt Service Routine, in order to properly service this interrupt.

This bit- field will be cleared (set to "0") after the local µP has read the "Tx E3 LAPD Status/ Interrupt" register.

Bit 4-Rx E3 (Framer) Interrupt Status

This "Read- Only" bit field indicates whether or not a "Receive E3 Framer block" interrupt request is pending.

If this bit- field is "0", then no "Receive E3 Framer block" interrupt request is pending.

However, if this bit- field is "1" then a "Receive E3 Framer block" interrupt request is pending and awaiting service. Since the Receive E3 Framer has several "potential" interrupt sources, the user should include reads to the following reg-

isters, during the Interrupt Service Routine, in order to determine the exact cause of the interrupt.

- Rx E3 Interrupt Status Register-1- (Address = 12h)
- Rx E3 Interrupt Status Register-2- (Address = 13h)

This bit- field will be cleared (set to "0") after the local µP has read the "Rx E3 Interrupt Status" register that contains the bit- field which is associated with the interrupting condition.

Bit 3-Tx CP (Cell Processor) Interrupt Status

This "Read- Only" bit- field indicates whether or not a "Transmit Cell Processor block" interrupt is pending.

If this bit- field is "0", then no "Transmit Cell Processor block" interrupt request is pending.

However, if this bit- field is "1", then a "Transmit Cell Processor block" interrupt request is pending and awaiting service. Since the Transmit Cell Processor has only one potential interrupt source (Data Path Integrity Error Occurrence), the user should still include a read to the "Tx CP Control/ Interrupt Register (Address = 72h), in the Interrupt Service Routine, in order to properly service this interrupt.

This bit- field will be cleared (set to "0") after the local µP has read the "Tx CP Control/ Interrupt" register.

Bit 2-Rx CP (Cell Processor) Interrupt Status

This "Read- Only" bit field indicates whether or not a "Receive Cell Processor block" interrupt request is pending.

If this bit- field is "0", then no "Receive Cell Processor block" interrupt request is pending.

However, if this bit- field is "1" then a "Receive Cell Processor block" interrupt is pending and awaiting service. Since the Receive Cell Processor has several "potential" interrupt sources, the user should include a read to the "Rx CP Interrupt Status" Register (Address = 61h), during the Interrupt Service Routine, in order to determine the exact cause of the interrupt.

This bit- field will be cleared (set to "0") after the local µP has read the "Rx CP Interrupt Status" register.

Bit 1-Tx Utopia Interrupt Status

This "Read- Only" bit field indicates whether or not a "Transmit Utopia Interface block" interrupt request is pending.

If this bit- field is "0", then no "Transmit Utopia Interface block" interrupt request is pending.

However, if this bit- field is "1", then a "Transmit Utopia Interface block" interrupt request is pending and awaiting service. Since the Transmit Utopia Interface Block has multiple potential interrupt sources, the user should include a read to the "Tx Utopia Interrupt/ Status Register" (Address = 80h) in the Interrupt Service Routine, in order to determine the exact cause of the interrupt.

This bit- field will be cleared (set to "0") after the local µP has read the "Tx UT Interrupt/ Status" register.

Bit 0-Rx Utopia Interrupt Status

This "Read- Only" bit field indicates whether or not a "Receive Utopia Interface block" interrupt request is pending.

If this bit- field is "0", then no "Receive Utopia Interface block" interrupt request is pending.

However, if this bit- field is "1", then a "Receive Utopia Interface block" interrupt request is pending and awaiting service. Since the Receive Utopia Interface block has multiple potential interrupt sources, the user should include a read to the "Rx UT Interrupt Enable/ Status Register" (Address = 7Dh), in order to determine the exact cause of the interrupt.

This bit- field will be cleared (set to "0") after the local µP has read the "Rx UT Interrupt Enable/ Status" register.

3.3.2.7 Test Cell Control and Status Register

Address = 06h, Test Cell Control and Status Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	Test Cell Enable	Unused	One Shot Test	One Shot Done	PRBS Lock		
RO	R/W	RO	R/W	RO	R/W	RO	RO
0	0	0	0	0	0	0	0

Bit 4-Test Cell (Generator/Receiver) Enable

This "Read/Write" bit-field allows the user to perform some testing on the UNI, by activating the "Test and Diagnostic" section. Once the user has activated this section, then the "internal" Test Cell Generator and Test Cell Receiver will be active. The Test Cell Generator will generate cells in accordance with a "traffic pattern" as dictated by the user in his/her selection within the "One Shot Test" bit field. The Test Cell Generator will generate test cells that contain the header byte pattern, as specified (by the user) in the "Test Cell Header Byte-1 through 4" registers. The payload portion of each of these test cells will be "filled with" data generated by a Pseudo-Random Byte Sequence (PRBS) generator.

The Test Cell Receiver functions as the "Test Cell Sink" and bit-error analyzer. The Test Cell Receiver will recognize each of these test cells by their header byte patterns. Further, the Test Cell Receiver will attempt to analyze the payload data (within each of these test cells) by acquiring "PRBS Lock" on the data. Once the Test Cell Receiver has "PRBS Lock" on this test cell payload data, then it can perform error-checking and error-reporting on this data. The "Test and Diagnostic" section of the UNI performs error reporting by updating the "Test Cell Error Accumulator" registers.

Writing a "1" to this bit-field enables the "Test Cell Generator/Receiver". Writing a "0" disables the "Test Cell Generator/Receiver".

For more information on these features, please see Section 4.3.

Bit 2-One Shot Test

This "Read/Write" bit-field allows the user to specify which of two "traffic options" that he/she would like the test cells to be generated. The UNI "Test and Diagnostic" section supports the following traffic options:

- "One Shot" Mode-A-single burst of 1024 Test Cells are generated
- "Continuous" Mode-A-continuous stream of Test Cells are generated for the duration that the "Test Cell Generator/Receiver" receiver are enabled.

Setting this bit-field to "1", followed by a "0" to "1" transition in the "Test Cell Enable" bit field (Bit 4 of this register), causes the Test Cell Generator to operate in the "One Shot" Mode, and generate a single burst of 1024 test cells. Afterwards the Test Cell Generator will halt, and will cease the production of new test cells, until the next "0" to "1" transition occurs in the "Test Cell Enable" bit field.

Conversely, setting this bit-field to "0", followed by a "0" to "1" transition in the "Test Cell Enable" bit field, causes the Test Cell Generator to operate in the "Continuous" Mode. When the Test Cell Generator is operating in the "Continuous" mode, it will produce a "continuous" stream of Test Cells, for the duration that the "Test Cell Enable" bit-field is set to "1".

Bit 1-One Shot Done

This "Read-Only" bit-field allows the user to monitor the status of the Test Cell Generator, while it is operating in the "One Shot" Mode. This bit-field will be set to "1", when the Test Cell Generator has completed its generator of the "burst" of the 1024 test cells. Conversely, this bit-field will be set to "0" while "test cell generation" is "in process".

Note: This bit-field has no meaning if the Test Cell Generator is operating in the "Continuous" Mode.

Bit 0-PRBS (Pseudo- Random Byte Sequence) Lock

This "Read- Only" bit field indicates whether or not the "Test Cell Receiver" has acquired "PRBS Lock" with the payload data of the incoming test cells. Once the "Test Cell Receiver" has acquired "PRBS Lock" with this data, then it can begin to perform error- checking on the incoming test cells.

Address = 07h, Future Use							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***

3.3.2.9 Test Cell Header Byte- 1

Address = 08h, Test Cell Header Byte- 1							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Test Cell Header Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	1

The "Read/ Write" bit- fields, within this register; along with those bit- fields within the "Test Cell Header Byte- 2 through - 4" registers; allows the user to define the header byte patterns for each of the 'test cells" that will be generated by the Test Cell Generator. This particular register allows the user to define the pattern for the first octet of these test cells.

3.3.2.10 Test Cell Header Byte- 2

Address = 09h, Test Cell Header Byte- 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Test Cell Header Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	1	0

The "Read/ Write" bit- fields, within this register; along with those bit- fields within the "Test Cell Header Byte- 1, - 3 and - 4" registers; allows the user to define the header byte patterns for each of the "test cells" that will be generated by the Test Cell Generator. This particular register allows the user to define the pattern for the second octet of these test cells.

3.3.2.11 Test Cell Header Byte- 3

Address = 0Ah, Test Cell Header Byte- 3							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Test Cell Header Byte 3							

Address = 0Ah, Test Cell Header Byte- 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	1	0	0	1	1

The "Read/Write" bit- fields, within this register; along with those bit- fields within the "Test Cell Header Byte - 1, - 2 and - 4" registers; allows the user to define the header byte patterns for each of the "test cells" that will be generated by the Test Cell Generator. This particular register allows the user to define the pattern for the third octet of these test cells.

3.3.2.12 Test Cell Header Byte- 4

Address = 0Bh, Test Cell Header Byte- 4							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Header Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	1	0	0

The "Read/Write" bit- fields, within this register; along with those bit- fields within the "Test Cell Header Byte- 1 through - 3" registers; allows the user to define the header byte patterns for each of the "test cells" that will be generated by the Test Cell Generator. This particular register allows the user to define the pattern for the fourth octet of these test cells.

3.3.2.13 Test Cell Error Accumulator-MSB

Address = 0Ch, Test Cell Error Accumulator-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Error-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

These "Reset- upon- Read" bit fields, along with those of the "Test Cell Error Accumulator-LSB" Register (Address = 0Dh), contains a 16- bit representation of the number of erred test cells that have been detected by the "Test Cell Receiver" since the last read of these registers. This register contains the upper- byte value for this 16- bit expression.

Note: The contents of these registers are valid only if the Test Cell Receiver has acquired "PRBS Lock" with the payload data of the test cells that it has received.

3.3.2.13 Test Cell Error Accumulator-LSB

Address = 0Dh, Test Cell Error Accumulator-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Error-Low Byte							

Address = 0Dh, Test Cell Error Accumulator-LSB							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

These "Reset- upon- Read" bit fields, along with those of the "Test Cell Error Accumulator-MSB" Register (Address = 0Ch), contains a 16- bit representation of the number of erred test cells that have been detected by the "Test Cell Receiver" since the last read of these registers. This register contains the lower- byte value for this 16- bit expression.

Note: The contents of these registers are valid only if the Test Cell Receiver has acquired "PRBS Lock" with the payload data of the test cells that it has received.

3.3.2.15 Rx E3 Configuration and Status Register- 1

Address = 0Eh, Rx E3 Configuration and Status Register- 1							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxPLDType[2:0]			RxFERF Algo	RxTMark Algo	RxPLDExp[2:0]		
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

Bit 7-5 RxPLDType[2:0] (Received Payload Type[2:0])

These three "Read- Only" bit- fields contain the "Payload Type" value within the MA byte of the most recently received E3 frame.

Note: The "Payload Type Msmatch" interrupt will be generated if the contents of these bit- fields differ from that of the "Expected Payload Types" in Bits 2 through 0 within this Register.

Bit 4 RxFERF Algo

This "Read/ Write" bit- field allows the user to select one of the two RxFERF Declaration Algorithms:

Writing a "0" to this bit- field selects the following "RxFERF Declaration" algorithm:

- The Receive E3 Framer declares a "Far End Receive Failure" (FERF) if the "FERF" bit- field, within the MA byte is set to "1" for 3 consecutive incoming E3 Frames. Likewise, the Receive E3 Framer will negate the "Far End Receive Failure" condition if the "FERF" bit- field, within the MA byte is set to "0" for 3 consecutive incoming E3 Frames.

Writing a "1" to this bit- field selects the following "RxFERF Declaration" algorithm:

- The Receive E3 Framer declares a "Far End Receive Failure" (FERF) if the "FERF" bit- field, within the MA byte is set to "1" for 5 consecutive E3 Frames. Likewise, the Receive E3 Framer will negate the "Far End Receive Failure" condition if the "FERF" bit- field, within the MA byte is set to "0" for 5 consecutive incoming E3 Frames.

Bit 3 Rx TMark Algorithm

This "Read/ Write" bit- field allows the user to select the number of consecutive incoming E3 frames, that the "Timing Marker" bit- field (within the MA byte- field) must be of a given logic state, before it is "validated" by the Receive E3 Framer. Once the Receive E3 Framer has "validated" the state of the "Timing Marker" bit- field, then it will write this logic state into Bit 1 (RxTMark) within the "Rx E3 Configuration & Status Register (Address = 0Fh).

Writing a "0" into this bit- field causes the Receive E3 Framer to "validate" the Timing Marker value after receiving 3 consecutive incoming E3 frames, with the "Timing Marker" bit- field of a given value. Writing a "1" into this bit- field causes the Receive E3 Framer to validate the "Timing Marker" value after receiving 5 consecutive incoming E3 frames, with the "Timing Marker" bit- field of a given value.

Bits 2-0: RxPLDExp[2:0]

This "Read/ Write" bit- field allows the user to specify the "Payload Type" that he/ she expects in the MA bytes, of each incoming E3 frame.

If the Receive E3 Framer detects a "Payload Type" that differs from the values within these bit- fields, then the UNI will generate the "Payload Type Mismatch" interrupt.

3.3.2.17 Rx E3 Status and Configuration Register- 2

Address = 0Fh, Rx E3 Configuration and Status Register- 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCF Algo	RxLCF	RxOOF	RxLOS	RxAIS	RxPLD Type UnStab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

Bit 7-RxLCF Algo (Loss of Frame Declaration Algorithm)

This "Read/ Write" bit- field allows the user to select the "LCF" (Loss of Frame) Declaration criteria, that will be used by the Receive E3 Framer. Writing a "0" to this bit- field configures the Receive E3 Framer to declare an LCF condition, after it has been in the OOF condition for 24 frame periods (3 ms). Writing a "1" to this bit- field configures the Receive E3 Framer to declare an LCF condition, after it has been in the OOF condition for 8 frame periods (1 ms).

Bit 6-RxLCF (Loss of Frame Declaration)

This "Read- Only" bit- field indicates whether or not the Receive E3 Framer is currently in the "Loss of Frame" (LCF) condition. If this bit- field is set to "1", then the Receive E3 Framer is currently in the LCF condition. Conversely, if this bit- field is set to "0", then the Receive E3 Framer is currently not in the LCF condition.

Bit 5-RxOOF (Out of Frame Declaration)

This "Read- Only" bit field indicates whether or not the Receive E3 Framer is currently experiencing an "Out of Frame" (OOF) condition. The Receive E3 Framer will declare an OOF condition if it has detected errors in the frame alignment bytes (FA1 and FA2) in four consecutive frames. If this bit- field is set to "1", then the Receive E3 Framer has declared, and is continuing to experience an OOF condition. If this bit- field is set to "0", then the Receive E3 Framer is currently not experiencing an OOF condition.

Bit 4-RxLOS (Loss of Signal Declaration)

This "Read- Only" bit- field indicates whether or not the Receive E3 Framer is currently experiencing a "Loss of Signal" (LOS) condition. The Receive E3 Framer will declare an LOS condition if it has detected a string of 32 consecutive "0s", via the RxPOS and RxNEG input pins. If this bit- field is set to "1", then the Receive E3 Framer has declared, and is continuing to experience an LOS condition. If this bit- field is set to "0", then the Receive E3 Framer is currently not experiencing an LOS condition.

Bit 3-RxAIS (Alarm Indication Status Declaration)

This "Read- Only" bit- field indicates whether or not the Receive E3 Framer is currently experiencing an "AIS" condition. The Receive E3 Framer will declare an AIS condition if it has detected two consecutive E3 frames, that each contain less than seven (7) "0s" . If this bit- field is set to "1", then the Receive E3 Framer has declared, and is continuing to experience an AIS condition. If this bit- field is set to "0", then the Receive E3 Framer is currently not experiencing an AIS condition.

Bit 2-RxPLDType UnStab

This "Read- Only" bit- field indicates whether or not the Receive E3 Framer has been receiving a consistent "Payload Type" value (within the MA Byte- Field); in the last 5 consecutive incoming E3 frames.

If the Receive E3 Framer has detected a change in the "Payload Type" value, within the last 5 incoming E3 frames, then it will set this bit- field to "1". If the "Payload Type" value has been consistent in the last 5 E3 frames, then the Receive E3 Framer will set this bit- field to "0".

Bit 1-Rx TMark

This "Read- Only" bit- field reflects the "most recently" validated "Timing Marker" value. The Receive E3 Framer will validate the "Timing Marker" state, after it has detected a "user- selectable" number of consecutive incoming E3 frames with a consistent "Timing Marker" value. The user makes this selection by writing the appropriate value to Bit 3 (RxTMarkAlgo) within the "Rx E3 Configuration/ Status Register (Address = 0Eh).

Bit 0-RxFERF (Far End Receive Failure)

This "Read- Only" bit- field indicates whether or not the Receive E3 Framer is experiencing an "FERF" (Far- End- Receive- Failure) condition. The Receive E3 Framer will declare a "FERF" condition, if it has received a "user- selectable" number of consecutive E3 frames, with the FERF bit- field (within the MA byte) set to "1". This "user- selectable" number is either 3 or 5 E3 frames. Conversely, the Receive E3 Framer will "negate" the FERF declaration, if it has received this "user- selectable" number of consecutive E3 frames, with the FERF bit- field set to "0".

If this bit- field is set to "1", then the Receive E3 Framer has declared an FERF condition. If this bit- field is set to "0", then the Receive E3 Framer has not declared an FERF condition.

Please see Sections 6.3.3.3.1 and 7.1.2.3.3, for a more detailed discussion on the meaning of the FERF bit- field, within the E3 frame.

3.3.2.18 Rx E3 Interrupt Enable Register- 1

Address = 10h, Rx E3 Interrupt Enable Register- 1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused			CCFA Interrupt Enable	COF Interrupt Enable	LCF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit 4-Change of Frame Alignment Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change of Frame Alignment" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt.

Bit 3-COF (Out of Frame) Interrupt Enable

This "Read/ Write" bit field allows the user to enable or disable the "Change in Out- of- Frame (COF) status" interrupt.

Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the "OOF" Condition, please see Section 7.1.2.2.1.

Bit 2-LOF (Loss of Frame) Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in Loss- of- Frame (LOF) status" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the "LOF" Condition, please see Section 7.1.2.2.1.

Bit 1-LOS (Loss of Signal) Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in LOS condition" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the LOS Condition, please see Section 7.1.2.3.1.

Bit 0-AIS Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in AIS condition" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the AIS Condition, please see Section 7.1.2.3.2.

3.3.2.19 Rx E3 Interrupt Enable Register- 2

Address = 11h, Rx E3 Interrupt Enable Register- 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	EM- Byte Error Interrupt Enable	Framing Byte Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	0	0	0	0	0	0	0

Bit 6-TTB Change Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in Trail Trace Buffer Message" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on Trail Trace Buffer messages, please see Section 7.1.2.7.

Bit 5-Received LAPD Message Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Received LAPD Message frame" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on this interrupt, please see Section 7.1.2.9.10.

Bit 4-FEBE (Far- End Block Error) Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Far- End- Block Error (FEBE)" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the "FEBE" Interrupt condition, please see Section 7.1.2.9.11.

Bit 3-FERF (Far- End Receive Failure) Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in FERF Condition" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on the "Change in FERF Condition" interrupt, please see Section 7.1.2.3.3.

Bit 2-EM Byte Error Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "EM Byte Error" interrupt. Setting this bit- field to

"1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on this interrupt, please see Section 7.1.2.9.5.

Bit 1-Framing Byte Error Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Framing Byte Error" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on this interrupt, please see Section 7.1.2.9.12.

Bit 0-Receive Payload Type Mismatch Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Receive Payload Type Mismatch" interrupt. Setting this bit- field to "1" enables this interrupt. Setting this bit- field to "0" disables this interrupt. For more information on this interrupt, please see Section 7.1.2.9.9.

3.3.2.20 Rx E3 Interrupt Status Register- 1

Address = 12h, Rx E3 Interrupt Status Register- 1							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Bit 4-COFA (Change of Frame Alignment) Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "Change of Frame Alignment" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "Change of Frame Alignment" interrupt if it has detected a change in frame alignment; in the incoming E3 frames.

Bit 3-OOF (Receive E3 Framer) Interrupt Status

This "Reset Upon Read" bit- field is set to "1" if the Receive E3 Framer has detected a "Change in the Out- of- Frame (OOF) Condition", since the last time this register was read. Therefore, this bit- field will be asserted under either of the following two conditions:

1. When the Receive E3 Framer has detected the appropriate conditions to declare an "OOF" Condition.
2. When the Receive E3 Framer has transitioned from the "OOF" Condition (Frame Acquisition Mode) into the "In- Frame" Condition (Frame Maintenance mode).

For more information of the OOF Condition, please see Section 7.1.2.2.1.

Bit 2-LOF (Loss of Frame) Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if a "Change in LOF Condition" interrupt has occurred since the last

read of this register.

The Receive E3 Framer will generate the "Change in LCF Condition" interrupt is response to either of the following two occurrences.

1. Whenever the Receive E3 Framer transitions from the "OOF Condition" state into the "LOF Condition" state, within the "E3 Framing Acquisition/ Maintenance" algorithm (per Figure 60).
2. Whenever the Receive E3 Framer transitions from the "FA1, FA2 Octet Verification" state to the "In- frame" state, within the "E3 Framing Acquisition/ Maintenance" algorithm (per Figure 60).

Bit 1-LOS (Loss of Signal) Interrupt Status

This "Reset Upon Read" bit will be set to "1", if the Receive E3 Framer has detected a

"Change in the LOS Status" condition, since the last time this register was read. This bit- field will be asserted under either of the following two conditions:

1. When the Receive E3 Framer detects the occurrence of an LOS Condition (e.g., the occurrence of 32 consecutive "spaces" in the incoming E3 data stream), and
2. When the Receive E3 Framer detects the end of an LOS Condition (e.g., when the Receive E3 Framer detects a string 32 bits that does not contain a string of four consecutive "0s").

The local µP can determine the current state of the LOS condition by reading bit 6 of the "Rx E3 Configuration and Status" Register (Address = 0Eh).

For more information in the "LOS of Signal (LOS) Alarm, please see Section 7.1.2.3.1.

Bit 0-AIS Interrupt Status

This "Reset Upon Read" bit field will be set to "1", if the Receive E3 Framer has detected a "Change in the AIS" condition, since the last time this register was read. This bit- field will be asserted under either of the following two conditions:

1. When the Receive E3 Framer first detects an AIS Condition in the incoming

Bit 4-Idle Condition Interrupt Status, and

2. When the Receive E3 Framer has detected the end of an "AIS Condition".

The local µP can determine the current state of the AIS condition by reading bit 7 of the "Rx E3 Configuration and Status" Register (Address = 0Eh).

For more information on the "AIS Condition" please see Section 7.1.2.3.2.

3.3.2.21 Rx E3 Interrupt Status Register- 2

Address = 13h, Rx E3 Interrupt Status Register- 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	EM- Byte Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Bit 6—TTB Change Interrupt Status (Receipt of New Trail Trace Buffer Message interrupt)

This "Reset- upon- Read" bit- field will be set to "1" if a "Receipt of New Trail Trace Buffer Message" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "Receipt of New Trail Trace Buffer Message" interrupt, if it receives an

E3 frame in which the value of the TR byte- field is of the form "1xxxxxxxb". A TR byte- field value of this form is identified as the "frame start marker".

Please see Section 7.1.2.9.8 for a more detailed discussion of this interrupt.

Bit 5-Received LAPD Message Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "Receipt of New LAPD Message frame" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate this "Receipt of New LAPD Message frame" interrupt when the LAPD Receiver has received a complete LAPD Message frame from the "Far- End" LAPD Transmitter.

Please see section 7.1.2.9.10 for a more detailed discussion of this interrupt.

Bit 4-FEBE (Far- End Block Error) Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "FEBE" (Far- End- Block Error) interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "FEBE" interrupt anytime it detects a "1" in the FEBE bit- field within an incoming E3 frame.

Please see Section 7.1.2.9.11 for a more detailed discussion of this interrupt.

Bit 3-FERF Interrupt Status

This "Reset Upon Read" bit will be set to '1' if the Receive E3 Framer has detected a "Change in the Rx FERF" Condition, since the last time this register was read.

This bit- field will be asserted under either of the following two conditions.

1. When the Receive E3 Framer first detects the occurrence of an Rx FERF Condition (all X- bits are set to '0').
2. When the Receive E3 Framer detects the end of the Rx FERF Condition (all X- bits are set to '0').

For more information on the Rx FERF (Yellow Alarm) condition, please see Section 7.1.2.3.3.

Bit 2-BIP (BIP- 8) Byte Error Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "BIP- 8 Error" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "BIP- 8 Error" interrupt if it has concluded that it has received an errored E3 frame, from the "Far- End" Terminal. Please see Section 7.1.2.9.5 for a more detailed discussion of this interrupt.

Bit 1-Framing Byte Error Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "Framing Byte Error" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "Framing Byte Error" interrupt if it has detected an error in the FA1 or FA2 bytes, on an incoming E3 frame. Please see Section 7.1.2.9.12 for a more detailed discussion of this interrupt.

Bit 0-Rx Pld Ms Interrupt Status

This "Reset- upon- Read" bit- field will be set to "1" if the "Payload Type Msmatch" interrupt has occurred since the last read of this register.

The Receive E3 Framer will generate the "Payload Type Msmatch" interrupt when it detects that the values, within the Payload Type bit- fields of the incoming E3 frame, has changed from that of the previous E3 frame. Please see

Section 7.1.2.9.9 for a more detailed discussion on this interrupt.

3.3.2.22 Rx NR Byte Register

Address = 14h, Rx NR Byte Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx NR Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the value of the NR byte, within the most recently received E3 frame. Please see Section 6.3.2.1.5 for a more detailed discussion on this register.

3.3.2.23 Rx GC Byte Register

Address = 15h, Rx GC Byte Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx GC Byte Register							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the value of the GC byte, residing in the most recently received E3 frame.

3.3.2.24 Rx TTB- 0 Register

Address = 16h, Rx TTB- 0 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 0							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the "frame start marker" byte of the 16 byte Trail Trace Buffer Message that has been received from the "Far- End" Terminal, via the TR byte- field within the incoming E3 frames. The remaining bytes, of this Trail Trace Buffer Message can be found in the "Rx TTB- 1" through "Rx TTB- 15" registers.

The data in this register is typically of the form [1, C6, C5, C4, C3, C2, C1, C0]. The "1" in the MSB position identifies this byte as being the "frame start marker" (e.g., the first byte within the 16 byte Trail Trace Buffer Message). The remaining bits: C0-C6 contain the CRC- 7 value that was calculated over the previous 16 byte Trail Trace Buffer Message.

Note: The XRT7234 E3 UNI will not compute or verify this CRC-7 value. It is up to the user's hardware and/or software to compute and verify this value.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.25 Rx TTB-1 Register

Address = 17h, Rx TTB-1 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte-1							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read-Only" register contains the second (2nd) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far-End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.26 Rx TTB-2 Register

Address = 18h, Rx TTB-2 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte-2							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read-Only" register contains the third (3rd) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far-End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.27 Rx TTB-3 Register

Address = 19h, Rx TTB-3 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte-3							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read-Only" register contains the fourth (4th) byte within the 16 byte Trail Trace Buffer Message, that has been

received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.28 Rx TTB- 4 Register

Address = 1Ah, Rx TTB- 4 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Receive Trail Trace Byte- 4							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the fifth (5th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.29 Rx TTB- 5 Register

Address = 1Bh, Rx TTB- 5 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Receive Trail Trace Byte- 5							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the sixth (6th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.30 Rx TTB- 6 Register

Address = 1Ch, Rx TTB- 6 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Receive Trail Trace Byte- 6							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the seventh (7th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.31 Rx TTB- 7 Register

Address = 1Dh, Rx TTB- 7 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 7							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the eighth (8th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.32 Rx TTB- 8 Register

Address = 1Eh, Rx TTB- 8 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 8							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the ninth (9th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.33 Rx TTB- 9 Register

Address = 1Fh, Rx TTB- 9 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 9							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the tenth (10th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.34 Rx TTB- 10 Register

Address = 20h, Rx TTB- 10 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 10							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the eleventh (11th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.35 Rx TTB- 11 Register

Address = 21h, Rx TTB- 11 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 11							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the twelfth (12th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.36 Rx TTB- 12 Register

Address = 22h, Rx TTB- 12 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 12							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the thirteenth (13th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.37 Rx TTB- 13 Register

Address = 23h, Rx TTB- 13 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 13							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the fourteenth (14th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.38 Rx TTB- 14 Register

Address = 24h, Rx TTB- 14 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 14							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the fifteenth (15th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.39 Rx TTB- 15 Register

Address = 25h, Rx TTB- 15 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Trail Trace Byte- 15							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This "Read- Only" register contains the sixteenth (16th) byte within the 16 byte Trail Trace Buffer Message, that has been received from the "Far- End" Terminal. This register typical contains an ASCII character that is required for the E.164 numbering format.

For more information on the use of this register, please see Section 7.1.2.7.

3.3.2.40 Rx E3 LAPD Control Register

Address = 26h, Rx E3 LAPD Control Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused						DL From NR	RxLAPD Enable
RO	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	0	0

Bit 1-DL From NR

This "Read/Write" bit-field allows the user to specify whether the LAPD Receiver should retrieve the bytes, comprising the incoming LAPD Message frame, from the NR byte-field, or from the GC byte-field, within each incoming E3 frame.

Writing a "1" configures the LAPD Receiver to retrieve the incoming LAPD Message frame octets from the NR byte-field, within each incoming E3 frame. Writing a "0" configures the LAPD Receiver to retrieve the incoming LAPD Message frame octets from the GC byte.

Bit 0-RxLAPD Enable

This "Read/Write" bit-field allows the user to enable or disable the LAPD Receiver, for reception of incoming LAPD Message frames from the "Far-End" LAPD Transmitter.

Writing a "1" to this bit-field enables the LAPD Receiver. Writing a "0" to this bit-field disables the LAPD Receiver.

3.3.2.41 Rx E3 LAPD Status Register

Address = 27h, Rx E3 LAPD Status Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused	Rx Abort	RxLAPD Type[1:0]		Rx CR Type	Rx FCS Error	End Of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Bit 6-Rx Abort

This "Read-Only" bit-field indicates whether or not the LAPD Receiver is currently detecting an abort sequence (e.g., a string of 7 consecutive "1s").

This bit-field is set to "1" if the LAPD Receiver is currently detecting an abort sequence in the incoming LAPD Channel. Conversely, this bit-field is set to "0" if the LAPD Receiver has not detected an abort sequence, since the last read of this register.

Bit 5, 4-RxLAPD Type[1:0]

These two "Read-Only" bit-fields combine to indicate the type and size of LAPD Message frame that has been

received by the LAPD Receiver. The following table relates the contents of these bit- fields to the LAPD Message type/ size.

RxLAPDType[1:0]	LAPD Message Frame Type	PMDL Message Size (Information Section)
00	Test Signal Identification Type	76 Bytes
01	Idle Signal Identification Type	76 Bytes
10	CL Path Identification Type	76 Bytes
11	ITU- T Path Identification Type	82 Bytes

Bit 3-Rx C/R Type

This "Read- Only" bit- field indicates the state of the C/ R bit- field, within octet # 2 of the most recently received LAPD Message frame.

Bit 2-Rx FCS Error

This "Read- Only" bit- field indicates whether or not the LAPD Receiver has detected an FCS (Frame Check Sequence) error, in the most recently received LAPD Message frame. This bit- field is set to "0" if the LAPD Receiver does not detect an FCS error in this LAPD Message frame. Conversely, this bit- field is set to "1" if the LAPD Receiver does detect an FCS error in this LAPD Message frame.

For a more detailed discussion on the LAPD Receiver's handling of the FCS bytes, please see Section 7.1.2.5.

Bit 1-EndOfMessage

The LAPD Receiver will assert this "read- only" bit- field, when it has received a complete LAPD Message frame. This bit- field, along with the "Receipt of New LAPD Message frame" interrupt, serves to inform the local µP that the "Receive LAPD Message" buffer contains a new PMDL message that needs to be read and processed.

This bit- field is cleared (to "0") upon reading this register.

Bit 0-Flag Present

The LAPD Receiver will assert this "read- only" bit- field when it is currently detecting the Flag Sequence octet (7Eh) in the incoming LAPD channel (e.g., either the GC or the NR byte- field, within each E3 frame). The LAPD Receiver will negate this bit- field when it is no longer receiving the Flag Sequence octet in the incoming LAPD channel.

3.3.2.42 Tx E3 Configuration Register

Address = 28h, Tx E3 Configuration Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re- transmit	TxLAPD Type[1:0]		DLinNR	NoData Link	TxAIS Enable	TxLOS Enable	MARx
R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
1	0	0	0	1	0	0	1

Bit 7-Auto Retransmit

This "Read/ Write" bit- field allows the user to configure the LAPD Transmitter to either transmit the LAPD Message frame only once; or repeatedly at one- second intervals.

Writing a "0" to this bit- field configures the LAPD Transmitter to transmit the LAPD Message frame once. Afterwards,

the LAPD Transmitter will halt transmission, until it has commanded to transmit another LAPD Message frame.

Writing a "1" to this bit- field configures the LAPD Transmitter to transmit the LAPD Message frame repeatedly at one second intervals. In this configuration, the LAPD Transmitter will repeat its transmission of the LAPD Message frame until it has been disabled.

Bit 6, 5–TxLAPD Type[1:0]

These two "Read/ Write" bit- fields allow the user to specify the type and size of LAPD Message frame that he/ she wishes to transmit to the "Far End" LAPD Receiver. The following table relates the contents of these bit- fields to the LAPD Message frame type.

TxLAPDType[1:0]	LAPD Message Frame Type	PMDL Message Size (Information Section)
00	Test Signal Identification Type	76 Bytes
01	Idle Signal Identification Type	76 Bytes
10	CL Path Identification Type	76 Bytes
11	ITU- T Path Identification Type	82 Bytes

Bit 4-BLinNR

This "Read/ Write" bit- field allows the user to specify whether the LAPD Transmitter should insert the "out- bound" LAPD Message frame octets into the NR byte- field, or in the GC- byte- field, within each outbound E3 frame.

Writing a "1" configures the LAPD Transmitter to insert the octets of the outbound LAPD Message frame into the NR byte- field, within each outbound E3 frame. Writing in "0" configures the LAPD Transmitter to insert the octets of the outbound LAPD Message frame into the GC byte- field, within each outbound E3 frame.

Bit 3-No Data Link (LAPD Transmitter Enable/ Disable)

This "Read/ Write" bit- field allows the user to enable or disable the LAPD Transmitter.

Writing a "1" to this bit- field causes the LAPD Transmitter to be disabled, and to not insert any outbound LAPD Mes- sage frame octets into the NR or GC byte- fields of the outbound E3 frames. Conversely, writing a "0" to this bit- field enables the LAPD Transmitter, for transmission of any outbound LAPD Message frames.

Bit 2–TxAIS Enable

This "Read/ Write" bit- field allows the user to command the Transmit E3 Framer to transmit an AIS pattern, upon demand.

Writing a "0" to this bit- field allows the Transmit E3 Framer to transmit internally generated data (e.g., the ITU- T G832 compatible E3 frames with ATM cell data) to the "Far- End" Terminal. Writing a "1" to this bit- field causes the Transmit E3 Framer to transmit an all "1s" pattern to the "Far- End" Terminal.

Note: If the Transmit E3 Framer is transmitting an AIS pattern to the "Far- End" Terminal, then it is not transmit- ting any E3 frames or ATM cell data. Consequently, if this command is invoked, the "Far End" Terminal will experience an "OOF" (Out of Frame) and an "LCD" (Loss of Cell Delineation) Condition.

Bit 1–TxLOS Enable

This "Read/ Write" bit- field allows the user to command the Transmit E3 Framer to transmit an LOS pattern, upon demand.

Writing a "0" to this bit- field allows the Transmit E3 Framer to transmit internally generated data (e.g., the ITU- T G832 compatible E3 frames with ATM cell data) to the "Far- End" Terminal. Writing a "1" to this bit- field causes the Transmit E3 Framer to transmit an "All 0s" pattern to the "Far- End" Terminal.

Note: If the Transmit E3 Framer is transmitting an LOS pattern to the Far- End Terminal, then it is not transmit- ting any E3 frames or ATM cell data. Consequently, the "Far- End" Terminal will experience an "LOS" (Loss of Signal), "OOF" (Out of Frame), and an "LCD" (Loss of Cell Delineation) condition.

Bit 0 - MARx (FERF and FEBE bit- field Loopback)

This "Read/ Write" bit- field allows the user to specify whether the value of the FERF and FEBE bit- fields, in the "outbound" E3 frames; should be based upon "Receive E3 Framer" conditions or upon the content of the "Tx MA Byte" register (Address = 2Ah).

FERF and FEBE values are based upon Receive E3 Framer Conditions

If the user selects "Receive E3 Framer" conditions, then the Transmit E3 Framer will set and clear the FERF and FEBE bit- fields in response to the following conditions.

- FERF Bit- field
  - If the Receive E3 Framer (on the same UNI chip) is currently experiencing an LOS, AIS, or LCF condition, then the Transmit E3 Framer will set the "FERF" bit- field (in the "outbound" E3 frame) to "1". Conversely, if the Receive E3 Framer is not experiencing any of these conditions, then the Transmit E3 Framer will set the FERF bit- field (in the "outbound" E3 frame) to "0".

- FEBE Bit- field
  - If the Receive E3 Framer detects a BIP- 8 error in the "incoming" E3 frame, then the Transmit E3 Framer will set the "FEBE" bit- field (in the "outbound" E3 frame) to "1". Conversely, if the Receive E3 Framer does not detect a BIP- 8 error in the "incoming" E3 frame, then the Transmit E3 Framer will set the "FEBE" bit- field (in the E3 "outbound" E3 frame) to "0".

FEBE and FERF values are based upon the contents of the "Tx MA Byte" register

If the user selects the contents of the "Tx MA Byte" register, then whatever value has been written into bit 7 (FERF), within the "Tx MA Byte" register (Address = 2Ah); will be the value of the "FERF" bit- field, in the "outbound" E3 frame. Likewise, whatever value has been written into Bit 6 (FEBE) within the "Tx MA Byte" register, will be the value of the "FEBE" bit- field, in the "outbound" E3 frame.

Writing a "1" into Bit 0 (MAx) within the "Tx E3 Configuration" register configures the Transmit E3 Framer to set the "FERF" and "FEBE" bit- fields (in the "outbound" E3 frames) to values based upon "Receive E3 Framer" conditions. Writing a "0" into this bit- field configures the Transmit E3 Framer to set the "FEBE" and "FEBE" bit- fields (in the "outbound" E3 frames) to the values written into bit- fields 6 and 7 within the "Tx MA Byte" register.

3.3.2.43 Tx GC Byte Register

Address = 29h, Tx GC Byte Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit GC Byte							
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 "Read/ Write" byte- field allows the user to specify the contents of the GC byte- field in each outbound E3 frame.

Note: The contents of this register is ignored, if the LAPD Transmitter is enabled and has been configured to insert the comprising octets of an "outbound" LAPD Message frame into the GC byte- field of each outbound E3 frame

(e.g., if DLinNR = "0").

3.3.2.44 Tx MA Byte Register

Address = 2Ah, Tx MA Byte Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit MA Byte							
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker
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 "Read/Write" byte-fields allows the user to specify the contents of the MA byte-field in each outbound E3 frame.

Note: The values written into bit-fields 6 (FEBE) and 7 (FERF) are inserted into "outbound" E3 frames, only if bit-field 0 (MAx) within the "Tx E3 Configuration" Register (Address = 28h) is set to "0". Otherwise, the Transmit E3 Framer will set the FERF and FEBE values, within each "outbound" E3 frame, to values based upon "Receive E3 Framer" conditions.

3.3.2.45 Tx NR Byte Register

Address = 2Bh, Tx NR Byte Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit NR Byte							
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 "Read/Write" byte-field allows the user to specify the contents of the NR byte-field in each outbound E3 frame.

Note: The contents of this register is ignored, if the LAPD Transmitter is enabled and has been configured to insert the comprising octets of an "outbound" LAPD Message frame into the NR byte-field of each outbound E3 frame (e.g., if DLinNR = "1").

3.3.2.46 Tx TTB-0 Register

Address = 2Ch, Tx TTB-0 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte-0							

Address = 2Ch, Tx TTB- 0 Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	0	0	0

This "Read/Write" byte- field, along with the "Tx TTB- 1" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the first of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

This particular byte- field should contain the pattern "[1, C6, C5, C4, C3, C2, C1, C0]" where C6 through C0 are the results of a CRC- 7 calculation over the previous 16- byte frame.

Note: The XR- T7234 E3 UNI will not compute this CRC- 7 value. It is up to the user's hardware and/or software to compute this value, prior to writing it into this register.

3.3.2.47 Tx TTB- 1 Register

Address = 2Dh, Tx TTB- 1 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 1							
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 "Read/Write" byte- field, along with the "Tx TTB- 0" and "Tx TTB- 2" through "Tx TTB- 15" register allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the second of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 2 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.48 Tx TTB- 2 Register

Address = 2Eh, Tx TTB- 2 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 2							
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 "Read/Write" byte- field, along with the "Tx TTB- 0", "Tx TTB- 1" and "Tx TTB- 3" through "Tx TTB- 15" register allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the third of a set of 16 E3 Frames, the Transmit E3

Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1, and Tx TTB- 3 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.49 Tx TTB- 3 Register

Address = 2Fh, Tx TTB- 3 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte- 3							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0" through "Tx TTB- 2" and "Tx TTB- 4" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the fourth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1, Tx TTB- 2 and Tx TTB- 4 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.50 Tx TTB- 4 Register

Address = 30h, Tx TTB- 4 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte- 4							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 3" and "Tx TTB- 5" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the fifth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 3 and Tx TTB- 5 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.51 Tx TTB- 5 Register

Address = 31h, Tx TTB- 5 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 5							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 4" and "Tx TTB- 6" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the sixth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 4 and Tx TTB- 6 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.52 Tx TTB- 6 Register

Address = 32h, Tx TTB- 6 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte - 6							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 5" and "Tx TTB- 7" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the seventh of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 5 and Tx TTB- 7 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.53 Tx TTB- 7 Register

Address = 33h, Tx TTB- 7 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 7							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 6" and "Tx TTB- 8" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the eighth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 6 and Tx TTB- 8 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.54 Tx TTB- 8 Register

Address = 34h, Tx TTB- 8 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte- 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

This "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 7" and "Tx TTB- 9" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the ninth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 7 and Tx TTB- 9 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.55 Tx TTB- 9 Register

Address = 35h, Tx TTB- 9 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte- 9							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 8" and "Tx TTB- 10" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the tenth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 8 and Tx TTB- 10 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.56 Tx TTB- 10 Register

Address = 36h, Tx TTB- 10 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 10							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 9" and "Tx TTB- 11" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the eleventh of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 9 and Tx TTB- 11 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.57 Tx TTB- 11 Register

Address = 37h, Tx TTB- 11 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 11							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 10" and "Tx TTB- 12" through "Tx TTB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the twelfth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 10 and Tx TTB- 12 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.58 Tx TTB- 12 Register

Address = 38h, Tx TTB- 12 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 12							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address = 38h, Tx TIB- 12 Register							
0	0	0	0	0	0	0	0

This "Read/ Write" byte- field, along with the "Tx TIB- 0 through Tx TIB- 11" and "Tx TIB- 13" through "Tx TIB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the thirteenth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TIB- 1 through Tx TIB- 11 and Tx TIB- 13 through Tx TIB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.59 Tx TIB- 13 Register

Address = 39h, Tx TIB- 13 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte - 13							
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 "Read/ Write" byte- field, along with the "Tx TIB- 0 through Tx TIB- 12", "Tx- TIB- 14", and "Tx TIB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the fourteenth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TIB- 1 through Tx TIB- 12, Tx TIB- 14 and Tx TIB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.60 Tx TIB- 14 Register

Address = 3Ah, Tx TIB- 14 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit Trail Trace Buffer Byte - 14							
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 "Read/ Write" byte- field, along with the "Tx TIB- 0 through Tx TIB- 13" and "Tx TIB- 15" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the fifteenth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TIB- 1 through Tx TIB- 13 and Tx TIB- 15 are used to transmit 15

ASCII characters required for the E.164 numbering format.

3.3.2.61 Tx TTB- 15 Register

Address = 3Bh, Tx TTB- 15 Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Trail Trace Buffer Byte- 15							
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 "Read/ Write" byte- field, along with the "Tx TTB- 0 through Tx TTB- 14" registers allows a user to define a "Trail Access Point Identifier" sequence of bytes, that will be transmitted to the "Far- End" Terminal. The "Far- End" Receiving Terminal will use this sequence of bytes to verify that it is connected to the proper "Transmitting Terminal". The Transmit E3 Framer will take the contents of these 16 registers, and insert them into the TR byte of the outbound E3 frame. In the sixteenth of a set of 16 E3 Frames, the Transmit E3 Framer will read in the contents of this register, and insert it into the TR byte- field, within the very next outbound E3 frame.

The contents of this register, along with Tx TTB- 1 through Tx TTB- 15 are used to transmit 15 ASCII characters required for the E.164 numbering format.

3.3.2.62 Tx FA1 Error Mask Register

Address = 3Ch, Tx FA1 Error Mask Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit FA1 Byte Error Mask							
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 "Read/ Write" bit- field allows the user to insert errors into the Framing Alignment octet, FA1 of each "outbound" E3 frame. The user may wish to do this for equipment testing purposes. Prior to transmission, the Transmit E3 Framer reads in the FA1 byte, and performs an XOR operation with it and the contents of this register. The results of this operation are written back into the FA1 octet position, in each outbound E3 frame. Consequently, if the user does not wish to inject errors into the FA1 octet of the "outbound" E3 frames, he/ she must insure that the contents of this register are set to all "0s" (the default value).

3.3.2.63 Tx FA2 Error Mask Register

Address = 3Dh, Tx FA2 Error Mask Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit FA2 Byte Error Mask							
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 "Read/ Write" bit- field allows the user to insert errors into the Framing Alignment octet, FA2 of each "outbound" E3 frame. The user may wish to do this for equipment testing purposes. Prior to transmission, the Transmit E3 Framer reads in the FA2 byte, and performs an XOR operation with it and the contents of this register. The results of this operation are written back into the FA2 octet position, in each outbound E3 frame. Consequently, if the user does not wish to inject errors into the FA2 octet of the "outbound" E3 frames, he/ she must insure that the contents of this register are set to all "0s" (the default value).

3.3.2.64 Tx EM Byte Error Mask Register

Address = 3Eh, Tx EM Byte Error Mask Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Transmit EM Byte Error Mask							
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 "Read/ Write" bit- field allows the user to insert errors into EM (Error Monitor) octet of each "outbound" E3 frame. The user may wish to do this for equipment testing purposes. Prior to transmission, the Transmit E3 Framer reads in the EM byte, and performs an XOR operation with it and the contents of this register. The results of this operation are written back into the EM octet position, in each outbound E3 frame. Consequently, if the user does not wish to inject errors into the EM octet of the "outbound" E3 frames, he/ she must insure that the contents of this register are set to all "0s" (the default value).

3.3.2.65 Tx E3 LAPD Status/ Interrupt Register

Address = 3Fh, Tx E3 LAPD Status/ Interrupt Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused				TxDL Start	TxDL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
RO	RO	RO	RO	R/ W	RO	R/ W	RUR
0	0	0	0	0	0	0	0

Bit 3-TxDL Start

This "Read/ Write" bit- field allows the user to command the LAPD Transmitter to do the following.

- Scan through the PMDL Message, within the "Transmit LAPD Message" buffer, and search for a string of five (5) consecutive "1s". The LAPD Transmitter will then insert (or "stuff") a "0" into the PMDL Message data, immediately following any string of 5 consecutive "1s".
- Read in this "stuffed" PMDL Message from the "Transmit LAPD Message" buffer, and encapsulate it into a LAPD

Message frame.

- Fragment the resulting LAPD Message frame into octets.
- Insert these octets into either the GC byte- field or the NR byte- field (depending upon the user's selection) in each "outbound" E3 frame.

A "0" to "1" transition, in this bit- field commands the LAPD Transmitter to initiate the "above- mentioned" procedure.

- Once the user has commanded the LAPD Transmitter to start transmission, the LAPD Transmitter will repeat the "above- mentioned" process once each second; and will insert flag sequence octets into the "outbound" LAPD channel, during the idle periods between transmissions.

Bit 2—TxDL Busy

This "Read- Only" bit- field allows the user to poll or monitor the status of the LAPD Transmitter to see if it has completed its transmission of the LAPD Message frame. The LAPD Transmitter will set this bit- field to "1", while it is in the process of transmitting the LAPD Message frame. However, the LAPD Transmitter will clear this bit- field to "0" once it has completed its transmission of the LAPD Message frame.

Bit 1–TxLAPD Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "LAPD Message frame Transmission Complete" interrupt.

Writing a "0" to this bit- field disables this interrupt. Writing a "1" to this bit- field enables this interrupt.

Bit 0–TxLAPD Interrupt Status

This "Reset- upon- Read" bit- field allows the user to determine if the "LAPD Message Frame Transmission Complete" interrupt has occurred since the last read of this register. If this bit- field contains a "1" then the "LAPD Message Frame Transmission Complete" interrupt has occurred since the last read of this register. Conversely, if this bit- field contains a "0" then it has not.

3.3.2.66 PMON LCV Event Count Register-MSB

Address = 40h, PMON LCV Event Count Register-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LCV Count–High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON LCV Event Count Register-LSB" (Address = 41h) contains a 16- bit representation of the number of "Line Code Violations" that have been detected by the Receive E3 Framer, since the last read of these registers. This register contains the MSB (or Upper- Byte) value of this 16 bit expression.

3.3.2.67 PMON LCV Event Count Register-LSB

Address = 41h, PMON LCV Event Count Register-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LCV Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON LCV Event Count Register - MSB (Address 40h), contain a 16 bit representation of the number of "Line Code Violations" that have been detected by the Receive E3 Framer, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16 bit expression.

3.3.2.68 PMON Framing Byte Error Event Count Register-MSB

Address = 42h, PMON Framing Byte Error Event Count Register - MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Framing Byte Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Framing Byte Error Event Count Register-LSB" (Address = 43h) contains a 16 bit representation of the number of "Framing Byte Errors" (e.g., in the FA1 and FA2 octets) that have been detected by the Receive E3 Framer, since the last read of these registers. This register contains the MSB (or Upper Byte) value of this 16 bit expression.

Note: If the user is interfacing the local µP/µC to the Mcroprocessor Interface of the UNI chip over an 8 bit Data Bus; then immediately after reading this register, the local µP/µC can also read the contents of the "PMON Framing Bit Error Event Count Register-LSB" by reading the "PMON Holding Register" (Address = 56h).

3.3.2.69 PMON Framing Byte Error Event Count Register-LSB

Address = 43h, PMON Framing Byte Error Event Count Register-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Framing Byte Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Framing Byte Error Event Count Register-MSB" (Address = 42h) contains a 16 bit representation of the number of "Framing Bytes Errors" (e.g., in the FA1 and FA2 octets) that have been detected by the Receive E3 Framer, since the last read of these registers. This register contains the LSB (or Lower Byte) value of this 16 bit expression.

Note: If the user is interfacing the local µP/µC to the Mcroprocessor Interface of the UNI chip over an 8 bit Data Bus; then immediately after reading this register, the local µP/µC can also read the contents of the "PMON Framing Bit Error Event Count Register-MSB" by reading the "PMON Holding Register" (Address = 56h).

3.3.2.70 PMON Received FEBE Event Count Register-MSB

Address = 44h, PMON Received FEBE Event Count Register-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Received FEBE Event Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the PMON Received FEBE Event Count Register-LSB" (Address = 45h) contains a 16 bit representation of the number of number of E3 frames (received by the Receive E3 Framer) that contain a "1" in the FEBE bit- field; since the last read of these registers. This register contains the MSB (or Upper Byte) value of this 16 bit expression.

3.3.2.71 PMON Received FEBE Event Count Register-LSB

Address = 45h, PMON Received FEBE Event Count Register-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Received FEBE Event Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the PMON Received FEBE Event Count Register - MSB" (Address = 44h) contains a 16 bit representation of the number of number of E3 frames (received by the Receive E3 Framer) that contain a "1" in the FEBE bit- field; since the last read of these registers. This register contains the LSB (or Lower Byte) value of this 16 bit expression.

3.3.2.72 PMON Framer BIP- 8 Error Event Count Register-MSB

Address = 46h, PMON Framer BIP- 8 Error Event Count Register-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Framer BIP- 8 Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Framer BIP- 8 Error Event Count Register-LSB" (Address = 47h) contains a 16- bit representation of the number of E3 frames, containing EM byte errors, that have been detected by the Receive E3 Framer, since the last read of these registers. This registers contains the MSB (or Upper Byte) value of this 16- bit expression.

3.3.2.73 PMON Framer BIP- 8 Error Event Count Register-LSB

Address = 47h, PMON Framer BIP- 8 Error Event Count Register-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Framer BIP- 8 Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Framer BIP- 8 Error Event Count Register-MSB" (Address = 46h) contains a 16- bit representation of the number of E3 frames, containing EM byte errors, that have been detected by the Receive E3 Framer, since the last read of these registers. This registers contains the LSB (or Lower Byte) value of this 16- bit expression.

3.3.2.74 PMON Received Single- Bit HEC Error Count-MSB

Address = 48h, PMON Received Single- Bit HEC Error Count-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
S- HEC Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Single HEC Error Count-LSB" register (Address = 49h) contains a 16 bit representation of the number of "Single bit HEC Errors" that have been detected by the Receive Cell Processor, since the last read of these registers. This register contains the MSB (or Upper byte) value of this 16- bit expression.

3.3.2.75 PMON Received Single- Bit HEC Error Count-LSB

Address = 49h, PMON Received Single- Bit HEC Error Count-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
S- HEC Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Single HEC Error Count-MSB" register (Address = 48h) contains a 16 bit representation of the number of "Single bit HEC Errors" that have been detected by the Receive Cell Processor, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16- bit expression.

3.3.2.76 PMON Received Multiple- Bit HEC Error-MSB

Address = 4Ah, PMON Received Multiple- Bit HEC Error-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
M- HEC Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Multiple HEC Error Count-LSB" register (Address = 4Bh) contains a 16 bit representation of the number of "Multiple- bit HEC Errors" that have been detected by the Receive Cell Processor, since the last read of these registers. This register contains the MSB (or Upper byte) value of this 16- bit expression.

3.3.2.77 PMON Received Multiple- Bit HEC Error-LSB

Address = 4Bh, PMON Received Multiple- Bit HEC Error-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
M- HEC Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Multiple HEC Error Count - MSB" register (Address = 4Ah) contains a 16 bit representation of the number of "Multiple- bit HEC Errors" that have been detected by the Receive Cell Processor, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16- bit expression.

3.3.2.78 PMON Received Idle Cell Count-MSB

Address = 4Ch, PMON Received Idle Cell Count-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Idle Cell Count-LSB" register (Address = 4Dh) contains a 16 bit representation of the number of "Idle Cells" that have been detected by the Receive Cell

Processor, since the last read of these register. This register contains the MSB (or Upper Byte) value of this 16- bit expression.

3.3.2.79 PMON Received Idle Cell Count-LSB

Address = 4Dh, PMON Received Idle Cell Count-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Idle Cell Count-MSB" register (Address = 4Ch) contains a 16 bit representation of the number of "Idle Cells" that have been detected by the Receive Cell Processor, since the last read of these register. This register contains the LSB (or Lower Byte) value of this 16- bit expression.

3.3.2.80 PMON Received Valid Cell Count-MSB

Address = 4Eh, PMON Received Valid Cell Count-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Valid Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Valid Cell Count-LSB" register (Address = 4Fh) contains a 16 bit representation of the number of "User (or Assigned) Cells" that have been detected by the Receive Cell Processor, since the last read of these register. This register contains the MSB (or Upper Byte) value of this 16- bit expression.

3.3.2.81 PMON Received Valid Cell Count-LSB

Address = 4Fh, PMON Received Valid Cell Count-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Valid Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Received Valid Cell Count-MSB" register (Address = 4Eh) contains a 16 bit representation of the number of "User (or Assigned) Cells" that have been detected by the Receive Cell Processor, since the last read of these register. This register contains the LSB (or Lower Byte) value of this 16- bit expression.

3.3.2.82 PMON Discarded Cell Count-MSB

Address = 50h, PMON Discarded Cell Count-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Cell Drop Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Discarded Cell Count- LSB" register (Address = 51h) contains a 16 bit representation of the number of cells that have been discarded by the Receive Cell Processor, since the last read of these registers. This register contains the MSB (or Upper byte) value of this 16 bit expression.

Please note that this expression includes Idle cells, cells with HEC byte errors, and cells filtered or removed by the User Cell Filter.

3.3.2.83 PMON Discarded Cell Count-LSB

Address = 51h, PMON Discarded Cell Count-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Cell Drop Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Discarded Cell Count- MSB" register (Address = 50h) contains a 16 bit representation of the number of cells that have been discarded by the Receive Cell Processor, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16 bit expression.

Please note that this expression includes Idle cells, cells with HEC byte errors, and cells filtered or removed by the User Cell Filter.

3.3.2.84 PMON Transmitted Idle Cell Count-MSB

Address = 52h, PMON Transmitted Idle Cell Count-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Idle Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Transmitted Idle Cell Count-LSB" register (Address = 53h)

contains a 16- bit representation of the number of "Idle Cells" that have been generated and transmitted by the Transmit Cell Processor, since the last read of these registers. This register contains the MSB (or Upper byte) value of this 16 bit expression.

3.3.2.85 PMON Transmitted Idle Cell Count-LSB

Address = 53h, PMON Transmitted Idle Cell Count-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Idle Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Register- upon- Read" register, along with the "PMON Transmitted Idle Cell Count-MSB" register (Address = 52h) contains a 16- bit representation of the number of "Idle Cells" that have been generated and transmitted by the Transmit Cell Processor, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16 bit expression.

3.3.2.86 PMON Transmitted Valid Cell Count-MSB

Address = 54h, PMON Transmitted Valid Cell Count-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Valid Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Transmitted Valid Cell Count-LSB" register (Address = 55h) contains a 16- bit representation of the number of "User (or Assigned) Cells" that have been generated and transmitted by the Transmit Cell Processor, since the last read of these registers. This register contains the MSB (or Upper byte) value of this 16 bit expression.

3.3.2.87 PMON Transmitted Valid Cell Count-LSB

Address = 55h, PMON Transmitted Valid Cell Count-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Valid Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

This "Reset- upon- Read" register, along with the "PMON Transmitted Valid Cell Count-MSB" register (Address = 54h) contains a 16- bit representation of the number of "User (or Assigned) Cells" that have been generated

and transmitted by the Transmit Cell Processor, since the last read of these registers. This register contains the LSB (or Lower byte) value of this 16 bit expression.

3.3.2.88 PMON Holding Register

Address = 56h, PMON Holding Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
PMON Hold Value							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register is of use if the user is operating the UNI in the 8- bit µP Access Mode. When the µP reads out a particular PMON Counter, One Second Accumulator, or Test Cell Error Accumulator (16 bit registers), it will read out one of two 8- bit registers. The contents of the other 8- bit register will be stored in this register.

For more information on this operation, please see Section 3.5.

3.3.2.89 One Second Error Status Register

Address = 57h, One Second Error Status Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused						Errored Sec	Severe Errored Sec
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Bit 1-Errored Second

This "Read- Only" bit- field indicates whether or not there were any errors during the last one second interval. If this bit- field is "0" then there were no errors during the last one second interval. If this bit- field is "1", then there was at least one error during the last one second interval.

Bit 0-Severe Errored Second

This "Read- Only" bit- field indicates whether or not the bit- error rate, of the last one second interval, exhibited a BER (bit error rate) exceeding 10- 3.

A "0" in this bit- field indicates that the BER for the last one- second interval was less than 10- 3.

Conversely, a "1" in this bit- field indicates that the BER for the last one- second interval exceeds 10- 3.

3.3.2.90 LCV- One Second Accumulator Register-MSB

Address = 58h, LCV- One Second Accumulator Register-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*

Address = 58h, LCV- One Second Accumulator Register-MSB							
LCV 1 Sec-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "LCV- One Second Accumulator Register-LSB" (Address = 59h) presents a 16- bit representation of the number of Line Code Violations that have been detected by the Receive E3 Framer, during the last one second interval. This register presents the MSB (Upper- byte) value of this expression.

3.3.2.91 LCV- One Second Accumulator Register-LSB

Address = 59h, LCV- One Second Accumulator Register-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LCV 1 Sec-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "LCV- One Second Accumulator Register-LSB" (Address = 58h) presents a 16- bit representation of the number of Line Code Violations that have been detected by the Receive E3 Framer, during the last one second interval. This register presents the LSB (Lower- byte) value of this expression.

3.3.2.92 Framer BIP- 8 Error-One Second Accumulator Register-MSB

Address = 5Ah, Framer BIP- 8 Errors-One Second Accumulator Register-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Framer BIP- 8 Error 1 Sec-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "Frame BIP- 8 Errors-One Second Accumulator Register-LSB" (Address = 5Bh) presents a 16- bit representation of the number of E3 frames, containing EM byte errors, that have been detected by the Receive E3 Framer, during the last one second interval. This register presents the MSB (Upper- byte) value of this expression.

3.3.2.93 Framer BIP- 8 Errors-One Second Accumulator Register-LSB

Address = 5Bh, Framer BIP- 8 Errors-One Second Accumulator Register-LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*

Address = 5Bh, Framer BIP- 8 Errors-One Second Accumulator Register-LSB							
Framer BIP- 8 Error 1 Sec-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "Frame BIP- 8 Errors-One Second Accumulator Register - MSB" (Address = 5Ah) presents a 16- bit representation of the number of E3 frames, containing EM byte errors, that have been detected by the Receive E3 Framer, during the last one second interval. This register presents the LSB (Lower- byte) value of this expression.

3.3.2.94 HEC Errors-One Second Accumulator Register-MSB

Address = 5Ch, HEC Errors-One Second Accumulator Register-MSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
HEC Errors 1 Sec-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "HEC Errors-One Second Accumulator Register-LSB" (Address = 5Dh) presents a 16- bit representation of the number cells with HEC errors that have been detected by the Receive Cell Processor, during the last one second interval. This register presents the MSB (Upper- byte) value of this expression.

3.3.2.95 HEC Errors-One Second Accumulator Register-LSB

Address = 5Dh, HEC Errors - One Second Accumulator Register - LSB							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
HEC Errors 1 Sec-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

This register, along with "HEC Errors-One Second Accumulator Register-MSB" (Address = 5Ch) presents a 16- bit representation of the number cells with HEC errors that have been detected by the Receive Cell Processor, during the last one second interval. This register presents the LSB (Lower- byte) value of this expression.

3.3.2.96 Rx CP Configuration Register

Address = 5Eh, Rx CP Configuration Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LCD	RDPChkPat	RDPChk Pat Enable	C Discard	OAM Check Bit	De-Scramblel Enable	Rx Coset Enable	HEC Error Ignore\ Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address = 5Eh, Rx CP Configuration Register							
1	0	0	1	1	1	1	0

Bit 7-LCD (Loss of Cell Delineation)

This "Read Only" bit- field indicates whether or not the Receive Cell Processor is currently experiencing a "Loss of Cell Delineation".

If this bit- field is "0", then the Receive Cell Processor is currently NOT experiencing a "Loss of Cell Delineation" and is properly delineating the ATM cell data that it receives from the Receive E3 Framer.

If this bit- field is "1", then the Receive Cell Processor is currently experiencing a "Loss of Cell Delineation" and is NOT properly delineating the ATM cell data that it receives from the Receive E3 Framer.

For more information on Cell Delineation by the Receive Cell Processor, please see Section 7.2.2.1.

Bit 6-RDPChk (Receive "Data Path Integrity Check") Pattern

The "Read/ Write" bit- field allows the user to select which of two possible "Data Path Integrity Check" patterns that the Receive Cell Processor will insert into the fifth octet of each cell that is written into the RxFIFO.

The "Data Path Integrity Check" pattern options are:

- An alternating pattern of "55h"/ "AAh".
- A constant pattern of "55h".

Writing a "0" to this bit- field selects the alternating pattern. Writing a "1" to this bit- field selects the constant pattern.

Note: This bit- field is ignored if Bit 5 (of this register) is set to "0".

Bit 5-RDPChk (Receive "Data Path Integrity Check") Pattern Enable

This "Read/ Write" bit- field allows the user to enable or disable the insertion of the "Data Path Integrity Check" pattern into the 5th octet of each cell that is written into the RxFIFO.

Writing a "0" into this bit- field disables the insertion of the "Data Path Integrity Check" pattern into the 5th octet of the cell (e.g., the cell, with its HEC byte, will be written into the RxFIFO).

Conversely, writing a "1" into this bit- field enables this features (e.g., the HEC byte of each cell will be overwritten by the "Data Path Integrity Check" pattern). The "Data Path Integrity Check" pattern, that is written into the cell depends upon the setting of Bit 6 (RDPChk) within this register.

For more information on this topic, please see Section 7.2.2.6.

Bit 4-IC (Idle Cell) Discard

This "Read/ Write" bit- field allows the user to configure the Receive Cell Processor to either discard or retain Idle Cells. If the user configures the Receive Cell Processor to discard Idle Cells, then the Idle Cells will be discarded and will NOT be written into the Rx FIFO. If the user configures the Receive Cell Processor to retain Idle Cells, then all Idle Cells will be retained and can be (depending upon the User Cell Filter settings) written to the Rx FIFO.

Writing a "0" to this bit- field configures the Receive Cell Processor to retain Idle Cells. Writing a "1" to this bit- field configures the Receive Cell Processor to discard Idle Cells.

For more information on the handling of Idle Cells by the Receive Cell Processor, please see Section 7.2.2.3.1.

Bit 3-OAM Check Bit

This "Read/ Write" bit- field allows the user to configure the Receive Cell Processor to "check" the next OAM cell, that it receives. Specifically, this means that the Receive Cell Processor, upon identifying an incoming OAM cell, will copy the header and payload bytes of this cell to the "Received OAM Cell" buffer (in on- chip RAM). If the user does not

configure the Receive Cell processor to perform an "OAM Cell Check", the OAM cell will simply be treated like any other user cell, as it is processed through the User Cell Filter, where it can be discarded or written to the Rx FIFO.

Writing a "0" to this bit-field disables the "OAM Cell Check" feature. Writing a "1" to this bit-field enables this feature.

Bit 2-De-Scramble Enable

This "Read/Write" bit-field allows the user to enable or disable the Cell Descrambler, within the Receive Cell Processor. When the Cell Descrambler is enabled, the Receive Cell Processor will "presume" that the payload portion of each incoming cell has been scrambled by the "far-end" Transmit Cell Processor. Therefore, the Receive Cell Processor will modify the contents of the cell payload accordingly. If the Cell Descrambler is disabled, then the Receive Cell Processor will perform NO modifications to the payload byte, of the incoming cells.

Writing a "0" to this bit-field disables the Cell De-Scrambler. Writing a "1" to this bit-field enables the Cell Descrambler.

For more information on Cell Scrambling and Cell Descrambling, please see Sections 6.2.2.2 and 7.2.2.5.

Bit 1-Rx Coset Enable

This "Read/Write" bit-field allows the user to configure the Receive Cell Processor to account for (or to not account for) the "far-end" Transmit Cell Processor's modulo-2 addition of the Coset polynomial: x6 + x4 + x2 + 1 to the "original" HEC byte, during HEC byte calculation and insertion.

If the user configures the Receive Cell Processor to account for the Coset Polynomial, then the Receive Cell Processor will go through the following procedure during HEC byte verification:

- Recompute the "Original" HEC (CRC-8) byte, based upon the values of bytes 1 through 4 in the received cell.
- Modulo-2 add the Coset Polynomial to the CRC-8 byte, thereby creating the "HEC byte".
- Compare the locally computed HEC byte with the fifth octet of the incoming cell.

If the user configures the Receive Cell Processor to NOT account for the Coset Polynomial, then the Receive Cell Processor will go through the following procedure during HEC byte verification:

- Recompute the HEC byte, based upon the values of bytes 1 through 4 in the received cell.
- Compare the locally computed HEC byte with the fifth octet of the incoming cell.

Writing a "0" to this bit-field configures the Receive Cell Processor to NOT account for the Coset Polynomial. Writing a "1" to this bit-field configures the Receive Cell Processor to account for the Coset Polynomial.

Bit 0-HEC Error Ignore

This "Read/Write" bit-field allows the user to configure the Receive Cell Processor to either discard or retain cells with HEC byte errors.

If the user configures the Receive Cell Processor to discard these errored cells (the default condition), then all incoming cells containing single-bit (when the Receive Cell Processor is operating in the "Detection Mode") or multi-bit errors in their header bytes, will be discarded and will NOT be written to the Rx FIFO. (Note: If the Receive Cell Processor is operating in the "Correction Mode", then those cells that contain single-bit errors will be corrected, via the HEC Byte Verification Algorithm, and will not be discarded).

If the user configures the Receive Cell Processor to retain these errored cells, then all incoming cells containing single-bit or multi-bit errors in their headers, will NOT be discarded, and may (depending upon the Idle or User cell filter settings) be written to the Rx FIFO.

Writing a "0" to this bit-field disables this feature (e.g., the Receive Cell Processor will discard errored cells). Writing a "1" to this bit-field enable this features (e.g., the Receive Cell Processor will retain errored cells.)

Note: For more information on this feature, please see Section 7.2.2.2.

3.3.2.97 Rx CP Additional Configuration Register

Address = 5Fh, Rx CP Additional Configuration Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused		User Cell Filter Discard	User Cell Filter Enable	Corr Thresh[1]	Corr Thresh[0]	Corr Enable	Unused
RO	RO	R/W	R/W	R/W	R/W	R/W	RO
0	0	0	0	1	1	1	0

Bit 5 User Cell Filter Discard

This "Read/Write" bit-field allows the user to specify which valid cells (e.g., non-Idle cells) are to be discarded by the User Cell Filter.

Writing a "0" to this bit-field causes the User Cell Filter to discard all user cells NOT matching the header byte patterns, as defined in the "Rx CP User Cell Filter Pattern Header byte" registers and the "Rx CP User Cell Filter Mask Header byte" registers.

Writing a "1" to this bit-field causes the User Cell Filter to discard all users cells MATCHING the header byte patterns, as defined in the "Rx CP User Filter Cell Pattern Header byte" registers and the "Rx CP User Cell Filter Mask Header byte" registers.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

Bit 4 User Cell Filter Enable

This "Read/Write" bit-field allows the user to enable or disable the User (or Assigned) Cell Filter. If the User Cell Filter is disabled then all non-Idle Cells will be written the Rx FIFO, within the Receive Utopia Interface block. However, if the User Cell Filter is enabled, then only those user cells, specified by the following parameters; will be written into the Rx FIFO.

- The contents of bit-field number 5, within this Register (User Cell Filter Discard).
- The contents of the four "Rx CP User Cell Filter Pattern Header Byte" registers (Address = 6Ah through 6Dh), and
- The contents of the four "Rx CP User Cell Filter Mask Header Byte" registers (Address = 6Eh through 71h)

Writing a "0" to this bit-field disables the User Cell Filter. Writing a "1" enables the User Cell Filter.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

Bits 3 and 2 Correction Threshold[1, 0]

These two "Read/Write" bit-fields allow the user to define the Correction Threshold, "M", as specified below. For more information on Correction Thresholds, please see Section 7.2.2.2.

- Correction Threshold[1, 0] = 0, 0, then M = 0

The Receive Cell Processor, while performing HEC Byte Verification, will always operate in the "Correction" mode.

- Correction Threshold[1, 0] = 0, 1, then M = 1

The Receive Cell Processor, while performing HEC Byte Verification, must detect a single "error- free" cell before it will transition from the "Detection" mode to the "Correction" mode.

- Correction Threshold[1, 0] = 1, 0, then M = 3

The Receive Cell Processor, while performing HEC Byte Verification, must detect 3 consecutive "error- free" cells before it will transition from the "Detection" mode to the "Correction" mode.

- Correction Threshold[1, 0] = 1, 1, then M = 7

The Receive Cell Processor, while performing HEC Byte Verification, must detect 7 consecutive "error- free" cells before it will transition from the "Detection" mode to the "Correction" mode.

Bit 1-Correction Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Correction" Mode, within the "HEC Byte Verification" Algorithm. Specifically, if the user disables the "Correction" mode, then the Receive Cell Processor, while performing HEC byte verification, will only operate in the "Detection" Mode (e.g., cells with single- bit errors are NOT corrected, and are subject to discard).

Writing a "0" to this bit- field disables the "Correction" mode. Writing a "1" to this bit- field enables the "Correction" Mode.

For more information on the Correction Mode, within the HEC Byte Verification Algorithm, please see Section 7.2.2.2.

3.3.2.98-Rx CP Interrupt Enable Register

Address = 60h, Rx CP Interrupt Enable Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused					OAM Interrupt Enable	LCD Interrupt Enable	HEC Error Interrupt Enable
RO	RO	RO	RO	RO	R/ W	R/ W	R/ W
0	0	0	0	0	0	0	0

Bit 2-OAM (Cell Received) Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Received OAM Cell" interrupt.

Writing a "0" to this bit- field disables the "Received OAM Cell" interrupt. Writing a "1" enables this interrupt.

Bit 1—"Change in LCD (Loss of Cell Delineation) Condition" Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Change in Loss of Cell Delineation Condition" interrupt.

Writing a "0" to this bit- field disables the "Change in Loss of Cell Delineation Condition" interrupt. Writing a "1" enables this interrupt.

Bit 0—"Detection of HEC Byte Error" Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Detection of HEC Byte Error" interrupt.

Writing a "0" to this bit- field disables the "Detection of HEC Error" interrupt. Writing a "1" enables this interrupt.

3.3.2.99 Rx CP Interrupt Status Register

Address = 61h, Rx CP Interrupt Status Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused					OAM Interrupt Status	LCD Interrupt Status	HEC Error Interrupt Status
RO	RO	RO	RO	RO	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Bit 2-OAM (Cell Received) Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Received OAM Cell" interrupt has occurred since the last read of this register. This interrupt will occur if the "Receive OAM Cell" buffer has a new OAM cell that needs to be read and processed by the local µP.

If this bit- field is "0" then the "Received OAM Cell" interrupt has NOT occurred since the last read of this register. If this bit- field is "1", then the "OAM Cell Received" interrupt has occurred since the last read of this register.

For more information on this interrupt, please see Section 7.2.2.4.

Bit 1-"Change in LCD (Loss of Cell Delineation) Condition" Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Change in Loss of Cell Delineation Condition" interrupt has occurred since the last read of this register. The Receive Cell Processor will generate this interrupt if either of the following two events occur.

1. If the Receive Cell Processor (while operating in the "SYNC" state) detects too many consecutive cells with HEC byte errors, and declares itself to be in the "HUNT" state. At this point, the Receive Cell Processor will not be delineating cells; and will cease to write anymore cells into the RxFIFO.
2. Once the Receive Cell Processor has transitioned from the "PRESYNC" state and into the "SYNC" state, within the "Cell Delineation" Algorithm (See Figure 71).

If this bit- field is "0", then the "Change in Loss of Cell Delineation Condition" interrupt has NOT occurred since the last read of this register. If this bit- field is "1", then the "Change in Loss of Cell Delineation Condition" interrupt has occurred since the last read of this register.

For more information on this interrupt and cell delineation, please see Section 7.2.2.1.2.

Bit 0–"Detection of HEC Byte Error" Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Detection of HEC Byte Error" interrupt has occurred since the last read of this register. This interrupt will occur if the Receive Cell Processor detects a single- bit or multi- bit HEC byte error in an incoming cell that it receives from the Receive E3 Framer.

If this bit- field is "0", then the "Detection of HEC Byte Error" interrupt has NOT occurred since the last read of this register. If this bit- field is "1", then the "Detection of HEC Byte Error" interrupt has occurred since the last read of this register.

3.3.2.100 - Rx CP Idle Cell Pattern Header-Byte 1

Address = 62h, Rx CP Idle Cell Pattern Header-Byte 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Pattern-Byte 1							
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 "Read/Write" register along with the "Rx CP Idle Cell Pattern Header-Bytes, 2 through 4" registers are used to specify, to the Receive Cell Processor, the header byte patterns for Idle Cells. The Receive Cell Processor will use this information to identify the Idle Cells from the stream of cells that it receives from the Receive E3 Framer. The purpose of this particular register (along with the "Rx CP Idle Cell Mask Header-Byte 1" register) is to allow the user to define the pattern for header byte 1 of the Idle Cells.

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.101-Rx CP Idle Cell Pattern Header-Byte 2

Address = 63h, Rx CP Idle Cell Pattern Heade-Byte 2							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Pattern-Byte 2							
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 "Read/Write" register along with the "Rx CP Idle Cell Pattern Header - Bytes, 1, 3 and 4" registers are used to specify, to the Receive Cell Processor, the header byte patterns for Idle Cells. The Receive Cell Processor will use this information to identify the Idle Cells from the stream of cells that it receives from the Receive E3 Framer. The purpose of this particular register (along with the "Rx CP Idle Cell Mask Header-Byte 2" register) is to allow the user to define the pattern for header byte 2 of the Idle Cells.

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.102-Rx CP Idle Cell Pattern-Byte 3

Address = 64h, Rx CP Idle Cell Pattern Header-Byte 3							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Pattern-Byte 3							
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 "Read/Write" register along with the "Rx CP Idle Cell Pattern Header - Bytes, 1, 2, and 4" registers are used to specify, to the Receive Cell Processor, the header byte patterns for Idle Cells. The Receive Cell Processor will use this information to identify the Idle Cells from the stream of cells that it receives from the Receive E3 Framer. The purpose of this particular register (along with the "Rx CP Idle Cell Mask Header - Byte 3" register) is to allow the user to define the pattern for header byte 3 of the Idle Cells.

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.103 Rx CP Idle Cell Pattern Header-Byte 4

Address = 65h, Rx CP Idle Cell Pattern Header - Byte 4							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Pattern-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This "Read/Write" register along with the "Rx CP Idle Cell Pattern Header - Bytes, 1 through 3" registers are used to specify, to the Receive Cell Processor, the header byte patterns for Idle Cells. The Receive Cell Processor will use this information to identify the Idle Cells from the stream of cells that it receives from the Receive E3 Framer. The purpose of this particular register (along with the "Rx CP Idle Cell Mask Header-Byte 4" register) is to allow the user to define the pattern for header byte 4 of the Idle Cells.

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.104 Rx CP Idle Cell Mask Header-Byte 1

Address = 66h, Rx CP Idle Cell Mask Header-Byte 1							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Mask Header-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in byte 1 of the incoming Idle cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP Idle Cell Pattern Header-Byte 1" register (Address = 62h) by the Idle Cell Filter, when the Receive Cell Processor is trying to determine if an incoming cell is an Idle Cell or not.

Writing a "1" into a particular bit- field in this register, forces the Receive Cell Processor to check and compare the corresponding bit in within octet 1 of the incoming cell with the corresponding bit- field in the "Rx CP Idle Cell Pattern Header-Byte 1" register.

Writing a "0" into a particular bit- field, causes the Receive Cell Processor to treat the corresponding bit within octet 1 in the incoming cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit in byte 1 of the incoming cell with the corresponding bit- field in the "Rx CP Idle Cell Pattern Header-Byte 1" register.)

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.105 Rx CP Idle Cell Mask Header-Byte 2

Address = 67h, Rx CP Idle Cell Mask Header-Byte 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Mask Header-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in byte 2 of the incoming cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP Idle Cell Pattern Header - Byte 2" register (Address = 63h) by the Idle Cell Filter, when the Receive Cell Processor is trying to determine if an incoming cell is an Idle Cell or not.

Writing a "1" into a particular bit- field in this register, forces the Receive Cell Processor to check and compare the corresponding bit within octet 2 of the incoming cell with the corresponding bit in the "Rx CP Idle Cell Pattern Header-Byte 2" register.

Writing a "0" into a particular bit- field, causes the Receive Cell Processor to treat the corresponding bit within octet 2 in the incoming cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit in byte 2 of the incoming cell with the corresponding bit in the "Rx CP Idle Cell Pattern Header-Byte 2" register.)

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.106 Rx CP Idle Cell Mask Header-Byte 3

Address = 68h, Rx CP Idle Cell Mask Header-Byte 3							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Mask Header-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in byte 3 of the incoming Idle cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP Idle Cell Pattern Header-Byte 3" register (Address = 64h) by the Idle Cell Filter, when the Receive Cell Processor is trying to determine if an incoming cell is an Idle Cell or not.

Writing a "1" into a particular bit- field in this register, forces the Receive Cell Processor to check and compare the corresponding bit within octet 3 of the incoming cell with the corresponding bit in the "Rx CP Idle Cell Pattern Header-Byte 3" register.

Writing a "0" into a particular bit- field, causes the Receive Cell Processor to treat the corresponding bit within octet 3 in the incoming cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit within octet 3 of the incoming cell with the corresponding bit in the "Rx CP Idle Cell Pattern Header-Byte 3" register.)

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.107 Rx CP Idle Cell Mask Header-Byte 4

Address = 69h, Rx CP Idle Cell Mask Header-Byte 4							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Mask Header-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in byte 4 of the incoming Idle cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP Idle Cell Pattern Header-Byte 4" register (Address = 65h) by the Idle Cell Filter, when the Receive Cell Processor is trying to determine if an incoming cell is an Idle Cell or not.

Writing a "1" into a particular bit- field in this register, forces the Receive Cell Processor to check and compare the corresponding bit within octet 4 of the incoming cell with the corresponding bit in the "Rx CP Idle Cell Pattern Header-Byte 4" register.

Writing a "0" into a particular bit- field, causes the Receive Cell Processor to treat the corresponding bit within octet 1 in the incoming cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit within octet 4 of the incoming cell with the corresponding bit- field in the "Rx CP Idle Cell Pattern Header-Byte 4" register.)

For more information on Idle Cell Handling, please see Section 7.2.2.3.1.

3.3.2.108 Rx CP User Cell Filter Pattern Header-Byte 1

Address = 6Ah, Rx CP User Cell Filter Pattern Header-Byte 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Header Pattern-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The User (or Assigned) cell filtering criteria is defined based upon the contents of 8 read/write registers. These registers are the four "Rx CP User Cell Filter Pattern Header Byte" Registers, and the four "Rx CP User Cell Filter Mask Header Byte" registers. This "Read/Write" register, along with the "Rx CP User Cell Filter Mask Header-Byte 1" register (Address = 6Eh) allows the user to define the User (or Assigned) cell filtering criteria for octet 1 of the incoming user cell. The user will write in the header byte pattern for octet 1, that he/she wishes to use as part of the User Cell Filtering criteria, into this register. The user will also write in a value to the "Rx CP User Cell Filter Mask Header - Byte 1" register, that indicates which bits within the first octet of the incoming cell are to be compared with the contents of this register.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

3.3.2.109 Rx CP User Cell Filter Pattern Header-Byte 2

Address = 6Bh, Rx CP User Cell Filter Pattern Header-Byte 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Header Pattern-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The User (or Assigned) cell filtering criteria is defined based upon the contents of 8 read/write registers. These registers are the four "Rx CP User Cell Filter Pattern Header Byte" Registers, and the four "Rx CP User Cell Filter Mask Header Byte" registers. This "Read/Write" register, along with the "Rx CP User Cell Filter Mask Header-Byte 2" register (Address = 6Fh) allows the user to define the User (or Assigned) cell filtering criteria for octet 2 of the incoming user cell. The user will write in the header byte pattern for octet 2, that he/she wishes to use as part of the User Cell Filtering criteria, into this register. The user will also write in a value to the "Rx CP User Cell Filter Mask Heade-Byte 2" register, that indicates which bits within the second octet of the incoming cell are to be compared with the contents of this register.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

3.3.2.110 Rx CP User Cell Filter Pattern Header-Byte 3

Address = 6Ch, Rx CP User Cell Filter Pattern Header-Byte 3							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Header Pattern-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The User (or Assigned) cell filtering criteria is defined based upon the contents of 8 read/write registers. These registers are the four "Rx CP User Cell Filter Pattern Header Byte" Registers, and the four "Rx CP User Cell Filter Mask Header Byte" registers. This "Read/Write" register, along with the "Rx CP User Cell Filter Mask Header-Byte 3" register (Address = 70h) allows the user to define the User (or Assigned) cell filtering criteria for octet 3 of the incoming user cell. The user will write in the header byte pattern for octet 3, that he/she wishes to use as part of the User Cell Filtering criteria, into this register. The user will also write in a value to the "Rx CP User Cell Filter Mask Header-Byte 3" register, that indicates which bits within the third octet of the incoming cell are to be compared with the contents of this register.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

3.3.2.111 Rx CP User Cell Filter Pattern Header-Byte 4

Address = 6Dh, Rx CP User Cell Filter Pattern Header - Byte 4							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Header Pattern-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The User (or Assigned) cell filtering criteria is defined based upon the contents of 8 read/write registers. These registers are the four "Rx CP User Cell Filter Pattern Header Byte" Registers, and the four "Rx CP User Cell Filter Mask Header Byte" registers. This "Read/Write" register, along with the "Rx CP User Cell Filter Mask Header-Byte 4" register (Address = 71h) allows the user to define the User (or Assigned) cell filtering criteria for octet 4 of the incoming user cell. The user will write in the header byte pattern for octet number 4, that he/she wishes to use as part of the User Cell Filtering criteria, into this register. The user will also write in a value to the "Rx CP User Cell Filter Mask Header-Byte 4" register, that indicates which bits within the fourth octet of the incoming cell are to be compared with the contents of this register.

For more information on the User Cell Filter, please see Section 7.2.2.3.2.

3.3.2.112 Rx CP User Cell Filter Mask Header-Byte 1

Address = 6Eh, Rx CP User Cell Filter Mask Header-Byte 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Mask Header-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in octet 1 of the incoming user cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP User Cell Filter Pattern Header—Byte 1" register (Address = 6Ah) by the User Cell Filter, when determining whether a given user cell is to be filtered out or written to the Rx FIFO.

Writing a "1" into a particular bit- field in this register, forces the User Cell Filter to check and compare the corresponding bit within octet 1 of the incoming user cell with the corresponding bit- field in the "Rx CP User Cell Filter Pattern Header-Byte 1" register.

Writing a "0" into a particular bit- field, causes the User Cell Filter to treat the corresponding bit within octet 1 in the incoming user cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit- field in octet 1 of the incoming user with the corresponding bit in the "Rx CP User Cell Filter Pattern Header-Byte 1" register.)

For more information on User Cell Filtering, please see Section 7.2.2.3.2.

3.3.2.113 Rx CP User Cell Filter Mask Header-Byte 2

Address = 6Fh, Rx CP User Cell Filter Mask Header-Byte 2							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Mask Header-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in octet 2 of the incoming user cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP User Cell Filter Pattern Header—Byte 2" register (Address = 6Bh) by the User Cell Filter, when determining whether a given user cell is to be filtered out or written to the Rx FIFO.

Writing a "1" into a particular bit- field in this register, forces the User Cell Filter to check and compare the corresponding bit within octet 2 of the incoming user cell with the corresponding bit in the "Rx CP User Cell Filter Pattern Header-Byte 2" register.

Writing a "0" into a particular bit- field, causes the User Cell Filter to treat the corresponding bit within octet 2 in the incoming user cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit within octet 2 of the incoming user with the corresponding bit- field within the "Rx CP User Cell Filter Pattern Header - Byte 2" register.)

For more information on User Cell Filtering, please see Section 7.2.2.3.2.

3.3.2.114 Rx CP User Cell Filter Mask Header-Byte 3

Address = 70h, Rx CP User Cell Filter Mask Header-Byte 3							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Mask Header-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in octet 3 of the incoming user cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP User Cell Filter Pattern Header—Byte 3" register (Address = 6Ch) by the User Cell Filter, when determining whether a given user cell is to be filtered out or written to the Rx FIFO.

Writing a "1" into a particular bit- field in this register, forces the User Cell Filter to check and compare the corresponding bit within octet 3 of the incoming user cell with the corresponding bit in the "Rx CP User Cell Filter Pattern Header-Byte 3" register.

Writing a "0" into a particular bit- field, causes the User Cell Filter to treat the corresponding bit within octet 3 in the incoming user cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit in octet 3 of the incoming user with the corresponding bit- field in the "Rx CP User Cell Filter Pattern Header-Byte 3" register.)

For more information on User Cell Filtering, please see Section 7.2.2.3.2.

3.3.2.115 Rx CP User Cell Filter Mask Header-Byte 4

Address = 71h, Rx CP User Cell Filter Mask Header-Byte 4							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Rx User Cell Mask Header-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

This "Read/Write" register allows the user to specify which bit(s), in octet 4 of the incoming user cell (in the Receive Cell Processor) are to be checked against the corresponding bit(s) in the "Rx CP User Cell Filter Pattern Header-Byte 4" register (Address = 6Dh) by the User Cell Filter, when determining whether a given user cell is to be filtered out or written to the Rx FIFO.

Writing a "1" into a particular bit- field in this register, forces the User Cell Filter to check and compare the corresponding bit in octet 4 of the incoming user cell with the corresponding bit in the "Rx CP User Cell Filter Pattern Header-Byte 4" register.

Writing a "0" to a particular bit, causes the User Cell Filter to treat the corresponding bit within octet 4 in the incoming user cell as a "don't care" (e.g., to forgo the comparison between the corresponding bit in octet 4 of the incoming user with the corresponding bit- field in the "Rx CP User Cell Filter Pattern Header-Byte 4" register.)

For more information on User Cell Filtering, please see Section 7.2.2.3.2.

3.3.2.116 Tx CP Control/ Interrupt Register

Address = 72h, Tx CP Control/ Interrupt Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Scrambler Enable	Coset Enable	HEC Insert Enable	TDP Check Pattern	GFC Insert Enable	TDP Error Interrupt Enable	Idle Cell HEC Calc Enable	TDP Error Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
1	1	1	1	0	0	1	0

The Transmit Cell Processor Control Register allows the user to control many aspect of the operation of the Transmit Control Processor. The description of each bit- field, within this register follows.

Bit 7-Scrambler Enable

This "Read/Write" bit- field allows the user to enable or disable the Cell Scrambler, within the Transmit Cell Processor. When the Cell Scrambler is enabled, it will "scramble" the payload bytes of each cell prior to its transmittal to the Transmit E3 Framer. When the Cell Scrambler is disabled, then the Transmit Cell Processor will not scramble the payload portion of each cell. Instead, the Transmit Cell Processor will transmit cells to the Transmit E3 Framer, with the cell payload data as received from the Tx FIFO.

Writing a "0" to this bit- field disables the Cell Scrambler. Writing a "1" enables the Cell Scrambler.

For more information on the Cell Scrambler, please see Section 6.2.2.2.

Bit 6-Coset Enable

This "Read/Write" bit- field allows the user to enable or disable the modulo- 2 addition of the Coset polynomial,

x6 + x4 + x2 + 1 to a newly computed HEC byte. Once this polynomial has been added to the existing HEC byte, this new modification to the contents of the HEC byte, will now be referred to as the "HEC byte".

Writing a "0" to this bit- field disables this addition. Writing a "1" to this bit- field enables this addition.

For more information into the function/ purpose of the Coset polynomial, please see Section 6.2.2.1.3.

Bit 5-HEC Byte Insert Enable-Assigned Cells

This "Read/ Write" bit- field allows the user to enable or disable the calculation and insertion of the HEC byte into user or assigned cells.

Writing a "0" into this bit- field disables the Transmit Cell Processor from calculating and inserting the HEC byte into each outgoing user or assigned cell. Writing a "1" into this bit- field enables this feature.

For more information on this bit selection, please see Section 6.2.2.1.1.

Bit 4-TDPChk Pat (Transmit Data Path Integrity Check Pattern Selection)

The Transmit Cell Processor is always checking for a specific (Data Path Integrity Check) pattern in the fifth octet of each cell that it reads from the Tx FIFO. This pattern will exist in the 5th octet of the cell, prior to the insertion of the HEC byte. This "Read/ Write" bit- field allows the user to specify the octet pattern that the Transmit Cell Processor should be checking for. The following table relates the contents of this bit field to the octet pattern expected by the Transmit Cell Processor.

TDPChk Pat	Result
0	Transmit Cell Processor expects an alternating "55h"/ "AAh" pattern for the value of the fifth octet of the cells received from the Tx FIFO.
1	Transmit Cell Processor expects a constant "55h" pattern for the value of the fifth octet of the cells received from the Tx FIFO.

For more information on this feature, please see Section 6.2.2.6.

Bit 3-GFC Nibble- Field Insert Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Transmit GFC Nibble Field" Serial Input port (TxGFC). If the user enables this input port, then he/ she can externally insert the value of the GFC Nibble- field into each outbound valid (e.g., user or OAM) cell. If this port remains disabled, then the GFC Nibble field value will remain as written into the Transmit Utopia Interface block, by the ATM Layer processor.

Writing a "0" into this bit- field disables this serial port. Writing a "1" into this bit- field enables this serial port.

Note: The "Transmit GFC Nibble- field" serial input port will not insert the GFC nibble- field value into Idle Cells.

For more information on this bit selection, please see Section 6.2.2.3.

Bit 2—TDP (Transmit Data Path Integrity Test) Error Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Data Path Integrity Test" interrupt. Writing a "0" to this bit- field disables this interrupt. Likewise, writing a "1" to this bit- field enables this interrupt.

Bit 1-HC (Idle Cell) HEC Byte Calculation Enable

This "Read/ Write" bit allows the user to enable or disable the calculation and the insertion of the HEC byte into each outbound Idle Cell.

Writing a "0" into this bit- field disables the "HEC Byte Calculation and Insertion into Idle Cells" feature. Writing a "1" into this bit- field enables this feature.

Note: If this feature is disable, then the Transmit Cell Processor will, instead, write the contents of the "Tx CP Idle Cell Pattern Header Byte 5" register (Address = 7Ah) into the fifth octet position of each Idle cell.

For more details into the operation of Idle Cells, please see Section 6.2.2.1.2.

Bit 0-TDP (Transmit Data Path Integrity Check) Error Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Data Path Integrity Test" interrupt has occurred since the last reading of the Tx CP Control Register. This interrupt will occur if the Transmit Cell Processor detects a byte- pattern, in the fifth octet position of a cell read from the TxFIFO, that differs from the expected "Data Path Integrity Check" pattern.

A "1" in this bit- field indicates that this interrupt has occurred since the last reading of the Tx CP Control Register. A "0" in this bit- field indicates that this interrupt has not occurred.

For more details on the Data Path Integrity Check, please see Section 6.2.2.6.

3.3.2.117 Tx CP OAM Cell Register

Address = 73h, Tx CP OAM Cell Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Send OAM	Unused						
Sem.	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Bit 7-Send OAM (Cell)

A "0" to "1" transition in this bit- field will cause the Transmit Cell Processor to read in the contents of the "Transmit OAM Cell Buffer" (located at 136h through 16Bh in on- chip RAM), and transmit this information as a cell to the Transmit E3 Framer.

For more information on OAM cell processing please see Section 6.2.2.4.

3.3.2.118 Tx CP HEC Byte Error Mask Register

Address = 74h, Tx CP HEC Byte Error Mask Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
HEC Error Mask							
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 "Read/ Write" byte- field allows the user to insert errors into the HEC byte of each "outbound" cell (from the Transmit Cell Processor block). Prior to transmission to the Transmit E3 Framer, the Transmit Cell Processor will perform an XOR operation with the HEC byte of each cell and the contents of this register; and will write the results of this operation back into the 5th octet position of each cell. Therefore, if the user does not wish to insert errors into the HEC byte of each cell, he/ she should insure that the contents of this register is set to "00h" (the default value). For more information in the purpose/ use of this register, please see Section 6.2.2.1.4.

3.3.2.119 Future Use

Address = 75h, Future Use							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*

3.3.2.120 Tx CP Idle Cell Pattern Header-Byte 1

Address = 76h, Tx CP Idle Cell Pattern Header-Byte 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern Header-Byte 1							
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 "Read/Write" byte- field allows the user to specify the contents of the first header byte of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 00h.

3.3.2.121 Tx CP Idle Cell Pattern Header-Byte 2

Address = 77h, Tx CP Idle Cell Pattern Header-Byte 2							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern Header-Byte 2							
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 "Read/Write" byte- field allows the user to specify the contents of the second header byte of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 00h.

3.3.2.122 Tx CP Idle Cell Pattern Header-Byte 3

Address = 78h, Tx CP Idle Cell Pattern Header - Byte 3							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern Header-Byte 3							

Address = 78h, Tx CP Idle Cell Pattern Header - Byte 3							
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 "Read/Write" byte-field allows the user to specify the contents of the third header byte of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 00h.

3.3.2.123—Tx CP Idle Cell Pattern Header-Byte 4

Address = 79h, Tx CP Idle Cell Pattern Header - Byte 4							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern Header-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This "Read/Write" byte-field allows the user to specify the contents of the first header byte of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 01h.

3.3.2.124–Tx CP Idle Cell Pattern Header-Byte 5

Address = 7Ah, Tx CP Idle Cell Pattern Header-Byte 5							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern Header-Byte 5							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	1	0	0	1	0

This "Read/Write" byte-field allows the user to specify the contents of the fifth header byte of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 52h.

Note: If the user enables the "Idle Cell HEC Byte Calculation Enable" features, then the Transmit Cell Processor will compute and insert the HEC byte into the 5th octet position, in lieu of using the contents of this register.

3.3.2.125–Tx CP Idle Cell Payload Register

Address = 7Bh, Tx CP Idle Cell Payload Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Idle Cell Pattern-Payload Byte							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	1	1	0	1	0

This "Read/Write" byte-field allows the user to specify the contents of the payload portion of the Idle Cells that are to be generated by the Transmit Cell Processor. The default value of this byte is 5Ah.

Note: The payload portion of each Idle Cell will consist of contents of this register, replicated 48 times.

3.3.2.126 Utopia Configuration Register

Address = 7Ch, Utopia Configuration Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused		Hand Shake Mode	SPHY/ MPHY*	CellOf 52 Bytes	TxFIFO Depth[1]	TxFIFO Depth[0]	Utopia Width 16
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

This register allows the user to control many aspects of the operation of the Transmit Utopia Interface block. The description of each of these bit- fields within this registers follows.

Bit 5-Handshake Mode

This "Read/ Write" bit- field allows the user to configure the Transmit and Receive Utopia Interface blocks to operate in either the "Octet Level" or "Cell Level" Handshake Modes.

Writing a "0" to this bit- field configures both the Transmit and Receive Utopia Interface blocks to operate in the "Octet-Level" Handshake Mode. Writing a "1" to this bit- field configures both of these blocks to operate in the "Cell Level" Handshake Mode.

For more information on "Octet- Level" and "Cell- Level" Handshake Mode operations, please see Section 6.1.2.2.1.

Bit 4-S- PHY/ M- PHY (Utopia Operating Mode)*

This "Read/ Write" bit- field allows the user to configure the UNI chip to operate in either the Single- PHY or Multi- PHY modes. When the UNI chip is operating in Single PHY Mode, it is configured such that the ATM Layer Processor will be writing ATM cell data to and reading ATM cell data from it, and no other UNI ICs. Consequently, in Single PHY Mode, the UNI IC will not be checking for a valid Address on the Utopia Address Bus. It will simply support read and write operations, with the ATM Layer Processor, based upon the Utopia Data Bus signal (e.g., TxEnB, TxSoC, TxClav, RxEnB, RxSoC and RxClav).

When the UNI chip is operating in Multi- PHY Mode, it is now configured such that the ATM Layer Processor will be writing ATM cell data to and reading ATM cell data from it and, possibly numerous other UNI ICs. Therefore, in this mode, the UNI IC will be checking for a valid Address, on the Utopia Address bus, prior to performing any reads or writes from the ATM Layer Processor. Unless the UNI IC detects it own Address, on the Utopia Address bus, it will ignore the Utopia Data Bus signals (e.g., TxEnB, TxSoC, TxClav, RxEnB).

Writing a "0" to this bit- field will configure the UNI to operate in the Multi- PHY mode. Writing a "1" to this bit- field will configure the UNI to operate in the Single- PHY mode. This configuration selection applies to both the Transmit Utopia Interface and Receive Utopia Interface blocks. The default Utopia Operating Mode is Multi- PHY.

For more information on Single- PHY and Multi- PHY operation, please see Section 6.1.2.3.

Bit 3-CellOf52Bytes

This "Read/ Write" bit- field allows the user to configure the Cell Size (e.g., Number of octets per cell) that the Transmit and Receive Utopia Interface blocks will process over the Transmit and Receive Utopia Data Buses; as summa-

rized below.

CellOf52Bytes	Number of Octets per Cell
0	53 Bytes-When the Utopia Data Bus width is configured to be 8 bits
54 Bytes-When the Utopia Data Bus width is configured to be 16 bits	
1	52 Bytes-Independent of the configured Utopia Data bus width

For more information on this parameter, please see Sections 6.1.2.1.3.

Bits 2, 1,-TFIFODepth[1, 0]

These two "Read/ Write" bit- fields allows the user to configure the Operating Depth of the Tx FIFO, as summarized below

TxFIFODepth[1, 0]	Operating Depth of Tx FIFO
00	16 cells
01	12 cells
10	8 cells
11	4 cells

Bit 0-UtWidth16-Utopia Data Width

This "Read/ Write" bit- field allows the user to configure the width of the Utopia Data bus (for both the Transmit and Receive Utopia Interface blocks) to be either 8 bits or 16 bits.

Writing a "0" to this bit- field will configure the Utopia Data bus width (for both Transmit and Receive directions) to be 8 bits. Writing a "1" to this bit- field will configure the Utopia Data bus width to be 16 bits.

For more information into this option, please see Section 6.1.2.1.2.

3.3.2.127-Receive Utopia Interrupt Enable/ Status Register

Address = 7Dh, Rx UT Interrupt Enable/ Status Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxFIFO Reset	RxFIFO Overflw Interrupt Enable	Unused	RCCCA Interrupt Enable	RxFIFO Overflw Interrupt Status	Unused	RCCCA Interrupt Status
RO	R/ W	R/ W	RO	R/ W	RO	RO	RUR
0	0	0	0	0	0	0	0

Bit 6-RxFIFO Reset

This "Read/Write" bit-field allows the user to command a reset of the RxFIFO, within the Receive Utopia Interface block; without having to command a reset of the entire chip. Commanding a reset of the RxFIFO will clear out all of its contents. Additionally, it will reset the pointer to cells within the RxFIFO. Writing a "1" to this bit-field will cause the Rx FIFO to be reset.

Bit 5-RxFIFO Overrun Interrupt Enable

This "Read/Write" bit-field allows the user to enable or disable the "Rx FIFO Overrun Condition" interrupt.

Writing a "0" to this bit-field disables this interrupt. Writing a "1" to this bit-field enables this interrupt.

Bit 3-RCOCA Interrupt Enable

This "Read/Write" bit-field allows the user to enable or disable the "Rx FIFO Change of Cell Alignment" interrupt.

Writing a "0" to this bit-field disables this interrupt. Writing a "1" to this bit-field enables this interrupt.

Bit 2-RxFIFO Overrun Interrupt Status

This "Read-Only" bit-field indicates whether or not the "Rx FIFO Overrun Condition" interrupt has occurred since the last read of this register.

A "0" in this bit-field indicates that the "Rx FIFO Overrun Condition" interrupt has not occurred since the last read of this register.

A "1" in this bit-field indicates that the "Rx FIFO Overrun Condition" interrupt has occurred since the last read of this register.

Note: This bit-field is only cleared if the RxFIFO has been depleted.

For more information on this interrupt condition, please see Section 7.3.2.3.

Bit 0-RCOCA (Receive Utopia Interface block - Change of Cell Alignment) Interrupt Status

This "Reset-upon-Read" bit-field indicates whether or not the "Rx FIFO Change of Cell Alignment interrupt has occurred since the last read of this register.

A "0" in this bit-field indicates that the "Rx FIFO Change of Cell Alignment" interrupt has not occurred since the last read of this register.

A "1" in this bit-field indicates that the "Rx FIFO Change of Cell Alignment" interrupt has occurred since the last read of the register.

For more information on this interrupt condition, please see Section 7.3.2.3.

3.3.2.128-Receive Utopia Address Register

Address = 7Eh, Receive Utopia Address Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			Receive Utopia Address				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bits 4 through 0 of this register are "Read/Write" bit-fields that allows the user to assign a specific "Utopia Address" value to this Receive Utopia Interface block. This register is only important when the UNI is running in Multi-PHY operation. For more information on this register and the Receive Utopia Address bus, please see Section 7.3.2.2.2.2.

3.3.2.129 Receive Utopia FIFO Status Register

Rx UT FIFO Status Register (Address = 7Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused						RxFIFO Full	RxFIFO Empty
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	1

Bit 1—RxFIFO Full

This "Read- Only" bit- field indicates whether or not the Rx FIFO (within the Receive Utopia Interface block) is full. If this bit- field is "0", then the Rx FIFO is NOT full. However, if this bit- field is "1", then the Rx FIFO is full.

Bit 0—RxFIFO Empty

This "Read- Only" bit- field indicates whether or not the Rx FIFO (within the Receive Utopia Interface block) is empty. If this bit- field is "0", then the Rx FIFO is NOT empty. However, if this bit- field is "1", then the Rx FIFO is empty.

3.3.2.130 Transmit Utopia Interrupt/ Status Register

Address = 80h, Tx UT Interrupt/ Status Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
TxFIFO Reset	Discard Upon Parity Error	Tx Parity Interrupt Enable	TxFIFO Overflw Interrupt Enable	TCOCA Interrupt Enable	Tx Parity Interrupt Status	TxFIFO Overflw Interrupt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Bit 7—TxFIFO Reset

This "Read/ Write" bit- field allows the user to command a reset of the TxFIFO, within the Transmit Utopia Interface block; without having to command a reset of the entire chip. Writing a "1" to this bit- field will cause the Tx FIFO to be reset.

Bit 6—Discard (Cell) Upon Parity Error (Transmit Utopia Interface block)

This "Read/ Write" bit- field allows the user to configure the Transmit Utopia Interface block to discard or retain cells containing parity error(s), as detected by the Transmit Utopia Interface block. Writing a "0" to this bit- field configures the Transmit Utopia Interface block to retain all cells (including the errored cells) and ultimately write these cells into the Tx FIFO. Writing a "1" to this bit- field configures the Transmit Utopia Interface block to discard every cell that contains a parity error. For more information on this selection please see Section 6.1.2.1.4.

Bit 5—Tx Parity Interrupt Enable (Transmit Utopia Interface block)

This "Read/ Write" bit- field configures the Transmit Utopia Interface block to generate the "Detection of Parity Error" interrupt, if it detects a parity error on the Transmit Utopia Data bus.

Writing a "0" to this bit- field disables this interrupt. Writing a "1" to this bit- field enables this interrupt.

Bit 4—TxFIFO Overrun Interrupt Enable

This "Read/ Write" bit- field configures the Transmit Utopia Interface block to generate the "Tx FIFO Overrun

Condition" interrupt if it detects an overrun condition in the TxFIFO.

Writing a "0" to this bit- field disables this interrupt. Writing a "1" to this bit- field enables this interrupt.

Bit 3–TxFIFO Change of Cell Alignment Interrupt Enable

This "Read/ Write" bit- field configures the Transmit Utopia Interface block to generate the "Tx FIFO Change of Cell Alignment" interrupt if it detects the receipt of a "Runt" cell.

Writing a "0" to this bit- field disables this interrupt. Writing a "1" to this bit- field enables this interrupt.

Bit 2–Tx Parity Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Detection of Parity Error Condition (Tx Utopia)" interrupt has occurred since the last read of this register.

A "0" in this bit- field indicates that the "Detection of Parity Error Condition" interrupt has not occurred since the last read of this register.

A "1" in this bit- field indicates that the "Detection of Parity Error Condition" interrupt has occurred since the last read of this register.

For more information on this interrupt condition, please see Section 6.1.2.1.4.

Bit 1–TxFIFO Overrun Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Tx FIFO Overrun Condition" interrupt has occurred since the last read of this register.

A "0" in this bit- field indicates that the "Tx FIFO Overrun Condition" interrupt has not occurred since the last read of this register.

A "1" in this bit- field indicates that the "Tx FIFO Overrun Condition" interrupt has occurred since the last read of this register.

For more information on this interrupt condition, please see Section 6.1.2.4.

Bit 0–TxFIFO Change of Cell Alignment Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Tx FIFO Change of Cell Alignment" interrupt has occurred since the last read of this register.

A "0" in this bit- field indicates that the "Tx FIFO Change of Cell Alignment" interrupt has not occurred since the last read of this register.

A "1" in this bit- field indicates that the "Tx FIFO Change of Cell Alignment" interrupt has occurred since the last read of the register. This interrupt will occur if the Transmit Utopia Interface block detects a "Runt" cell.

For more information on this interrupt condition, please see Section 6.1.2.4.

3.3.2.131-Future Use

Address = 81h, Future Use							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
0	0	0	0	0	0	0	0

3.3.2.132 Transmit Utopia Address Register

Address = 82h, Transmit Utopia Address Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused			Transmit Utopia Address[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bits 4 through 0 of this register are "Read/Write" bit-fields that allows the user to assign a specific "Utopia Address" value to this Transmit Utopia Interface block. This register is only important when the UNI is running in Multi-PHY operation. For more information on this register and the Transmit Utopia Address bus, please see Section 6.1.2.3.2.

3.3.2.133 Transmit Utopia FIFO Status Register

Address = 83h, Transmit Utopia FIFO Status Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused						TxFIFO Full	TxFIFO Empty
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	1

Bit 1-TxFIFO Full

This "Read Only" bit-field indicates whether or not the Tx FIFO (within the Transmit Utopia interface block) is full. If this bit-field contains a "0" then the Tx FIFO is NOT full. If this bit-field contains a "1", then the Tx FIFO IS full.

Bit 0-TxFIFO Empty

This "Read Only" bit-field indicates whether or not the Tx FIFO (within the Transmit Utopia interface block) is empty. If this bit-field contains a "0" then the Tx FIFO is NOT empty. If this bit-field contains a "1", then the Tx FIFO IS empty.

3.3.2.134 Line Interface Drive Register

Address = 84h, Line Interface Drive Register							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused		REQB	TAOS	Encodis	TxLev	RLoop	LLoop
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit 5-REQB (Receive Equalization Bypass Control)

This "Read/Write" bit-field allows the user to control the state of the REQB output pin of the UNI device. This output pin is intended to be connected to the REQB input pin of the XR-T7295E E3 Line Receiver IC. If the user forces this

signal to toggle "high", then the XR-T7295E device will cause its newly received E3 line signal to "by-pass" some equalization circuitry within the XR-T7295E device. Conversely, if the user forces this signal to toggle "low", then the XR-T7295E device will route its newly received E3 line signal through its internal equalization circuitry.

Writing a "1" to this bit-field causes the UNI device to toggle the REQB output pin "high". Writing a "0" to this bit-field causes the UNI device to toggle the REQB output pin "low".

For information on the criteria that should be used when deciding whether to bypass the equalization circuitry or not, please consult the "XR-T7295E E3 Integrated Line Receiver" data sheet.

Note: If the customer is not using the XR-T7295E E3 Line Receiver IC, then he/she can use this bit-field and the REQB output pin for other purposes.

Bit 4-TACS (Transmit All Ones Signal)

This "Read/Write" bit-field allows the user to control the state of the TACS output pin of the UNI device. This output pin is intended to be connected to the TACS input pin of the XR-T7296 E3 Line Transmitter IC. If the user forces this signal to toggle "high", then the XR-T7296 device will transmit an "All Ones" pattern onto the line. Conversely, if the user commands this output signal to toggle "low" then the XR-T7296 E3 Line Transmitter IC will proceed to transmit data based upon the pattern that it receives via the TxPOS and TxNEG output pins (of the UNI IC).

Writing a "1" to this bit-field will cause the TACS output pin to toggle "high". Writing a "0" to this bit-field will cause this output pin to toggle "low".

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field, and the TACS output pin for other purposes.

Bit 3-Encodis (HDB3 Encoder Disable)

This "Read/Write" bit-field allows the user to control the state of the Encodis output pin of the UNI device. This output pin is intended to be connected to the Encodis input pin of the XR-T7296 E3 Line Transmitter IC. If the user forces this signal to toggle "high", then the "internal HDB3 encoder" (within the XR-T7296 device) will be disabled. Conversely, if the user command this output signal to toggle "low", then the "internal HDB3 encoder" (within the XR-T7296 device) will be enabled.

Writing a "1" to this bit-field causes the UNI IC to toggle the "Encodis" output pin "high". Writing a "0" to this bit-field will cause the UNI IC to toggle this output pin "low".

Note:

1. The HDB3 encoder, within the XR-T7296 device, is not to be confused with the HDB3 encoding capable that exists within the Transmit E3 Framer block of the UNI IC.
2. The user is advised to disabled the HDB3 encoder (within the XR-T7296 IC) if the Transmit and Receive E3 Framers (within the UNI) are configured to operate in the HDB3 line code.
3. If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the "Encodis" output pin for other purposes.

Bit 2-TxLev (Transmit Level Select Output)

This "Read/Write" bit-field allows the user to control the state of the "TxLev" output pin of the UNI device. This output pin is intended to be connected to the TxLev input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will increase the amplitude of its output signal on the line, in order to drive the signal over cable lengths of 225 ft or greater. Therefore, the user is recommended to set this output high, if he/she is driving E3 lines signal over 225 ft (or more) of cable.

Conversely, if the user is driving E3 line signals over less than 225 ft of cable, then he/she is recommended to set this output pin low.

Writing a "1" to this bit-field commands the UNI to toggle the TxLev output "high". Writing a "0" to this bit-field commands the UNI to toggle this output signal "low".

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the TxLev output pin for other purposes.

Bit 1-RLOOP (Remote Loopback)

This "Read/Write" bit-field allows the user to control the state of the RLOOP output pin of the UNI device. This output pin is intended to be connected to the RLOOP input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will start operating in the "Remote Loopback Mode". Conversely, if the user commands this signal to toggle "low", then the XR-T7296 device will be operating in the "Normal" mode.

Writing a "1" into this bit-field commands the UNI to toggle the "RLOOP" output signal "high". Writing a "0" into this bit-field commands the UNI to toggle this output signal "low".

For a detailed description of the XR-T7296 E3 Line Transmitter's operation during "Remote Loopback", please see the "XR-T7296 E3/STS-1, E3 Integrated Line Transmitter's Data Sheet.

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the RLOOP output pin for other purposes.

Bit 0-LLOOP

This "Read/Write" bit-field allows the user to control the state of the LLOOP output pin of the UNI device. This output pin is intended to be connected to the LLOOP input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will start operating in the "Local Loopback Mode". Conversely, if the user commands this signal to toggle "low", then the XR-T7296 device will be operating in the "Normal" mode.

Writing a "1" into this bit-field commands the UNI to toggle the "LLOOP" output signal "high". Writing a "0" into this bit-field commands the UNI to toggle this output signal "low".

For a detailed description of the XR-T7296 E3 Line Transmitter's operation during "Local Loopback", please see the "XR-T7296 E3/STS-1, E3 Integrated Line Transmitter's Data Sheet.

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the LLOOP output pin for other purposes.

3.3.2.135-Line Interface Scan Register

Address = 85h, Line Interface Scan Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused					DMO	RLOL	RLOS
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Bit 2-DMO (Drive Monitor Output)

This "Read-Only" bit-field indicates the logic state of the DMO input pin of the UNI device. This input pin is intended to be connected to the DMO output pin of the XR-T7296 E3 Line Transmitter IC. If this bit-field contains a logic "1", then the DMO input pin is "high". The XR-T7296 E3 Line Transmitter IC will set this pin "high" if the drive monitor circuitry (within the XR-T7296 device) has not detected any bipolar signals at the MTIP and MRING inputs (of the XR-T7296 device) within the last 128 ± 32 bit periods.

Conversely, if this bit-field contains a logic "0", then the DMO input pin is "high". The XR-T7296 E3 Line Trans-

mitter IC will set this pin "low" if bipolar signals are being detected at the MTIP and MRING input pins.

Note: If this customer is not using the XR- T7296 E3 Line Transmitter IC, then he/ she can use this input pin for a variety of other purposes.

Bit 1-RLOL (Receive Loss of Lock)

This "Read- Only" bit- field indicates the logic state of the RLOL input pin of the UNI device. This input pin is intended to be connected to the RLOL output pin of the XR- T7295E E3 Line Receiver IC. If this bit- field contains a logic "1", then the RLOL input pin is "high". The XR- T7295E E3 Line Receiver IC will set this pin "high" if the phase- locked- loop circuitry (within the XR- T7295E device) has lost "lock" with the incoming E3 data- stream and is not properly recovering clock and data.

Conversely, if this bit- field contains a logic "0", then the RLOL input pin is "low". The XR- T7295E E3 Line Receiver IC will hold this pin "low" as long as this "phase- locked- loop" circuitry (within the XR- T7295E device) is properly "locked" onto the incoming E3 data- stream, and is properly recovering clock and data from this data- stream.

For more information on the operation of the XR- T7295E E3 Line Receiver IC, please consult the "XR- T7295E E3 Integrated Line Receiver" data sheet.

Note: If the customer is not using the XR- T7295E E3 Line Receiver IC, then he/ she can use this bit- field, and the RLOL input pin for other purposes.

Bit 0-RLOS (Receive Loss of Signal)

This "Read- Only" bit- field indicates the logic state of the RLOS input pin of the UNI device. This input pin is intended to be connected to the RLOS output pin of the XR- T7295E E3 Line Receiver IC. If this bit- field contains a logic "1", then the RLOS input pin is "high". The XR- T7295E device will toggle this signal "high" if it (the XR- T7295E E3 Line Receiver IC) is not detecting a sufficient amount of signal energy on the line, from the incoming E3 data- stream. This event indicates that the XR- T7295E device may be experiencing a Loss of Signal (LOS) condition.

Conversely, if this bit- field contains a logic "0", then the RLOS input pin is "low". The XR- T7295E device will hold this signal "low" if it is detecting a sufficient amount of signal energy on the line, from the incoming E3 data- stream.

For more information on the operation of the XR- T7295E E3 Line Receiver IC, please consult the "XR- T7295 E3 Integrated Line Receiver" data sheet.

3.4 The "Loss of Clock Enable" Feature

The timing for the Microprocessor Interface section is asynchronous to the 34.368 MHz signal, which is applied to either the RxLineClk or the TxInClk input pins. However, the "on- chip" registers are updated by circuitry that is driven by one of these two clock signals. Hence, if the UNI device experiences a "Loss of Clock signal" event such that neither the TxInClk nor the RxLineClk signal are present, then the read or write operations with the UNI Microprocessor Interface section could cease to function.

In order to prevent this phenomenon, the UNI device offers a "Loss of Clock" (LOC) protection feature that allows the Microprocessor Interface section to at least complete or terminate an "in- process" Read or Write cycle (with the local µP) should this "Loss of Clock" event occur. The "LOC" circuitry consists of a ring oscillator that continuously checks for signal transitions at the TxInClk and RxLineClk input pins. If a "Loss of Clock" signal event occurs such that no transitions are occurring on these pins, then the LOC circuitry will automatically assert the Rdy_Dtck signal in order to complete (or terminate) the current "Read" or "Write" cycle with the UNI Microprocessor Interface section.

The user may enable or disable this "LOC Protection" feature by writing to Bit 6 (Disable LOC) within the "UNI

Operating Mode" register, as depicted below.

UNI Operating Mode Register (Address = 00h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	Disable LOC	Int LOS Enable	Reset By Reg	Cell Loopback	Line Loopback	TimRef Sel(1)	TimRef Sel(0)
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	1	0	0	0	0	0

Writing a "1" to this bit- field disables this "LOC Protection" feature. Writing a "0" to this bit- field enables this feature.

Note: The "Ring Oscillator" can be a source of noise, within the UNI chip. Hence, there may be a situation where the user will wish to disable the "LOC Protection" feature.

3.5 Using the PMON Holding Register

If the Microprocessor Interface section is configured to operate over an 8- bit data bus; then the local µP will be able to read from and write to the UNI on- chip registers, 8 bit per (read or write) cycle. Since most of the UNI on- chip registers contain 8- bits; communicating with the local µP, over an 8- bit data bus, is not much of an inconvenience. However, all of the PMON registers, within the UNI IC, contain 16 bits. Consequently, any reads of the PMON registers, will require two read cycles.

The XR- T7234 E3 UNI includes a feature that will make reading a PMON register a slightly less complicated task. The UNI chip address space contains a register known as the "PMON Holding" register, which is located at 56h. Whenever the local µP (while operating over an 8- bit data bus with the Microprocessor Interface of the UNI) reads in an 8- bit value of a given PMON registers (e.g., either the upper- byte or the lower byte value of the PMON register); the other 8- bit value of that PMON register will automatically be accessible by reading the PMON Holding register.

Hence, whenever the Microprocessor Interface is configured to operate over an 8- bit data bus, anytime the local µP is trying to read in the contents of a PMON register; the first read access must be made directly to one of the 8- bit values of the PMON registers (e.g., for example: the PMON Received Single- Bit HEC Error Count - MSB, Address = 48h). However, the second read can always be made to a constant location in system memory; the PMON Holding Register.

3.6 The Interrupt Structure within the UNI Microprocessor Interface Section

The XR- T7234 UNI device is equipped with a sophisticated Interrupt Servicing Structure. This Interrupt Structure includes an Interrupt Request output, INT*, numerous Interrupt Enable Registers and numerous Interrupt Status Registers. The Interrupt Servicing Structure, within the UNI contains two levels of hierarchy. The top level is at the "functional block" level (e.g., the Receive E3 Framer, Transmit E3 Framer, Receive Cell Processor, etc.) 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 UNI, are capable of generating Interrupt Requests to the local µC/ µP. The UNI device Interrupt Structure has been carefully designed to allow the user to quickly determine the exact source of the interrupt (with minimal latency) which will aid the local µP/ µC in determining which interrupt service routine to call up in order to eliminate the condition(s) causing the interrupt.

Table 5 lists all of the possible conditions that can generate interrupts, within each functional block of the UNI.

Table 5. List of all of the Possible Condition that can Generate Interrupts within the XR-T7234 UNI Device

Functional Block	Interrupting Condition
Transmit Utopia Block	• Detection of Parity Errors • Change of Cell Alignment • Tx FIFO Overrun
Transmit Cell Processor	• Data path integrity error occurrence
Transmit E3 Framer	• LAPD Message Frame Transfer Complete
Receive E3 Framer	• Change of State on Receive LOS, OOF, LOF, AIS Detection • Change in FERF Condition • Change in Frame AlignmentReceipt of New LAPD Message frame • EM (BIP-8) Byte Errors • FEBE Indications Detection of Framing Byte Errors • Receipt of New Trail Trace Buffer Message • Payload Type Msmatch Change
Receive Cell Processor	• HEC Byte Errors • OAM Cell Received • Loss of Cell Delineation
Receive Utopia Block	• Rx FIFO Overrun • Change of Cell Alignment
UNI Chip Level	• One-Second Interrupt

The XR-T7234 UNI Interrupt Block comes equipped with the following registers to support the servicing of this wide array of potential "interrupt request" sources. Table 6 lists these registers and their corresponding addresses, within the UNI.

Table 6. A Listing of the XR-T7234 UNI Device Interrupt Block Registers

Address Location	Register
04h	UNI Interrupt Enable Register
05h	UNI Interrupt Status Register

Table 6. A Listing of the XR- T7234 UNI Device Interrupt Block Registers

Address Location	Register
10h	Receive E3 Framer Interrupt Enable Register- 1
11h	Receive E3 Framer Interrupt Enable Register- 2
12h	Receive E3 Framer Interrupt Status Register- 1
13h	Receive E3 Framer Interrupt Status Register- 2
3Fh	Transmit E3 LAPD Status/ Interrupt Register
60h	Receive Cell Processor Interrupt Enable Register
61h	Receive Cell Processor Interrupt Status Register
72h	Transmit Cell Processor Control/ Interrupt Register
7Dh	Receive Utopia Interrupt Enable/ Status Register
80h	Transmit Utopia Interrupt Enable/ Status Register

General Flow of UNI Chip Interrupt Servicing

When any of the conditions, presented in Table 4 A occurs, (if their Interrupt is enabled), then the UNI will generate an interrupt request to the local µP/µC by asserting the active- low interrupt request output pin, INT*. Shortly after the local µP/µC has detected the activated INT* signal, it will enter into the appropriate "user- supplied" interrupt service routine. The first task for the local 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., XR- T7234 UNI Device), if multiple peripheral devices exist in the user's system. However, once the "interrupting peripheral" device has been identified, the next task for the local µP/µC is to determine exactly what feature or functional section within the device requested the interrupt.

Determine the Functional Block(s) Requesting the Interrupt

If the interrupting device turns out to be the XR- T7234 E3 UNI, then the local µC/µP must determine which 'functional block' (within the UNI) requested the interrupt. Hence, upon reaching this state, one of the very first things that the local µC/µP must do within the user supplied "UNI" interrupt service routine, is to perform a read of the UNI Interrupt Status Register (Address = 05h) within the XR- T7234 UNI device. The bit format of the UNI Interrupt Status Register is presented below.

UNI Interrupt Status Register: Address = 05h							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec. Interrupt Status	Tx E3 Framer Interrupt Status	Rx E3 Framer Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO

The UNI Interrupt Status Register, presents the "interrupt request" status of each functional block, within the chip. The purpose of the UNI Interrupt Status Register is to help the local µP/µC 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 5. Once the local µP/µC 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 UNI further supports the Functional Block hierarchy by providing the UNI Interrupt Enable Register (Address = 04h). The bit format of this register is identical to that for the UNI Interrupt Status register, and is presented below for the sake of completeness.

UNI Interrupt Enable Register: Address = 04h							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	One Sec. Interrupt Enable	Tx E3 Framer Interrupt Enable	Rx E3 Framer Interrupt Enable	Tx CP Interrupt Enable	Rx CP Interrupt Enable	Tx Utopia Interrupt Enable	Rx Utopia Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The UNI Interrupt Enable Register allows the user to individually enable or disable the interrupt requesting capability of the functional blocks, within the UNI. If a particular bit field, within this register contains the value "0"; then the corresponding functional block has been disabled from 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 UNI, contains anywhere from 1 to 12 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.

Interrupt Service Routine Branching: After Reading the UNI Interrupt Status Register

The contents of the UNI Interrupt Status Register identify which of 7 functional blocks (within the UNI IC) have requested interrupt service. The local µP 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 a bit- field in the UNI Interrupt Status Register. Additionally, this table also presents a list and addresses of corresponding on- chip Registers that the Interrupt Service Routine should branch to and read; based upon the Interrupting Functional Block.

Table 7. Interrupt Service Routine Guide

Interrupting Functional Block	The Next Registers to be Read During the Interrupt Service Routine	Register Address
Receive E3 Framer	Rx E3 Interrupt Status Register- 1	12h
	Rx E3 Interrupt Status Register- 2	13h
Receive Cell Processor	Rx CP Interrupt Status Register	61h
Receive Utopia Interface	Rx UT Interrupt Enable/ Status Register	7Dh
Transmit Utopia Interface	Tx UT Interrupt Enable/ Status Register	80h
Transmit Cell Processor	Tx CP Register	72h
Transmit E3 Framer	Tx E3 LAPD Status/ Interrupt Register	3Fh
One Second Interrupt	User's Option	

Once the local µP/µC has read the register that corresponds to the "interrupting source" within the UNI, then the following things will happen.

1. The "asserted interrupt status" bit- fields within this register will be reset upon read.
2. The "asserted" bit- field, within the UNI Interrupt Status Register will be reset.

3. The UNI device will negate the INT* (Interrupt Request) output.

3.6.1 Automatic Reset of Interrupt Enable Bits

Occasionally, the user's system (which includes the UNI device), may experience a fault condition, such that a "UNI Interrupt Condition" will continuously exist. If this particular interrupt condition has been enabled (within the UNI) then the UNI device will generate an interrupt request to the local µP/µC. Afterwards, the local µP/µC will attempt to service this interrupt by reading the UNI Interrupt Status Register and the subsequent "source" level interrupt status register. Additionally, the local µP/µC will attempt to perform some "system- related" tasks in order to try to resolve those conditions causing the interrupt. After the local µP/µC has attempted all of these things, the UNI IC will negate the INT* output. However, because the system fault still remains, the conditions causing the UNI to issue this interrupt request, also still exists. Consequently, the UNI device will generate another interrupt request, which forces the local µP/µC to "once again" attempt to service this interrupt. This phenomenon quickly results in the local µP/µC being "tied up" in a continuous cycle of executing this one interrupt service routine. Consequently, the local µP/µC (along with portions of the overall system) now becomes non- functional.

In order to prevent this phenomenon from ever occurring, the UNI IC allows the user to automatically reset the "interrupt enable" bits, following their activation. The user can implement this feature by writing the appropriate value to Bit 5 (Int En Reset) within the UNI I/O Control Register, as depicted below:

UNI I/O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	IntEn Reset	ATM/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Writing a "1" to this bit- field configures the UNI to automatically disable a given interrupt, following its activation. Writing a "0" to this bit- field configures the UNI to leave the "Interrupt Enable" bits enabled, following interrupt activation.

If a user opts to implement the "Automatic Reset of Interrupt Enable Bits" feature, then he/ she might wish to configure the local µP/µC to go back and "re- enable" these interrupts at a later time.

3.6.2 One Second Interrupts

The UNI Interrupt Status Register, and the UNI Interrupt Enable Register each contain a bit- field for the "One Second" Interrupt. If this interrupt is enabled (within the UNI Interrupt Enable register), then the UNI device will automatically generate an interrupt requests to the local µP/µC repeatedly at one- second intervals. At a minimum, the "user's" interrupt service routine must service this interrupt by reading the UNI Interrupt Status Register (Address = 05h). Once the local µP/µC has read this register, then the following things will happen.

1. The "One- Second Interrupt" bit- field, within the UNI Interrupt Status Register, will be reset to "0".
2. The UNI will negate the INT* (Interrupt Request) output.

The purpose of providing this "One Second" interrupt is to allow the local µP/µC the opportunity to perform certain task at one- second intervals. The user can accomplish this by including the performance of these various tasks as a part of the Interrupt Service Routine, for the "One- Second" type interrupt. Some of these tasks could include:

- Reading in the contents of the "One- Second" Performance Monitor registers

- Reading various other Performance Monitor registers.
- Writing a new PMDL Message into the "Transmit LAPD Message" Buffer.

After the LAPD Transmitter has been enabled and commanded to initiate transmission of the LAPD Message frame (containing the PMDL Message, residing within the "Transmit LAPD Message" buffer); the LAPD Transmitter will continue to re-transmit this same LAPD Message frame, repeatedly at one-second intervals; until it has been disabled. If the user writes in a new PMDL message, into the LAPD Transmit buffer, immediately following the occurrence of a "One-Second" interrupt, then he/she can be sure that this "write activity" will not interfere with this periodic transmission of the LAPD Message.

Notes regarding the UNI Interrupt Enable and UNI Interrupt Status Registers:

1. The UNI Interrupt Enable Registers allows the user to globally disable all potential interrupts within a given functional block; by writing a '0' into the appropriate bit-field of this register. However, the UNI Interrupt Enable Register does not allow the user to globally enable all potential interrupts within a given functional block. In other words, enabling a given functional block does not automatically enable all of its potential interrupt sources. Those potential interrupt sources that have been disabled at the "source level" will remain disabled, independent of the status of their associated functional blocks.
2. The UNI Interrupt Enable Register is set to 00h upon power up or reset. Therefore, the user will have to write some "1s" to this registers in order to enable some of the interrupts.

The remainder of the registers, presented in Table 6 will be presented in the discussion of their corresponding functional blocks. These discussions will present more details about the interrupt causes and how to properly service them.

3.7 Interfacing the UNI to an Intel type Microprocessor

The UNI can be interfaced to both "Intel"-type and "Motorola" type microprocessors/microcontrollers. The following sections will provide one example for each type of processor. This section discusses how to interface the XR-T7234 E3 UNI to the 8051 microcontroller.

The 8051 Microcontroller

The 8051 family of microcontrollers is manufactured by Intel and comes with a variety of amenities. Some of these amenities include:

- on chip serial port
- four 8 bit I/O port (P0 - P3)
- two 16 bit timers
- 4k bytes of ROM
- 128 bytes of RAM

Figure 6 presents a block diagram of the 8051 Mcrocontroller, and Figure 7 presents the pin out of this device.

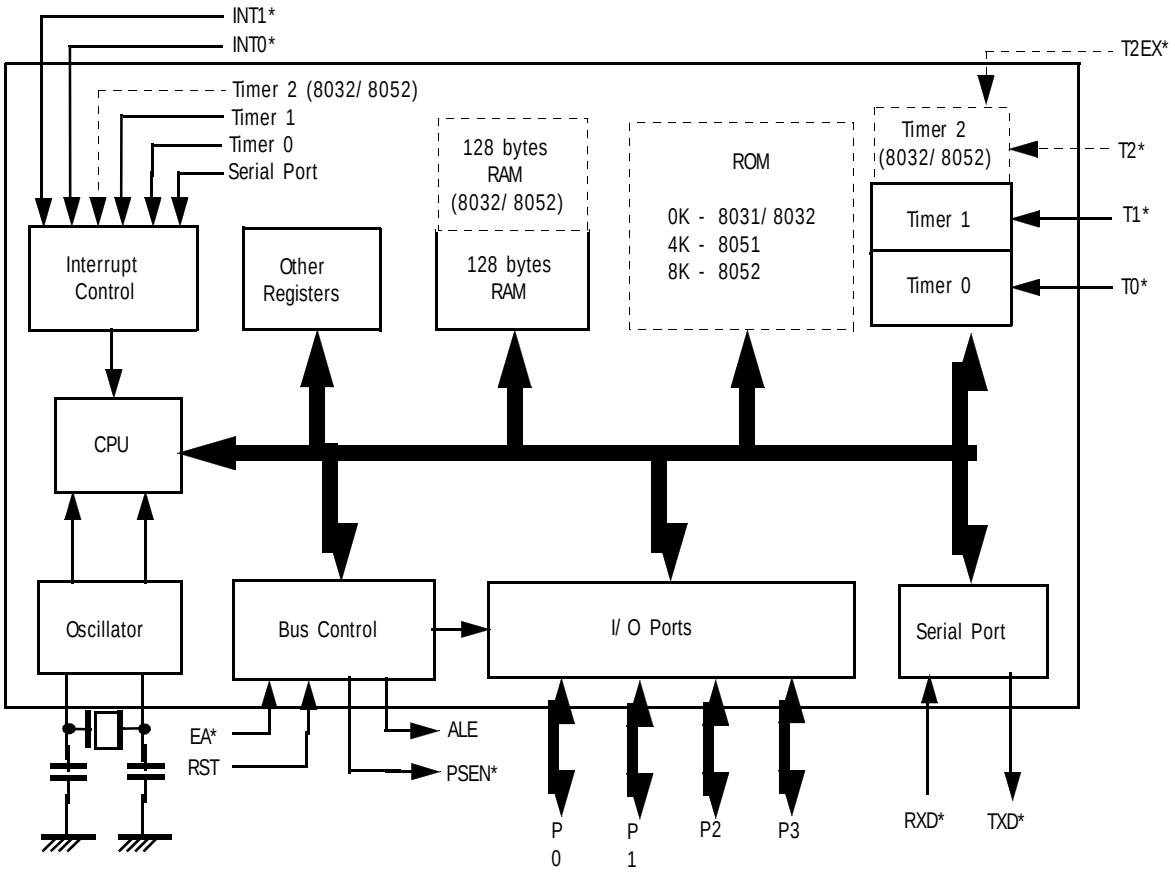

Figure 6. Block Diagram of the 8051 Mcrocontroller

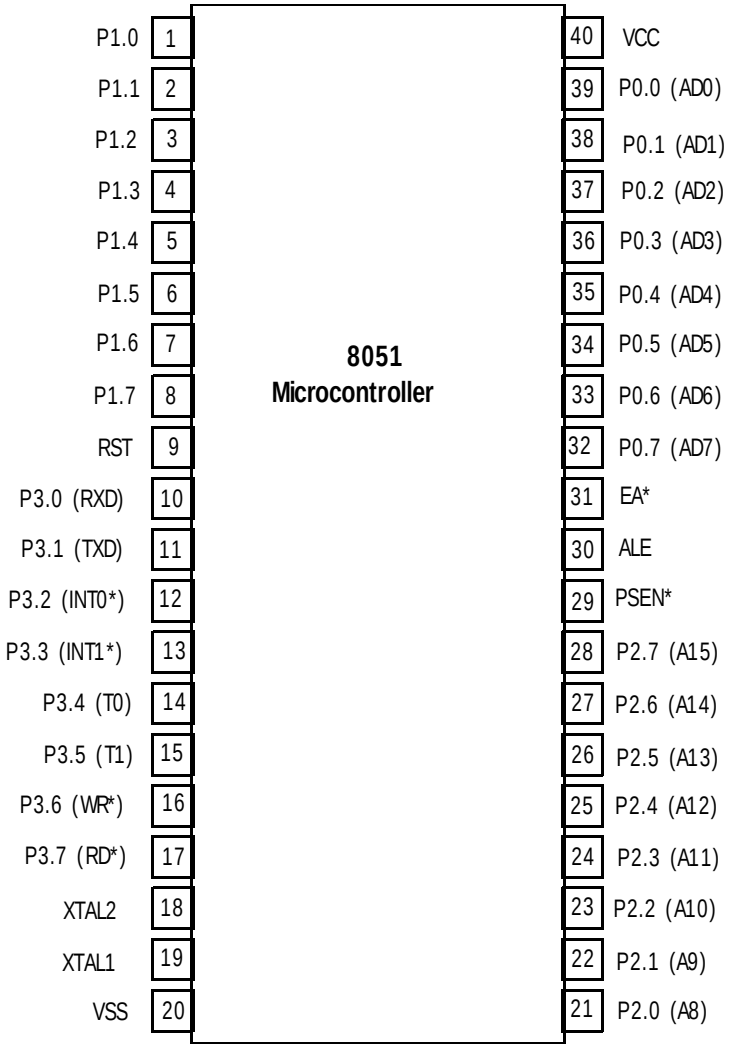

Figure 7. Pin out of the 8051 Mcrocontroller

The 8051 µC consists of 4 8- bit I/O ports. Some of these ports have alternate functions as will be discussed below.

Port 0 (P0.0 - P0.7)

This port is a dual- purpose port on pins 32 - 39 of the 8051 IC. In minimal component designs, it is used as a general purpose I/O port. For larger designs with external memory, it becomes a multiplexed address and data bus (AD0-AD7).

Port 1 (P1.0 - P1.7)

Port 1 is a dedicated port on pins 1 - 8. The pins, designated at P1.0, P1.1, P1.2,..., are available for interfacing as required. No alternative functions are assigned for Port 1 pins; thus they are used solely for interfacing external devices. Exceptions are the 8032/ 8052 ICs, which use P1.0 and P1.1 either as I/O lines or as external inputs to the third timer.

Port 2 (P2.0 - P2.7)

Port 2 (Pins 21-28) is a dual- purpose port that can function as general purpose I/O, or as the high byte of the address bus for designs with external code memory of more than 256 bytes of external data memory (A8 -

A15).

Port 3

Port 3 is a dual- purpose port on pins 10 - 17. In addition to functioning as general purpose I/O, these pins have multiple functions. Each of these pins have an alternate purpose, as listed in Table 8, below.

Table 8. Alternate Functions of Port 3 Pins

Bit	Name	Alternate Function
P3.0	RXD	Receive Data for Serial Port
P3.1	TXD	Transmit Data for Serial Port
P3.2	INT0*	External Interrupt 0
P3.3	INT1*	External Interrupt 1
P3.4	T0	Timer/ Counter 0 External Input
P3.5	T1	Timer/ Counter 1 External Input
P3.6	WR*	External Data Memory Write Strobe
P3.7	RD*	External Data Memory Read Strobe

The 8051 also has numerous additional pins which are relevant to interfacing to the XR- T7234 E3 UNI or other peripherals. These pins are:

ALE-Address Latch Enable

If Port 0 is used in its alternate mode (e.g., as the data bus and the lower byte of the address bus), then ALE is the signal that latches the address into an external register during the first half of a memory cycle. Once this is done, the Port 0 lines are then available for data input or output during the second half of the memory cycle, when the data transfer takes place.

INT0 (P3.2) and INT1* (P3.3)*

INT0* and INT1* are external interrupt request inputs to the 8051 µC. Each of these interrupt pins support "direct interrupt" processing. In this case, the term "direct" means that if one of these inputs are asserted, then program control will automatically branch to a specific (fixed) location in code memory. This location is determined by the circuit design within the 8051 µC IC and cannot be changed. Table 9 presents the location (in code memory) where the program control will branch to, if either of these inputs are asserted.

Table 9. Interrupt Service Routine Locations (in Code Memory) for INT0* and INT1*

Interrupt	Location
INT0*	0003H
INT1*	0013H

Therefore, if the user is using either one of these inputs as an interrupt request input, then the user must insure that the appropriate interrupt service routine or unconditional branch instruction (to the interrupt service routine) is located at one of these address locations.

If the 8051 µC is required to interface to external components in the data memory space of sizes greater than 256 bytes, then both Ports 0 and Port 2 must be used as the address and data lines. Port 0 will function as a multiplexed address/ data bus. During the first half of a memory cycle, Port 0 will operate as the lower address byte. During the second half of the memory cycle, Port 0 will operate as the bi- directional data bus. Port 2 will be used as the upper address byte. ALE and the use of a 74HC373 transparent latch device can be used to demultiplex the Address and Data bus signals.

Figure 8 presents a schematic illustrating how the XR- T7234 E3 UNI can be interfaced to the 8051 µC.

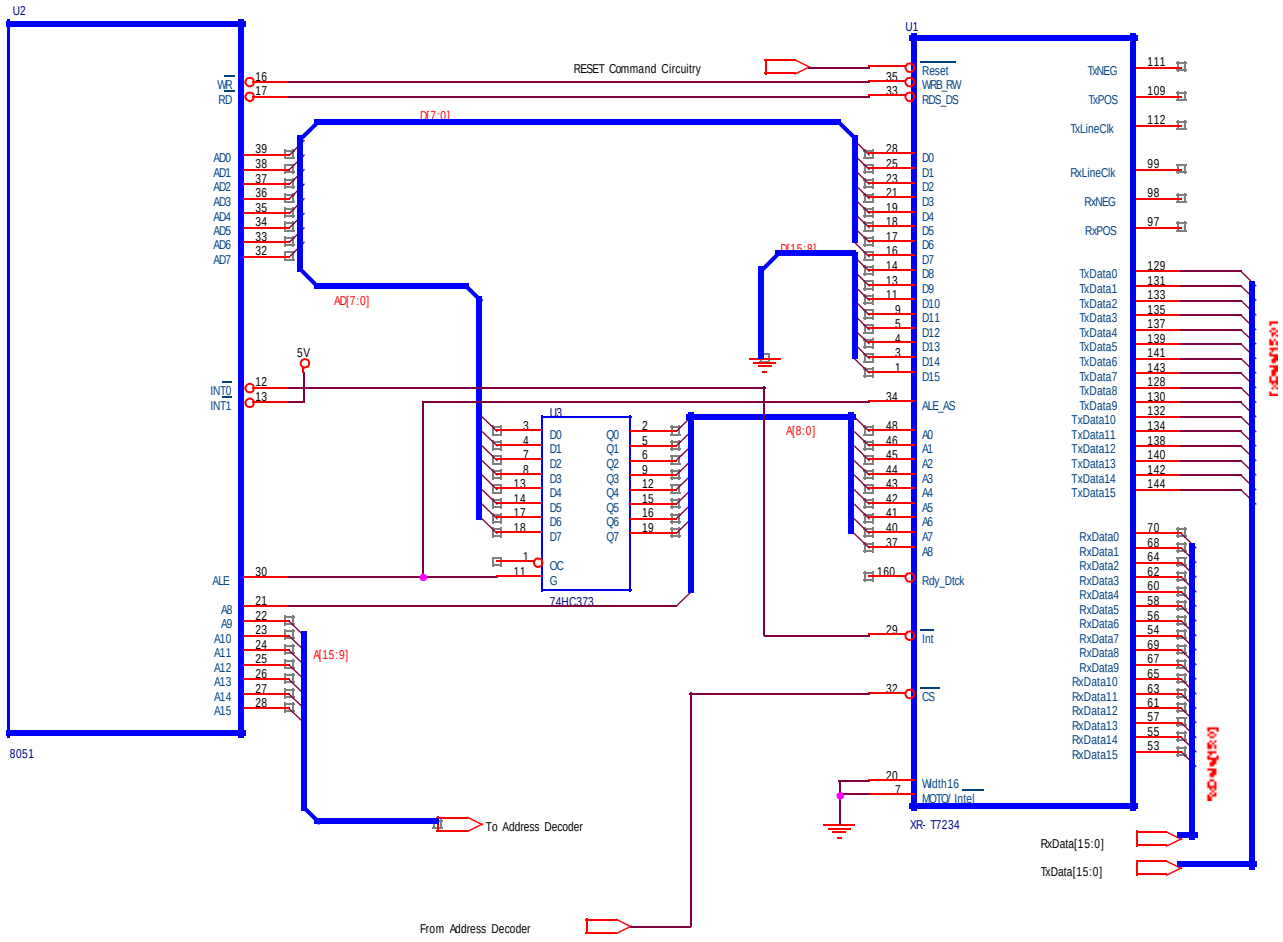

Figure 8. Schematic Depicting how to Interface the XR- T7234 E3 UNI to the 8051 Microcontroller

The circuitry in Figure 8 will function as follows during a UNI requested interrupt. The UNI device would request an interrupt from the CPU by asserting its "active- low" INT* output pin. This will cause the INT0* input pin of the CPU to go "low". When this happens the 8051 CPU will finish executing its current instruction, and will then branch pro-

gram control the UNI interrupt service routine. In the case of Figure 8, the interrupt service routine will be located in 0003H in code memory. The 8051 CPU does not issue an Interrupt Acknowledge signal back to the UNI. It will just begin processing through the UNI's interrupt service routine. Once the CPU has eliminated the cause(s) of the interrupt request, the UNI's INT* pin will be negated (go "high") and the CPU will return from the interrupt service routine and resume normal operation.

3.8 Interfacing the UNI to a Motorola type Microprocessor

This section discusses how to interface the XR- T7234 E3 UNI to the 68000 microprocessor.

Figure 9 presents a schematic on how to interface the XR- T7234 E3 UNI to the MC68000 microprocessor, over a 16 bit data bus.

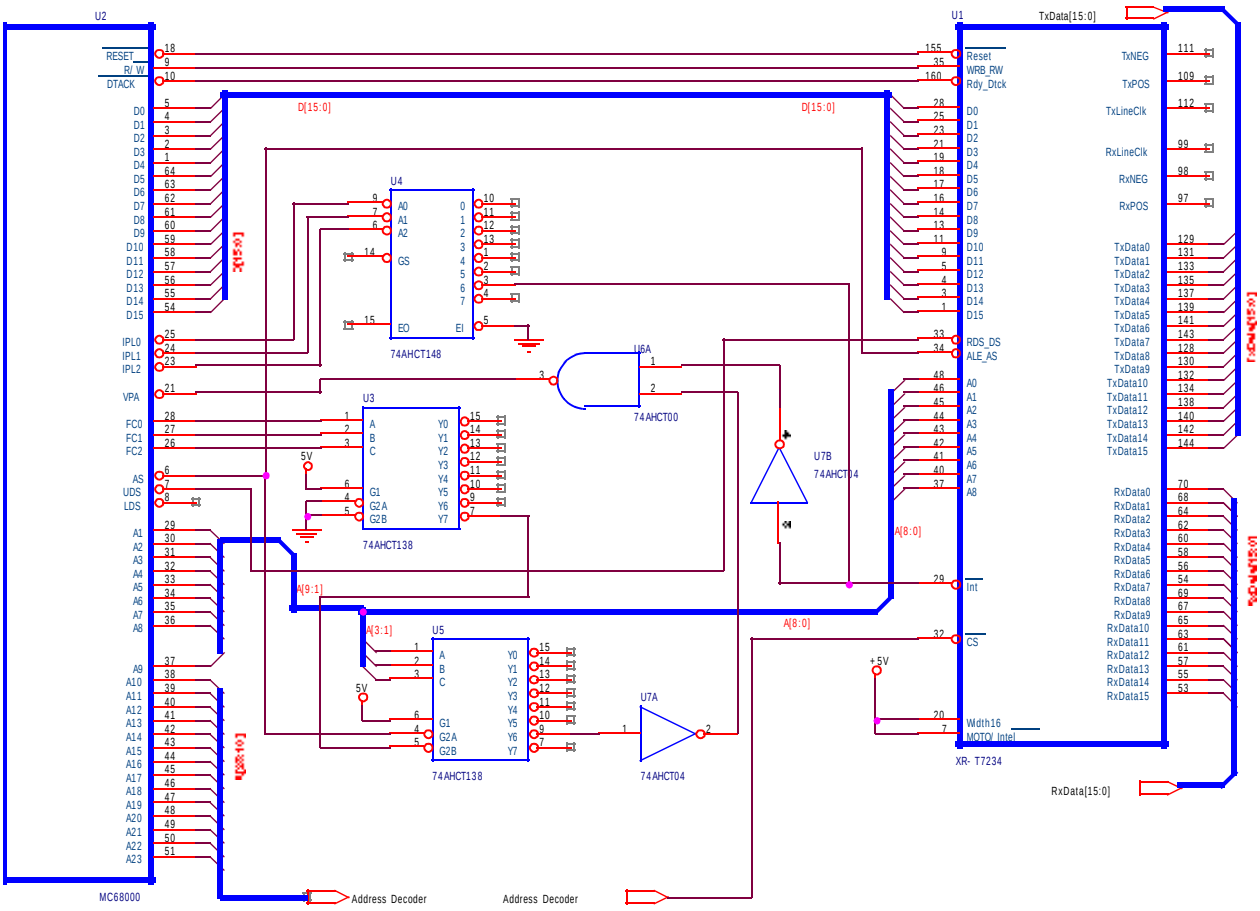

Figure 9. Schematic Depicting how to Interface the XR- T7234 E3 UNI to the MC68000 Microprocessor.

In general, the approach to interfacing these two devices is pretty straightforward. However, the user must be aware of the fact that the XR- T7234 E3 UNI does not provide an interrupt vector to the MC68000, during an "Interrupt Acknowledge" Cycle. Therefore, the user must configure his/ her design to support "auto- vectored" interrupts. "Auto-

vectored" interrupt processing is a feature offered by the MC68000 Family of microprocessors, where, if the microprocessor knows (prior to any IACK cycle) the "Interrupt Level" of this current interrupt, and that the "interrupting" peripheral does not support vectored interrupts, then the µP will generate its own Interrupt Vector. The schematic shown in Figure 9, has been configured to support "auto- vectored" interrupts.

Functional Description of Circuit in Figure 9

When the XR- T7234 E3 UNI generates an interrupt, the INT* output will toggle "low". This will force Input 6, of the "Interrupt Priority Encoder" chip (U4), to also toggle "low". In response to this, the Interrupt Priority Encoder chip will set its three outputs to the following states: A2 = '0', A1 = '0' and A0 = '1' (which is the number 6 in "inverted" binary format). The state of three output pins will be read by the "active- low" interrupt request inputs of the µP (IPL2, IPL1, and IPL0). When the 68000 µP detects this value at its three interrupt request inputs, it will know two things:

1. An interrupt request has been issued by one of the peripheral devices
2. The interrupt request is a "Level 6" interrupt request (due to the values of the A2 - A0 outputs from the "Interrupt Priority Encoder" IC).

Once the 68000 µP has determined these two things it will initiate an "Interrupt Acknowledge" (IACK) cycle by doing the following:

1. Identify this new bus cycle as an interrupt service routine by setting all of its Function Code output pins (FC2-FC0) "high".
2. Placing the Interrupt Level on the Address bus output pins A[3:1].

When the 68000 µP has toggled all of its Function Code output pin "high", the "Function Code Decoder" chip (U3) will read this value from the FC2 - FC0 pins as being the binary value for 7. As result, U3 will assert its "active- low" Y7 output pin. At the same time, the address lines A[3:1] are carrying the current "Interrupt Level" of this IACK cycle (level = 6, or '1 1 0' in this example) and applying this value to the A, B, and C inputs of the "IACK Level Decoder" chip (U5). Initially, all of the outputs of U5 are "tri- stated." Due to the fact that its "active- low" G2A and G2B inputs are negated (e.g., at a logic "high"). However, when the 68000 µP begins the IACK cycle, it will assert its Address Strobe (AS*) signal. This action will result in asserting the G2A input pin of U5. Additionally, since the Function Code Decoder chip has also asserted its Y7 output pin this will, in turn assert the G2B input pin of U5. At this point, the output of U5 will no longer be "tri- stated". U5 will read in the contents of its A, B, and C inputs, and assert the appropriate output pin. In this case, since U5 has the binary value of "6" applied to its input, it will, in turn assert its "active- low" Y6 output pin. The combination of the IntB* output (of the XR- T7234) and Y6 (from U5) being asserted will cause U6A to assert the active- low VPA* (Valid Peripheral Address) input pin of the 68000 µP. Anytime the 68000 detects its VPA* pin being asserting during an IACK cycle, it know that this is an Auto- vectored Interrupt Cycle. When the 68000 is operating in an "Auto- vectored" Interrupt Cycle, it knows that it will not receive an interrupt vector from the peripheral device (e.g., the XR- T7234 UNI in this case), and that it must generate its own vector. In the very next bus cycle, the 68000 µP is going to implement a "pseudo- read" of the data bus. However, in reality no data will be read from the XR- T7234 device. The 68000 µP will instead have determined that since this current IACK is an Auto- vectored-Level 6 Interrupt cycle; which corresponds to Vector Number 30, within the 68000 µP's Exception Vector Table. Vector Number 30 corresponds to an Address Space of 78h, in the 68000 µP's address space. In the case of this example, the user is required to place an unconditional branch statement (to the location of the XR- T7234 Interrupt Service Routine) at 78h in system level memory.

Table 10 presents the Auto- vector Interrupt Table (e.g., the relationship between the Interrupt Level and the corre-

sponding location in memory for this unconditional branch statement) for the MC68000 µP.

Table 10. Auto- vector Interrupt Table for the MC68000 Microprocessor

Interrupt Level	Vector Number	Address Space
1	25	064h
2	26	068h
3	27	06Ch
4	28	070h
5	29	074h
6	30	078h
7	31	07Ch

4.0 The UNI Test and Diagnostic Section

The "Test and Diagnostic" Section, within the XRT7234 E3 UNI offers a significant amount of on-chip "self-test" capability. This "self-test" capability of the XRT7234 E3 UNI is briefly itemized below.

- The XRT7234 E3 UNI can be configured to operate in one of two loopback modes: Line and Cell.
- The UNI contains an on-chip Test Cell Generator that is capable of generating Test Cells with "user-specified" header byte patterns. The Test Cell Generator also uses an "on-chip" PRBS generator to fill in the bytes for the payload portion of these test cells.
- The "Test and Diagnostic" section, within the UNI allows the user to route these test cells through the UNI, while operating in the "Line-side" test mode.
- The Test and Diagnostic section includes a Test Cell Receiver that is capable of identifying, terminating, and evaluating the "post-loopback" test cells.
- The Test Cell Receiver will also report the occurrence of errors, by incrementing an on-chip "Test Cell Error Accumulation" register.

The UNI chip's Test and Diagnostic Section allows the user to run a wide variety of diagnostic tests, based upon his/her selection of the following three parameters:

- The type of Loopback modes
- Line Side or System Side Testing
- Test Cell Generator/Receiver configuring

Each of these "Test and Diagnostic" parameters are discussed in detail, below.

4.1 The UNI Chip's Loopback Modes

The XRT7234 E3 UNI IC allows the user to configure it into one of two loopback modes:

- Line Loopback Mode
- Cell Loopback Mode

The following sections define each of these loopback modes, and discusses how to configure the UNI to operate in these modes.

4.1.1 The "Line Loopback" Mode

When the UNI is operating in the Line Loopback Mode, the output of the Transmit E3 Framer (e.g., TxPOS and TxNEG) will be internally routed to the input pins of the Receive E3 Framer (e.g., RxPOS and RxNEG). Figure 10 presents an illustration of the UNI chip operating in the Line Loopback mode.

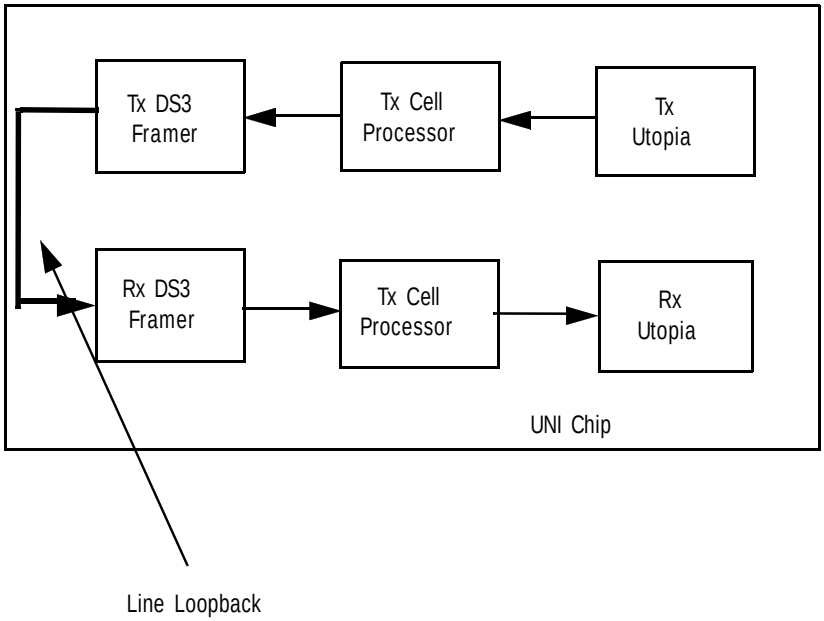

Figure 10. Illustration of the UNI operating in the Line Loopback Mode.

The user can configure the UNI chip to operate in the Line Loopback mode, by writing the appropriate data to the UNI Operating Mode Register (Address = 00h), as depicted below.

UNI Operating Mode Register (Address = 00h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
	Disable LOC	Internal LOS Enable	Reset by Reg.	Cell Loop-back	Line Loopback	TimRefSel[1, 0]	
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "1" to bit 2 of the UNI Operating Mode Register, configures the UNI to operate in the "Line Loopback" mode. Writing a "0" to this bit- field disables the Line Loopback mode.

4.1.2 The Cell Loopback Mode

When the UNI is configured to operate in the "Cell Loopback" Mode, then ATM cells, that are delineated and pass through the Receive Cell Processor will be routed directly (internally) the Tx FIFO (within the Transmit Utopia Interface block). Once these cells arrive at the TxFIFO, they will be read- in and further processed by the Transmit Cell Processor. These cell will ultimately be routed onto the "outbound" E3 line via the Transmit E3 Framer. Figure 11, presents an illustration of the UNI chip operating in the "Cell Loopback" Mode.

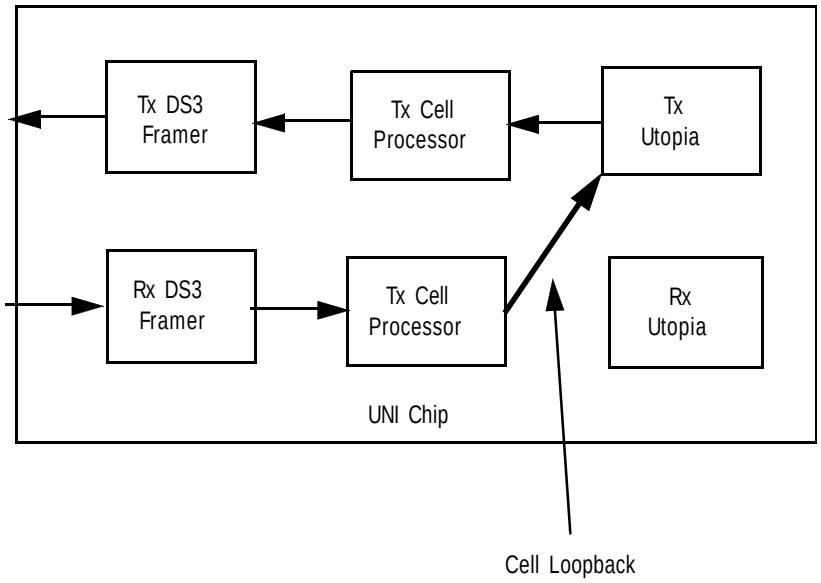

Figure 11. An Illustration of the UNI chip operating in "Cell Loopback" Mode.

The user can configure the UNI chip to operate in the "Cell Loopback" mode, by writing the appropriate data to the UNI Operating Mode Register (Address = 00h), as depicted below.

UNI Operating Mode Register (Address = 00h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	Disable LOC	Internal LOS Enable	Reset by Reg.	Cell Loopback	Line Loopback	TimRefSel[1, 0]	
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "1" to bit 3 of the UNI Operating Mode Register, configures the UNI to operate in the "Cell Loopback" mode. Writing a "0" to this bit-field disables the "Cell Loopback" mode.

4.2 Line-Side/System-Side Tests

The current version of the XRT7234 E3 UNI chip supports "Line-Side" Testing, but not "System-Side" testing. However, for the sake of completeness, both of these test modes are briefly discussed below.

Future versions of the UNI Chip will allow the user to generate test cells and run tests in either the "Line Side" Mode or in the "System Side" Mode.

4.2.1 Line-Side Tests

In "Line Side" Testing, the UNI chip will generate some test cells, and will transmit these cells either on or out towards the E3 "Line" (hence the name "Line-side" tests). At some point, these test cells will be looped-back into the Receive Path, where they will ultimately be terminated, and evaluated by the Test Cell Receiver. These Line-Side tests are intended to be conducted while the UNI is operating in the "Line-loopback" mode (see Section 4.1). However, the Line Side tests can also be conducted while the system (external to the UNI device), implements an "External

Loopback" mode. In this case, no UNI loopback mode would be configured, and the user's system would implement this "External Loopback" by routing the "Transmit E3 Line data" (from the UNI), back into the RxPOS and RxNEG inputs of the UNI device.

Note: The Cell Loopback Mode cannot be used in the "Line- Side" Tests.

If the user selects a "Line Side" test, then the "Test Cell Generator" will generate and insert test cells into the TxFIFO (within the UNI). These test cells will be read out of the TxFIFO by the Transmit Cell Processor, and are ultimately be transmitted out onto the E3 line. Eventually (depending upon the type of Loopback chosen), these test cells will be routed back into the "Receive Path" of the UNI device. Once in the "Receive Path", those test cells that reach the RxFIFO will be identified by their header byte patterns, and collected by the "Test Cell Receiver" where they will be checked for bit errors. The features and characteristics of the Test Cell Generator and the Test Cell Receiver is discussed in detail in Section 4.3.

The configuration for the Line- side Test, while the UNI is configured to operate in the Line and "External" Loopback modes are illustrated in Figures 12 and 13, respectively.

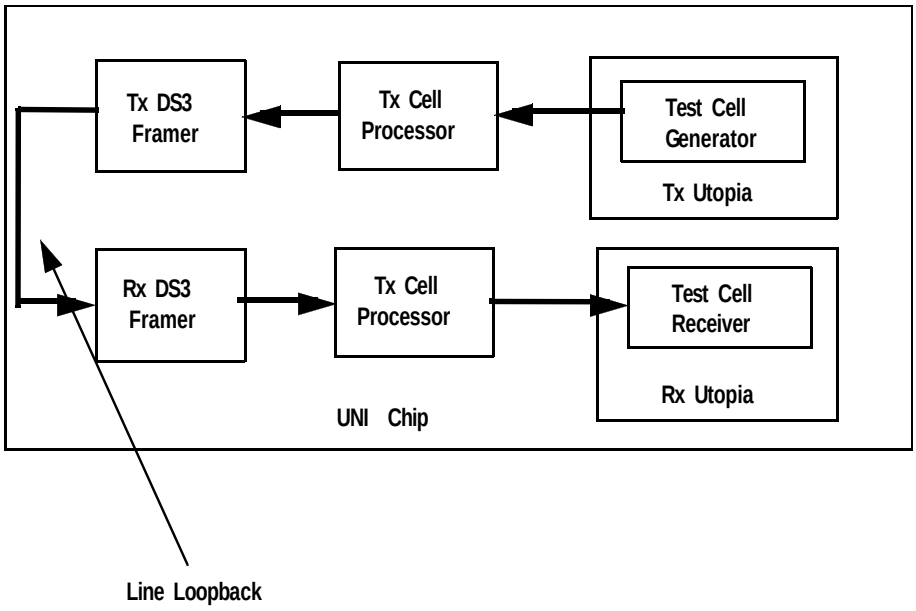

Figure 12. Illustration of Line Side Test, while the UNI is Configured to operate in 'Line Loopback" Mode.

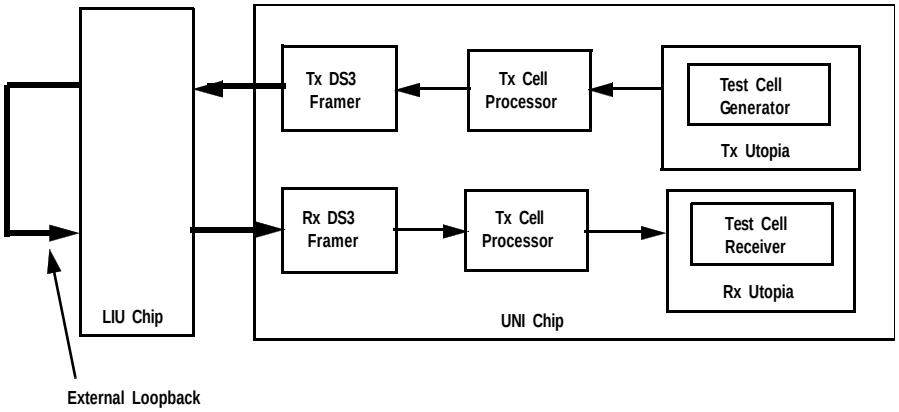

Figure 13. Illustration of Line Side Test, while the UNI is configured to operate in the "External Loopback" Mode.

4.2.2 "System- Side" Testing (Not Presently Supported by the XR- T7234 E3 UNI)

If the user selects a "System side" test then the "Test Cell Generator" will generate and insert the test cells into the RxFIFO, where they can be read out and processed by the ATM Layer processor, via the Receive Utopia Interface block. At some point, these test cells will be looped back into the "Transmit Path" (of the UNI device), via some externally implemented "Utopia Loopback" mode. Once in the "transmit path", those test cells that reach the TxFIFO (of the UNI) will be identified by their header byte patterns, and collected by the "Test Cell Receiver" where they will be checked for bit errors. Figure 14 presents an illustration of a System Side Test Configuration, while the UNI System is operating in a "Utopia Loopback Mode", via the ATM Layer Processor.

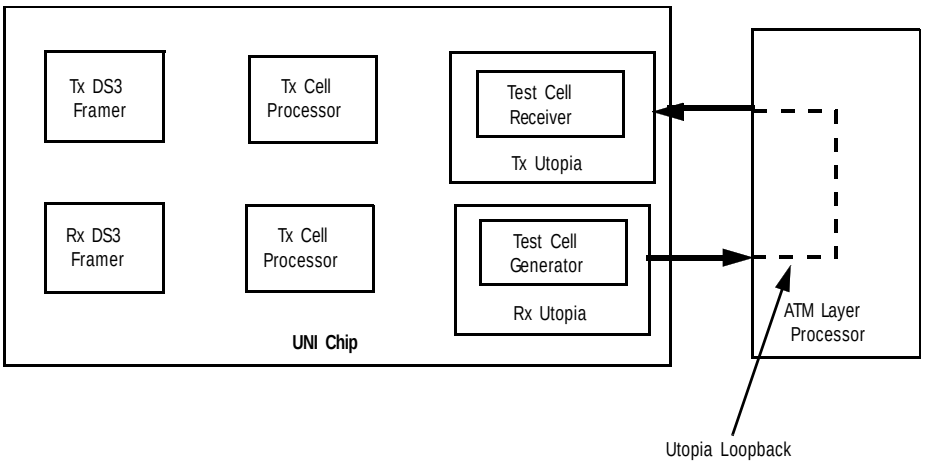

Figure 14. Illustration of System Side Test, while the UNI System is configure to operating in Utopia Loopback Mode.

Note:

1. The System- side test is not supported by this version of the XR- T7234 E3 UNI IC.
2. The "Utopia Loopback" mode, as depicted in Figure 14, must be implemented by the user's system level hardware.

4.3 Operating the Test Cell Generator/ Receiver

Sections 4.1 discussed the various loopback modes that are available within the UNI device. Section 4.2 discussed Line Side Testing; where the "internal" Test Cell Generator can be enabled to produce test cells, and write them into the TxFIFO. These cells will be read out of the TxFIFO by the Transmit Cell Processor, and routed towards the E3 line. Section 4.1 and 4.2 also mentioned that these cells could be "looped" back into the Receive path (of the UNI), at various points depending upon the Loopback Mode selected. When operating the UNI in the Line Side Test Mode, then the following loopback options are available.

- Line Loopback
- External Loopback (see Figure 13)

As these test cells proceed through the Receive path (after traversing the loopback point), they will eventually arrive at the RxFIFO (within the Receive Utopia Interface); where they will be identified, collected and analyzed by the Test Cell Receiver.

The next two sections discuss the operation of the Test Cell Generator and the Test Cell Receiver, within the UNI Test and Diagnostic Section.

4.3.1 Characteristics of the Test Cell Generator

The Test Cell Generator has the following characteristics:

- It allows the user to specify the header byte patterns of these test cells.
- The payload bytes within these test cells will be filled by an internal Pseudo- Random Byte Sequence (PRBS) Pattern Generator.
- It allows the user to select one of two "Test Cell Traffic" generating options. These options are.
  - The "One Shot" Mode
  - The "Continuous" Mode
- It generates Test Cells and writes them into the TxFIFO (within the Transmit Utopia Interface block); where they will be read from and processed throughout the UNI.

Each of these characteristics of the Test Cell Generator are described in greater detail below.

4.3.1.1 Specifying the Header Byte Pattern of Test Cells

The user can specify his/her choice for the header byte patterns of these Test Cells, by writing the "desired" header byte patterns into the "Test Cell Header Byte" registers 1 through 4 (Address = 08h through 0Bh); as depicted below.

Test Cell Header Byte- 1 Register (Address = 08h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Header Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Test Cell Header Byte- 2 Register (Address = 09h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Header Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Test Cell Header Byte- 3 Register (Address = 0Ah)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Header Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Test Cell Header Byte- 4 Register (Address = 0Bh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Test Cell Header Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Writing the Test Cell Header bytes into these registers accomplishes two things:

a. It configures the Test Cell Generator to produce Test Cells with these header byte patterns; and

b. It informs the Test Cell Receiver of the "header bytes" patterns of these Test Cells, in order to help it to identify and collect these cells for error- checking purposes.

4.3.1.2 The Payload Bytes of the Test Cells

The Test Cell Generator will automatically fill the payload portion of these Test Cells with bytes that are generated by an internal Pseudo- Random Byte Sequence (PRBS) generator. These PRBS generated bytes will ultimately used by the Test Cell Receiver in order to perform error- checking of the "post- routed" Test Cells.

4.3.1.3 Test Cell Generator–Test Cell Traffic Options

The user can configure the Test Cell Generator to generate "Test Cells" based upon one of two traffic options: The "One Shot" Mode, or the "Continuous" Mode. The user can make this selection by writing the appropriate value to Bit 2 of the Test Cell Control and Status Register (Address = 06h); as depicted below.

Test Cell Control and Status Register (Address = 06h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			Test Cell Enable	Line/ System	One Shot Test	One Shot Done	PRBS Lock

Test Cell Control and Status Register (Address = 06h)							
R/W	R/W	R/W	R/W	R/W	R/W	RO	RO

The user can configure the "Test Cell Generator" to operate in the "One Shot" mode by writing a "1" into this bit- field. Conversely, the user can configure the "Test Cell Generator" to operate in the "Continuous" Mode by writing a "0" into this bit- field.

4.3.1.3.1 The "One Shot" Mode

If the Test Cell Generator is configured to operate in the "One Shot" mode, then upon enabling the Test Cell Generator (by inducing a "0" to "1" transition in the "Test Cell Enable" field of this register), the Test Cell Generator will generate a "single burst" of 1024 test cells. Afterwards, the Test Cell Generator will stop and will not produce any more test cells until the next "0" to "1" transition in the "Test Cell Enable" bit- field.

If the Test Cell Generator is operating in the "One Shot" Mode, the user can determine its "progress" in its "Test Cell" production by reading Bit 1 (One Shot Done) within the "Test Cell Control and Status Register"; as depicted below.

Test Cell Control and Status Register (Address = 06h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			Test Cell Enable	Line/ System	One Shot Test	One Shot Done	PRBS Lock
R/W	R/W	R/W	R/W	R/W	R/W	RO	RO

If this "Read- Only" bit- field contains a "1" then the Test Cell Generator has completed its generation of the latest burst of 1024 Test Cells. However, if this bit- field contains a "0", then generation of this burst of Test Cells is still in-process.

Note: The contents within Bit 1 (One Shot Done) is only relevant if the user is operating the Test Cell Generator in the "One Shot" Mode.

4.3.1.3.2 The "Continuous" Mode

If the Test Cell Generator is configured to operate in the "Continuous" mode, then upon enabling the Test Cell Generator (by writing a "1" into the "Test Cell Enable" bit- field in this register), the Test Cell Generator will produce a continuous stream of Test Cells for the duration that the Test Cell Generator is enabled.

4.3.1.4 Enabling the "Test Cell Generator"

The user can Enable the "production" of Test Cells by writing a "1" to Bit 4 (Test Cell Enable) within the Test Cell Control and Status Register, as depicted below.

Test Cell Control and Status Register (Address = 06h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***

Test Cell Control and Status Register (Address = 06h)							
Unused			Test Cell Enable	Line/ System	One Shot Test	One Shot Done	PRBS Lock
R/W	R/W	R/W	R/W	R/W	R/W	RO	RO

Writing a "1" to this bit- field enables the "Test Cell Generator/ Receiver". Writing a "0" disables the "Test Cell Generator/ Receiver".

Once the Test Cell Generator has been enabled, it will begin to either produce a continuous stream of cells (if configured to operate in the "Continuous" mode), or a single burst of 1024 cells (if configured to operate in the "One Shot" mode.)

4.3.2 Characteristics of the Test Cell Receiver

The Test Cell Receiver has the following characteristics:

- It checks the header bytes of all cells arriving at the RxFIFO (within the Receive Utopia Interface Block), in order to determine if they are (or are not) Test Cells.
- It reads in those cells that have identified as "Test Cells" from the RxFIFO
- Acquires and Maintains "PRBS Lock" with the payload data, within these Test Cells.
- Reports the occurrences of errors.

4.3.2.1 Identifying the Test Cells

The Test Cell Receiver will monitor those cells that reach the RxFIFO, within the Receive Utopia Interface block, and, from that "stream of incoming cells" identify and collect the test cells. The Test Cell Receiver will use the "user- specified" header byte patterns, as written into the "Test Cell Header Byte" registers - 1 through 4 (Address = 08h to 0Bh), in order identify these test cells.

4.3.2.2 Acquiring and Maintaining "PRBS Lock"

During Test Cell production, the Test Cell Generator will fill in the "payload portions" of each test cell with bytes that are generated by a "Pseudo- Random Byte Sequence (PRBS) generator. Consequently, the data within the cell payload bytes (of these test cells) follows a pre- defined sequence.

After the Test Cell Receiver has started to "collect" these test cells from the RxFIFO, it will "strip off" the cell header bytes and will begin to evaluate their payload bytes. One of the first things that the Test Cell Receiver will try to do is to look for this "pre- defined (PRBS) sequence" within this test cell payload data. Once the Test Cell Receiver has found this "pre- defined" sequence within the test cell payload data, it will inform the user of this fact by asserting the

"PRBS Lock" bit- field, within the "Test Cell Control and Status" Register; as depicted below.

Test Cell Control and Status Register (Address = 06h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused			Test Cell Enable	Line/ System	One Shot Test	One Shot Done	PRBS Lock
R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	RO	RO
			1	0	x	x	1

As long as the Test Cell Receiver has found (and continues to find) this pre- defined sequence in the incoming Test Cell payload data, it will keep this bit- field asserted (e.g., at a logic "1").

4.3.2.3 Evaluating the Test Cell Payload Data and Reporting Errors

Once the Test Cell Receiver has acquired "PRBS Lock" with the contents of the incoming Test Cell payload data; then it can begin to compare this data with the "pre- defined" PRBS pattern of data, as produced by the PRBS Generator (within the Test Cell Generator). If the Test Cell Receiver detects any discrepancies between the Test Cell Payload bytes, and the "pre- defined" PRBS pattern, then it will increment to the "Test Cell Error Accumulator" Registers. The "Test Cell Error Accumulator" register actually consists of two 8- bit "Reset upon Read" registers, as depicted below.

Test Cell Error Accumulator-MSB (Address = 0Ch)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Test Cell Errors-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR

Test Cell Error Accumulator-LSB (Address = 0Dh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Test Cell Errors-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR

These two register combine to form a 16- bit expression of the number of bit- errors that the Test Cell Receiver has detected within the payload bytes of the "incoming" test cells. The "Test Cell Error Accumulator-MSB" register contains the "upper" byte value of this 16- bit expression. Likewise, the "Test Cell Error Accumulator-LSB" register contains the "lower" byte value of this 16- bit expression.

Since these registers are "Reset upon Read", they contain the number of "test cell payload" bit errors, that have been detected, by the Test Cell Receiver, since the last read of these registers.

5.0 Line Interface Drive and Scan Section

The "Line Interface Drive and Scan" Section, of the XR- T7234 E3 UNI consists of 5 output pins, three input pins, a "Read/ Write" register, and a "Read Only" register.

The purpose of the "Line Interface Drive and Scan" section, is to allow the user to monitor and exercise control over many aspects of the XR- T7295E E3 Line Receiver IC and the XR- T7296 E3 Line Transmitter IC, without having to develop the necessary "off- chip" glue logic.

Figure 15 presents a circuit schematic that depicts how the XR- T7234 E3 UNI should be interfaced to the XR- T7295E/ XR- T7296 E3 Line Interface Unit devices.

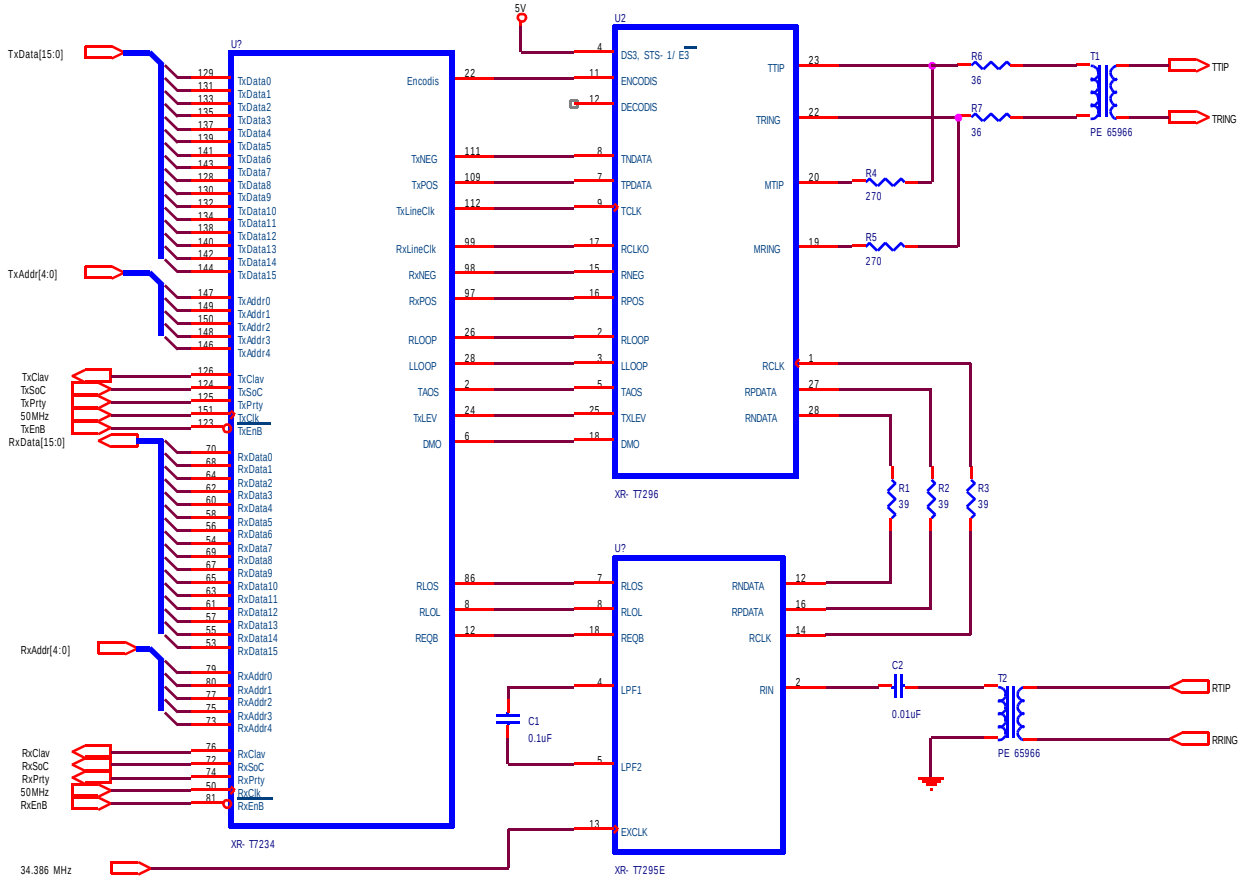

Figure 15. Circuit Schematic Illustrating how the XR- T7234 E3 UNI should be interfaced to the XR- T7295/ XR- T7296 E3 Line Interface Unit devices.

As mentioned above, the Line Interface Drive and Scan Section, consists of five output pins and three input pins. The logic state of the output pins are controlled by the contents within the "Line Interface Drive" Register, as depicted below.

Line Interface Drive Register (Address = 84h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		REQB	TAOS	Encodis	TxLev	RLoop	LLoop
RO	RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W

Line Interface Drive Register (Address = 84h)							
0	0	0	0	0	0	0	0

The role of each of these bit- fields and their corresponding output pins are depicted below.

The logic state of the input pins can be monitored by reading the contents of the "Line Interface Scan" Register, as depicted below.

5.1 Bit- Fields with the Line Interface Drive Register

Bit 5-RECQB-(Receive Equalization Bypass Control)

This "Read/ Write" bit- field allows the user to control the state of the "RECQB" output pin of the UNI device. This output pin is intended to be connected to the "RECQB" input pin of the XR- T7295E E3 Line Receiver IC. If the user forces this signal to toggle "high", then the XR- T7295E device will cause its newly received E3 line signal to "by- pass" some equalization circuitry within the XR- T7295E device. Conversely, if the user forces this signal to toggle "low", then the XR- T7295E device will route its newly received E3 line signal through its internal equalization circuitry.

Writing a "1" to this bit- field causes the UNI device to toggle the "RECQB" output pin "high". Writing a "0" to this bit- field causes the UNI device to toggle the "RECQB" output pin "low".

For information on the criteria that should be used when deciding whether to bypass the equalization circuitry or not, please consult the "XR- T7295E E3 Integrated Line Receiver" data sheet.

Note: If the customer is not using the XR- T7295E E3 Line Receiver IC, then he/ she can use this bit- field and the 'RECQB' output pin for other purposes.

Bit 4-TACS-(Transmit All Ones Signal)

This "Read/ Write" bit- field allows the user to control the state of the "TACS" output pin of the UNI device. This output pin is intended to be connected to the "TACS" input pin of the XR- T7296 E3 Line Transmitter IC. If the user forces this signal to toggle "high", then the XR- T7296 device will transmit an "All Ones" pattern onto the line. Conversely, if the user commands this output signal to toggle "low" then the XR- T7296 E3 Line Transmitter IC will proceed to transmit data based upon the pattern that it receives via the TxPOS and TxNEG output pins (of the UNI IC).

Writing a "1" to this bit- field will cause the "TACS" output pin to toggle "high". Writing a "0" to this bit- field will cause this output pin to toggle "low".

Note: If the customer is not using the XR- T7296 E3 Line Transmitter IC, then he/ she can use this bit- field, and the "TACS" output pin for other purposes.

Bit 3-Encodis-(HDB3 Encoder Disable)

This "Read/ Write" bit- field allows the user to control the state of the "Encodis" output pin of the UNI device. This output pin is intended to be connected to the "Encodis" input pin of the XR- T7296 E3 Line Transmitter IC. If the user forces this signal to toggle "high", then the "internal HDB3 encoder" (within the XR- T7296 device) will be disabled. Conversely, if the user commands this output signal to toggle "low", then the "internal HDB3 encoder" (within the XR- T7296 device) will be enabled.

Writing a "1" to this bit- field causes the UNI IC to toggle the "Encodis" output pin "high". Writing a "0" to this bit- field will cause the UNI IC to toggle this output pin "low".

Note:

1. The HDB3 encoder, within the XR- T7296 device, is not to be confused with the HDB3 encoding capabilities that exists within the Transmit E3 Framer block of the UNI IC.
2. The user is advised to disabled the HDB3 encoder (within the XR- T7296 IC) if the Transmit and Receive E3 Framers (within the UNI) are configured to operate in the HDB3 line code.
3. If the customer is not using the XR- T7296 E3 Line Transmitter IC, then he/ she can use this bit- field and the "Encodis" output pin for other purposes.

Bit 2 - TxLev - (Transmit Level Select Output)

This "Read/Write" bit-field allows the user to control the state of the "TxLev" output pin of the UNI device. This output pin is intended to be connected to the "TxLev" input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will increase the amplitude of its output signal on the line, in order to drive the signal over cable lengths of 225 ft or greater. Therefore, the user is recommended to set this output "high", if he/she is driving E3 lines signal over 225 ft (or more) of cable.

Conversely, if the user is driving E3 line signals over less than 225 ft of cable, then he/she is recommended to set this output pin "low".

Writing a "1" to this bit-field commands the UNI to toggle the TxLev output "high". Writing a "0" to this bit-field commands the UNI to toggle this output signal "low".

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the "TxLev" output pin for other purposes.

Bit 1 - RLOOP - (Remote Loopback)

This "Read/Write" bit-field allows the user to control the state of the "RLOOP" output pin of the UNI device. This output pin is intended to be connected to the "RLOOP" input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will start operating in the "Remote Loopback Mode". Conversely, if the user commands this signal to toggle "low", then the XR-T7296 device will be operating in the "Normal" mode.

Writing a "1" into this bit-field commands the UNI to toggle the "RLOOP" output signal "high". Writing a "0" into this bit-field commands the UNI to toggle this output signal "low".

For a detailed description of the XR-T7296 E3 Line Transmitter's operation during "Remote Loopback", please see the "XR-T7296 DS3/STS-1, E3 Integrated Line Transmitter's Data Sheet.

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the 'RLOOP" output pin for other purposes.

Bit 0 - LLOOP

This "Read/Write" bit-field allows the user to control the state of the "LLOOP" output pin of the UNI device. This output pin is intended to be connected to the "LLOOP" input pin of the XR-T7296 E3 Line Transmitter IC. If the user commands this signal to toggle "high", then the XR-T7296 E3 Line Transmitter IC will start operating in the "Local Loopback Mode". Conversely, if the user commands this signal to toggle "low", then the XR-T7296 device will be operating in the "Normal" mode.

Writing a "1" into this bit-field commands the UNI to toggle the "LLOOP" output signal "high". Writing a "0" into this bit-field commands the UNI to toggle this output signal "low".

For a detailed description of the XR-T7296 E3 Line Transmitter's operation during "Local Loopback", please see the "XR-T7296 E3/STS-1, E3 Integrated Line Transmitter's Data Sheet.

Note: If the customer is not using the XR-T7296 E3 Line Transmitter IC, then he/she can use this bit-field and the "LLOOP" output pin for other purposes.

5.2 Bit-Fields within the Line Interface Scan Register

Address = 85h, Line Interface Scan Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused					DMO	RLOL	RLOS
RO	RO	RO	RO	RO	RO	RO	RO

Address = 85h, Line Interface Scan Register							
0	0	0	0	0	0	0	0

The meaning/ role of each of these bit- field and their corresponding input pins are defined below.

Bit 2- DMO- (Drive Monitor Output)

This "Read- Only" bit- field indicates the logic state of the "DMO" input pin of the UNI device. This input pin is intended to be connected to the "DMO" output pin of the XR- T7296 E3 Line Transmitter IC. If this bit- field contains a logic "1", then the "DMO" input pin is "high". The XR- T7296 E3 Line Transmitter IC will set this pin "high" if the drive monitor circuitry (within the XR- T7296 device) has not detected any bipolar signals at the MTIP and MRING inputs (of the XR- T7296 device) within the last 128 ± 32 bit periods.

Conversely, if this bit- field contains a logic "0", then the "DMO" input pin is "high". The XR- T7296 E3 Line Transmitter IC will set this pin "low" if bipolar signals are being detected at the "MTIP" and "MRING" input pins.

Note: If this customer is not using the XR- T7296 E3 Line Transmitter IC, then he/ she can use this register bit- field and input pin for a variety of other purposes.

Bit 1- RLOL- (Receive Loss of Lock)

This "Read- Only" bit- field indicates the logic state of the "RLOL" input pin of the UNI device. This input pin is intended to be connected to the "RLOL" output pin of the XR- T7295E E3 Line Receiver IC. If this bit- field contains a logic "1", then the RLOL input pin is "high". The XR- T7295E E3 Line Receiver IC will set this pin "high" if the phase- locked- loop circuitry (within the XR- T7295E device) has lost "lock" with the incoming E3 data- stream and is not properly recovering clock and data.

Conversely, if this bit- field contains a logic "0", then the "RLOL" input pin is "low". The XR- T7295E E3 Line Receiver IC will hold this pin "low" as long as this "phase- locked- loop" circuitry (within the XR- T7295E device) is properly "locked" onto the incoming E3 data- stream, and is properly recovering clock and data from this data- stream.

For more information on the operation of the XR- T7295E E3 Line Receiver IC, please consult the "XR- T7295 E3 Integrated Line Receiver" data sheet.

Note: If the customer is not using the XR- T7295E E3 Line Receiver IC, then he/ she can use this bit- field, and the "RLOL" input pin for other purposes.

Bit 0- RLOS- (Receive Loss of Signal)

This "Read- Only" bit- field indicates the logic state of the "RLOS" input pin of the UNI device. This input pin is intended to be connected to the "RLOS" output pin of the XR- T7295E E3 Line Receiver IC. If this bit- field contains a logic "1", then the RLOS input pin is "high". The XR- T7295E device will toggle this signal "high" if it (the XR- T7295E E3 Line Receiver IC) is not detecting a sufficient amount of signal energy on the line, from the incoming E3 data- stream. This event indicates that the XR- T7295E device may be experiencing a Loss of Signal (LOS) condition.

Conversely, if this bit- field contains a logic "0", then the "RLOS" input pin is "low". The XR- T7295E device will hold this signal "low" if it is detecting a sufficient amount of signal energy on the line, from the incoming E3 data- stream.

For more information on the operation of the XR- T7295E E3 Line Receiver IC, please consult the "XR- T7295E E3 Integrated Line Receiver" data sheet.

Note: Asserting the "RLOS" input pin will cause the UNI device to declare an LOS (Loss of Signal) condition. Therefore, the user is advised not to use the "RLOS" input pin as a general- purpose input pin.

6.0 Transmit Section

The purpose of the Transmit section of the XRT7234 E3 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 a public or leased E3 transport medium.

The Transmit section of the E3 UNI chip consists of the following blocks:

- Transmit Utopia Interface
- Transmit Cell Processor
- Transmit E3 Framer

The ATM Layer processor will write ATM Cell Data into the Transmit Utopia Interface Block of the UNI device. 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 Tx FIFO throughout this document); where it can be read and further processed by the Transmit Cell Processor. The Transmit Utopia Interface block will also perform some parity checking on the data that it receives from the ATM Layer processor, and will provide signaling to support data-flow control between the ATM Layer Processor and the Transmit Utopia Interface block.

The Transmit Cell Processor block will read in the ATM cells from the Tx FIFO. It 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 Cell Processor will insert this HEC byte into the 5th octet position within the cell. The Transmit Cell Processor 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 the Transmit E3 Framer. If the Tx FIFO (within the Transmit Utopia Interface block) is depleted and has no (user) cells available, then the Transmit Cell Processor will automatically 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—34.368 Mbps in this case). The Transmit Cell Processor has provisions to allow the user to generate and transmit an OAM cell via software control.

Note: The 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 E3 Framer will take the ATM cells that it has received from the Transmit Cell Processor, and insert this data into the payload portions of the E3 frame. The Transmit E3 Framer will also generate and insert overhead bytes that support framing, performance monitoring (parity bits), path maintenance data link as well as alarm and status information originating from the (Near-End) Receiver section of this UNI. The purpose of these alarm and status information bits is to alert the far-end equipment that the (Near End) UNI Receiver has detected some problems in receiving data from it. The Transmit E3 Framer supports the ITU-T G.832 Framing Format.

The following sections discuss the blocks comprising the Transmitter Portion of the E3 UNI in detail.

6.1 TRANSMIT UTOPIA INTERFACE BLOCK

6.1.1 Brief Description of the Transmit Utopia Interface

The Transmit Utopia Interface Block provides a "Utopia Level 2" compliant interface that allows the ATM Layer or ATM Adaptation Layer processors to interconnect to the UNI device. The ATM Layer processor will write ATM cell data into the UNI via the Transmit Utopia Interface block. The Transmit Utopia Interface block is capable or receiving ATM cell data at data rates of up to 800 Mbps. This interface will support both an 8 and 16 bit wide data bus. Since the ATM Layer processor writes ATM cell data into the Transmit Utopia Interface block at clock rates independent of the line bit rate (in this case, E3), the received data (from the ATM layer processor) is written into an internal FIFO. This FIFO will be referred to as the Tx FIFO throughout this document. The contents of the Tx FIFO will be read-in and further processed by the Transmit Cell Processor. Data-flow control between the ATM Layer processor

and the Transmit Utopia Interface block is provided by the TxClav pin. Figure 16 presents a simple illustration of the Transmit Utopia interface block and the associated pins.

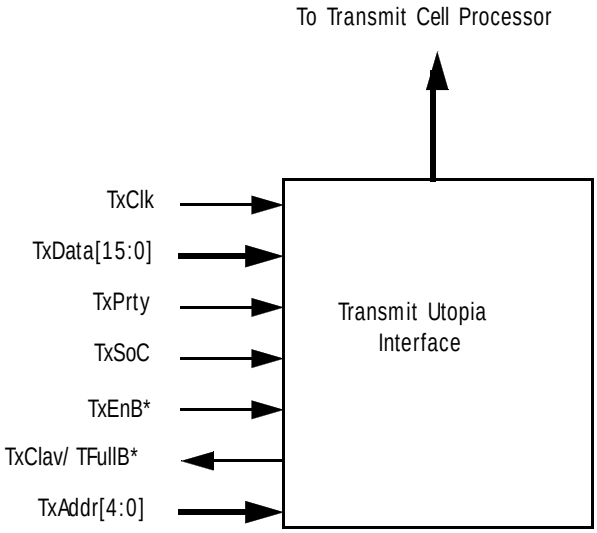

Figure 16. Simple Block Diagram of Transmit Utopia Interface

6.1.2 Functional Description of the Transmit Utopia Interface

The purposes of the Transmit Utopia interface block are to:

- Receive ATM cell data from the AAL or ATM Layer processor.
- Make these cells available to the Transmit Cell Processor block.
- Provide some form of flow control of cell data from the ATM Layer processor (via the TxClav output pin).
- Check the parity of the data received from the ATM Layer processor, with an option to discard errored cells.
- Detect and discard "Runt" cells, and resume normal operation afterwards.

The Transmit Utopia Interface block consists of the following sub- blocks.

- Transmit Utopia Input Interface
- Transmit Utopia Configuration/ Status Registers
- Transmit Utopia FIFO Manager
- Transmit Utopia Cell FIFO (Tx FIFO)

The Transmit Utopia Interface block consists of an input interface which complies to the "Utopia Level 2 interface specifications", and the Tx FIFO. The width of the Transmit Utopia data bus is user- configurable to 8 or 16 bits. The incoming data bytes or words (16 bits) are checked for odd- parity. The computed parity bit is then compared with that presented at the TxPrty input pin, while the corresponding data byte [word] is present at the TxData[15:0] input. Interrupts are generated upon error conditions. Cells with parity error may be dropped if enabled through a register setting.

The Transmit Utopia Interface block can be configured to process 52, 53, or 54 bytes per cell. If the Transmit Utopia Interface block detects a "runt" cell (e.g., a cell that is smaller than what the Transmit 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 depth of the TxFIFO is sixteen cells with the operating FIFO depth user- configurable to four, eight, twelve or sixteen cells by register settings. The incoming data (from the ATM Layer processor) is written into the Tx FIFO where it can be read- in and further processed by the Transmit Cell Processor. A FIFO manager maintains the

Tx FIFO and indicates FIFO empty, FIFO full, cell space available, etc. Figure 17 presents a functional block diagram of the Transmit Utopia Interface Block.

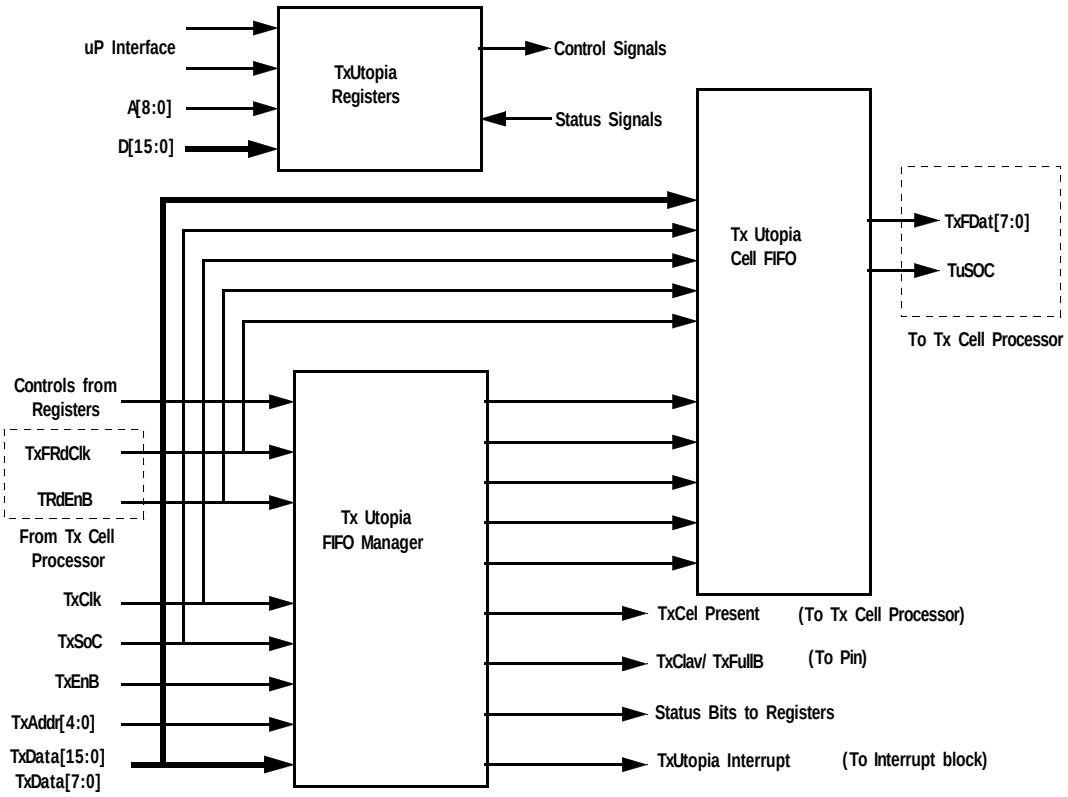

Figure 17. Functional Block Diagram of the Transmit Utopia Block

The following sections discuss each functional sub- block of the Transmit Utopia Interface Block in detail. These sections will discuss the many features associated with the Transmit Utopia Interface block as well as how the user can select/ configure these features in order to suit his/ her application needs. Detailed discussions of Single- PHY and Multi- PHY operation will each be presented in their own section even though it involves the use of all of these functional blocks.

6.1.2.1 Transmit Utopia Bus Input Interface

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

- 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)
- The handling of errored cells received from the ATM Layer processor

A discussion of the operation of the Transmit Utopia Bus Interface along with each of these options will be presented below.

6.1.2.1.1 The Pins of the Transmit Utopia Bus Interface

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

- TxData[15:0]–Transmit Utopia Data Bus Input pins

- TxAddr[4:0]–Transmit Utopia Address Bus Input pins
- TxClk–Transmit Utopia Interface block clock input pin
- TxSoC–Transmit "Start of Cell" indicator input pin
- TxPrty–Transmit Utopia–Odd Parity Input pin
- TxEnB*–Transmit Utopia Data Bus–Write Enable input pin
- TxClav/ TFullB*–TxFIFO Cell Available

Each of these signals are briefly discussed below.

TxData[15:0] –Transmit Utopia Data Bus input pins

The ATM Layer Processor will write its ATM Cell Data into the Transmit Utopia Interface block, by placing it, in a byte- wide (or 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" modes (See Section 6.1.2.1.2). If the "8- bit wide" mode is selected, then only the TxData[7:0] input pins are active and capable of receiving data. If the "16- bit wide" mode is selected, the all 16 input pins (e.g., TxData[15:0]) are active. The Transmit Utopia Data bus is tri- stated while the active- low TxEnB* (Transmit Utopia Data Bus–Write Enable) input signal is "high". Therefore, the ATM Layer processor must assert this signal (e.g., toggling TxEnB* "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, TxClk.

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 CellOf52Bytes option–See Section 6.1.2.1.3), following the latest assertion of the TxSoC (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 TxSoC pin. Once the Transmit Utopia Interface block reaches this condition, it will ignore the assertions of the TxEnB* pin, and will keep the Transmit Utopia Data bus input pins tri- stated until the ATM Layer Processor pulses the TxSoC 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 TxFIFO is less than that configured via the "CellOf52Bytes" option), then the Transmit Utopia Interface block will discard this cell, and resume normal operation.

Note: The 100 pin version of the XR- T7234 device allocates only 8 pins for the Transmit Utopia Data Bus: TxData[7:0]. The Transmit Utopia Data bus pins: TxData[15:8] are not available.

TxAddr[4:0]–Transmit Utopia Address Bus input pins

These input pins are used only when the UNI is operating in the Multi- PHY mode. Therefore, for more information on the Transmit Utopia Address Bus, please see Section 6.1.2.3.2.

TxClk–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 and the Transmit Utopia Address bus (for Multi- PHY operation) into the Transmit Utopia Interface block. This clock signal can run at frequencies of 25 MHz, 33 MHz, or 50 MHz.

TxEnB–Transmit Utopia Data Bus–Write Enable input pin*

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.

TxPrty–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 TxPrty pin, while the corresponding byte (or word) is present on the Transmit Utopia Data Bus.

TxSoC- Transmit Utopia "Start of Cell" Indicator input pin

The ATM Layer processor is expected to pulse this signal "high", for one clock period of TxClk, when the first byte (or word) of a new 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 TxSoC 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 "CellOf52Bytes" option- See Section 6.1.2.1.3) 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 TxSoC pin has been pulsed "high" once again. This phenomenon is more clearly defined in "Example- 1" below.

Further, if the ATM Layer Processor pulses the TxSoC pin, before the appropriate number of bytes (as configured via the "CellOf52Bytes" option- See Section 6.1.2.1.3), have been read in and processed by the Transmit Utopia Interface block, then a "runt" cell will have been detected. Whenever the Transmit Utopia Interface block detects a "runt" cell, it will generate a "Change in Cell Alignment" interrupt and will discard the "runt" cell. This phenomenon is more clearly defined in "Example- 2" 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 TxSoC 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, until the TxSoC pin has been asserted.

Example- 2

If the ATM Layer processor were to prematurely assert the TxSoC pin, (e.g., when the 52nd byte is present on the Transmit Utopia data bus, then the Transmit Utopia Interface block will interpret the previous 52 bytes of cell data as a "runt" cell. The Transmit Utopia Interface block will then generate a "Change of Cell Alignment" interrupt and will proceed to discard this runt cell.

*TxClav/ 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 6.1.2.2.1 for more information regarding this signal.

6.1.2.1.2 Selecting the Utopia Data Bus Width (Applies to the 160 Pin Version only)

The Utopia data bus width can be selected to be either 8 or 16 bits by writing the appropriate data to the Utopia Configuration Register, as shown below.

Utopia Configuration Register (Address = 7Ch)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused		Handshake Mode	M- PHY	CellOf52 Bytes	TxFIFODepth[1, 0]		UtWidth16
RO		R/ W	R/ W	R/ W	R/ W		R/ W

If the user chooses a Utopia Data Bus width of 8 bits, then only the Transmit Utopia Data input pins: TxData[7:0] will be active. (The input pins: TxData[15:8] will not be active). If the user chooses a Utopia Data bus width of 16 bits, then all of the Transmit Utopia Data input pins: TxData[15:0] will be active. The following table relates the value of

Bit 0 (UtWidth) within the Utopia Configuration Register, to the corresponding width of the Utopia Data bus.

Table 11. The Relationship between the contents of Bit Field 0 (UtWidth16) within the Utopia Configuration Register and the operating width of the Utopia Data bus

Value for UtWidth16	Width of Utopia Data Bus
0	8 bit wide Data Bus
1	16 bit wide Data Bus

Note:

1. The selection of this bit- field also effects the width of the Receive Utopia Data bus.
2. Upon power up or reset, the Utopia Data Bus width will be 8 bits. Therefore, the user must write a "1" to this bit in order to set the width of the Transmit Utopia (and the Receive Utopia) Data buses to 16 bits.
3. This option is only available for the 160 pin packaged version of the XR- T7234 device. The 100 pin device only contains Transmit Utopia Data Bus input pins; TxData[7:0]. TxData[15:8] are not available in the 100 pin version.

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

The UNI allows the user with to select the number of octets per cell that the Transmit Utopia Interface block will process, following each assertion of the TxSoC 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 to bit 3 (CellOf52 Bytes) within the Utopia Configuration Register, as depicted below.

Utopia Configuration Register (Address = 7Ch)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		Handshake Mode	M- PHY	CellOf52 Bytes	TxFIFODepth[1, 0]		UtWidth16
RO	R/ W	R/ W	R/ W	R/ W	R/ W		

The following table specifies the relationship between the value of this bit and the number of octets/ cell that the Transmit Utopia Interface block will process.

Table 12. The Relationship between the contents of Bit 3 (CellOf52Bytes) within the Utopia Configuration Register, and the number of octets per cell that will be processed by the Transmit and Receive Utopia Interface blocks

CellOf52 Bytes	Number of Bytes/ Cells
0	53 bytes when the Utopia Data Bus width is 8 bits. 54 bytes when the Utopia Data Bus width is 16 bits.
1	52 bytes, regardless of the configured width of the Utopia Data Bus

Note: This selection applies to both the Transmit Utopia and Receive Utopia interface blocks. Additionally, the shaded

selection reflects the default condition upon power up or reset.

6.1.2.1.4 Parity Checking and Handling of ATM Cell Data received from the ATM Layer Processor

The ATM Layer processor is expected to compute the odd parity bit for all bytes or words that it intends to write into the Transmit Utopia Interface block. The ATM Layer processor is then expected to apply the value of this parity bit to the TxPrty input pin of the UNI, while the corresponding byte (or word) is present on the Transmit Utopia data bus. The Transmit Utopia Interface block will independently compute the odd parity of the contents on the Transmit Utopia Data Bus. Afterwards, the Transmit Utopia Interface block will compare its calculated value for parity with that placed on the TxPrty input pin (by the ATM Layer processor). If these two values are equal, then the byte (or word) of data will be processed through the Transmit Utopia Interface block. However, if these two parity values are not equal, then the "Detection of Parity Error (Transmit Utopia Interface)" interrupt will occur, and the cell comprising this errored byte (or word) will be (optionally) discarded. The user can configure the Transmit Utopia Interface block to discard or retain this "errored" cell by writing the appropriate data to the Transmit Utopia Interrupt/ Status Register (Address = 80h) as depicted below.

Tx UTI Interrupt/ Status Register (Address = 80h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
TxFIFO Reset	Discard Upon PErr	TPerr IntEn	TxFIFO ErrIntEn	TCCCA IntEn	TPErr IntStat	TxFIFO OverInt Stat	TCCCA IntStat
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

If the user sets this bit- field to a "1", then the Transmit Utopia Input Interface block will discard the errored cell. If the user sets this bit- field to "0", then the Transmit Utopia Interface block will not discard the errored cell; and this cell will be written into the Tx FIFO.

6.1.2.2 Transmit Utopia FIFO Manager

The Tx FIFO Manager has the following responsibilities.

- Monitoring the fill level of the Tx FIFO, and providing the appropriate level of Flow Control of data between the Transmit Utopia Interface block and the ATM Layer processor.
- Detecting and discarding "Runt" cells and insuring that the Tx FIFO can resume normal operation following the removal of the runt cell.
- Insuring that the Tx FIFO can respond properly to an "Overrun" condition, by generating the "Tx FIFO Overrun Condition" interrupt, discarding the resulting "runt" or errored cell, and resuming proper operation afterwards.

Transmit Utopia FIFO Manager Features and Options

This section discusses the numerous features that are provided by the Transmit Utopia FIFO Manager. Additionally, this section discusses how the user can customize these features to suit his/ her application needs.

The Transmit Utopia FIFO Manager provides the user with the following options.

- Handshaking Mode (Octet Level vs Cell Level)
- User selected Operating Tx FIFO Depth
- Resetting the Tx FIFO
- Monitoring the Tx FIFO

6.1.2.2.1 Selecting the Handshaking Mode (Octet Level vs Cell Level)

The Transmit Utopia Interface block offers two different data flow control modes for data transmission between the ATM Layer processor and the UNI IC. These two modes are: "Octet- Level" Handshaking and "Cell- Level" Handshaking;

as specified by the Utopia Level 2, Version 8 Specifications, and are discussed below.

6.1.2.2.1.1 Octet- Level Handshaking

The UNI will be operating in the "Cell- Level" Handshaking Mode following power up or reset. Therefore, the user have to set bit 5 (Handshaking Mode) of the Utopia Configuration Register to "0" in order to configure the UNI into the "Octet- Level" Handshake mode. The main signal that is responsible for data flow control, between the ATM Layer processor and the Transmit Utopia Interface block is the TxClav output pin. The ATM Layer processor is expected to monitor the TxClav output pin in order to determine if it is OK to write data into the Tx FIFO. The TxClav output pins exhibits a role that is similar to CTS (Clear to Send) in RS- 232 based data transmission systems. As long as TxClav is at a logic "high", the ATM Layer processor is permitted to write more cell data bytes (or words) into the Transmit Utopia Interface block (and in turn, the TxFIFO). However, when the TxClav pin toggles "low", this indicates that the Tx FIFO can only accept 4 (or less) more write operations from the ATM Layer processor. Once the TxClav pin returns high, this indicates that the TxFIFO can accept more than 4 write operations from the ATM Layer processor, and that the ATM Layer processor can resume writing data to the Transmit Utopia Interface block.

In other words, if the Utopia Data bus is configured to be 8- bits wide, then the TxClav signal will toggle "low" when the TxFIFO can only accept 4 (or less) bytes of ATM cell data, from the ATM Layer processor. If the Utopia Data bus is configured to be 16- bits wide; then the TxClav signal will toggle "low" when the TxFIFO can only accept 8 (or less) bytes of ATM cell data from the ATM Layer processor.

Figure 18 presents a timing diagram illustrating the behavior of TxClav during writes to the Transmit Utopia Interface block, while operating in the Octet- Level Handshaking Mode.

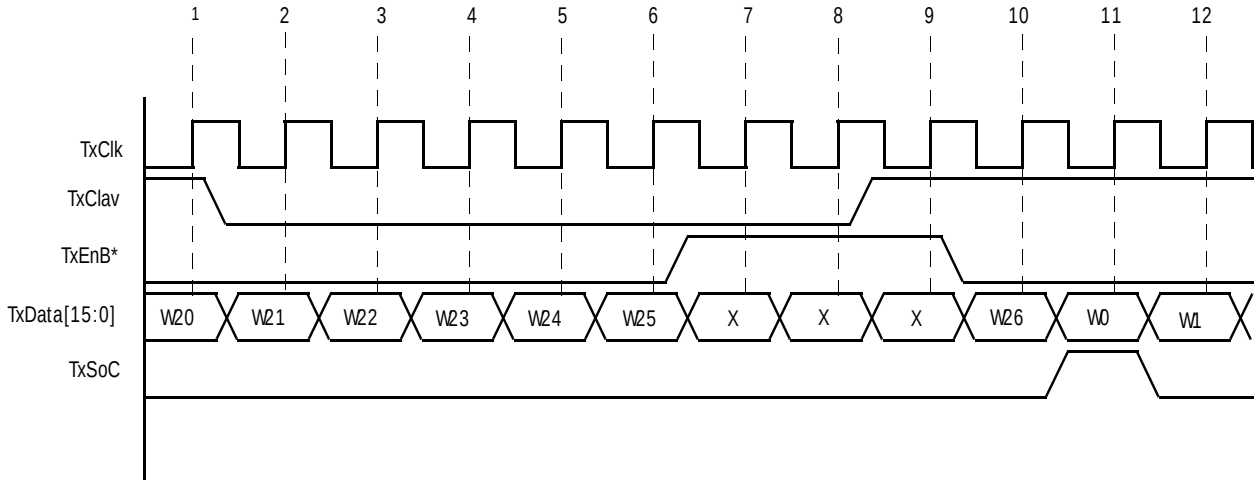

Figure 18. Timing Diagram of TxClav and various other signals during writes to the Transmit Utopia Interface block, while operating in the Octet- Level Handshaking Mode.

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 18 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

In Figure 18, TxClav is initially "high" during clock edge # 1. However, shortly after the ATM Layer processor writes in word W20, TxClav toggles "low", indicating that the TxFIFO is starting to fill up. The ATM Layer processor will detect this "negation of TxClav" during clock edge #2; while it is writing word W21 into the Transmit Utopia Interface block. At this point, the ATM Layer processor is only permitted to execute four more "write" operations with the Transmit Utopia Interface block. Therefore, the ATM Layer processor will proceed to write in words: W22, W23,

W24 and W25 before negating TxEnB*. The ATM Layer processor must keep TxEnB* negated until it detects that TxClav has once again returned "high". In Figure 20, TxClav is asserted after clock edge #8. The ATM Layer processor detects this transition in TxClav at clock edge #9; and subsequently, asserts TxEnB*. The ATM Layer resumes writing in more ATM cell data into the Transmit Utopia Interface block.

6.1.2.2.1.2 Cell- Level Handshaking

The UNI will be operating in the "Cell- Level" Handshaking mode following power up or reset. In the "Cell- Level" Handshaking mode, when the TxClav is at 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 TxClav toggles from "high" to "low", it indicates that the very next cell (following the one that is currently being written) cannot be accepted by the Tx FIFO. Conversely, once TxClav has returned to the logic "1" level, it indicates that at least one more full cell may be written into the Tx FIFO by the ATM Layer processor. As in the "Octet- Level" Handshake mode, the ATM Layer processor is expected to poll the TxClav output pin towards the end of transmission of the cell currently being written and to proceed with transmission of the next cell only if TxClav is at a logic "high".

The UNI can operate in either the "Octet- Level" or the "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 6.1.2.3.

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) within the Utopia Configuration Register, as depicted below.

Utopia Configuration Register (Address = 7Ch)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused		Handshake Mode	M- PHY	Cell Of 52 Bytes	TxFIFO Depth[1, 0]		UtWidth16
RO		R/W	R/W	R/W	R/W		R/W

The following table specifies the relationship between this bit and the corresponding Handshaking Mode.

Table 13. The Relationship between the contents in bit field 5 (Handshake Mode) within the Utopia Configuration Register and the Resulting Utopia Interface Handshake Mode.

Value	Utopia Interface Handshake Mode
0	The Utopia Interfaces operate in the "Octet Level" handshake mode.
1	The Utopia Interfaces operate in the "Cell Level" handshake mode.

Note:

1. The Handshaking Mode selection applies to both the Transmit Utopia Interface and Receive Utopia Interface blocks.
2. Since Multi- PHY mode operation requires the use of "Cell- Level" Handshaking; this bit- field is ignored if the UNI is operating in the Multi- PHY mode.
3. Finally, the UNI will be operating in the "Cell- Level" Handshaking Mode upon power up or reset. Therefore, the user must write a "0" to this bit- field in order to configure the UNI into the "Octet Level Handshaking" mode.

Figure 19 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.

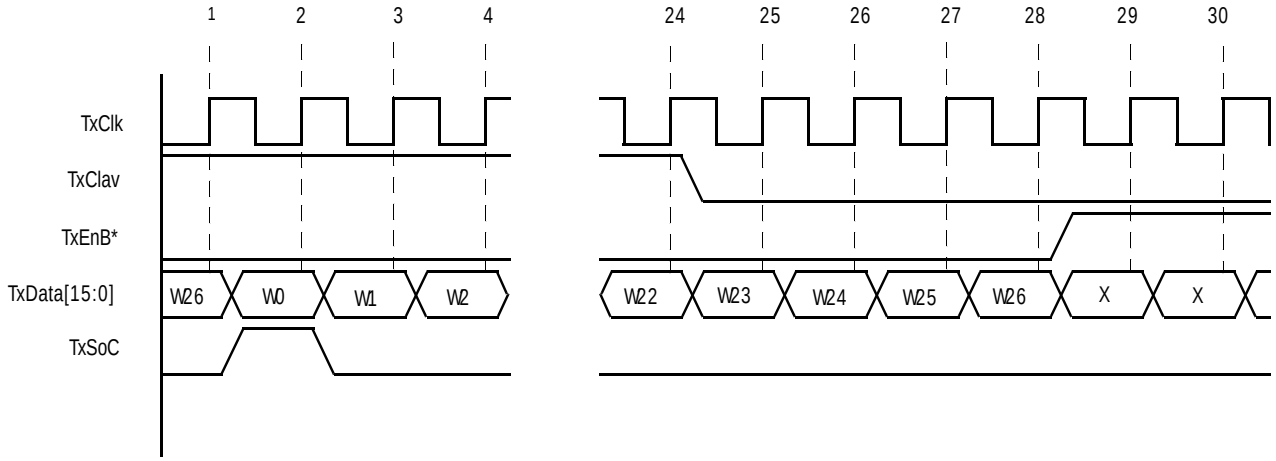

Figure 19. Timing Diagram of various Transmit Utopia Interface block signals, when the Transmit Utopia Interface block is operating in the "Cell Level Handshaking" mode.

Notes regarding Figure 19:

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 19 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

In Figure 19, 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, TxClav toggles "low". In the "Cell- Level" Handshaking mode, this means that ATM Layer processor is not permitted to write in the subsequent cell (e.g., the cell which is to follow the one that is currently being written 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 proceeds to write in Words W23 through W26 and then negates the TxEnB* signal after clock edge #28. At this point, the ATM Layer processor must wait until TxClav toggle "high" once again; before writing in the next ATM cell.

6.1.2.2.2 Selecting the Operating Depth of the Tx FIFO

The physical depth of the Tx FIFO is 16 cells. However, for various reasons the user may wish to operate with a smaller FIFO depth. Therefore, the UNI allows the user to select an operating depth of 4, 8, 12 or the full 16 cells. The user can make this selection by writing the appropriate data to Bits 1 and 2 (TxFIFODepth[1, 0]) within the Utopia Configuration Register, as depicted below.

Utopia Configuration Register: Address = 7Ch							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		Handshake Mode	M- PHY	CellOf52 Bytes	TxFIFODepth[1, 0]		UtWidth16
RO		R/ W	R/ W	R/ W	R/ W		R/ W

The following table presents the values for both Bits 1 and 2 (within the Utopia Configuration Register) and the

corresponding operating depth of the TxFIFO.

Table 14. The Relationship between TxFIFODepth[1:0] within the Utopia Configuration Register and the Operating Depth of the TxFIFO

Bit 2	Bit 1	Operating Depth of the Transmit FIFO
0	0	16 cells
0	1	12 cells
1	0	8 cells
1	1	4 cells

The operating depth of the Transmit FIFO will be 16 cells upon power up or reset. Therefore, the user must write the appropriate data to these two bit- fields in order to change this parameter.

6.1.2.2.3 Resetting the Tx FIFO via Software Command

The UNI allows the user to reset the Tx FIFO, via software command, without the need to implement a master reset of the entire UNI device. This can be accomplished by writing the appropriate data to bit 7 (TxFIFO Reset) within the Transmit Utopia Interrupt Enable/ Status Register as depicted below.

Tx UT Interrupt/ Status Register (Address = 80h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
TxFIFO Reset	Discard Upon PErr	TPerr IntEn	TxFIFO ErrIntEn	TCCCA IntEn	TPErr IntStat	TxFIFO OverInt Stat	TCCCA IntStat
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

6.1.2.2.5 Monitoring the Tx FIFO Status

The local µP has the ability to poll and monitor the status of the Tx FIFO via the Transmit Utopia FIFO Status Register (Address = 71h). The bit format of this register is presented below.

Tx UT FIFO Status Register (Address = 71h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused						TxFIFO Full	TxFIFO Empty
RO	RO	RO	RO	RO	RO	RO	RO

The following tables define the values for Bits 1 and 0 and their corresponding meaning.

TxFIFO Full	
TxFIFO Full (Bit 1)	***Meaning***
0	Tx FIFO is full, the ATM Layer processor risks causing an overrun if it writes to the TxFIFO now.
1	Tx FIFO is not full.

TxFIFO Empty	
TxFIFO Empty (Bit 0)	*Meaning*
0	Tx FIFO is not empty
1	Tx FIFO is empty. The Tx Cell Processor is currently generating IDLE cells

6.1.2.3 Utopia Modes of Operation (Single PHY and Multi- PHY operation)

The UNI chip can support both Single- PHY and Multi- PHY operation. Each of these operating modes are discussed below.

6.1.2.3.1 Single PHY Operation

The UNI chip will be operating in the Multi- PHY mode upon power up or reset. Therefore, the user must write a "1" to Bit 4 within the Utopia Configuration register (Address = 7Ch) in order to configure the UNI into the "Single- PHY" Mode.

Utopia Configuration Register (Address = 7Ch)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused		Handshake Mode	S- PHY/ M- PHY*	Cell Of 52 Bytes	TxFIFO Depth[1, 0]		UtWidth16
RO		R/ W	R/ W	R/ W	R/ W		R/ W

Writing a '1' to this bit- field configures the UNI to operate in the Single- PHY Mode. Writing a '0' configures the UNI to operate in the Multi- PHY Mode.

In Single- PHY operation, the ATM layer processor is pumping data into and receiving data from only one UNI

device, as depicted below in Figure 20.

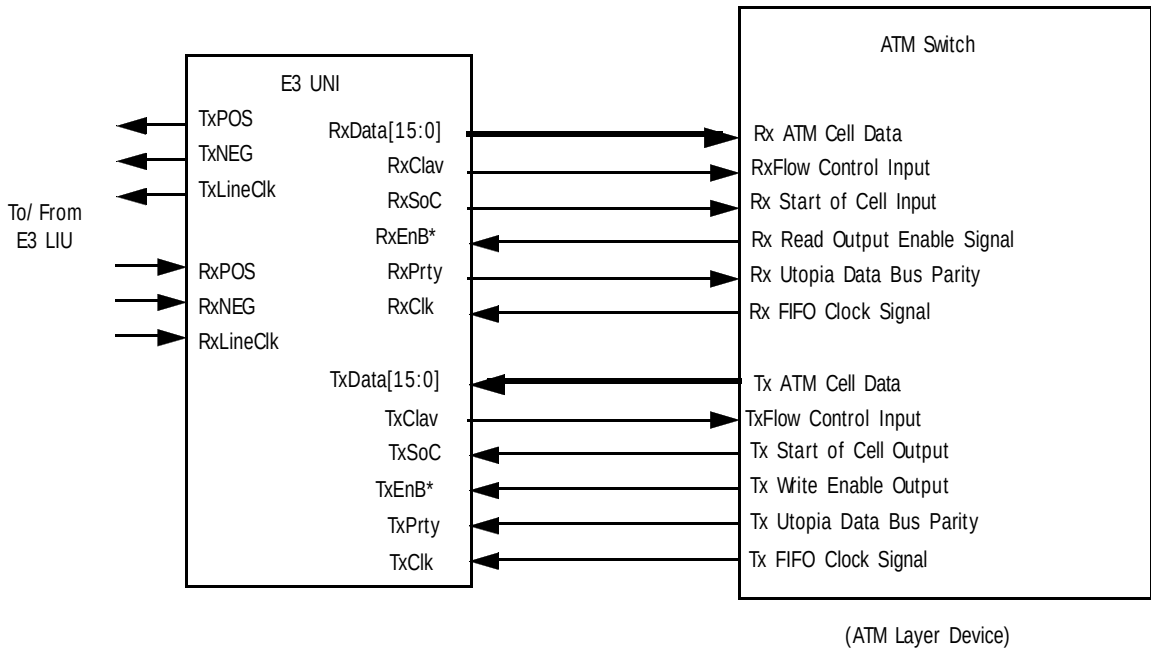

Figure 20. Simple Illustration of Single- PHY Operation

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 7.3.2.2.2.1. 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 TxClav output pin.

If the TxClav pin is "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 TxClav pin is "low", then the Tx FIFO is too full to accept anymore data and the ATM Layer Processor must wait until TxClav toggles "high" before writing any cell data to the Transmit Utopia Interface block.

Note: The actual meaning of TxClav toggling "low" depends upon whether the UNI is operating in the "Cell Level" or "Octet Level" handshake modes.

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 TxSoC pin "high" for one clock period of TxClk.

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

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

4. Assert the "Transmit Utopia Data Bus"-Write Enable Signal, TxEnb*.

This step should also be done concurrently with pulsing the TxSoC 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.

If the UNI is operating in the Octet- Level handshake mode, then the ATM Layer processor should check the level of

the TxClav signal, at least once for every four (4) writes of ATM cell data to the Transmit Utopia Interface block.

If the UNI is operating in the Cell- Level Handshake mode, then the ATM Layer Processor should check the level of the TxClav signal, as it nears completion of writing in a given cell.

The above- mentioned procedure is also depicted in Flow- Chart Form in Figure 21; and in Timing Diagram form in Figures 22 and 23.

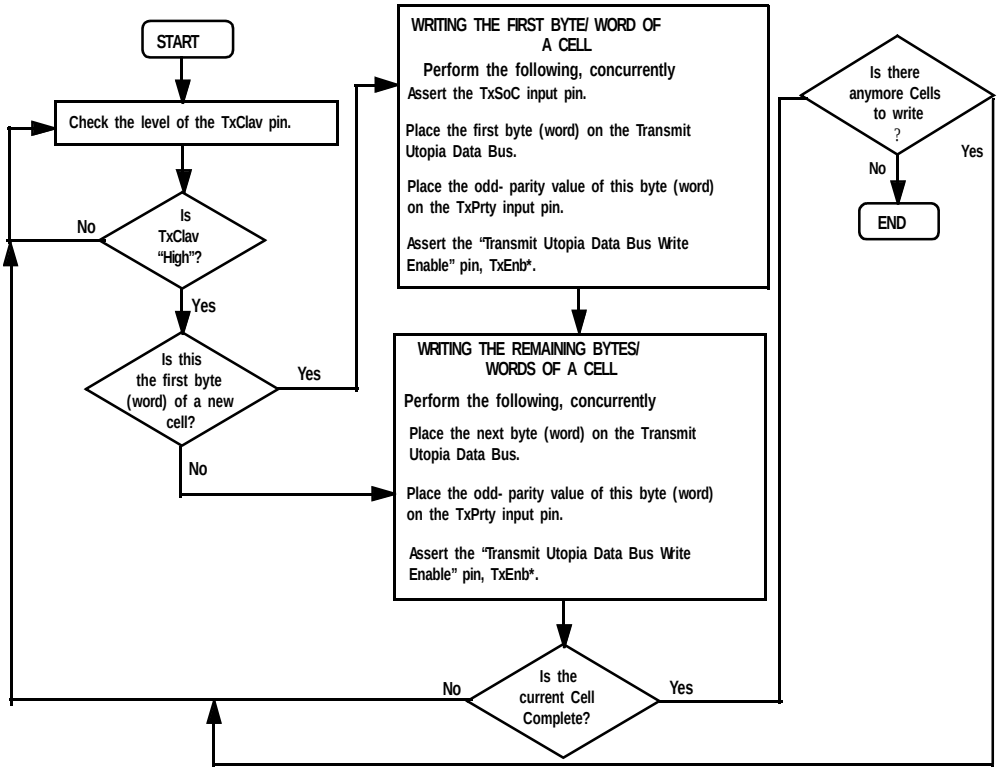

Figure 21. Flow Chart depicting the approach that the ATM Layer Processor should take when writing ATM Cell Data into the Transmit Utopia Interface block, when the UNI is operating in the Single PHY Mode.

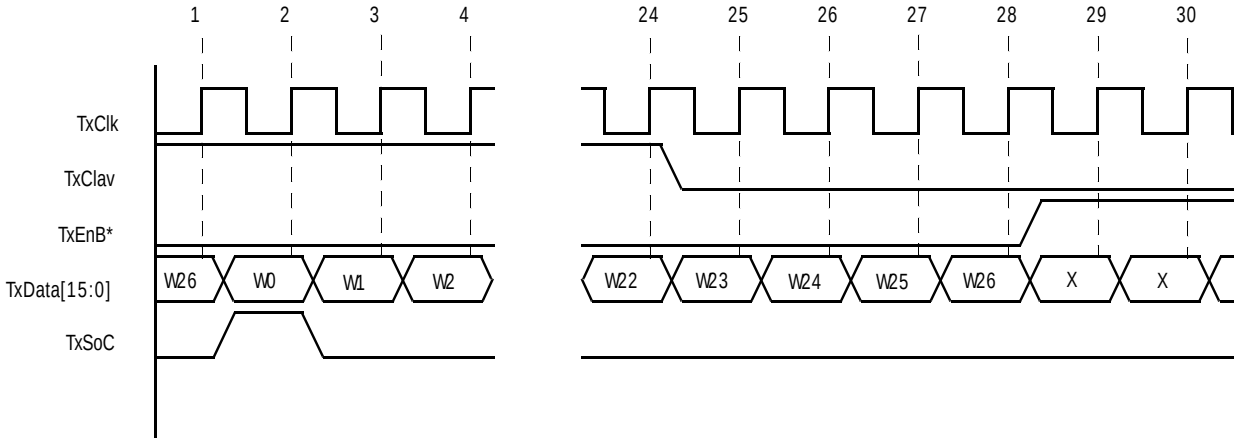

Figure 22. Timing Diagram of ATM Layer processor Transmitting Data to the UNI over the Utopia Data Bus, (Single- PHY Mode/ Cell- Level Handshaking).

Notes regarding Figure 22:

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 22 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.
3. The Transmit Utopia Interface Block is configured to operate in the Cell- Level Handshaking mode.

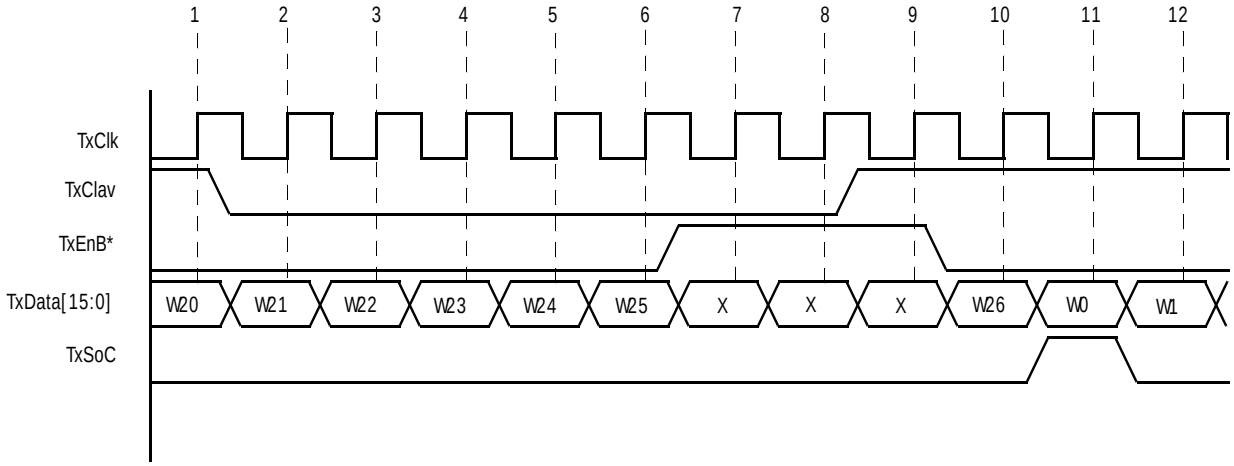

Figure 23. Timing Diagram of ATM Layer processor Transmitting Data to the UNI over the Utopia Data Bus (Single- PHY Mode/ Octet- Level Handshaking).

Notes regarding Figure 23:

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 23 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.
3. The Transmit Utopia Interface Block is configured to operate in the Octet- Level Handshaking Mode.

Final Comments on Single- PHY Operation

The important thing to note about the Single- PHY mode is that the TxClav 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 TxClav pin will have a slightly different role when the UNI is operating in the Multi- PHY mode.

The UNI, while operating in Single PHY mode, can be configured for either "Octet- Level" or "Cell Level" Handshaking. In either case, the ATM Layer processor is expected to poll the TxClav output pin before writing the next byte, word or cell to the TxFIFO.

6.1.2.3.2 Multi PHY operation

The UNI IC will be operating in the "Multi- PHY" mode upon power up or reset. In the "Multi- PHY" operating mode, the ATM Layer processor may be writing data into and reading data from several UNI devices in parallel. When the UNI 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 devices.
- Selecting which UNI (out of several possible UNI devices) to write ATM cell data to.

Each of these operations are 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 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 UNI device (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 "Utopia Interface Block is 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 "Tx Utopia Address Register" (Address = 82h) within its "host" UNI device. The bit- format of the "Tx Utopia Address Register" is presented below.

TX Utopia Address Register (Address = 82h)							
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	0	0	0	0	0

Likewise, the user assigns a "Utopia address" value to a particular "Receive Utopia Interface block", within one of the UNIs (in the "Multi- PHY" system) by writing this address value into the "Rx Utopia Address Register" (Address = 7Eh) within the "host" UNI device. The bit- format of the "Rx Utopia Address Register" is presented below.

Rx Utopia Address Register (Address = 7Eh)							
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	0	0	0	0	0

Note: The role of the Receive Utopia Interface block, in "Multi- PHY" operation is presented in Section 7.3.2.2.2.2.

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

When the UNI is operating in the "Multi- PHY" mode, 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, follows.

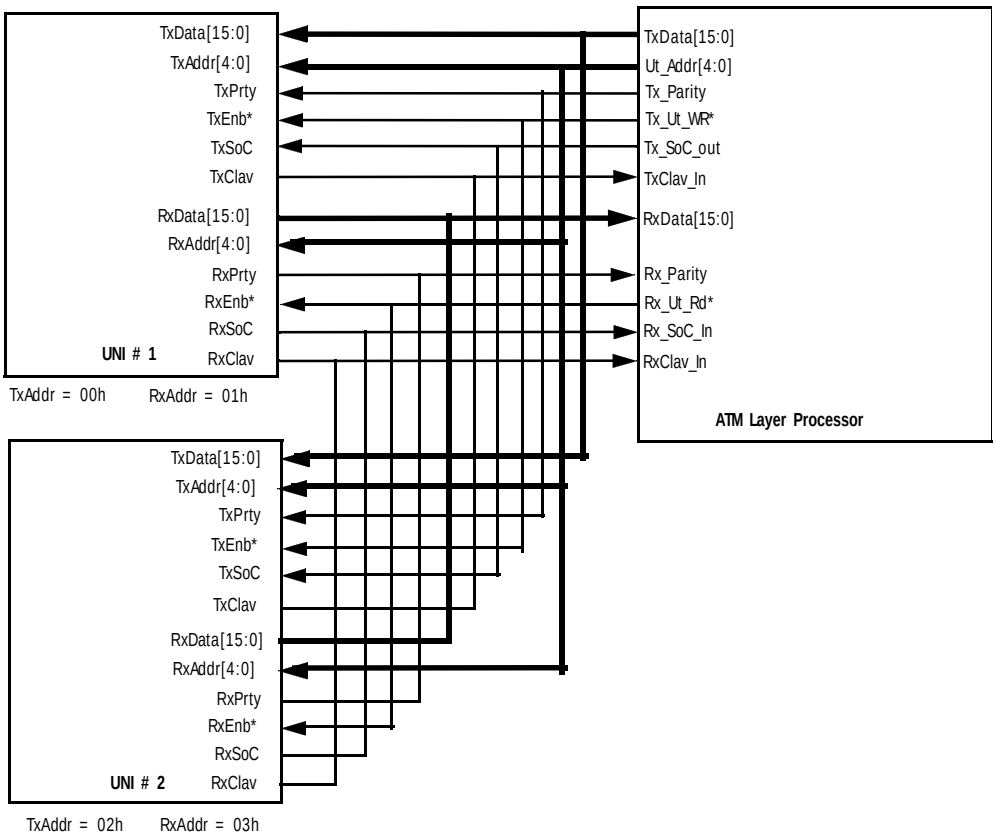

Figure 24. An Illustration of Multi- PHY Operation with UNI Devices #1 and #2

Figure 24 depicts a "Multi- PHY" system consisting of an 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 "TxClav" line, a common "RxClav" line, as well as common TxEnB*, RxEnB*, TxSoC and RxSoC 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 24.

The Utopia Address values, that have been assigned to each of the Transmit and Receive Utopia Interface blocks, within Figure 24, are listed below in Table 15.

Table 15. Utopia Address Values of the Utopia Interface blocks illustrated in Figure 24.

Block	Utopia Address Value
Transmit Utopia Interface block-UNI #1	00h
Receive Utopia Interface block-UNI #1	01h
Transmit Utopia Interface block-UNI #2	02h
Receive Utopia Interface block-UNI #2	03h

Recall, that the Transmit Utopia Interface blocks were assigned these addresses by writing these values into the "Tx Utopia Address Register" (Address = 82h) within their "host" UNI device. The discussion of the Receive Utopia

Interface blocks, within UNIs #1 and #2 is presented in Section 7.3.2.2.2.2.1.

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 TxEnB* input pin (if it is not asserted already).

The UNI device (being "polled") will know that this is only a "polling" operation, if the TxEnB* input pin is asserted, prior to detecting its Utopia Address on the "Utopia Address" bus.

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 "TxClav_in" input pin (see Figure 24).

The UNI devices role in the "polling" operation

UNI #2 will sample the signal levels placed on its Tx Utopia Address input pins (TxAddr[4:0]) on the rising edge of its "Transmit Utopia Interface block" clock input signal, TxClk. 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 = 82h).

If these values do not match, (e.g., TxAddr[4:0] ¼02h) then UNI #2 will keep its "TxClav" output signal "tri- stated"; and will continue to sample its "Transmit Utopia Address bus input" pins; with each rising edge of TxClk.

If these two values do match, (e.g., TxAddr[4:0] = 02h) then UNI #2 will drive its "TxClav" output pin to the appropriate level, reflecting its TxFIFO "fill- status". Since the UNI is automatically operating in the "Cell Level Handshaking" mode, while it is operating in the "Multi- PHY" mode; the UNI will drive the TxClav 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 "TxClav" 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 "TxClav" output signal "tri- stated". Therefore, when UNI #2 is driving its "TxClav" output pin to the appropriate level; it will be driving the entire "TxClav" line, within the "Multi- PHY" system. Consequently, UNI#2 will also be driving the "TxClav_in" input pin of the ATM Layer processor (see Figure 24).

If UNI #2 drives the "TxClav" 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 TxClav line "high" (during "polling"), then the ATM Layer processor will know that it can write cell data into the Transmit Utopia Interface block, of UNI # 2.

Figure 25 presents a timing diagram, that depicts the behavior of the ATM Layer processor's and the UNI's signals

during polling.

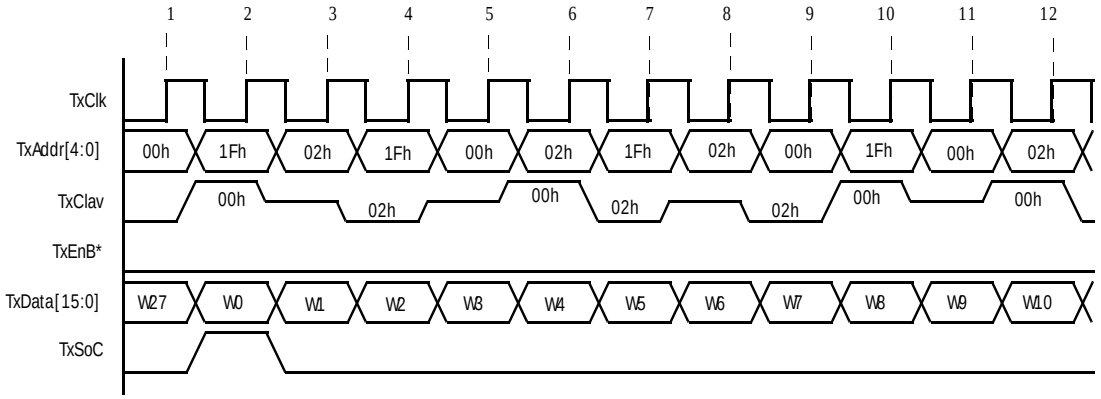

Figure 25. Timing Diagram illustrating the Behavior of various signals from the ATM Layer processor and the UNI, during Polling.

Notes regarding Figure 25:

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 25 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 (TxAddr[4:0] = 00h) during this "polling process".
4. The TxFIFO, within UNI#2's Transmit Utopia Interface block (TxAddr[4:0] = 02h) is incapable of receiving any additional ATM cell data from the ATM Layer processor. Hence, the TxClav line will be driven "low" whenever this particular Transmit Utopia Interface block is "polled".
5. The Transmit Utopia Address of 1Fh, is not associated with any UNI device, within this "Multi- PHY" system. Hence, the TxClav line is tri- stated whenever this address is "polled".

Note: Although Figure 24 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 7.2.2.2.2.2 for a discussion on the Receive Utopia Interface block during Multi- PHY operation.

6.1.2.3.2.2 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 "Utopia Address Bus".
2. Negate the TxEnB* 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.
3. Assert the TxEnB* signal.
4. Assert the TxSoC input pin.
5. Begin applying the ATM Cell data in a byte- wide (or word- wide) manner to the Transmit Utopia Data Bus.

Figure 26 presents a flow- chart that depicts the "UNI Device Selection and Write" process in Multi- PHY operation.

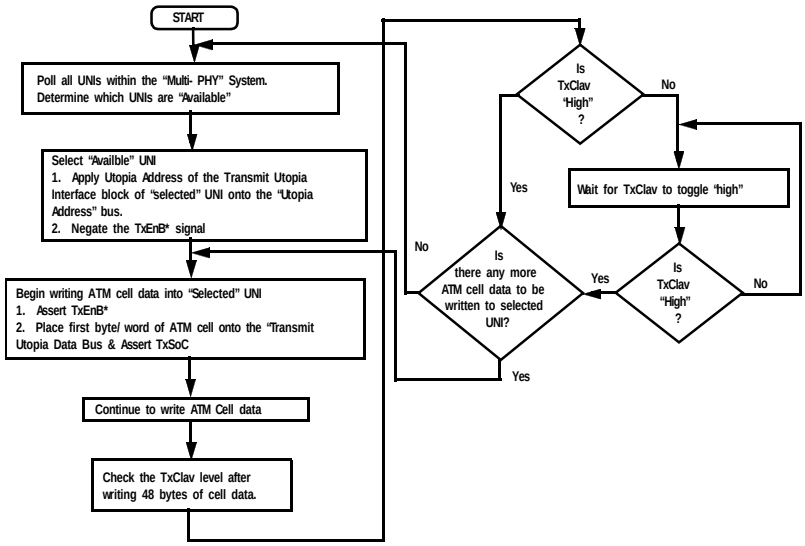

Figure 26. Flow- Chart of the "UNI Device Selection and Write Procedure" for the Multi- PHY Operation.

Figure 27 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.

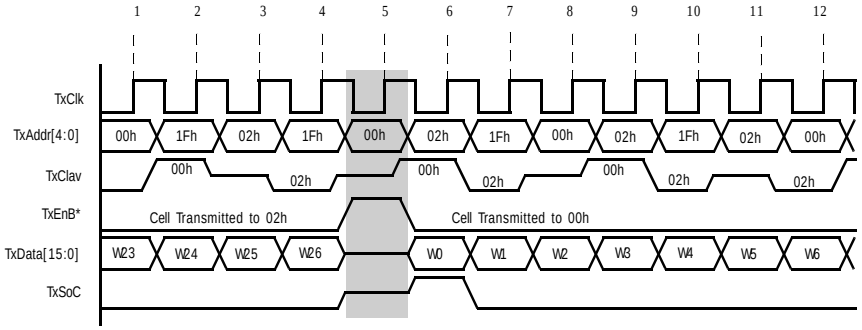

Figure 27. Timing Diagram of the Transmit Utopia Data and Address Bus signals, during the 'Multi- PHY" UNI Device Selection and Write Operations.

Notes regarding Figure 27:

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 (e.g., W0 through W26) for each ATM cell.

In Figure 27, the ATM Layer processor is initially writing ATM cell data to the Transmit Utopia Interface block within UNI #2 (TxAddr[4:0] = 02h). However, the ATM Layer processor is also polling the Transmit Utopia Interface block within UNI #1 (TxAddr[4:0] = 00h) and some "non- existent" device at TxAddr[4:0] = 1Fh. 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 (TxAddr[4:0] = 02h). The ATM Layer processor will indicate its intentions to select a new UNI device for writing by negating the TxEnB* signal, at clock edge #5 (see the shaded portion of Figure 27). 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 (TxAddr[4:0] = 00h).
2. The TxEnB* 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.

6.1.2.4 Transmit Utopia Interrupt Servicing

The Transmit Utopia Interface block will generate interrupts upon the following conditions:

- Detection of parity errors
- Change of cell alignment (e.g., the detection of "runt" cells)
- TxFIFO Overrun

If one of these conditions occur and if that particular condition is enabled for interrupt generation, then when the local µP/µC reads the UNI Interrupt status register, as shown below, it should read "0xxxxx1xb" (where the b suffix denotes a binary expression, and the "x" denotes a "don't care" value).

UNI Interrupt Status Register (Address = 05h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec Interrupt Status	Tx E3 Interrupt Status	Rx E3 Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO
0	x	x	x	x	x	1	x

At this point, the local µC/µP has determined that the Transmit Utopia Interface block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly.

The next step in the interrupt service routine should be to determine which of the three Transmit Utopia Interface Block interrupt conditions has occurred and is causing the interrupt request. In order to accomplish this, the local µP/µC should now read the Tx UT Interrupt Enable/Status Register, which is located at address 80h within the UNI device. The bit format of this register is presented below.

Tx UT Interrupt Enable / Status Register (Address-80h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx FIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCCCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCCCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

The "Tx UT Interrupt Enable/Status" Register has eight bit- fields. However, only six of these bit fields are relevant to interrupt processing. Bits 0-2 are the interrupt status bits and bits 3-5 are the interrupt enable bits; for the Transmit Utopia Interface block. Each of these "interrupt processing relevant" bit fields are defined below.

Bit 0-TCCCA Interrupt Status-Transmit Utopia Change of Cell Alignment Condition

If the ATM Layer Processor asserts the TxSoC input pin prior to writing the contents of a complete cell (as config-

ured via the CellCf52Bytes option) on the Transmit Utopia Data Bus, then the Transmit Utopia Interface block will interpret this newly received cell data as a "runt" cell. When the Transmit Utopia Interface block detects a "runt" cell, it will generate the "Transmit Utopia Change of Cell Alignment Condition" interrupt, and the "runt" cell will be discarded. The Transmit Utopia Interface Block will indicate that it is generating this kind of interrupt by asserting Bit 0 (TCOCA Interrupt Status) within the Transmit Utopia Interrupt Enable/ Status Register, as depicted below:

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
TxFIFO Reset	Discard Upon Par t	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR
x	x	x	x	1	x	x	1

Bit 1–Tx FIFO Overrun Interrupt Status

If the Tx FIFO is filled to capacity, and if the ATM Layer processor attempts to write any additional data to the Tx FIFO, some of the data within the Tx FIFO will be overwritten, and in turn lost. If the Transmit Utopia Interface block detects this condition, and if this interrupt condition has been enabled, then the UNI will assert the INT* pin to the local µP/µC. Additionally, the UNI will set bit- field 1, (TxFIFO Overrun Interrupt Status) within the Tx Utopia Interrupt Enable/ Status Register to "1", as depicted below:

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
TxFIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR
x	x	x	1	x	x	1	x

Bit 1 of the Tx UT Interrupt Enable/ Status register will be reset or cleared upon the local µP/µC reading this register. This action will also negate bit 3 within the UNI Interrupt Status Register and the INTB* output pin, unless other outstanding interrupt conditions are awaiting service.

Bit 2–TPErr Interrupt Status-Detection of Parity Error via the Transmit Utopia Interface block

The ATM Layer processor is expected to compute and present the odd- parity value of each byte or word of ATM Cell data that it intends place on the Transmit Utopia Data bus. As the ATM Layer processor is writing ATM cell data into the Transmit Utopia Interface block, it will place the value of this parity bit at the TxPrty input pin of the UNI device while the corresponding byte (or word) is present on the Transmit Utopia data bus. The Transmit Utopia Interface block will read the contents of the Transmit Utopia Data Bus, and will independently compute the odd- parity value of that byte or word. Afterwards, the Transmit Utopia Interface block will then compare its computed parity value with that presented at the TxPrty input (by the ATM Layer processor). If these two parity values are different then a "Transmit Utopia Parity error" has been detected. If this interrupt condition has been enabled, then the UNI will generate the "Detection of Parity Error" interrupt. Additionally, the UNI will set bit- field 2 (Tx UT Parity Error

Interrupt Status), within the Transmit Utopia Interrupt Enable/ Status Register to "1", as depicted below.

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Tx FIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCCCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCCCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR
x	x	1	x	x	1	x	x

Once the local µP/µC has read the contents of the Tx UT Interrupt Enable/ Status register, then bit 3 of the UNI Interrupt Status Register, bit 2 of the Tx UT Interrupt Enable/ Status register, and the INTB* output pin will all be negated, unless outstanding interrupts conditions are awaiting servicing.

Bit 3-TCCCA Interrupt Enable-Transmit Utopia Change of Cell Alignment Interrupt Enable

This "read/ write" bit- field allows the user to enable or disable the "Change of Cell Alignment" interrupt. The local microprocessor can enable this interrupt by writing a "1" to this bit- field. Upon power up or reset conditions, this bit-field will contain a "0". Therefore the default condition is for this interrupt to be disabled.

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Tx FIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCCCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCCCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

Bit 4-Tx FIFO ErrInt Enable-Tx FIFO Overrun Condition Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Tx FIFO Overrun" interrupt. The local microprocessor can enable this interrupt by writing a "1" to this bit. Upon power up or reset conditions, this bit will contain a "0". Therefore the default condition is for this interrupt to be disabled. The local microprocessor must write a "1" to this bit in order to enable this interrupt.

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Tx FIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCCCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCCCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

Bit 5-TPerr Interrupt Enable-Detection of Parity Error in Transmit Utopia block Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Detected Parity error" interrupt. The user can enable this interrupt by writing a "1" to this bit. Upon power up or reset conditions, this bit will contain a "0". Therefore the

default condition is for this interrupt to be disabled. The user must write a "1" to this bit in order to enable this interrupt.

Tx UT Interrupt Enable / Status Register (Address = 80h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx FIFO Reset	Discard Upon Parity Error	Tx UT Parity Error Interrupt Enable	Tx FIFO Overrun Interrupt Enable	TCCCA Interrupt Enable	Tx UT Parity Error Interrupt Status	Tx FIFO Overrun Interrupt Status	TCCCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

6.2 TRANSMIT CELL PROCESSOR

6.2.1 Brief Description of the Transmit Cell Processor

The Transmit Cell Processor reads in cells from the Transmit Utopia FIFO (Tx FIFO) within the Transmit Utopia Interface block. Immediately after reading in the cell from the TxFIFO, the Transmit Cell Processor will verify the "Data Path Integrity Check" pattern (located in octet # 5, within this cell). Afterwards, the Transmit Cell Processor optionally computes and inserts the HEC byte into each cell, and optionally scrambles the cell payload bytes. When the TxFIFO does not contain a full cell, the Transmit Cell Processor generates a programmable idle (or unassigned) cell and inserts it in the transmit stream. The Transmit Cell Processor provides the user with the ability to write an "outbound" OAM cell into the "Transmit OAM Cell" buffer, and to transmit this OAM cell, upon demand. Additionally, the Transmit Cell Processor is also equipped with a serial input port which allows the user to externally insert the value of the GFC (Generic Flow Control) field for each outbound valid cell. Figure 28 presents a simple illustration

of the Transmit Cell Processor block and the associated external pins.

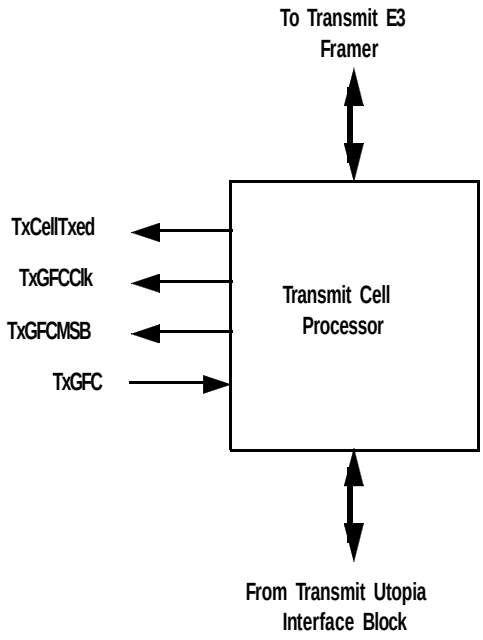

Figure 28. Simple Illustration of the Transmit Cell Processor Block and the Associated External Pins

Figure 29 presents a functional block diagram of the Transmit Cell Processor.

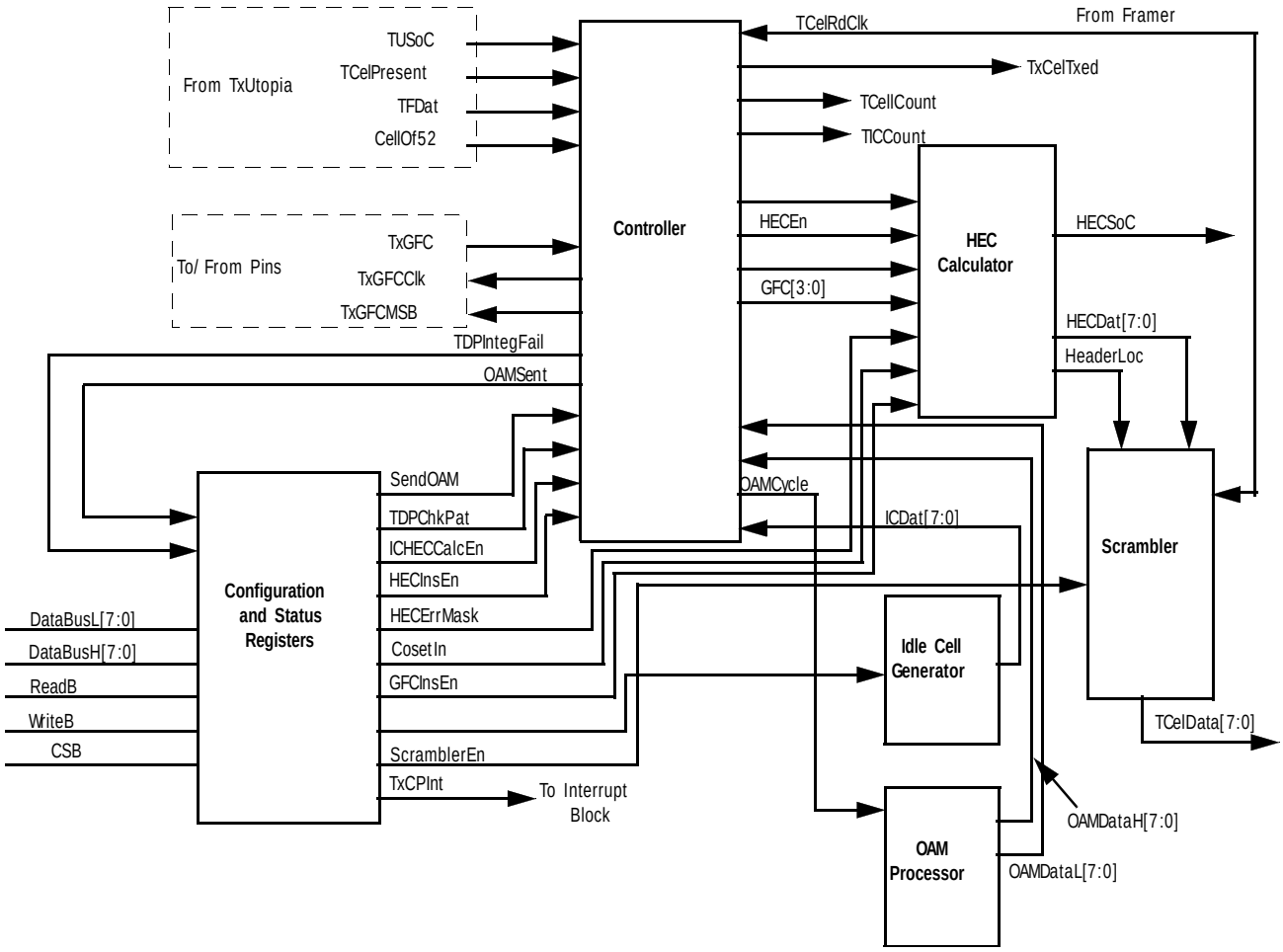

Figure 29. Functional Block Diagram of the Transmit Cell Processor

6.2.2 Functional Description of Transmit Cell Processor

The Transmit Cell Processor consists of the following functional blocks.

- Configuration and Status Register
- Controller
- HEC Calculator
- OAM Processor
- Cell Scrambler
- IDLE Cell Generator
- "Transmit GFC Nibble- field" serial input port

Most of these functional blocks will be discussed in some detail below. The Transmit Cell Processor will read in ATM Cell Data from the Tx FIFO. The first four bytes of each cell is loaded into the "HEC Byte calculator". The fifth byte of each user cell will be read- in and compared against a pre- defined "Data Path Integrity Check" pattern. While this "check" is being performed; the "HEC Byte Calculator" will take these first four bytes of the cell, and compute a HEC byte value. This HEC byte value will be written (or inserted) into the 5th octet position of the cell. Conse-

quently, the "Data Path Integrity Check" pattern will now be overwritten. Bytes 6 through 53 (the cell payload) of each cell, are sent onto the "Cell Scrambler" and are summarily "scrambled". Afterwards, the cell is reassembled (with the first four header bytes, the newly computed HEC byte and the scrambled payload bytes), and is routed to the Transmit E3 Framer.

When a complete cell is not available in the TxFIFO, a cell is created by "Idle Cell Generator". The user has the option of specifying the contents of the header and payload of these Idle Cells via the µP-accessible registers. The payload of the Idle Cell will be programmed with a repeating pattern of a byte contained within an on-chip register. From this point on, the Idle Cell is processed in the same manner as is an assigned (e.g., user or OAM) cell. A valid HEC byte is computed over the four bytes of the programmed idle cell header and is inserted into the fifth octet position. The user has the option to disable the HEC Byte Calculation and Insertion features for Idle cells, and the contents of the fifth- header byte programmed register may be transmitted directly.

The Transmit Cell Processor allows the user to transmit pre- programmed OAM cells upon demand. The content of this OAM cell is stored in an on- chip RAM location, which will be referred to as the "Transmit OAM Cell Buffer". When the local µP decides to transmit the OAM cell to the "Far End" Terminal, it writes a "1" to a certain register bit. The Transmit Cell Processor will then proceed to read in the contents of the "Transmit OAM Cell" buffer, and form a cell from this data. This OAM cell will be subsequently processed like any user or Idle cell (e.g., processed through the HEC Byte Calculator and Cell Scrambler) and then routed to the Transmit E3 Framer for transmission onto the E3 line.

As mentioned earlier, the Transmit Cell Processor will perform a "Data Path Integrity Check" on all user cells that it reads from the TxFIFO. More specifically, the Transmit Cell Processor will look for a specific data pattern that should be residing within octet #5 of these cells. The purpose of this test is to verify the integrity of the communication link throughout the "ATM Layer processor" system. This "Data Path Integrity Pattern" was written into the cell by the Receive Cell Processor of another UNI, prior to its entry into the "ATM Layer processor" system. If the Transmit Cell Processor detects a discrepancy between the contents of octet #5 and the expected pattern, then the Transmit Cell Processor will generate a "Data Path Integrity Check" error interrupt. After the Transmit Cell Processor has completed its checking of the "Data Path Integrity Check" pattern; within a given cell, it will (typically) overwrite this pattern with the HEC byte.

The Transmit Cell Processor will inform external circuitry when a cell has been transmitted from the Transmit Cell Processor to the Transmit E3 Framer, by pulsing the "TxCellTxed" output pin. Therefore, the "TxCellTxed" signal will be pulsing at a nominal rate of 80 kHz.

6.2.2.1 HEC Byte Calculation and Insertion

The "HEC Byte Calculator" takes the first four bytes of each cell and computes a CRC- 8 value via the generating polynomial $x^8 + x^2 + x + 1$. 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, instead insert this newly computed value into byte 5 of the cell before transmission. The user also has the following additional options regarding the "HEC Byte Calculator".

- HEC Byte Calculation and Insertion Enable/ Disable for user and OAM cells.
- HEC Byte Calculation and Insertion Enable/ Disable for Idle Cells.
- Inserting errors into the HEC byte, for chip/ equipment testing purposes.

The implementation and result of selecting each of these options are presented below.

6.2.2.1.1 Configuring the HEC Byte Calculator for User and OAM Cells

The user can enable or disable the "HEC Byte Calculation and Insertion" feature for user and OAM cells by writing the appropriate value to Bit 5 (HEC Insert Enable) within the "Tx CP Control/ Interrupt" Register, as depicted below.

Tx CP Control/ Interrupt Register (Address = 72h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*

Tx CP Control/ Interrupt Register (Address = 72h)							
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
1	1	x	1	0	0	1	0

If the user opts to disable this feature, then the HEC byte will not be computed and the contents within the fifth octet position of each cell (e.g., typically the "Data Path Integrity Check" pattern) will be transmitted to the Transmit E3 Framer block as is. The following table relates the content of this bit- field to the "HEC Byte Calculator's" handling of valid (e.g., user or OAM) cells.

Table 16. The Relationship between the contents of Bit- field 5 (HEC Ins Enable) within the Tx CP Control/ Interrupt Register, and the HEC Byte Calculator's handling of valid cells

HEC Insert Enable	Result
0	HEC Byte Calculation is disabled and the 5th byte is transmitted to the Transmit E3 Framer as is.
1	The HEC Byte is calculated and is inserted into the 5th octet position of each valid cell.

Upon power up or reset, the "HEC Byte Calculator and Insertion" feature is enabled. The user must write a "0" to this bit in order to disable this operation.

6.2.2.1.2 Configuring the "HEC Byte Calculator and Insertion" Feature for Idle Cells

The user can separately enable or disable the "HEC Byte Calculation and Insertion" feature for the outbound Idle Cells. The user can exercise this option by writing the appropriate value to bit 1 (Idle Cell HEC CalEn) within the "Tx CP Control/ Interrupt" Register, as depicted below.

Tx CP Control/ Interrupt Register (Address = 72h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
1	1	1	1	0	0	x	0

This "Read/ Write" bit- field allows the user to enable or disable the "Calculation and Insertion" of the HEC byte into the Idle Cell as illustrated below. If the user chooses to disable this feature, then the 5th octet of the Idle Cells will be transmitted to the Transmit E3 Framer block as programmed in the "Tx CP Idle Cell Pattern Header-Byte 5" register (Address = 7Ah). Table 17 relates the content of this bit- field to the "HEC Byte Calculator's" handling of Idle Cells.

Table 17. The Relationship between the content within Bit 1 (IC HEC Calc En) within the "Tx CP Control/ Interrupt" Register and the resulting handling of Idle Cells, by the "HEC Byte Calculator"

Idle Cell HEC CalEn	Result
0	The entire programmed Idle Cell header is transmitted without Modification

Table 17. The Relationship between the content within Bit 1 (IC HEC Calc En) within the "Tx CP Control/ Interrupt" Register and the resulting handling of Idle Cells, by the "HEC Byte Calculator"

Idle Cell HEC CalEn	Result
1	The HEC byte is calculated, via the first four bytes of the header, and is inserted into the fifth octet position within each Idle Cell.

Upon power up or reset, the Transmit Cell Processor will be configured such that the HEC bytes will be calculated and inserted each Idle Cell. The user must write a "0" to this bit- field in order to disable this feature.

6.2.2.1.3 Modulo-2- Addition of Coset Polynomial to the HEC Byte Value

When enabled, the HEC Byte Calculator takes the first four bytes of each cell and computes a CRC- 8 value via the generating polynomial x8 + x2 + x + 1. The BISDN Physical Layer specifications (ITU Recommendations I.432) specifies that this CRC- 8 (or HEC) value can optionally be modulo- 2 added to the polynomial x6 + x4 + x2 + 1; and inserting the result of this calculation into the fifth byte of each cell. The purpose of this option is to provide protection against bit slips. This protection is not required in transmission systems that ensure adequate one's density. However, this operation does provide protection against all zeros cells that could be passed to the ATM Layer during a loss of signal condition on the transmission medium. The ATM Forum UNI specifications also requires this operation.

The user can enable or disable this modulo- 2 addition, by writing the appropriate value to bit 6 (Coset Enable) within the "Tx CP Control/ Interrupt" Register, as depicted below.

Tx CP Control/ Interrupt Register (Address = 72h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
1	x	1	1	0	0	1	0

A "1" in this bit- field will enable this modulo addition. Conversely, a "0" in this bit- field will disable this operation.

Upon power up or reset, the Transmit Cell Processor will be configured to modulo- 2 add the Coset polynomial to the HEC byte prior to insertion into the cell. The user must write a "0" into this bit- field in order to disable this operation.

6.2.2.1.4 Inserting Errors into the HEC Byte via Software Control

The XR- T7234 E3 UNI allows the user to insert errors into the HEC bytes of "outbound" cells in order to support equipment testing. One such test that the user may wish to run is to verify is that the HEC byte verification (e.g., error detection and/ or correction) features of some "Far- End" terminal equipment is functioning properly. The user would conduct this test by transmitting cells with erroneous HEC byte values to the "unit under test" (UUT). The user can exercise this option by writing the appropriate data into the Tx CP Error Mask register, which is located at address 62h within the UNI.

Tx CP HEC Byte Error Mask Register; (Address = 74h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
HEC Error Mask Byte							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Transmit Cell Processor automatically XORs the HEC Byte (or each "outbound" cell) with the contents of this register. The results of this operation is written back into the fifth octet position of each of these cells. Therefore, if the user does not wish to inject errors into the HEC byte, he/she should insure that the contents of this register is 00h, the default value.

6.2.2.2 The Cell Scrambler

The Cell Scrambler takes bytes 6 through 53 of each cell (the payload) and scrambles the contents of these bytes. The purpose of scrambling the cell payload bytes is to reduce the possibility of the contents of the cell payload mimicking patterns that are used for framing and cell delineation purposes. The scrambler generating polynomial is $x^{43} + 1$. The user can enable or disable the Cell Scrambler by setting or clearing bit 7 (Scrambler Enable) within the "Tx CP Control/Interrupt" Register, as depicted below:

Tx CP Control/Interrupt Register (Address = 72h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
x	1	1	1	0	0	1	0

A "1" in this bit-field enables the Cell Scrambler. Conversely, a "0" in this bit-field disables the Cell-Scrambler.

Upon power up or reset, the Cell Scrambler function will be enabled. Therefore, the user must write a "0" to this bit in order to disable cell scrambling.

6.2.2.3 GFC Nibble-Field Serial Input Port (Only Available in the 160 pin package version)

The first four bits in the first header byte of each cell are allocated for carrying "Generic Flow Control" (GFC) information. The user can externally insert his/her own values for the GFC nibble-field into each outbound valid (user or OAM) cell, via a serial input port. The user will activate this serial input port (the "Transmit GFC-Nibble-field" serial input port) by writing a "1" to bit 3 (GFC Insert Enable) within the "Tx CP Control/Interrupt" Register, as depicted below:

Tx CP Control/Interrupt Register (Address = 72h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR
1	1	1	1	x	0	1	0

Once the user has activated the "Transmit GFC Nibble-field" serial input port, it will accept the 4 bit GFC value via the TxGFC input pin during each valid cell processing period. The TxGFC serial input port will be expecting the bits of the GFC nibble-field in descending order (MSB first). The GFC bits are clocked into the serial input port via the rising edge of the clock output signal, TxGFCClk. Since these four bits must be provided for each cell; TxGFCClk will provide four clock edges during each cell processing period. The "Transmit GFC Nibble-field" Serial input port will also provide a "framing pulse" in the form of the TxGFCMSB output pin pulsing "high". This output pin will pulse "high" when the Transmit Cell Processor is ready to receive the MSB (most significant bit) of the GFC nibble. Figure 30

presents a timing diagram illustrating the role of each of these signals during GFC nibble insertion.

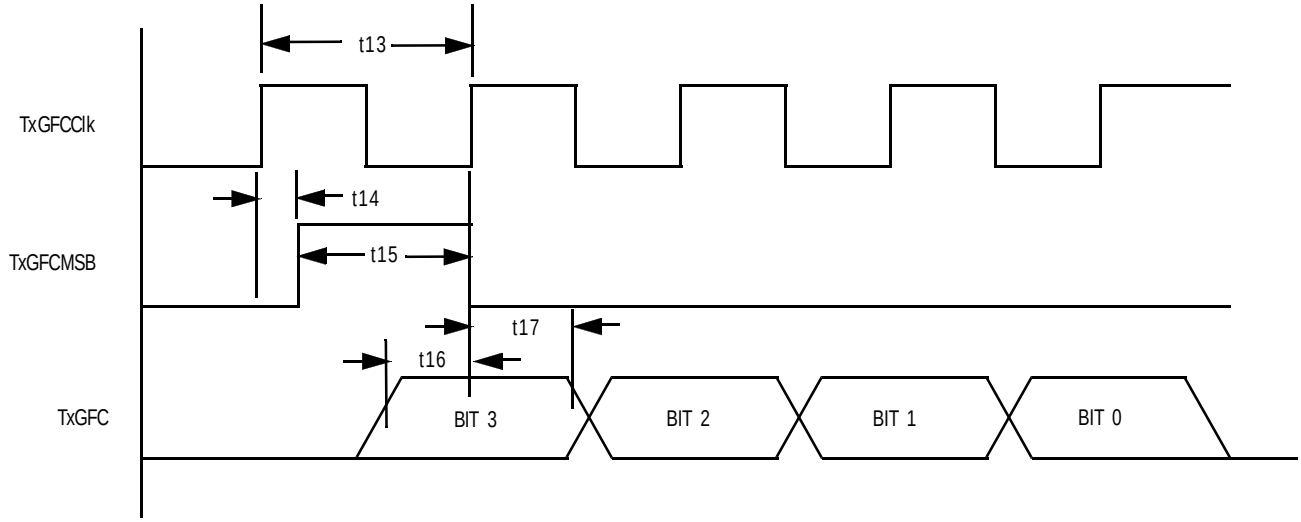

Figure 30. Behavior of TxGFC, TxGFCClk, and TxGFCMSB during GFC Nibble insertion into the "Outbound" Valid Cell

Note:

1. The "Transmit GFC Nibble- field" serial input port will only insert the GFC value into valid (e.g., user or OAM) cells. It will not insert the GFC value into Idle Cells.
2. The "Transmit GFC Nibble- field" serial input port is only available in the 160 pin package version of the XR- T7234 E3 UNI.

6.2.2.4 OAM Cell Processing

The UNI chip provides on- chip RAM space for the storage of the complete contents (header and payload) of an OAM cell. This RAM space is known as the "Transmit OAM Cell" buffer (consisting of 54 bytes) and is located at 136h through 16Bh within the UNI address space. Therefore, in order to "load" the OAM cell into the "Transmit OAM Cell" buffer. The local µP must write this data into this address location within the UNI IC, via the Microprocessor Interface. Afterwards, whenever the user wishes to transmit the OAM cell, the local µP must to write a "1" to bit 7 (SendOAM) within the Tx CP OAM Register as depicted below.

Tx CP OAM Register (Address = 73h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
SendOAM	Unused						
Semaphore	RO	RO	RO	RO	RO	RO	RO

If the local µP writes a "1" bit 7 (or "1xxxxxxxb") to the Tx CP OAM Register; then the Transmit Cell Processor will read- in the contents of the "Transmit OAM Cell" buffer, and form it into a cell. This OAM cell will then be routed to the HEC Byte Calculator and Cell Scrambler within the Transmit Cell Processor block, prior to transmittal to the Transmit E3 Framer. Bit 7 of the Tx CP OAM Register will be reset (to "0") upon completion of the transmission of the OAM cell. The user may also poll this bit in order to determine whether or not the OAM cell has been sent.

The user can monitor the number of valid cells (e.g., user and OAM cells) that have been generated and transmitted to the Transmit E3 Framer. The Transmit Cell Processor increments the contents of the "PMON Transmitted Valid Cell Count (MSB and LSB)" Registers (Address = 54h, and 55h) for each valid cell that it generates. These two registers are "Reset- upon- Read" registers that when concatenated present a 16- bit representation of the total number of

"valid cells" generated and transmitted by the Transmit Cell Processor, since the last read of these registers. The bit- format of these two registers follows.

PMON Transmitted Valid Cell Count - MSB (Address = 54h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Valid Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Transmitted Valid Cell Count -LSB (Address = 55h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Tx Valid Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

6.2.2.5 Idle Cell Processing

Whenever the TxFIFO (within the Transmit Utopia Interface block) does not contain a complete cell, the Transmit Cell Processor will automatically generate and process Idle Cells. The user can customize the contents of these Idle Cells or he/ she can use the default values, that are provided by the UNI chip. The user can customize the contents of these Idle Cells by programming six different registers:

- Tx CP Idle Cell Pattern-Header Byte 1
- Tx CP Idle Cell Pattern-Header Byte 2
- Tx CP Idle Cell Pattern-Header Byte 3
- Tx CP Idle Cell Pattern-Header Byte 4
- Tx CP Idle Cell Pattern-Header Byte 5
- Tx CP Transmit Cell Payload

Table 18 presents the Bit Format of each of these Registers and Table 19 presents the Address and Default values of these cells.

Table 18. Bit Format of the Tx CP Idle Cell Pattern - Header Bytes and Tx CP Cell Payload" Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Tx CP Idle Cell Pattern-Header Byte 1	Transmit Idle Cell Pattern-Header Byte 1							
Tx CP Idle Cell Pattern-Header Byte 2	Transmit Idle Cell Pattern-Header Byte 2							
Tx CP Idle Cell Pattern-Header Byte 3	Transmit Idle Cell Pattern-Header Byte 3							
Tx CP Idle Cell Pattern-Header Byte 4	Transmit Idle Cell Pattern-Header Byte 4							
Tx CP Idle Cell Pattern-Header Byte 5	Transmit Idle Cell Pattern-Header Byte 5							
Tx CP Idle Cell Payload	Transmit Idle Cell Payload							

Table 19. Address and Default Values of the Tx CP Idle Cell Pattern Registers

Address	Register	Default Value
76h	Tx CP Idle Cell Pattern-Header Byte 1	00h
77h	Tx CP Idle Cell Pattern-Header Byte 2	00h
78h	Tx CP Idle Cell Pattern-Header Byte 3	00h
79h	Tx CP Idle Cell Pattern-Header Byte 4	01h
7Ah	Tx CP Idle Cell Pattern-Header Byte 5	00h
7Bh	Tx CP Idle Cell Payload	00h

The role of the registers for Idle Cell Pattern-Bytes 1 through 4 is quite straightforward. When the Transmit Cell Processor opts to generate an Idle cell, it will read in the content of these registers, and send these values onto the HEC Byte Calculator. Consequently, the contents of the "Transmit Idle Cell Pattern-Header Byte 5" will likely be overwritten by the HEC Byte Calculator in the Idle Cell, unless the HEC Byte Calculator has been disabled (See Section 6.2.2.1.2). The payload portion of these Idle Cells is defined by the contents of the Transmit Idle Cell Payload Register (Address = 7Bh), repeated 48 times. When the Transmit Cell Processor reads in this register to form the cell payload, the resulting payload will be sent on to the Cell Scrambler and is (optionally) scrambled just like any assigned cell.

The UNI will keep track of the number of Idle cells that have been generated and transmitted to the Transmit E3 Framer. The Transmit Cell Processor increments the contents of the "PMON Transmitted Idle Cell Count (MSB and LSB)" Registers (Address = 52h and 53h) for each Idle Cell that is generated and transmitted. These two registers are 'Reset- upon- Read' registers that, when concatenated, presents a 16- bit representation of the total number of idle cells generated and transmitted since the last time these registers were read. The bit format of these two registers follow.

PMON Transmitted Idle Cell Count - MSB (Address = 52h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Idle Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Transmitted Idle Cell Count-LSB (Address = 53h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Tx Idle Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

6.2.2.6 Data Path Integrity Check

The Transmit Cell Processor provides for some performance monitoring of the communication link between the various UNIs, over the "ATM Switching System". This performance monitoring feature is referred to as the "Data

Path Integrity Check".

The Receive Cell Processor, or some equivalent entity, within a UNI device, will (after performing HEC byte verification) write a "Data Path Integrity Check" pattern into each cell prior to its being read and processed by the ATM Layer processor. This cell (with the "Data Path Integrity Check" pattern) will be routed through the ATM switch, and possibly throughout the Wide Area Network (WAN); before arriving to the Transmit Utopia Interface block of a given XR-T7234 E3 UNI. The Transmit Cell Processor will read in this cell from the Tx FIFO, and will, prior to inserting a new HEC byte into the cell, read in the fifth octet, from the Tx FIFO and check it for a specific pattern or value. The user can configure the Transmit Cell Processor to check for either a constant "55h" pattern or an alternating pattern of "55h" and "AAh" for each cell. The user can also configure the Transmit Cell Processor to generate an interrupt if a Data Path Integrity Test fails. The user can accomplish all of this by writing the appropriate data to the "Tx CP Control/ Interrupt" Register (Address = 72h). The bit format (with the relevant bit fields shaded) of this register is shown below.

Note:

1. The "Data Path Integrity Check" feature is disabled if the Transmit (and Receive) Utopia Interface blocks have been configured handle 52 byte cells.
2. This "Data Path Integrity Test" is only performed on user cells. The Transmit Cell Processor does not perform this test on OAM or Idle Cells.

Tx CP Control/ Interrupt Register (Address = 72h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR

The role that each of these "shaded" bit field play is presented below.

Bit 4-TDPChk Pat-Test Data Path Integrity Check Pattern

The Transmit Cell Processor is always checking for a specific pattern in the fifth octet of a user cell retrieved from the Tx FIFO. This "Read/ Write" bit allows the user to specify the octet pattern that the Transmit Cell Processor should be checking for. The following table relates the contents of this bit field to the octet pattern expected by the Transmit Cell Processor.

Table 20. The Relationship between the contents of Bit 4 (TDPChk Pat) within the "Tx CP Control/ Interrupt" Register, and the "Data Path Integrity Check" Pattern that the Transmit Cell Processor will look for in the 5th octet of each incoming user cell

TDPChk Pattern	"Data Path Integrity Pattern" Expected by the Transmit Cell Processor
0	Transmit Cell Processor expects an alternating "55h"/ "AAh" pattern for the value of the fifth octet of the cells received from the Tx FIFO.
1	Transmit Cell Processor expects a constant "55h" pattern for the value of the fifth octet of the cells received from the Tx FIFO.

The remaining shaded bits are "Interrupt service" related and will be discussed in the following section.

6.2.2.7 Transmit Cell Processor Interrupt Servicing

The Transmit Cell Processor generates interrupts upon the detection of an error in the "Data Path Integrity Check" pattern.

If this condition occurs, and if that particular is enabled for interrupt generation, then the UNI will generate the "Data Path Integrity Check Pattern Error" interrupt. Afterwards, when the local µP/ µC reads the UNI Interrupt Status Register, as shown below, it should read "0xxx1xxxb" (where the b suffix denotes a binary expression, and the "x" denotes a

"don't care" value).

UNI Interrupt Status Register (Address = 05h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	One Sec. Interrupt Status	Tx E3 Interrupt Status	Rx E3 Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO
0	x	x	x	1	x	x	x

At this point, the local µC/µP has determined that the Transmit Cell Processor block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly.

Since the Transmit Cell Processor contains only one interrupt source, the Interrupt Service Routine, in this case should perform a read of the "Tx CP Control/ Interrupt" Register (Address = 72h), in order to verify and service this condition. The bit format of this register is presented below.

Tx CP Control/ Interrupt Register (Address = 72h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Scrambler Enable	Coset Enable	HEC Insert Enable	TDPChk Pattern	GFC Insert Enable	TDPErr Interrupt Enable	Idle Cell HEC CalEn	TDPErr Interrupt' Status
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RUR

This register contain 8 active bit- fields. However, only two of these bit- fields are relevant to Interrupt Processing. Bit 0 is an Interrupt Status bit, and Bit 2 is an Interrupt Enable bit.

Bit 2–TDPErrIntEn–"Test Data Path Integrity Check" Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the "Data Path Integrity Check Pattern Error" interrupt. Writing a "0" to this bit- field disables this interrupt. Likewise, writing a "1" to this bit- field enables this interrupt.

Bit 0–TDPErrIntStat–"Test Data Path Integrity Check" Interrupt Status

This "Reset- upon- Read" bit- field indicates whether or not the "Data Path Integrity Check Pattern Error" interrupt has occurred since the last reading of the "Tx CP Control/ Interrupt" Register. This interrupt will occur if the Transmit Cell Processor detects a byte- pattern, in the fifth octet position of each cell read from the TxFIFO, that differs from the expected "Data Path Integrity Check" pattern.

A "1" in this bit- field indicates that this interrupt has occurred since the last reading of the Tx CP Control/ Interrupt Register. A "0" in this bit- field indicates that this interrupt has not occurred.

Note: Once the local µP has read this register, Bit 0 (TDPerr Interrupt Status) will be reset to "0". Additionally, Bit 3 (Tx CP Interrupt Status) within the "UNI Interrupt Status" register will also be reset to "0".

6.3 TRANSMIT E3 FRAMER

6.3.1 Brief Description of the Transmit E3 Framer

The Transmit E3 Framer takes the incoming ATM Cell data from the Transmit Cell Processor and maps it into the payload portion of the outbound E3 frame. The Transmit E3 Framer supports the ITU- T G.832 compliant frame format. The Transmit E3 Framer operates at 34.368 MHz and framing is derived from one of three (3) possible input

clock signals. The framing overhead bytes are generated and inserted with the E3 payload bits in order to make up the complete E3 frame. The E3 frame is then encoded into either the Unipolar, AMI or HDB3 line codes as it is transmitted to the "Far- End" Terminal. The Transmit E3 Framer provides an interface that supports the transmission of path maintenance data link messages on the outbound E3 frames via the on- chip LAPD Transmitter. The Transmit E3 Framer also transmits the contents of 16 on- chip Trail Trace Buffer registers out to the "Far- End" Terminal. Different transmission conditions like AIS (Alarm Indication Signal), and the Yellow Alarm can be generated upon software command. Further, the LOS (Loss of Signal) condition can be simulated upon software command.

6.3.2 Definition of the ITU- T G.832 Compliant E3 Frame

The role of the various OH (overhead) bytes, within the ITU- T G.832 compliant E3 Frame, are best described by discussing the E3 Frame Format as a whole. The E3 Frame contains 537 bytes, of which 7 bytes are overhead and the remaining 530 bytes are "payload" bytes. The ATM cells are inserted into this 530 byte payload portion, for each frame.

These 537 octets are arranged in 9 rows of 60 columns each, except for the last three rows which contain only 59 columns. The frame repetition rate for this type of E3 frame, is 8000 frames per second, thereby resulting in the standard E3 bit- rate of 34.386 Mbps. Figure 31 presents an illustration of the ITU- T G.832 compliant E3 Frame Format.

<table>
<tr><th>FA1</th><th>FA2</th></tr>
<tr><td>EM</td><td rowspan="5">530 Octet Payload</td></tr>
<tr><td>TR</td></tr>
<tr><td>MA</td></tr>
<tr><td>NR</td></tr>
<tr><td>GC</td></tr>
<tr><td>1 byte</td><td>59 bytes</td></tr>
</table>

Figure 31. Illustration of the ITU-T G.832 Compliant E3 Frame Format

6.3.2.1 Definition of the Overhead Bytes

The seven (7) overhead bytes are shown in Figure 31, as FA1, FA2, EM, TR, MA, NR, and GC. Each of these Overhead Bytes are further defined below.

6.3.2.1.1 Frame Alignment (FA1 and FA2) Bytes

FA1 and FA2 are known as the frame alignment bytes. The Receiver E3 Framer, while trying to acquire or maintain framing with its incoming E3 frames, will attempt to locate these two bytes. FA1 is assigned the value of F6h, and FA2 is assigned 28h.

6.3.2.1.2 Error Monitor (EM) Byte

The EM byte contains the results of BIP-8 (Bit Interleaved Parity) calculations over an entire E3 frame. The Bit Interleaved Parity (BIP-8) byte field supports error detection, during the transmission of E3 frames, between the "Near-End" Transmit E3 Framer and the "Far End" Receive E3 Framer.

The Transmit E3 Framer will compute the BIP-8 value over the 537 octet structure, within each E3 frame. The resulting BIP-8 value is then inserted into the EM byte-field within the very next E3 frame. BIP-8 is an eight bit code in which the nth bit of the BIP-8 code reflects the even parity bit calculated with the nth bit of each of the 537 octets within the E3 frame. Thus, the BIP-8 value presents the results for 8 separate even-bit parity calculations.

The Receive E3 Framer will compute its own version of the EM byte for each E3 frame that it receives. Afterwards, it will compare the value of its "locally computed" EM byte with the EM byte that it receives in the very next E3 frame. If the two EM byte values are equal, then the Receive E3 Framer will conclude that this E3 frame was received in an "error-free" fashion. Further, the Receive E3 Framer will inform the "Far-End" Terminal of this fact by having the "Near-End" Transmit E3 Framer set the "FEBE" (Far-End-Block Error) bit, within the MA Byte of an outbound E3 frame, (to the "Far-End" Terminal) to "0". Please see Section 6.3.2.1.4 for a discussion of the MA Byte.

However, if the Receive E3 Framer detects an error in the incoming EM byte, then it will conclude that the corresponding E3 frame is errored. Further, the Receive E3 Framer will inform the "Far- End" Terminal (e.g., the source of this errored E3 frame) of this fact by having the "Near- End" Transmit E3 Framer set the "FEBE" bit, within an "outbound" E3 frame (destined for the "Far- End" Terminal) to "1".

A detailed discussion on the practical use of the EM byte is presented in Section 6.3.3.2.

6.3.2.1.3 The Trail Trace (TR) Byte

This byte- field is used to repetitively transmit a trail access point identifier so that a trail receiving terminal can verify its continued connection to the intended transmitter. The trail access point identifier uses the 16- byte numbering format as tabulated below.

Trail Trace Bits								
Byte Number	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
1 (Frame Start Marker)	1	C6	C5	C4	C3	C2	C1	C0
2	0	X	X	X	X	X	X	X
•	0	X	X	X	X	X	X	X
•	0	X	X	X	X	X	X	X
16	0	X	X	X	X	X	X	X

The first byte of this 16 byte string is a "frame start marker" and is typically of the form [1, C6, C5, C4, C3, C2, C1, C0]. The "1" in the MSB (most significant bit) of this first byte is used to identify this byte as the "frame start marker" (e.g., first byte of the 16- byte Trail Trace Buffer Sequence). The bits: C6 through C0 are the results of a CRC- 7 calculation over the previous 16- byte frame. The subsequent 15 bytes are used for the transport of 15 ASCII characters required for the E.164 numbering format.

6.3.2.1.4 Maintenance and Adaptation (MA) Byte

The MA byte is responsible for carrying the FERF (Far- End Receive Failure) and the FEBE (Far End Block Error status) indicators from one terminal to another. The MA byte- field also carries the "Payload Type", the "Payload Dependent" and the "Timing Marker" indicators. The byte format for the MA byte is presented below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker

Bit 7-FERF (Far- End Receive Failure)

If the Receive E3 Framer (at a "Near- End" Terminal) is experiencing problems receiving E3 frame data from a "Far- End" Terminal (e.g., an LOS, OOF or AIS condition), then it inform the "Far- End" Terminal of this fact by commanding the "Near- End" Transmit E3 Framer to set the "FERF" bit- field (within the MA byte- field) of an "outbound" E3 frame (which is destined for the "Far- End" Terminal) to "1". The "Near- End" Transmit E3 Framer will continue to set the FERF bit- fields (within subsequent "outbound" E3 frames) to "1" until the Receive E3 Framer no longer experiences problems in receiving the E3 frame data. If the "Far- End" Terminal receives a certain number of consecutive E3 Frames, with the FERF bit- field set to "1", then the "Far- End" Terminal will interpret this signaling as an indication of a "Far- End Receive Failure" (e.g., a problem with the "Near- End" Terminal).

Conversely, if the Receive E3 Framer (at a "Near- End" Terminal) is not experiencing any problems receiving E3 frame data from a "Far- End" Terminal; then it will also inform the "Far- End" Terminal of this fact by commanding the "Near- End" Transmit E3 Framer to set the "FERF" bit- field (within the MA byte- field) of an "outbound" E3 frame (which is destined for the "Far- End" Terminal) to "0". The "Far- End" Terminal will interpret this form of signaling as

an indication of a normal operation.

A detailed discussion into the practical use of the FERF bit- field is presented in Section 6.3.3.3.1.

Bit 6-FEBE (Far- End Block Error)

If a "Near- End" Receive E3 framer detects an error in the EM byte, within an incoming E3 frame that it has received from the "Far- End" Terminal, then it inform the "Far- End" Terminal of this error by commanding the "Near- End" Transmit E3 Framer to set the "FEBE" bit- field (within the MA byte- field) of an "outbound" E3 frame (which is destined for the "Far- End" Terminal) to "1". The "Far- End" Terminal will interpret this signaling as an indication that the E3 frames that it is transmitting back out to the "Near- End" Receive E3 Framer, are errored.

Conversely, if the "Near- End" Receive E3 Framer does not detect any errors in the EM byte, within the incoming E3 frame, then it will also inform the "Far- End" Terminal of this fact; by commanding the "Near- End" Transmit E3 Framer to set the "FEBE" bit- field of an outbound E3 frame; (which is destined for the "Far- End" Terminal) to "0".

A detailed discussion into the practical use of the FEBE bit- field is presented in Section 6.3.3.2.

Bits 5-3: Payload Type

These bit- fields indicates to the "Far- End" Receiver, what kind of data is being transported in the 530 bytes of E3 Frame payload data. Some of the defined payload type values are tabulated below.

Table 21. A listing of the various Payload Type Values and their corresponding meanings

Payload Type Value	Meaning
000	Unequipped
001	Equipped
010	ATM
011	SDH TU- 12s

Bits 2-1: Payload Dependent

Bit 0-Timing Marker

This bit- field is set to "0" to indicate that the timing source is traceable to a Primary Reference Clock. Otherwise, this bit- field is set to "1".

6.3.2.1.5 The Network Operator (NR) Byte

The NR byte or the GC byte can be configured to transport LAPD Message frame octets from the LAPD Transmitter to the LAPD Receiver (of another UNI chip) at a data rate of 64 kbps (1 byte per E3 frame).

If the user opts not to use the NR byte to "transport" these LAPD Message frames, then the Transmit E3 Framer will read in the contents of the "Tx NR Byte" Register (Address = 2Bh), and insert this value into the NR byte- field of each "outbound" E3 frame.

The Receive E3 Framer will read in the contents of the NR byte- field within each incoming E3 frame; and will write it into the "Rx NR Byte" Register. Consequently, the user can determine the value of the NR byte, within the most recently received E3 frame, by reading the "Rx NR Byte" Register (Address = 14h).

6.3.2.1.6 The General Purpose Communications Channel (GC) Byte

The NR byte or the GC byte can be configured to transport LAPD Message frames from the LAPD Transmitter to the LAPD Receiver (of another UNI chip) at a rate of 64 kbps (1 byte per E3 frame).

If the user opts not to use the GC byte to "transport" these LAPD Message frames, then the Transmit E3 Framer will

read in the contents of the "Tx GC Byte" Register (Address = 29h), and insert this value into the GC byte- field of each "outbound" E3 frame.

The Receive E3 Framer will read in the contents of the GC byte- field, within each incoming E3 frame, and will write it into the "Rx GC Byte" register. Consequently, the user can determine the value of the GC byte, within the most recently received E3 frame, by reading the "Rx GC Byte Register" (Address = 15h).

6.3.3 Functional Description of the Transmit E3 Framer Block

The Transmit E3 Framer receives ATM cells from the Transmit Cell Processor, and inserts this data into the payload portion of each "outbound" E3 frame. The Transmit E3 Framer proceeds to generate the OH bytes, and include these bytes with the E3 payload bytes in order to form the complete E3 frame. The Transmit E3 Framer will then encode this E3 Frame Data into a Unipolar Format (for transmission over optical fiber) or in a Bipolar Format (AMI or HDB3 line code) for transmission over a transformer- coupled copper medium, to a far away E3 Receiver Terminal via an LIU IC.

The Transmit E3 Framer also provides a serial input port to allow the user to insert his/ her own overhead (OH) bytes into the outbound E3 Frames. Finally, the Transmit E3 Framer allows the user to insert errors into some of the OH bytes and to transmit various alarm conditions upon software control in order to support equipment testing as well as transmitting the appropriate alarm signals as conditions warrant.

Figure 32 presents a simple illustration of the Transmit E3 Framer block, along with the associated external pins.

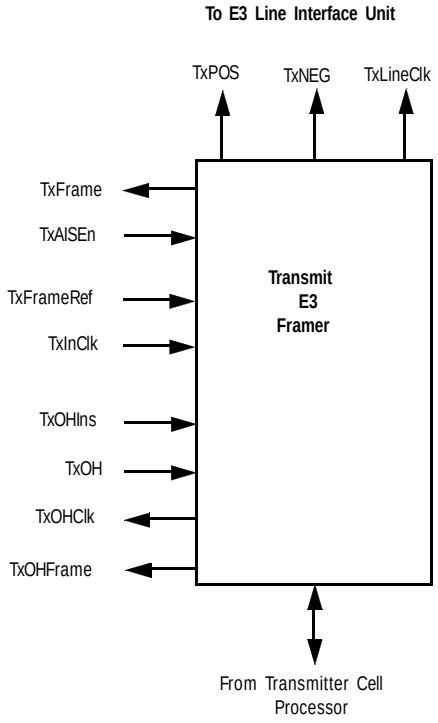

Figure 32. A Simple Illustration of the Transmit E3 Framer Block and the associated External Pins

Figure 33 presents a functional block diagram of the Transmit E3 Framer block.

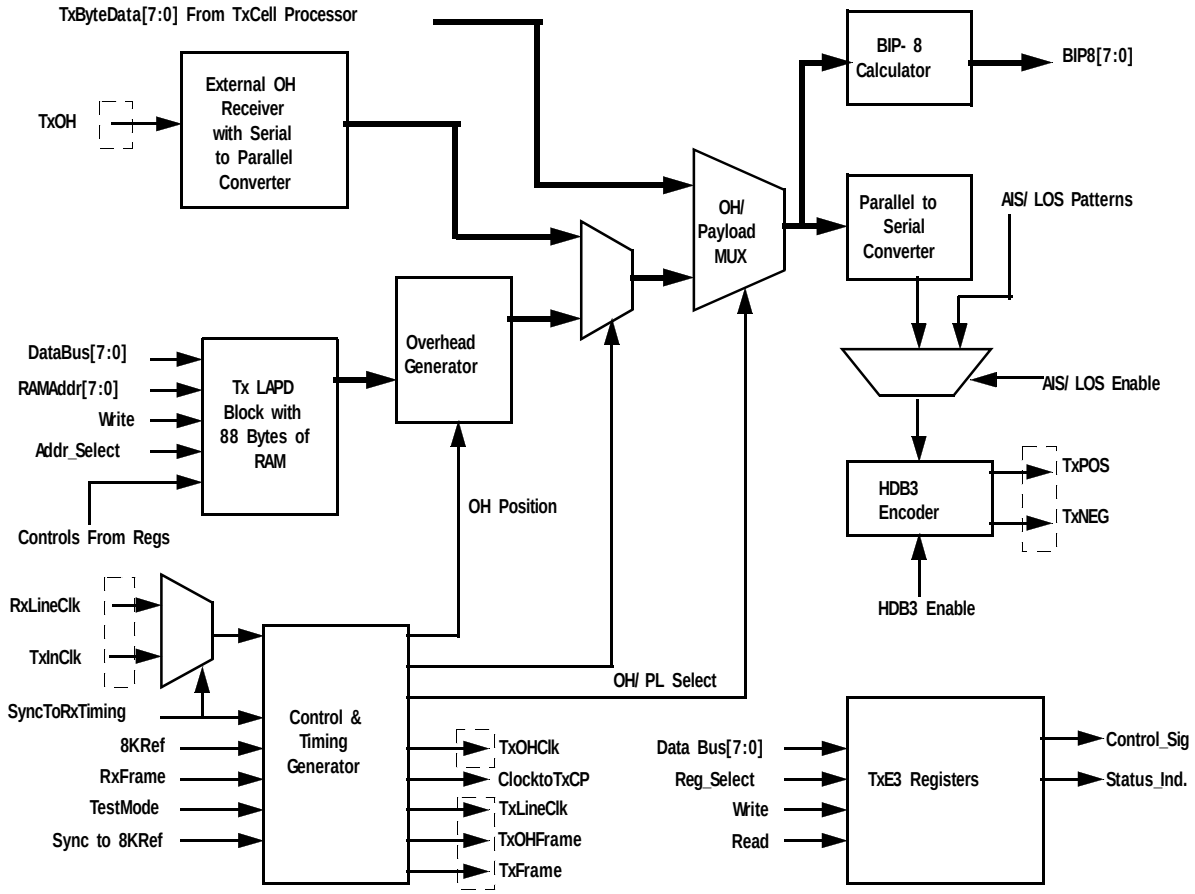

Figure 33. A Functional Block Diagram of the Transmit E3 Framer Block

6.3.3.1 The Error Monitor (EM) Byte Calculator

The purpose of the EM byte is to support error detection in the transmission of the E3 frames between the "Near-End" Transmit E3 Framer and the "Far End" Receive E3 Framer. The Transmit E3 Framer will compute the Bit-Interleaved Parity (BIP- 8) value for a given E3 frame, based upon the contents of the 537 octets, within that E3 frame. Afterwards, the Transmit E3 Framer will insert the resulting BIP- 8 value, into the "EM" byte- field of the very next outbound E3 frame.

For information on how the Receive E3 Framer processes the EM byte, please see Section 7.1.2.4.

6.3.3.2 Far End Block Error (FEBE)

The "Far- End Block Error" bit- field resides in bit 6, within the MA byte of each E3 frame, as depicted below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker

The purpose of the FEBE (Far End Block Error) bit- field (within the MA byte) is two- fold.

3. It allows a Terminal (which is transmitting E3 frames to a "Far- End" terminal) to determine whether or not the "Far- End" terminal is receiving its E3 frame data in an "error- free" fashion.

4. It allows a Terminal (which is receiving E3 frames from a "Far- End" terminal) to inform the "Far- End" Terminal when it is receiving "errored" E3 frames from that same "Far- End" Terminal.

The role of the FEBE bit- field is best presented in the practical example below.

Example

Consider a "Near- End" terminal that is communicating with a "Far- End" Terminal. This "Near- End" Terminal consists of the "Transmit E3 Framer" and "Receive E3 Framer" blocks within the XR- T7234 E3 UNI IC; as depicted in Figure 34.

The "Transmit E3 Framer" will generate and transmit E3 frames to the "Far- End" Terminal. Likewise, the "Receive E3 Framer" will receive and process E3 frames, originating from the "Far- End" Terminal. The "Near- End" Receive E3 Framer (e.g., the Receive E3 Framer on this particular chip) is going to verify the value of the EM bytes, within the incoming E3 frames (from the "Far- End" terminal). If the "Near- End" Receive E3 Framer detects no errors in the incoming EM bytes, then it will notify the "Far- End" Terminal of this fact, by having the "Near End" Transmit E3 Framer, set the "FEBE" bit (within the MA byte), within an "outbound" E3 frame (which is destined for the "Far- End" Terminal); to "0". This phenomenon is illustrated in Figures 34 and 35, below.

Figure 34 illustrates the "Near- End" Receive E3 Framer receiving an "error- free" E3 frame. In this figure, the locally computed EM byte of "5Ah" matches that received from the "Far- End" Terminal. Figure 35 illustrates the subsequent action of the "Near- End" Transmit E3 Framer, which will transmit an E3 frame, with the FEBE bit- field set to "0"; to the "Far- End" Terminal. This signaling indicates that the "Near- End" Receive E3 Framer has received an "error free" E3 frame from the "Far- End" Terminal.

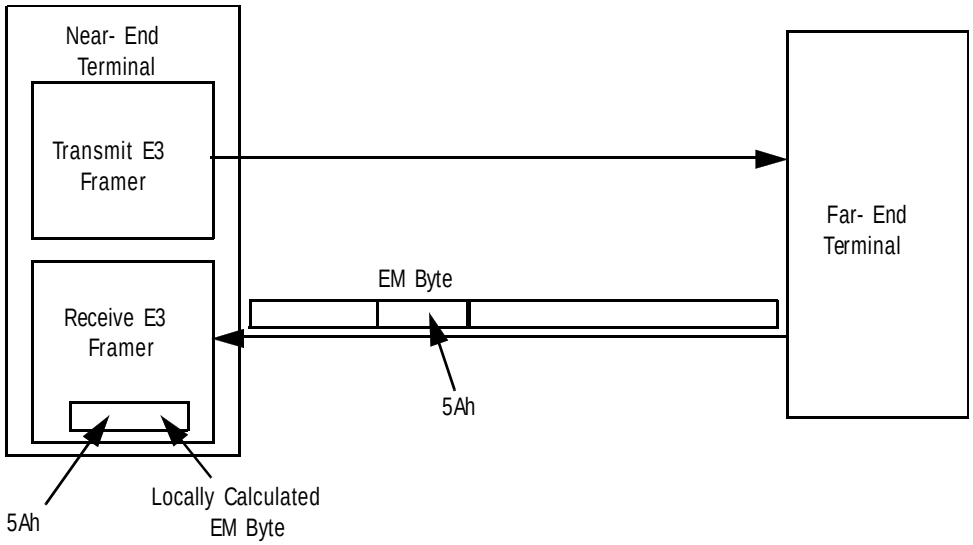

Figure 34. Illustration of the "Near- End" Receive E3 Framer, receiving an E3 Frame (from the "Far- End" Terminal) with a correct EM- byte.

Figure 35, Illustration of the "Near- End" Transmit E3 Framer, transmitting an E3 Frame (to the "Far- End" Terminal) with the FEBE bit (within the MA byte- field) set to "0".

Conversely, if the "Near- End" Receive E3 Framer detects an error in the incoming EM byte, then it will notify the "Far- End" Terminal of this fact, by having the "Near- End" Transmit E3 Framer, set the "FEBE" bit within an "outbound" E3 frame (which is destined for the "Far- End" Terminal); to "1". This phenomenon is illustrated in Figures 36 and 37.

Figure 36 illustrates the "Near- End" Receive E3 Framer receiving an "errored" E3 frame from the "Far- End" Terminal.

In this figure, the "Near- End" Receive E3 Framer is receiving an E3 frame, with an EM byte containing the value "5Ah". This value does not match the "locally computed" EM byte value of "5Bh". Consequently, there is an error in this E3 frame.

Figure 37 illustrates the subsequent action of the "Near- End" Transmit E3 Framer, which will transmit an E3 frame, with the FEBE bit- field set to "1"; to the "Far- End" Terminal. This signaling indicates that the "Near- End" Receive E3 Framer has received an "errored" E3 frame from the "Far- End" Terminal.

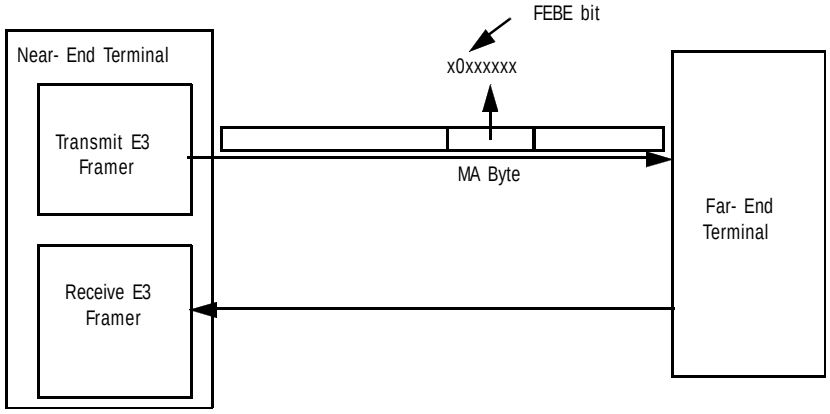

Figure 35. Illustration of the "Near- End" Receive E3 Framer, receiving an E3 Frame (from the "Far- End" Terminal) with an incorrect EM- byte.

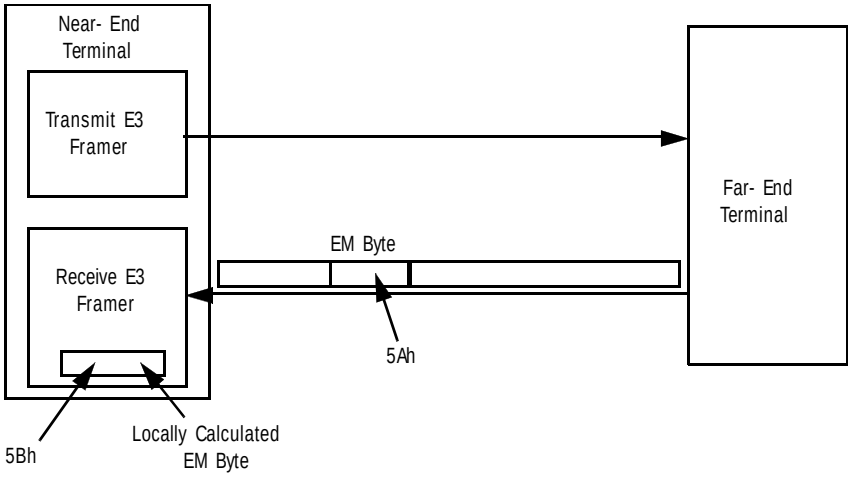

Figure 36. Illustration of the "Near- End" Transmit E3 Framer, transmitting an E3 Frame (to the "Far- End" Terminal) with the FEBE bit (within the MA byte- field) set to "1".

For information on how the Receive E3 Framer processes the FEBE bit- field, within each incoming E3 frame, please see Section 7.1.2.6.

Manipulating the "FEBE" bit- field

The user can manipulate the state of the "FEBE" bit- field, within an "outbound" E3 frame by executing the following two- step procedure.

1. Write a "0" into Bit 0 (MARx), within the "Tx E3 Configuration" register (Address = 28h); as depicted below.

Tx E3 Configuration Register (Address = 28h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Auto Re- Transmit	TxLAPDType[1:0]		DLinNR	NoData Link	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	x	x	x	x	0	1	0

This step will configure the Transmit E3 Framer to read in the contents of bit- field 6, within the "Tx MA Byte" register; and insert it into the "FEBE" bit- field of the "outbound" E3 frame.

2. Write the desired value for the "FEBE" bit- field into Bit 6 (FEBE) within the "Tx MA Byte" Register, as depicted below.

Tx MA Byte Register (Address = 2Ah)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
TxMA[7:0]							
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	x	0	1	0	x	x	x

6.3.3.3 Alarm Generation

The Transmit E3 Framer can be commanded to generate various types of alarm patterns, at will, upon software command. Some of these alarms conditions, and the approaches that are used to invoke these alarms are presented below.

6.3.3.3.1 Generating the FERF (Far- End Receive Failure) Indicator

The Transmit E3 Framer will transmit a "FERF" indication in response to the "Near- End" Receive E3 framer detecting an "LOS", AIS, or OOF condition. The Transmit E3 Framer can also transmit a "FERF" indication at the user's will, under software command.

Generating the FERF Indicator in response to "Adverse" Receive Conditions

If the Receive E3 Framer detects an LOS, AIS, or an OOF condition in the E3 frame data that it is receiving from a "Far- End" terminal; then the Receive E3 Framer will notify this "Far- End" Terminal of these problems by having the "Near- End" Transmit E3 Framer set the "FERF" bit- field, within an "outbound" E3 frame (which is destined for the "Far- End" Terminal) to "1". Conversely, if the Receive E3 Framer receives a "normal" E3 frame, from the "Far- End" Terminal; then the Receive E3 Framer will notify the "Far- End" Terminal of this fact by having the "Near- End" Transmit E3 Framer set the "FERF" bit- field, within an "outbound" E3 frame (which is destined for the "Far- End" Terminal) to "0".

Figure 38 through 41 illustrates this phenomenon.

Figure 38 illustrates the "Near- End" Receive E3 Framer experiencing an LOS condition in the E3 frame data that it is receiving from the "Far- End" Terminal. Figure 39, illustrates the subsequent action of the "Near- End" Transmit E3 Framer. In this case, the Transmit E3 Framer will set the "FERF" bit- field, in the outbound E3 frame (which is destined for the "Far- End" Terminal) to "1".

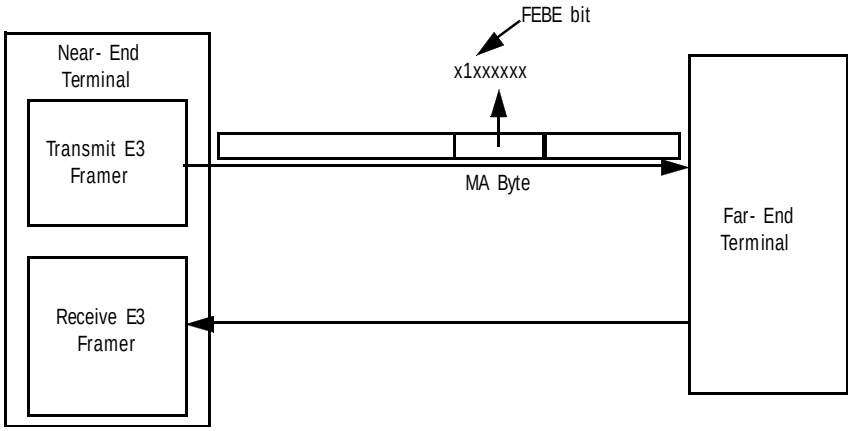

Figure 37. Illustration of the 'Near- End' Receive E3 Framer experiencing an LOS condition

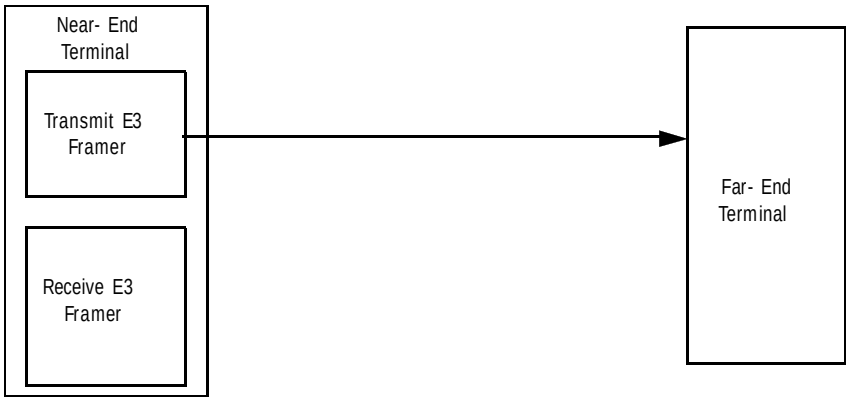

Figure 38. Illustration of the 'Near- End' Transmit E3 Framer, transmitting an E3 Frame (to the 'Far- End' Terminal) with the FERF bit (within the MA byte- field) set to "1".

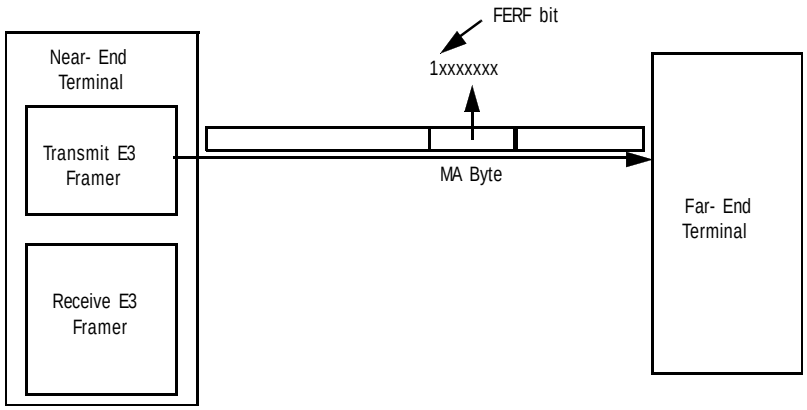

Figure 39. illustrates the "Near- End" Receive E3 Framer receiving a "normal" E3 frame from the "Far- End" Terminal.

Figure 41 illustrates the subsequent action of the "Near- End" Transmit E3 Framer. In this case, the Transmit E3 Framer will set the "FERF" bit- field, in the outbound E3 frame (which is destined for the "Far- End" Terminal) to "0".

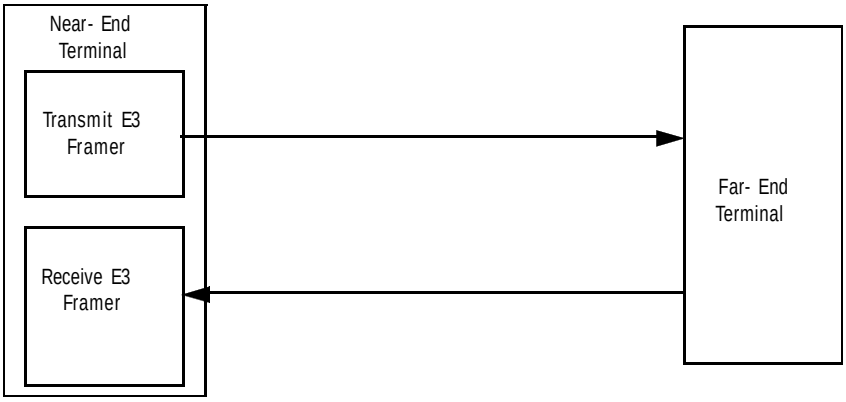

Figure 40. Illustration of the "Near- End" Receive E3 Framer receiving a proper E3 signal from the "Far- End" Terminal.

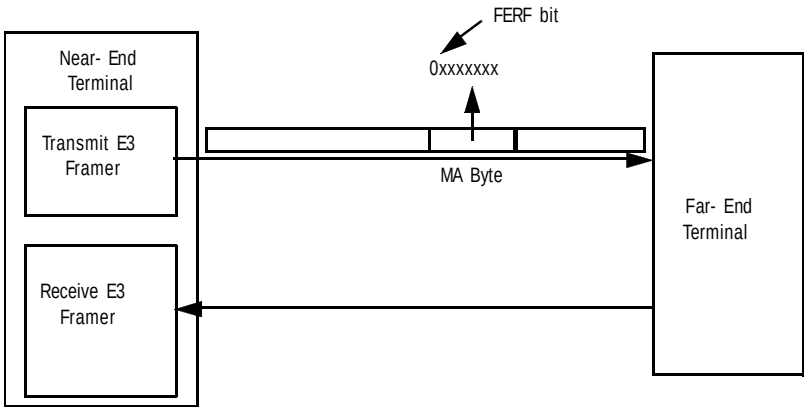

Figure 41. Illustration of the "Near- End" Transmit E3 Framer, transmitting an E3 frame (to the "Far- End" Terminal) with the FERF bit (within the MA byte- field) set to '0".

Manipulating the "FERF" bit- field

The user (or local µP) can manipulate the state of the "FERF" bit- field within an "outbound" E3 frame, by executing the following two- step procedure.

1. Write a "0" into Bit 0 (MARx) within the "Tx E3 Configuration" register (Address = 28h); as depicted below:

Tx E3 Configuration Register (Address = 28h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Auto Re-Transmit	TxLAPDType[1:0]		DLinNR	NoData Link	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	x	x	x	x	0	1	0

This step will configure the Transmit E3 Framer to read in the contents of Bit 7, within the "Tx MA Byte" register; and insert it into the "FERF" bit- field of the "outbound" E3 frame.

2. Write the desired value for the "FERF" bit into Bit 7 (FERF) within the "Tx MA Byte" register, as depicted below:

Tx MA Byte Register (Address = 2Ah)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***

Tx MA Byte Register (Address = 2Ah)							
TxMA[7:0]							
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	x	0	1	0	x	x	x

6.3.3.3.2 Generating the AIS (Alarm Indication Signal) Condition pattern

The Transmit E3 Framer can be commanded to generate an AIS condition pattern, at the user's will, upon software command.

If the user writes a "1" into Bit 2 (TxAISEn) within the "Tx E3 Framer Configuration" Register (Address = 28h), as depicted below:

Tx E3 Framer Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re-transmit	TxLAPDType[1:0]		DLinNR	NoData Link	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	X	X	X	X	1	0	X

then the Transmit E3 Framer will generate an all "1s" pattern. The "All Ones" pattern will be interpreted by the "Far-End" Receive E3 Framer as the AIS Condition. If the user writes a "0" into this bit- field then the Transmit E3 Framer will resume the transmission of the E3 frame OH and payload bytes.

Note:

1. Because the payload portion of the E3 frame is being overwritten by this "All One's" pattern, no ATM cell data will be transmitted to the "Far End" Receive E3 Framer, while this command is invoked.
2. The "All One's" pattern will also overwrite the Frame Alignment bytes, within each E3 Frame. Hence, commanding the Transmit E3 Framer to generate an "AIS" pattern will cause the "Far- End" Receive E3 Framer to lose framing and declare an OOF and LOF Conditions.

6.3.3.3.3 Generating the LOS (Loss of Signal) Condition

The Transmit E3 Framer can generate a signal that simulates an LOS condition, at the user's will, upon software command.

If the user writes a "1" into Bit 1 (TxLOSEn) within the Tx E3 Framer Configuration Register (Address = 28h), as depicted below:

Tx E3 Framer Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*

Tx E3 Framer Configuration Register (Address = 28h)							
Auto Re-transmit	TxLAPDType[1:0]		DLinNR	NoData Link	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	X	X	X	X	0	1	X

then the Transmit E3 Framer will generate an all "0s" pattern. This "all zeros" pattern will be interpreted by the "Far End" Receive E3 Framer as the LOS condition. If the user writes a "0" into this bit-field, then the Transmit E3 Framer will resume the transmission of the E3 frame OH and payload bytes.

Note:

1. Because the payload portion of the E3 frame is being overwritten by the "All Zero's" pattern, no ATM cell data will be transmitted to the "Far End" Receive E3 Framer, while this command is invoked.
2. The "All Zero's" pattern will also overwrite the Frame Alignment bytes, within each E3 Frame. Hence, commanding the Transmit E3 Framer to generate an "LOS" pattern will cause the "Far-End" Receive E3 Framer to lose framing and declare an OOF and LOF Conditions.

6.3.3.4 Trail Trace Buffers

The XRT7234 E3 UNI device contains 16 bytes worth of "Transmit" Trail Trace Buffers, and 16 bytes worth of "Receive" Trail Trace Buffers. The role of the "Transmit" Trail Trace Buffers, is described below. The role of the "Receive" Trail Trace Buffers are described in Section 7.1.2.7.

The XRT7234 E3 UNI device contains 16 "Transmit Trail Trace Buffer Registers (e.g., Tx TTB-0 through Tx TTB-15). The purpose of these registers are to provide a 16-byte "Trail Access Point Identifier" to the "Far-End" Receiving Terminal. The "Far-End" Receiving Terminal will use this information to verify that it is still receiving data from its intended transmitter. The specific use of these registers follows.

For "Trail Trace Buffer message" purposes, the Transmit E3 Framer will group 16 consecutive E3 Frames, into a "Trail Trace Buffer" super-frame. When the Transmit E3 Framer is generating the first E3 frame, within a "Trail Trace Buffer" super-frame, it will read in the contents of the "Tx TTB-0" Register (Address = 2Ch) and insert this value into the TR byte-field of this very first "outbound" E3 frame. When the Transmit E3 Framer is generating the very next E3 Frame, (e.g., the second E3 frame, within a "Trail Trace Buffer" super-frame), it will read in the contents of the "Tx TTB-1" Register (Address = 2Dh) and insert this value into the TR byte-field of this "outbound" E3 frame. As the Transmit E3 Framer is creating each subsequent E3 frame, within this "Trail Trace Buffer" super-frame, it will continue to increment to the very next "Transmit Trail Trace buffer" register. The Transmit E3 Framer will then read in the contents of this particular "Transmit Trail Trace" buffer register (e.g., Tx TTB-n) and insert this value into the TR byte-field of the very next outbound E3 frame. After the Transmit E3 Framer has created the 16th E3 frame, within a given "Trail Trace Buffer" super-frame (e.g., it has read in the contents of the "Tx TTB-15" register, and has inserted this value into the TR-byte of the 16th E3 frame); it will begin to create a new "Trail Trace Buffer" super-frame, by reading the contents of the "Tx TTB-0" register, and repeating the above-mentioned procedure.

The contents of the "Tx TTB-0" register will typically be of the form [1, C6, C5, C4, C3, C2, C1, C0]. The "1" in the MSB (most significant bit) position of this byte, is used to designate that this octet is the "frame start marker" (e.g., is the first of the 16 TR bytes, within a "Trail Trace Buffer" super-frame). The remaining Trail Trace Buffer registers (Tx TTB-1 through Tx TTB-15) will typically contain a "0" in their MSB (most significant bit) positions. The remaining bits within the "Tx TTB-0" register; C6 through C0 are the CRC-7 bits calculated over the contents of all 16 TR bytes, within the previous "Trail Trace Buffer" super-frame. The contents of the remaining "Transmit Trail Trace Buffer" registers (e.g., Tx TTB-1 through Tx TTB-15) will typically contain the 15 ASCII characters required for the E.164 numbering format.

Note:

1. The XRT7234 E3 UNI will not compute the CRC-7 value, to be written into the Tx TTB-0 Register. The user's system must compute this value prior to writing it into the Tx TTB-0 Register.

2. The user, when writing data into the Tx TTB registers, must take care to insure that only the Tx TTB- 0 register contains an octet with a "1" in the MSB (most significant bit) position. All remaining Tx TTB registers (e.g., Tx TTB- 1 through Tx TTB- 15) must contain octets with a "0" in the MSB position. The reason for this "cautionary" note is presented in Section 7.1.2.7

6.3.3.5 The LAPD Transmitter

The LAPD Transmitter allows the user to transmit path maintenance data link (PMDL) messages to the "Far- End" Receiver via the "outbound" E3 Frames. The user can configure the Transmit E3 Framer to transport the octets, comprising this outbound PMDL message, via the NR or the GC byte- fields, within each outbound E3 frame. Further, the user can configure the LAPD Transmitter to transmit a LAPD Message frame (containing a PMDL message) only once or repeatedly at one- second intervals.

6.3.3.5.1 The LAPD Message Frame

The on- chip LAPD Transmitter supports both the 76 byte and 82 byte length message formats, and the UNI allocates 88 bytes of on- chip RAM (e.g., the "Transmit LAPD Message" buffer) to store the PMDL message to be transmitted. Prior to transmission to the "Far- End" Terminal, the LAPD Transmitter will encapsulate the PMDL Message (e.g., the data which has been written into the Transmit LAPD Message Buffer) into an ITU- T Q.921 compliant (or LAPD) message frame. Figure 42 presents the byte format of the resulting LAPD Message frame.

Flag Sequence (8 bits)		
SAPI (6- bits)	C/ R	EA
TEI (7 bits)		EA
Control (8- bits)		
76 or 82 Bytes of Information (Payload)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8- bits)		

Figure 42. LAPD Message Frame Format

Where: Flag Sequence = 7Eh
SAPI + CR + EA = 3Ch or 3Eh
TEI + EA = 01h
Control = 03h

The following sections defines each of these bit/ byte- fields within the LAPD Message Frame Format.

Flag Sequence Byte

The Flag Sequence byte is assigned the value 7Eh, and is used to denote the boundaries of the LAPD Message Frame.

SAPI-Service Access Point Identifier

The SAPI bit- fields are assigned the value of "001111b" or 15_{10}.

TEI-Terminal Endpoint Identifier

The TEI bit- fields are assigned the value of 00h. The TEI field is used in N- ISDN systems to identify a terminal out of multiple possible terminal. However, since the UNI transmits data in a point- to- point manner, the TEI value is unimportant.

Control

The Control identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. The UNI assigned the Control byte the value 03h. Hence, the UNI will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The "Information Payload" is the 76 bytes or 82 bytes of data (e.g., the PMDL Message) that the user has written into the on- chip "Transmit LAPD Message" buffer (which is located at addresses 86h through DDh).

It is important to note that the user must write in a specific octet value into the first byte position within the Transmit LAPD Message buffer (located at Address = 86h, within the UNI). The value of this octet depends upon the type of LAPD Message frame/ PMDL Message that the user wishes to transmit. Table 22 presents a list of the various types of LAPD Message frames/ PMDL Messages that are supported by the XR- T7234 E3 UNI device; and the corresponding octet value that the user must write into the first octet position within the "Transmit LAPD Message buffer.

Table 22. The LAPD Message Type and the Corresponding value of the First Byte, to be written into the "Transmit LAPD Message" buffer

LAPD Message Frame Type	Value of First Byte to be written into the Transmit LAPD Message buffer (86h in the UNI Address Space)	PMDL Message Size
CL Path Identification	38h	76 bytes
IDLE Signal Identification	34h	76 bytes
Test Signal Identification	32h	76 bytes
ITU- T Path Identification	3Fh	82 bytes

Frame Check Sequence Bytes

The 16 bit FCS (Frame Check Sequence) is calculated over the LAPD Message Header bytes (e.g., the Flag Sequence, SAPI, TEI, and Control bytes) and the Information Payload bytes, by using the CRC- 16 polynomial, $x^{16} + x^{12} + x^5 + 1$. Afterwards, this FCS value is inserted into the two- octet "FCS value" position, within the LAPD Message frame.

6.3.3.5.2 The Operation of the LAPD Transmitter

If the user wishes to transmit a PMDL message via the LAPD Transmitter, he/ she must perform the following steps.

1. Specify the type of LAPD Message frame to be Transmitted (within the Transmit LAPD Message Buffer).
2. Write the PMDL Message into the remaining portion of the Transmit LAPD Message Buffer.
3. Specify the type of LAPD message frame to be transmitted (via Register Settings).
4. Specify which byte- field (within the E3 Frame) that the LAPD Message frame is to be transported on.
5. Specify whether the LAPD Transmitter should transmit this LAPD Message frame only once, or an indefinite number of times at one- second intervals.
6. Enable the LAPD Transmitter
7. (Initiate the Transmission)

Each of these steps are discussed in detail below.

1. Specify the type of LAPD Message frame to be Transmitted (within the Transmit LAPD Message Buffer)

The user must write in a specific octet value into the first octet position within the Transmit LAPD Message Buffer (e.g., at address location 86h within the UNI). This octet is referred to as the "LAPD Message frame ID' octet. The value of this octet must correspond to the type of LAPD Message frame, he/ she wishes to transmit. This octet will ultimately be used by the "Far- End" LAPD Receiver, in order to help it identify the type of LAPD message frame that it is receiving. Table 22 lists these octets and the corresponding LAPD Message types.

2. Write the PMDL Message into the remaining part of the Transmit LAPD Message Buffer.

The user must now write in his/ her PMDL Message into the remaining portion of the Transmit LAPD Message Buffer (e.g., addresses 87h through 135h within the UNI).

3. Specifying the Type of LAPD Message to be Transmitted (via Register Settings)

Once again, the user must specify which type of LAPD Message he/ she wishes to send. However, in this step the user is informing the LAPD Transmitter which type of message that he/ she wishes to transmit. As mentioned earlier, the user can transmit one of four different types of LAPD Messages. He/ she can accomplish this by writing the appropriate data to bits 1 and 2 of the Tx E3 Configuration Register (Address = 28h). The bit- for-

mat of this register is presented below.

Tx E3 Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re- transmit	TxLAPDType[1:0]		DLinNR	No DataLink	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The relationship between the contents of bit- fields 5 and 6, within the "Tx E3 Configuration" Register, and the LAPD Message type follows.

Table 23. Relationship between TxLAPDType[1:0] and the LAPD Message Type/ Size

TxLAPD Type[1:0]	LAPD Message Type/ Size
00	LAPD Message type is "Test Signal" type. The size of this message is 76 bytes
01	LAPD Message type is "Idle Signal Identification" type. The size of this message is 76 bytes.
10	LAPD Message type is "CL Path Identification Type". The size of this message is 76 bytes
11	LAPD Message type is "ITU Path Identification Type". The size of this message is 82 bytes.

Note: This register configuration is used by the "Near- End" LAPD Transmitter, in order to determine how to encapsulate the PMDL message into a LAPD Message frame. Therefore, this Message Type selected must correspond with the value of the first byte written into the "Transmit LAPD Message" buffer, as presented in Table 22.

4. Specifying which byte- field (within the E3 Frame) that the LAPD Message frame octets are to be transported on.

The Transmit E3 Framer allows the user to transport the LAPD Message frame octets via either the NR byte or via the GC byte- field, within each outbound E3 Frame. The user makes this selection by writing the appropriate value to bit- field 4 (DLinNR), within the "Tx E3 Configuration" Register, as depicted below.

Tx E3 Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re- transmit	TxLAPDType[1:0]		DLinNR	No DataLink	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

If the user writes a "0" into this bit- field, then the LAPD Transmitter will transmit the comprising octets of the outbound LAPD Message frame via the GC byte- field, within each outbound E3 frame. Additionally, the Transmit E3 Framer will insert the contents of the "Tx NR Byte" Register (Address = 2Bh) into the NR byte of each outbound E3 frame.

Conversely, if the user writes a "1" into this bit- field, then the LAPD Transmitter will transmit the outbound LAPD Message frame octets via the NR byte- field, within each outbound E3 frame. Additionally, the Transmit E3 Framer will insert the contents of the "Tx GC Byte" Register (Address = 29h) into the GC byte of each outbound E3 frame.

5. Specify whether the LAPD Transmitter should transmit the LAPD Message frame only once, or an indefinite number of times at one- second intervals.

The Transmit E3 Framer allows the user to configure the LAPD Transmitter to transmit this LAPD Message frame only once, or an indefinite number of times at one- second intervals. The user implements this configuration by writing

the appropriate value into Bit 7 (Auto Retransmit) within the "Tx E3 Configuration" register; as depicted below.

Tx E3 Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re- transmit	TxLAPDType[1:0]		DLinNR	No DataLink	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

If the user writes a "1" into this bit- field, then the LAPD Transmitter will transmit the LAPD Message frame repeatedly at one- second intervals until the LAPD Transmitter is disabled.

If the user writes a "0" into this bit- field, then the LAPD Transmitter will transmit the LAPD Message frame only once. Afterwards, the LAPD Transmitter will halt its transmission until the user invokes the "Transmit LAPD Message frame" command, once again.

6. Enable the LAPD Transmitter

Prior to the transmission of any data via the LAPD Transmitter, the user must enable the LAPD Transmitter. He/she can accomplish this by writing a "0" into Bit 3 (No DataLink) of the Tx E3 Configuration Register, as depicted below.

Tx E3 Configuration Register (Address = 28h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Auto Re- transmit	TxLAPDType[1:0]		DLinNR	No DataLink	TxAIS En	TxLOS En	MARx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

If the user writes a "0" into this bit- field, then the LAPD Transmitter will be enabled, and the LAPD Transmitter will begin to transmit a continuous stream of Flag Sequence octets (7Eh), via either the GC or the NR byte- field of each outbound E3 frame (depending upon the user's setting for Bit 4 of this register).

If the user writes a "1" into this bit- field, then the LAPD Transmitter will be disabled. The Transmit E3 Framer will then insert the contents of the Tx GC Byte Register, into the GC byte- field, for each outbound E3 frame. Likewise, the Transmit E3 Framer will also insert the contents of the Tx NR Byte Register, into the NR byte, for each outbound E3 frame. No transmission of data from the LAPD Transmitter will occur.

7. Initiate the Transmission

At this point, the user should have written the PMDL message into the on- chip Transmit LAPD Message Buffer. He/she should have specified the type of LAPD message that he/she wishes to transmit. Third, he/she should have specified whether the LAPD Transmitter will transport the LAPD Message frame octets via the GC byte- field or via the NR byte- field of each outbound E3 frame. Finally, the user should have enabled the LAPD Transmitter. The only thing remaining to do is to initiate the transmission of this message. The user initiates this process by writing a "1" to Bit 3 (TxDL Start) within the Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh); as depicted below.

Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused				TxDL Start	TxDL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	R/W

A "0" to "1" transition of Bit 3 (TxDL Start) in this register, initiates the transmission of LAPD Message frames. At this point, the LAPD Transmitter will begin to search through the PMDL message, which is residing within

the "Transmit LAPD Message" buffer. If the LAPD Transmitter finds any string of five (5) consecutive "1s" in the PMDL Message; then the LAPD Transmitter will insert a "0" immediately following these strings of consecutive "1s". This procedure is known as "stuffing". The purpose of "PMDL Message" stuffing is to insure that the user's PMDL Message does not contain strings of data that mimic the "Flag Sequence" byte (e.g., six consecutive "1s") or the "Abort Sequence" (e.g., seven consecutive "1s"). Afterwards, the LAPD Transmitter will begin to encapsulate the PMDL Message, residing in the Transmit LAPD Message buffer, into a LAPD Message Frame. Finally, the LAPD Transmitter will fragment the "outbound" LAPD Message frame into octets; and will begin to transport these octets via the GC or the NR byte- fields (depending upon the user's selection) of each outbound E3 frame.

While the LAPD Transmitter is transmitting this LAPD Message frame, the "TxDL Busy" (Bit 2) bit, within the Tx E3 LAPD Status/ Interrupt Register, will be set to "1"; as depicted below:

Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused				TxDL Start	TxDL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	R/W
0	0	0	0	1	1	x	x

This bit- field allows the user to "poll" the status of the LAPD Transmitter. Once the LAPD Transmitter has completed the transmission of the LAPD Message frame, this bit- field will toggle back to "0".

The user can configure the LAPD Transmitter to interrupt the local µC/µP upon completion of transmission of the LAPD Message frame, by setting bit- field "1" (TxLAPD Interrupt Enable) of the "Tx E3 LAPD Status/ Interrupt" register to "1", as depicted below:

Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused				TxDL Start	TxDL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	x

The purpose of this interrupt is to let the local µC/µP know that the LAPD Transmitter is available and ready to transmit a LAPD Message frame with a new PMDL message. Bit 0 (TxLAPD Interrupt Status) within the Tx E3 LAPD Interrupt/ Status Register will reflect the status for the LAPD Transmitter Interrupt.

Note: This bit- field will be reset upon reading this register.

Summary of Operating the LAPD Transmitter

Once the user has invoked the "TxDL Start" command, the LAPD Transmitter will do the following.

- Generate the four octets of the LAPD Message frame header (e.g., Flag Sequence, SAPI, TEI, Control, etc.) and insert it into the header octet positions within the LAPD Message frame.
- It will read in the contents of the "Transmit LAPD Message" buffer (e.g., the PMDL message data) and insert it into the "Information Payload" portion of the LAPD Message frame.
- Compute the 16- bit "Frame Check Sum" (FCS) of the LAPD Message Frame (e.g., of the LAPD Message header and the PMDL message data) and insert this value into the FCS value octet positions within the LAPD Message frame.
- Append a "trailer" Flag Sequence octet to the end of the LAPD Message frame (following the 16 bit FCS value).
- Fragment the resulting LAPD Message frame into octets; and begin inserting these octets, into either the GC or NR byte- field of the outbound E3 frame (depending upon the user's selection).
- Complete the transmission of the frame overhead, information payload, FCS value, and trailing Flag Sequence octets via the Transmit E3 Framer.

Once the LAPD Transmitter has completed its transmission of the LAPD Message frame, the UNI will generate an interrupt to the local µC/µP (if enabled). Afterwards, the LAPD Transmitter will either halt its transmission of LAPD Message frames or will proceed to retransmit the LAPD Message frame, repeatedly at one- second intervals. In between these transmission of the LAPD Message frame, the LAPD Transmitter will be sending an continuous stream of "Flag Sequence" bytes. The LAPD Transmitter will continue this behavior until the user has disabled the LAPD Transmitter by writing a "1" into bit 3 (No DataLink) within the Tx E3 Configuration Register.

Note: In order to prevent the user's data (the PMDL Message within the LAPD Message Frame) from mimicking the "Flag Sequence" byte or an Abort Sequence, the LAPD Transmitter will search through the PMDL Message data and insert a "0" into this data, immediately following the detection of five (5) consecutive "1s" (this "stuffing" occurs while the PMDL message data is being read in from the "Transmit LAPD Message buffer" and is being encapsulated into a LAPD Message frame. The "Far- End" LAPD Receiver (See Section 7.1.2.5) will have the responsibility of checking the "newly received" PMDL messages for a string of 5 consecutive "1s" and removing the subsequent "0" from the payload portion of the incoming LAPD message.

Figures 43 and 44 presents two flow chart diagrams. Figure 43 depicts the procedure (in "white boxes") that the user should use in order to transmit a PMDL Message via the LAPD Transmitter, when the LAPD Transmitter is configured to "retransmit" the LAPD Message frame, repeatedly at one second intervals. This figure also indicates (via

the "Shaded" boxes) what the LAPD Transmitter circuitry will do before and during message transmission.

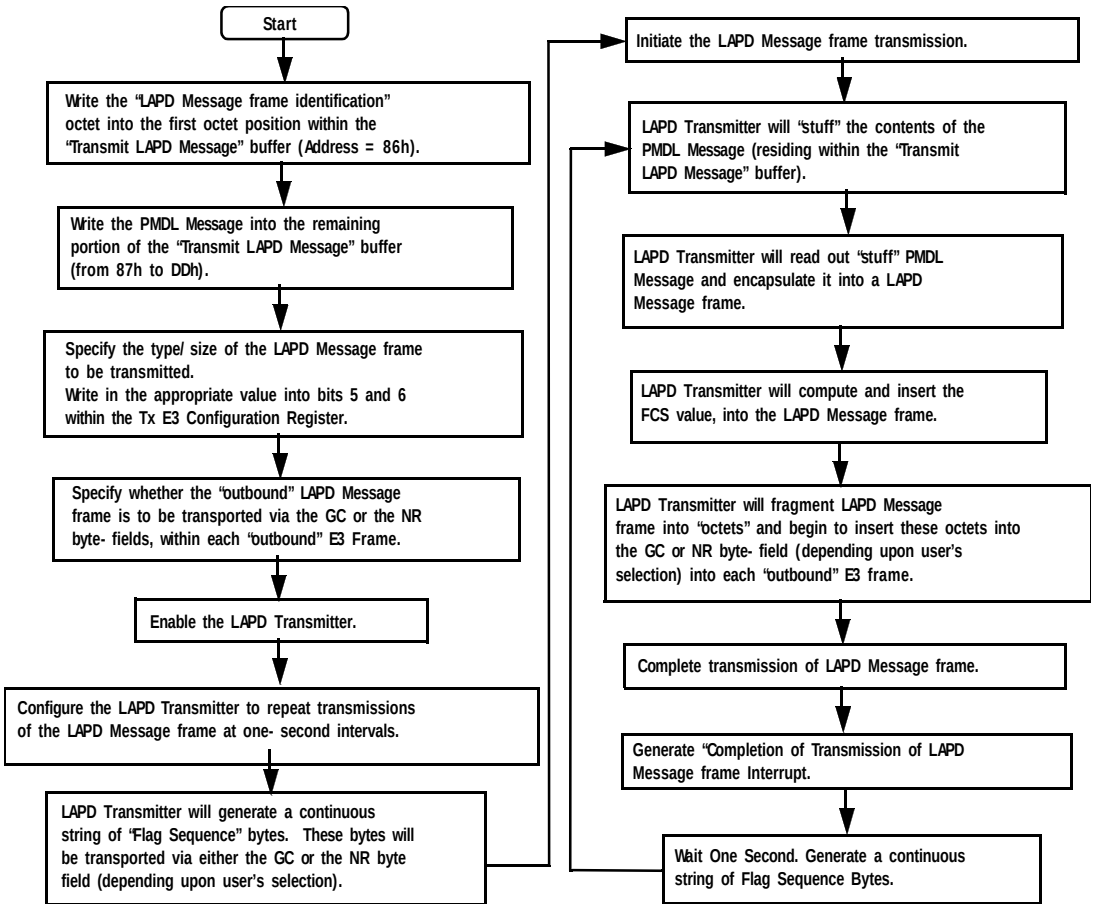

Figure 43. Flow Chart depicting how to use the LAPD Transmitter (LAPD Transmitter is configured to re- transmit the LAPD Message frame repeatedly at one- second intervals)

Figure 44 presents the procedure (in "white boxes") that the user should use in order to transmit a PMDL Message via the LAPD Transmitter, when the LAPD Transmitter is configured to transmit the LAPD Message frame only

once, and then halt transmission.

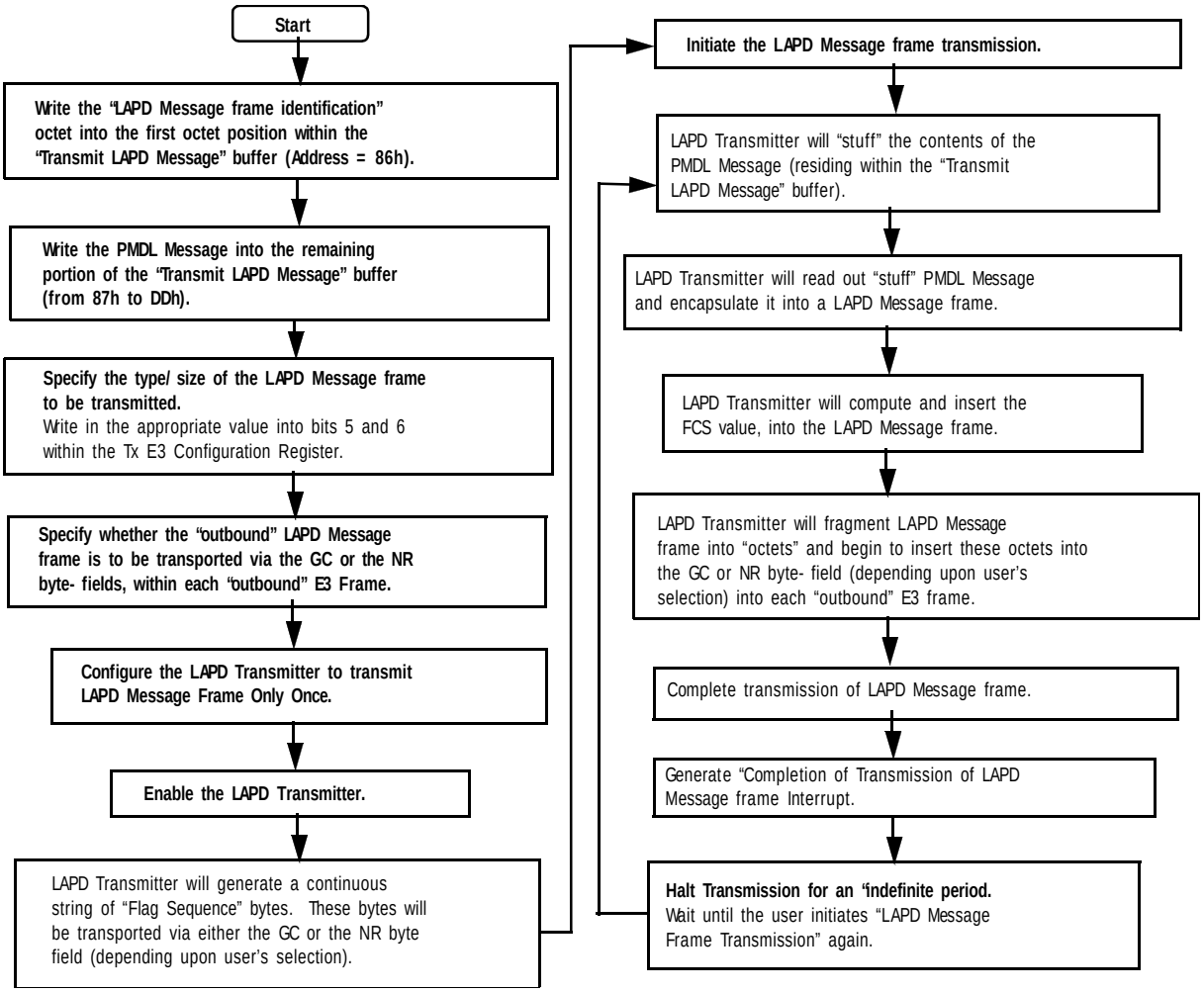

Figure 44. Flow Chart depicting how to use the LAPD Transmitter (LAPD Transmitter is configured to transmit the LAPD Message frame only once, and then halt)

The Mechanics of Transmitting New LAPD Message, if the LAPD Transmitter has been configured to retransmit the LAPD Message frame, repeatedly at one- second intervals

If the LAPD Transmitter has been configured to retransmit the LAPD Message repeatedly at one- second intervals, then it will do the following at one- second intervals.

- "Stuff" the PMDL Message.
- Read in the "stuffed" PMDL Message from the "Transmit LAPD Message" buffer.
- Encapsulate this "stuffed" PMDL Message into a LAPD Message frame.
- Transmit this LAPD Message frame to the "Far- End" Terminal.

If the user wishes to transmit another (e.g., a different) PMDL message to the "Far End" LAPD Receiver, he/ she will have to write this "new" message into the "Transmit LAPD Message" buffer, via the Microprocessor Interface section of the UNI. However, the user must take care when writing in this new PMDL message. If he/ she writes this

message into the "Transmit LAPD Message" buffer at the "wrong time" (with respect to these "one- second" LAPD Message frame transmissions), the user's action could interfere with these transmissions; thereby causing the LAPD Transmitter to transmit a "corrupted" message to the "Far- End" LAPD Receiver. In order to avoid this problem, while writing the new message into the "Transmit LAPD Message" buffer, the user should do the following.

1. Configure the UNI to automatically reset activated interrupts

The user can do this by writing a "1" into Bit 5 of the "UNI I/O Control" Register, as depicted below:

UNI I/O Control Register, (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LCC Rx	LCC Tx	IntEn Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

This action will prevent the LAPD Transmitter from generating its own "one- second" interrupts.

2. Enable the "One- Second" Interrupt

This can be done by writing a "1" into Bit 6 of the "UNI Interrupt Enable" Register, as depicted below:

UNI Interrupt Enable Register (Address = 04h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec Interrupt Enable	Tx E3 Framer Interrupt Enable	Rx E3 Framer Interrupt Enable	Tx CP Interrupt Enable	Rx CP Interrupt Enable	Tx Utopia Interrupt Enable	Rx Utopia Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	0	0

3. Write the new message into the "Transmit LAPD Message" buffer immediately after the occurrence of the "One- Second" Interrupt

By timing the writes to the "Transmit LAPD Message" buffer to occur immediately after the occurrence of the "One Second" Interrupt, the user avoids conflicting with the "one- second" transmission of the LAPD Message frame, and will transmit the correct (uncorrupted) PMDL message to the "Far End" LAPD Receiver.

6.3.3.6 Timing/ Framing Reference for the Transmit E3 Framer Block

The Transmit E3 Framer will generate and transmit E3 frame data, based upon one of three different timing or framing sources:

- Receive E3 Framer timing
- TxInClk input signal
- TxFrameRef input signal

The user can select which one of these three timing or framing source signals to use as the reference for the Transmit E3 Framer, by writing to Bits 0 and 1 (TimRefSel[1, 0]) within the UNI Operating Mode Register

(Address = 00h); as depicted below.

UNI Operating Mode Register: (Address = 00h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	Disable LOC	IntLOS Enable	Reset by Reg	Cell Loop Back	Line Loop Back	TimRefSel[1, 0]	
RO	R/W	R/W	R/W	R/W	R/W	R/W	

The following table lists the values of Bits 0 and 1 and the resulting timing source for the Transmit E3 Framer block.

Table 24. The Relationship between TimRefSel[1:0] (e.g., Bits 1 and 0 of the UNI Operating Mode Register), and the resulting Timing/ Framing Source for the Transmit E3 Framer Block

TimRefSel [1:0]		**Transmit E3 Framer Timing/ Frame Source**
0	0	TxInClk input signal. Framing is asynchronous upon "power on".
0	1	TxFrameRef input signal.
1	0	RxLineClk input signal–The Recovered Clock Signal from the incoming (Received) E3 line signal.
1	1	TxInClk input signal. Framing is asynchronous upon "power on".

Each of these "potential" timing and framing sources for the Transmit E3 Framer are discussed in greater detail below.

6.3.3.6.1 RxLineClk-Receive E3 Framer Timing

In this mode, the Transmit E3 Framer Timing is based upon the recovered clock input signal, RxLineClk, obtained by the Receive E3 Framer. E3 framing (e.g., when the Transmit E3 Framer begins to create a new frame) will be asynchronous with respect to any external timing signals.

This mode is convenient from the stand- point that it requires no external timing sources (to either the TxFrameRef or TxInClk pins). However, this configuration has one drawback: If the Receive E3 Framer block experiences an LOS condition, or somehow loses the recovered clock signal, then the Transmit E3 Framer block will essentially have no timing source.

Therefore, in this configuration, the operation of the Transmit E3 Framer block is dependent upon the actions of the "Near- End" Receive E3 Framer block and its incoming E3 data- stream signal.

6.3.3.6.2 TxInClk Input Signal

In this mode, the Transmit E3 Framer block will use the clock signal that is input at the TxInClk pin, as the timing reference. E3 framing (e.g., when the Transmit E3 Framer begins to create a new frame) will be asynchronous with respect to any external timing signals. If the user selects this mode, then he/ she must insure that a high quality 34.368 MHz clock signal is applied at this input.

The advantage of using this timing signal as the reference for the Transmit E3 Framer, over the RxLineClk signal; is that an LOS condition (or a loss of clock recovery event with the incoming E3 line signal) does not adversely affect the Transmit E3 Framer's operation.

6.3.3.6.3 TxFrameRef Input Signal

In this mode, the Transmit E3 Framer block will use the input signal, at the TxFrameRef input pin as the "framing reference". In other words, a rising edge at this input pin will cause the Transmit E3 Framer to begin its creation of a new E3 frame. Consequently, if the user selects this input signal as the framing reference signal, he/ she must supply a 8 kHz clock signal to this input pin.

6.3.3.7 Interfacing the Transmit E3 Framer to the Line

The XR- T7234 E3 ATM UNI is a digital device that takes ATM cell data from an ATM Layer processor, processes this cell data and ultimately maps this information into the payload portion of an outbound E3 frame. In most applications, the user will transmit this E3 frame data to the "Far- End" terminal over a "transformer- coupled" copper medium (e.g., coaxial cable) or over optical fiber. If the user wishes to transmit the E3 frame data over coaxial cable, then he/ she must interface the UNI device to a line interface unit (LIU) IC. An LIU is a device that has sufficient drive capability, along with the necessary pulse- shaping circuitry to be able to transmit a signal through the transmission medium in a manner that it can be reliably received by the "Far- End" receiver. Figure 45 presents a circuit schematic depicting the UNI interfacing to an LIU (XR- T7296 E3 Transmit LIU).

If the user wishes to transmit the E3 frame data over optical fiber, then he/ she must interface the UNI to an "Electrical- to- Optical" converter device.

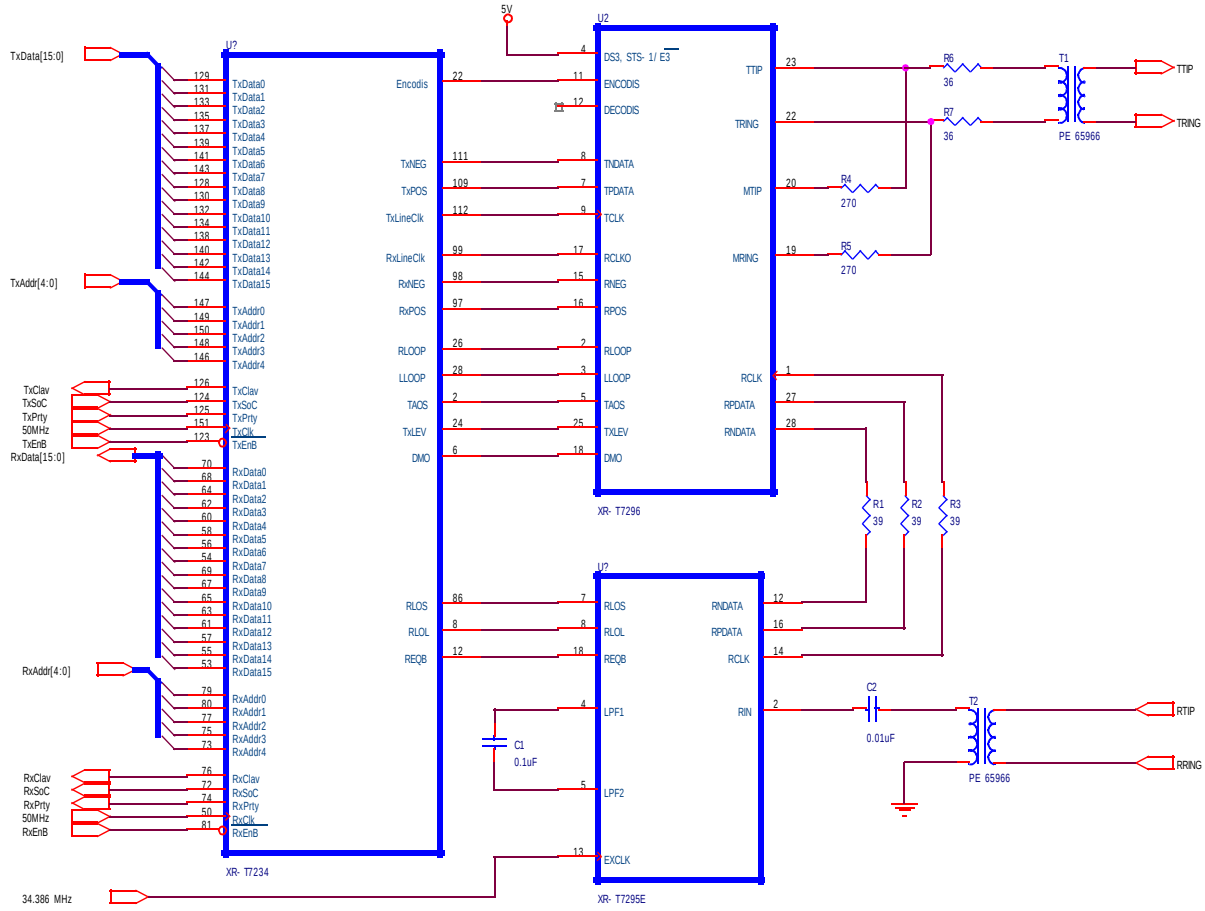

Figure 45. Schematic of the XR- T7234 E3 UNI interfacing to the XR- T7295E and XR- T7296 E3 Line Interface Unit devices.

6.3.3.7.1 Line Interface Output Mode

The UNI device can be configured to operate in one of two "Line Interface" output modes.

- Unipolar
- Bipolar

Each of these "Line Interface" output modes are described below.

6.3.3.7.1.1 The Unipolar Mode

The Transmit E3 Framer can transmit data to the LIU IC or other external circuitry via two different output modes: Unipolar or Bipolar. If the user selects Unipolar (or Single- Rail) mode, then the contents of the E3 Frame is output via the TxPOS pin in a NRZ (non- return- to- zero) format to external circuitry. The TxNEG pin will only be used to denote the frame boundaries. TxNEG will pulse "high" for one bit- period, at the start of each new E3 frame, and will remain "low" for the remainder of the frame period. Figure 46 presents an illustration of the TxPOS and TxNEG signals during data transmission while the UNI is operating in the Unipolar mode. This mode is sometimes referred to as the "Single Rail" mode because the data pulses only exist in one polarity: positive.

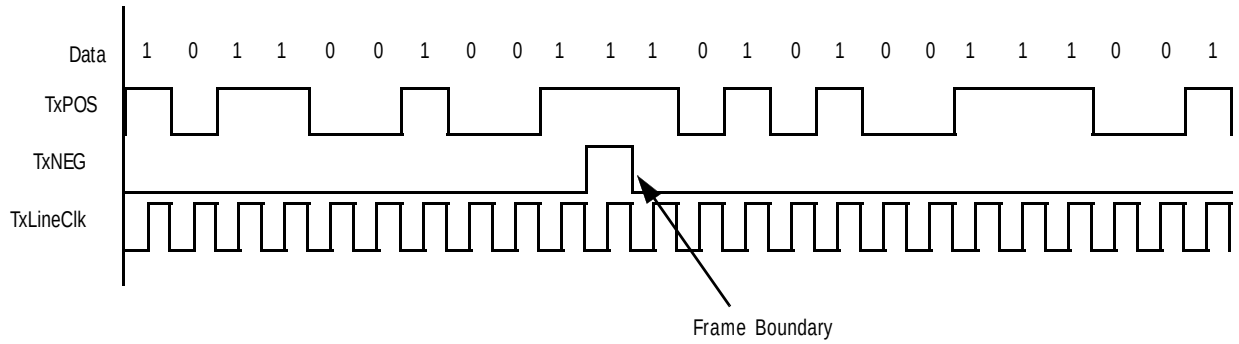

Figure 46. The Behavior of TxPOS and TxNEG signals during data transmission while the UNI is operating in the Unipolar Mode

Note: The user is advised not to operate the UNI in the Unipolar mode if he/ she wishes to transmit the E3 frame data to a "Far- End" E3 Terminal over some transformer- coupled copper medium. The user should, instead, use one of the Bipolar- mode line codes for this application. The Unipolar mode can be used to drive the line signals through optical fiber. In this case, the TxPOS output would be applied to an LED at the electrical/ optical interface circuitry.

The user can configure the UNI to operate in the Unipolar Mode, by writing a "1" to Bit 3 (Unipolar/ Bipolar*) within the UNI I/ O Control Register (Address = 01h), as shown below.

UNI I/ O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	0	x	x	1	x	x	0

6.3.3.7.1.2 The Bipolar Mode

When the UNI is operating in the Bipolar (or Dual- Rail) mode, then the contents of the E3 frame data is output via both the TxPOS and TxNEG pins. If the user chooses the Bipolar mode, then he/ she can transmit the E3 line signal to the "Far- End" Receiver via one of two different line codes: Alternate Mark Inversion (AMI) or High Density Bipolar- 3 (HDB3). Each one of these line codes will be discussed below. The Bipolar mode is sometimes referred to as the "Dual- Rail"

mode because the data pulses occur in two polarities: positive and negative. It is important to note that the bipolar mode is the "mode of choice", if the user is trying to transmit the E3 frame data over transformer- coupled coaxial cable. This is because the "bipolar" line codes introduce no dc component into the line signal. Hence, dc distortion is minimized. The role of the TxPOS, TxNEG and TxLineClk output pins, for the Bipolar mode are discussed below.

TxPOS-Transmit Positive Polarity Pulse:

The UNI device will assert this output to the LIU IC when it desires for the LIU to generate and transmit a "positive polarity" pulse to the "Far- End" Receiver.

TxNEG-Transmit Negative Polarity Pulse:

The UNI device will assert this output to the LIU IC when it desires for the LIU to generate and transmit a "negative polarity" pulse to the "Far- End" Receiver.

TxLineClk-Transmit Line Clock:

The LIU IC uses this signal, from the UNI, to sample the state of its TxPOS and TxNEG inputs. The results of this sampling dictates the type of pulse (positive polarity, zero, or negative polarity) that it will generate and transmit to the "Far- End" Receive E3 Framer.

The user can configure the UNI to operate in the Bipolar Mode, by writing a "0" to Bit 3 (Unipolar/ Bipolar*) within the UNI I/ O Control Register (Address = 01h), as shown below.

UNI I/ O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	0	x	x	0	x	x	0

6.3.3.7.1.2.1 The AMI Line Code

AMI or Alternate Mark Inversion, means that consecutive "one's" pulses (or marks) will be of opposite polarity, with respect to each other. The line code involves the use of three different amplitude levels" "+1", "0", "- 1". The "+1" and "- 1" amplitude signals are used to represent one's (or mark) pulses and the "0" amplitude pulses (or the absence of a pulse) are used to represent zeros (or space) pulses. The general rule for AMI is: if a given "mark" pulse is of "positive" polarity, then the very next "mark" pulse will be of "negative" polarity and vice versa. This alternating polarity relationship exists between two consecutive mark pulses, independent of the number of "zeros" that may exist between these two pulses. The Transmit E3 Framer takes the binary stream of E3 frame data and outputs this data to the LIU device, in the AMI line code format via the TxPOS and TxNEG output pins. Figure 47 presents an illustration of E3 frame data which is encoded into the AMI Line code. This figure illustrates this AMI encoded E3 frame as it would appear at the TxPOS and TxNEG pins of the UNI, as well as the output signal on the line.

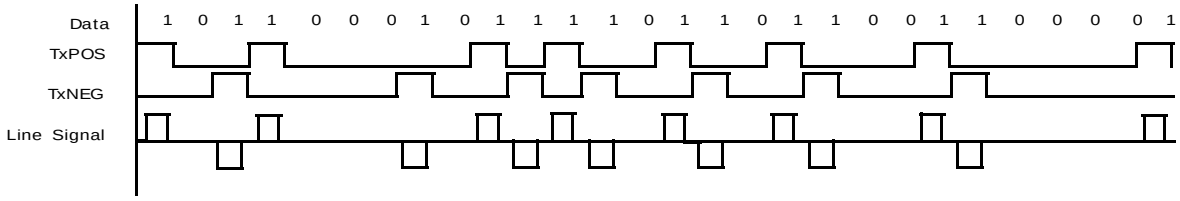

Figure 47. Illustration of AMI Line Code as output by the Transmit E3 Framer, within the UNI device.

The user can configure the UNI to operate with the AMI Line code by writing a "1" to Bit 4 (AMI/HDB3*) within the "UNI I/O Control" Register, as depicted below.

"UNI I/O Control" Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	x	1	0	x	x	0

Note: This bit- field is ignored if the UNI is configured to operate in the Unipolar Mode.

6.3.3.7.1.2.2 The HDB3 Line Code

The Transmit E3 Framer and the associated LIU IC combine the E3 frame data and timing information (originating from the TxLineClk signal) into the E3 line signal that is transmitted to the "Far- End" Terminal. The LIU at the "Far- End" Terminal has the task of recovering this data and timing information from the incoming pulses on the E3 line. Most E3 LIU devices, on the market today, contain clock and data recovery schemes that rely on the use of Phase- Locked- Loop (PLL) circuitry. The PLL circuitry, within the LIU, assists in the clock and data recovery process by "locking" onto the transitions in the E3 line signal. Therefore, these clock recovery schemes are vulnerable to the occurrence of a long stream of consecutive zeros (e.g., the absence of transitions in the line). Hence, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique that is used in the E3 transport medium is HDB3 encoding. In general, the HDB3 line code is very similar to the AMI line code: with the exception of the case when a long string of consecutive zeros occurs on the line. In HDB3 line coding, any string of 4 consecutive zeros will be replaced with either a "000V" or a "B00V" where "B" refers to a Bipolar pulse (e.g., a pulse with a polarity that is compliant with the AMI coding rule). And "V" refers to a Bipolar Violation pulse (e.g., a pulse with a polarity that violates the alternating polarity scheme of AMI). The decision between inserting an "000V" or a "B00V" is made to insure that an odd number of Bipolar (B) pulses exist between any two Bipolar Violation (V) pulses. When the UNI is configured to transmit E3 frame data in the HDB3 line code, the Transmit E3 Framer will take the E3 frame data and output it (onto the line) in the HDB3 format via the TxPOS and TxNEG output pins. Figure 48 presents an illustration of the Transmit E3 Framer taking E3 frame data (in a binary data stream format) and converting it into the HDB3 encoded format. The "HDB3- encoded" dual- rail data is then output to the LIU via the TxPOS and TxNEG output pins. Figure 48 also illustrates the resulting signal on the E3 line.

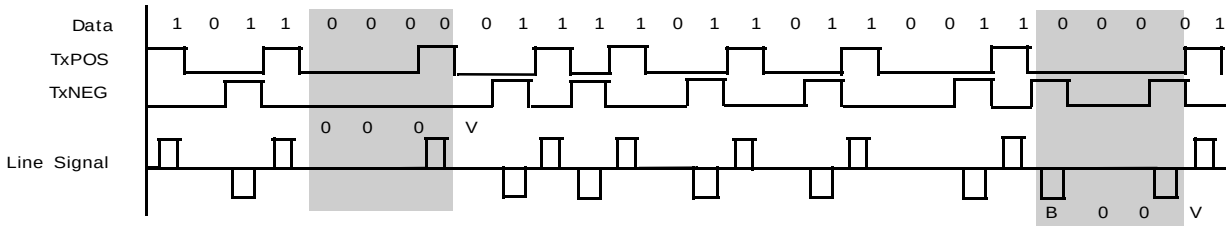

Figure 48. Illustration of two examples of HDB3 Encoding

Note: Figure 48 shows two examples of "zero- suppression" due to the HDB3 line code. In the first example, a string of 4 consecutive zeros is replaced by a "000V" pattern. In the second example, a string of 4 consecutive zeros is replaced by a "B00V" pattern.

The user can configure the UNI to operate with the HDB3 Line Code by writing a "0" to Bit 4 (AMI/ HDB3*) within the "UNI I/ O Control" Register, as depicted below.

UNI I/ O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	0	x	0	0	x	x	0

Note: This bit- field is ignored if the UNI is configured to operate in the Unipolar Mode.

6.3.3.7.2 TxLineClk Clock Edge Selection

The UNI allows the user to specify whether the E3 frame output data (via the TxPOS and TxNEG output pins) is to be updated on the rising or falling edges of the TxLineClk signal. This selection is made by writing to bit 2 (TxLineClk Inv) within the "UNI I/ O Control" Register, as depicted below.

UNI I/ O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	0	x	1	0	x	x	0

The following table relates the contents of this bit- field to the clock edge of TxLineClk that updates the data on the TxPOS and TxNEG output pins.

Table 25. The Relationship between the contents of Bit 2 (TxLineClk Inv) within the UNI I/O Control Register and the TxLineClk clock edge that TxPOS and TxNEG outputs are updated on

Bit 2	Result
0	**Rising Edge:** Outputs on TxPOS and/or TxNEG are updated on the rising edge of TxLineClk. See Figure 49 for timing relationship between TxLineClk, TxPOS and TxNEG signals, for this selection.
1	**Falling Edge:** Outputs on TxPOS and/or TxNEG are updated on the falling edge of TxLineClk. See Figure 50 for timing relationship between TxLineClk, TxPOS and TxNEG signals, for this selection.

Note: The user will typically make the selection based upon the "set- up" and "hold" time requirements of the "Transmit LIU" IC.

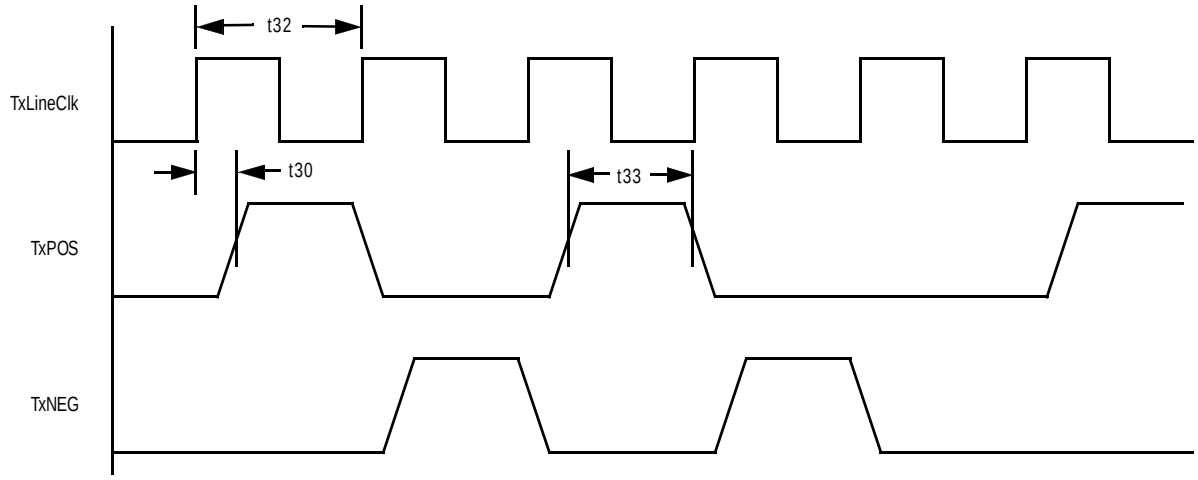

Figure 49. Waveform/ Timing Relationship between TxLineClk, TxPOS and TxNEG— TxPOS and TxNEG are configured to be updated on the rising edge of TxLineClk

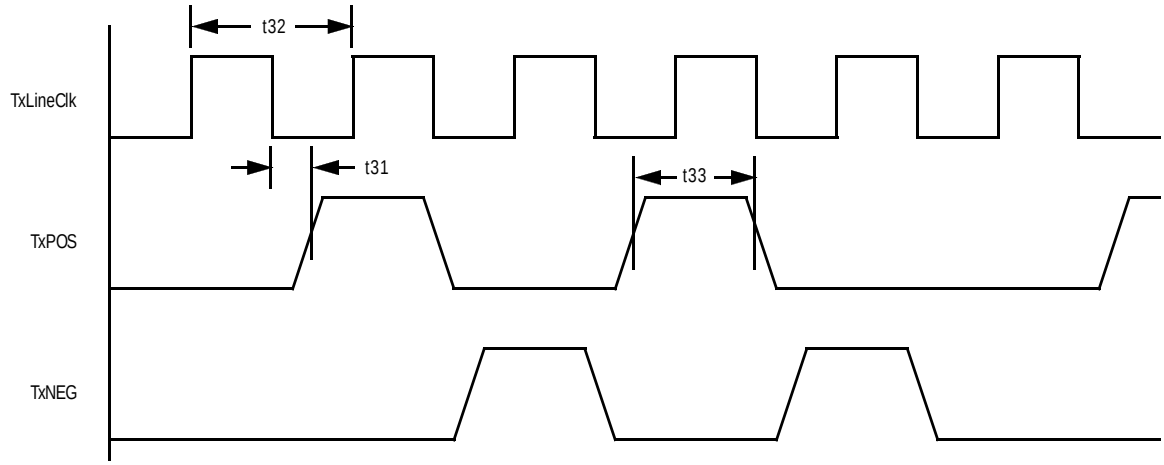

Figure 50. Waveform/ Timing Relationship between TxLineClk, TxPOS and TxNEG-TxPOS and TxNEG are configured to be updated on the falling edge of TxLineClk

6.3.4 The Transmit E3 Framer Serial Output Port

The TxOH Serial Input allows the user to insert his/ her value(s) for the overhead (OH) bytes into the outbound E3 Frames. The TxOH Serial Input Port is activated when the user asserts the TxOHIns pin (e.g., toggles this input pin "high"). Once this serial port is activated, then the user is expected to serially apply his/ her choices for the E3 OH bytes at the TxOH input pin. The TxOHClk output pin functions as a clock that will sample and latch data at the TxOH input pin on its rising edge. The frequency of this TxOHClk clock signal is approximately 448 kHz. The TxOHFrame output signal is provided to inform the user when the value for the MSB (most significant bit) of the FA1 byte is expected. The TxOH Serial Input port expects the user to apply his/ her value for OH bytes in the order as presented in Figure 31.

The TxOH Serial Input port also allows the user to selectively externally insert his/ her value for the E3 OH data (e.g., allowing some of the OH bytes to be internally generated, while inserting his/ her own values for the remaining overhead bits). This can be accomplished by "keeping track" of the number of clock pulses occurring since the last assertion of the TxOHFrame signal, and by asserting the TxOHIns at the time the TxOH serial input port would be expecting the "OH byte(s) that the user wishes to be externally inserted". The TxOHIns input should be negated for the remainder of the E3 frame sampling period, so that these OH bytes will be internally generated.

Figure 51 presents a timing diagram that illustrates the behavior of the TxOH serial interface signals during user

input of the E3 CH bits.

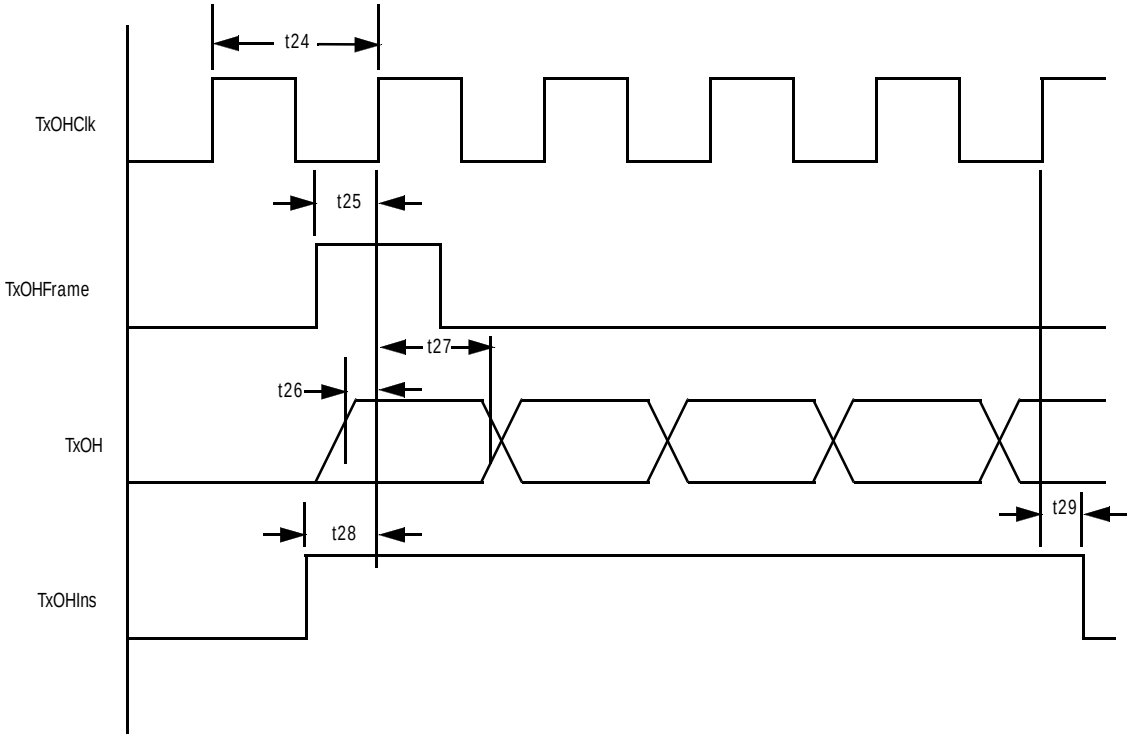

Figure 51. Timing Diagram illustrating the Behavior of the E3 CH Byte Serial Input Port, during user input of the CH bytes.

6.3.5 Transmit E3 Framer Interrupts

The Transmit E3 Framer block will generate an interrupt to the local µP/µC upon the occurrence of one condition:

- Completion of Transmission of LAPD Message frame

If this condition occurs, and if this particular condition has been enabled for interrupt generation, then when the local µP/µC reads the UNI Interrupt Status Register (during the early stages of Interrupt Processing), as shown below, it

should read 0x1xxxxxb (where the - b suffix denotes a binary expression, and the "x" denotes a "don't care" value).

UNI Interrupt Status Register (Address = 05h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	One Sec Interrupt Status	Tx E3 Framer Interrupt Status	Rx E3 Framer Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO
0	0	1	0	0	0	0	0

At this point, the local µP/µC has determined that the Transmit E3 Framer block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly. Since the Transmit E3 Framer only contains one possible source of interrupt, the Interrupt Service Routine should branch to the "Tx E3 LAPD Status/ Interrupt" Register (Address = 3Fh). The roles/ functions of the bit- fields, within this register, that are relevant to interrupt processing are described below.

Tx E3 LAPD Status/ Interrupt Register (Address = 3Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused				TxDL Start	TxDL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	X	X	X	X

This register has four (4) active bit- fields. However, only two of these bit- fields are relevant to interrupt processing. Bit 0 is an interrupt status bit and Bit 1 is an interrupt enable bit.

Bit 0–TxLAPD Interrupt Status

This "Reset Upon Read" bit- field is asserted once the LAPD Transmitter has completed transmission of a LAPD Message frame to the "Far- End" Receiver. Additionally, the UNI will notify the local µC/µP of this fact by asserting the INT* output pin to the local µC/µP. The purpose of this interrupt is to alert the local µC/µP that the LAPD Transmitter has completed the transmission of the LAPD message frame, and that it is ready and available to transmit another PMDL Message.

Bit 1–TxLAPD Interrupt Enable

This "Read/ Write" bit- field allows the user to enable/ disable interrupts generated due to the completion of transmitting a LAPD message frame to the "Far- End" Receiver. The user can enable this interrupt by writing a "1" to this bit- field. Conversely, writing a "0" into this bit- field disables this interrupt.

7.0 The Receive Section

The purpose of the Receiver Section of the XRT7234 E3 ATM UNI device is to allow a local ATM Layer (or ATM Adaptation Layer) processor to receive ATM cell data from a remote piece of equipment via a public or leased E3 transport medium.

The Receive Section of the E3 UNI chip consists of the following functional blocks:

- Receive E3 Framer
- Receive Cell Processor
- Receive Utopia Interface

The Receive E3 Framer will synchronize itself to this incoming E3 Data Stream (containing ATM cells) via the RxPOS, RxNEG, and RxLineClk input pins, and proceed to "strip off" and process the OH bytes of the E3 frame. Once all of the OH bytes have been removed, the payload portion of the received E3 Frame should consist of ATM cells, which are subsequently sent on to the Receive Cell Processor.

The Receive Cell Processor takes the "unframed" stream of ATM cell data and performs the following operations.

- Cell Delineation.
- HEC Byte Verification

It takes the first four octets of the cell (the header) and computes a HEC byte. The Receive Cell Processor will then compare this locally computed HEC byte value with that of the fifth octet, within the incoming cell. If the two HEC byte values are equal, then the cell is then retained for further processing. If the two HEC byte values are not equal, then most of the cells with single-bit errors are corrected. However, the cell is optionally discarded if multiple-bit errors are detected.

- Idle Cell Filtering

The Receive Cell Processor will detect and (optionally) remove Idle Cells. And can be configured to filter User and OAM cells.

- User Cell Filtering

The Receive Cell Processor can be configured to filter user or OAM cells, based upon their header byte patterns.

- The Receive Cell Processor will (optionally) de-scramble the payload portion of the cell (the 6th through the 53rd octet), and pack these octets in with the cell header bytes, and the HEC byte for transmission to the Receive Utopia Interface block.

The following sections discuss the blocks comprising the Receiver section of the E3 UNI in detail.

7.1 RECEIVE E3 FRAMER

7.1.1 Brief Description of the Receive E3 Framer

The Receive E3 Framer receives the E3 frame data from the line, via the RxPOS, RxNEG, and RxLineClk input pins; and it synchronizes itself to the incoming E3 data-stream. It decodes and frames the incoming data into E3 frames. The Receive E3 Framer supports the ITU-T G.832 compliant E3 framing format. It detects Line Code Violations (LCV), the Loss of Signal (LOS), the Alarm Indication Status (AIS), Out of Frame (OOF), and Loss of Frame (LOF) conditions. The Receive E3 Framer computes the BIP-8 values over a given E3 frame and compares it that within the BIP byte-field in the very next incoming E3 frame. It extracts and processes the E3 frame overhead bytes and provides them to a serial output port. Finally, the Receive E3 Framer will receive LAPD Message frames from the "Far-End" Transmit E3 Framer. Afterwards, the LAPD Receiver (within the Receive E3 Framer) will extract out the PMDL Messages from the LAPD Message frames, and will write it into the "Receive LAPD Message" buffer.

The Receive E3 Framer will detect and generate interrupts upon error conditions. The status of the Receive E3 Framer can be read by registers through the UNI- Microprocessor Interface. The Receive E3 Framer will route the contents of the E3 frame payload to the Receive Cell Processor.

Figure 52 presents a simple block diagram of the Receive E3 Framer along with the associated pins. Additionally, Figure 53 presents a more in- depth functional block diagram of the Receive E3 Framer.

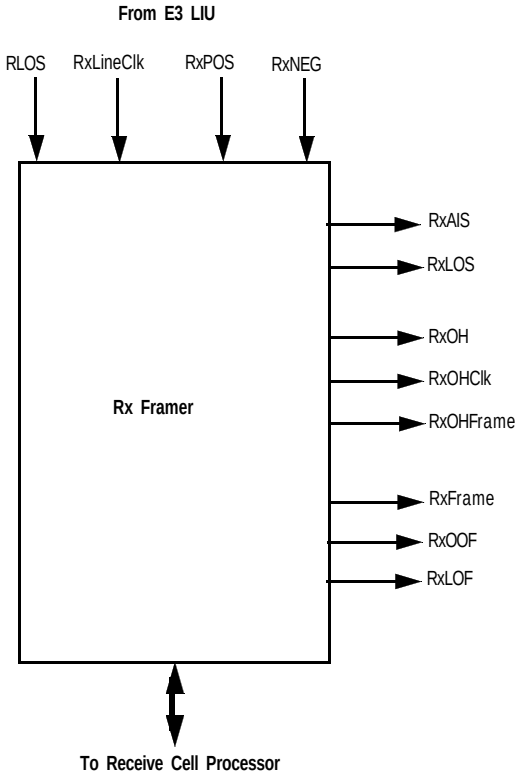

Figure 52. Simple Block Diagram of the Receive E3 Framer, with associated pins

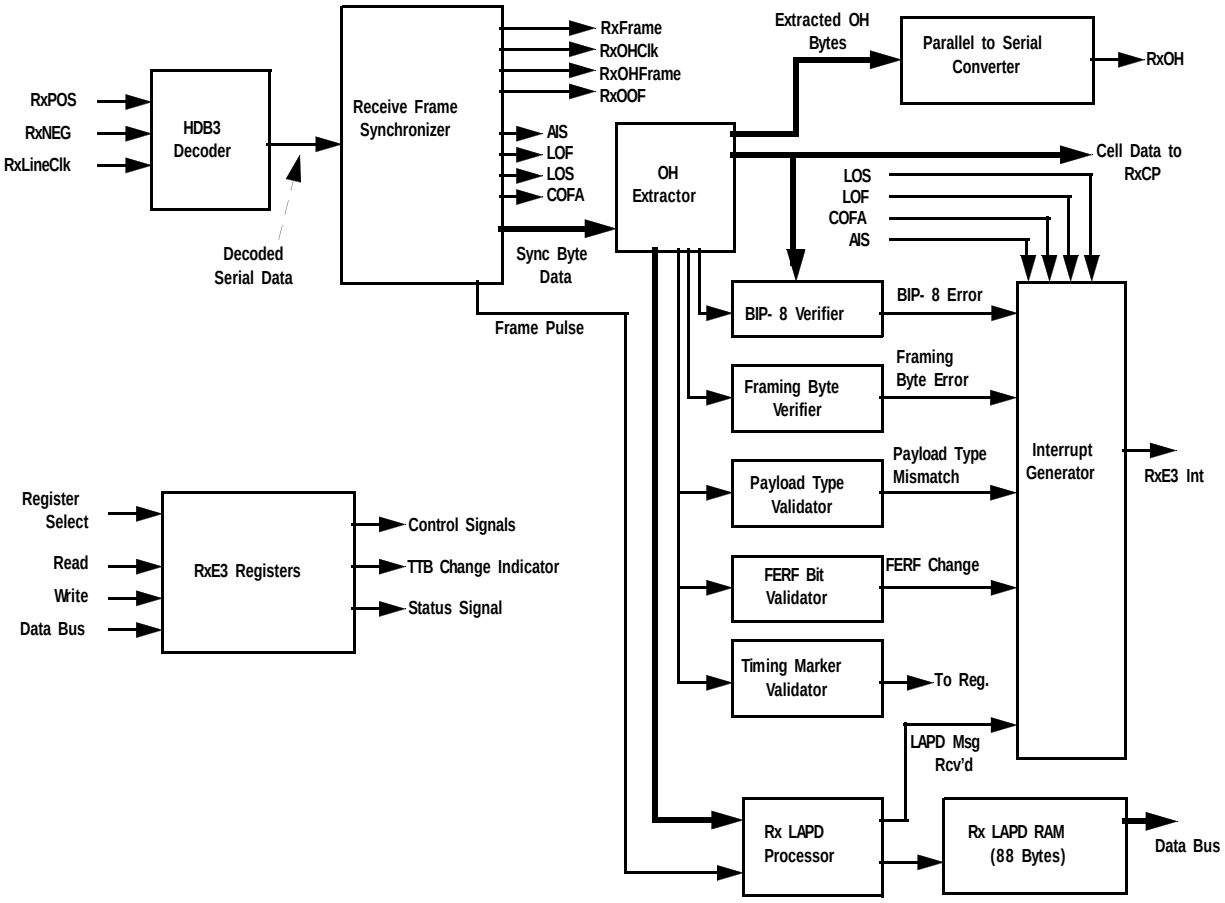

Figure 53. Functional Block Diagram of the Receive E3 Framer

7.1.2 DETAILED FUNCTIONAL DESCRIPTION OF THE RECEIVE E3 FRAMER

7.1.2.1 Receiving and Decoding Incoming E3 Data via the E3 Line

When incoming pulses, within the E3 line signal arrive at a "receiving" terminal, the E3 line signal is typically received by the "Line Interface Unit" (LIU) device. The LIU will recover the clock and data from the E3 line signal, and output this data to the Receive E3 Framer via the RxPOS, RxNEG, and RxLineClk input pins. Therefore, the Receive E3 Framer will receive both timing and data information from the incoming E3 data stream. The E3 timing information will be received via the RxLineClk input pin; and the E3 frame data information will be received via the RxPOS and RxNEG input pins. The Receive E3 Framer is capable of receiving E3 data pulses in either the "Unipolar" or "Bipolar" modes. If the Receive E3 Framer is operating in the "Bipolar" mode, then it can be configured to decode either AMI or HDB3 line code data. Each of these input formats and line codes will be discussed in detail, below.

7.1.2.1.1 Unipolar Mode Decoding

If the Receive E3 Framer is operating in the "Unipolar" Mode, then it will receive the Single Rail NRZ E3 data pulses via the RxPOS input pin. The Receive E3 Framer will also receive its timing signal via the RxLineClk signal. However, no data pulses will be applied to the RxNEG input pin. The Receive E3 Framer receives a logic "1" when a logic "1" level signal is present at the RxPOS input pin, during the sampling edge of the RxLineClk signal. Likewise, a logic "0" is received when a logic "0" level signal is applied to the RxPOS pin. Figure 54 presents an illustration of the behavior of the RxPOS, RxNEG, and RxLineClk pins when the UNI is receiving E3 frame data, while operating in the Unipolar Mode.

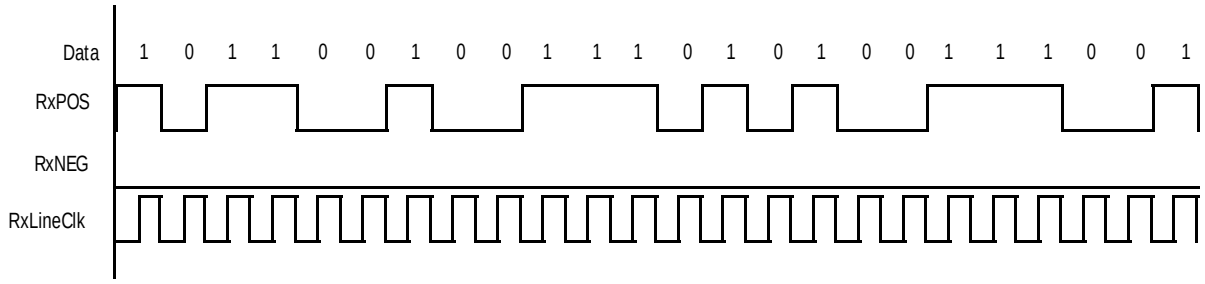

Figure 54. Behavior of the RxPOS, RxNEG, and RxLineClk signals during data reception of Unipolar Data

Note: Operating the UNI in the Unipolar Mode is recommended if the user intends to transmit and receive the E3 frame data over optical fiber. However, if the user intends to transmmit and receive the E3 data over a transformer-coupled copper medium (e.g., coaxial cable), then he/she is urged to operate the UNI in the "Bipolar" mode.

The user can configure the Receive E3 Framer to operate in either the Unipolar or the Bipolar Mode by writing the appropriate value into Bit 3 (Unipolar/Bipolar*) within the "UNI I/O Control" Register, as depicted below:

UNI I/O Control Register (Address = 01h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LCC Rx	LCC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	x	1	0	x	x	0

The following table relates the value within Bit 3 (Unipolar/Bipolar*) to the Receive E3 Framer Line Interface Input Mode.

Table 26. The Relationship between the contents of Bit 3 (Unipolar/Bipolar*) within the UNI I/O Control Register and the resulting Receive E3 Framer Line Interface Input Mode

Bit 3	Receive E3 Framer Line Interface Input Mode
0	**Bipolar Mode (Dual Rail):** E3 Frame data is receive via both the RxPOS and the RxNEG input pins.
1	**Unipolar Mode (Single- Rail):** E3 Frame data is received via only the RxPOS input pin. No E3 frame data is received via the RxNEG input pin.

Note:

1. The default condition is the Bipolar Mode
2. This selection also effects the Transmit E3 Framer Line Interface Output Mode

7.1.2.1.2 Bipolar Mode Decoding

If the Receive E3 Framer is operating in the Bipolar Mode, then it will receive the E3 data pulses via both the RxPOS and RxNEG inputs. Additionally, the Receive E3 Framer will also receive the E3 timing signal via the RxLineClk input pin. Figure 55 presents a circuit diagram illustrating how the Receive E3 Framer interfaces to the Line Interface Unit while the UNI is operating in the Bipolar Mode. If the Receive E3 Framer is operating in the "Bipolar" mode, then it can be configured to decode either the AMI or HDB3 line codes.

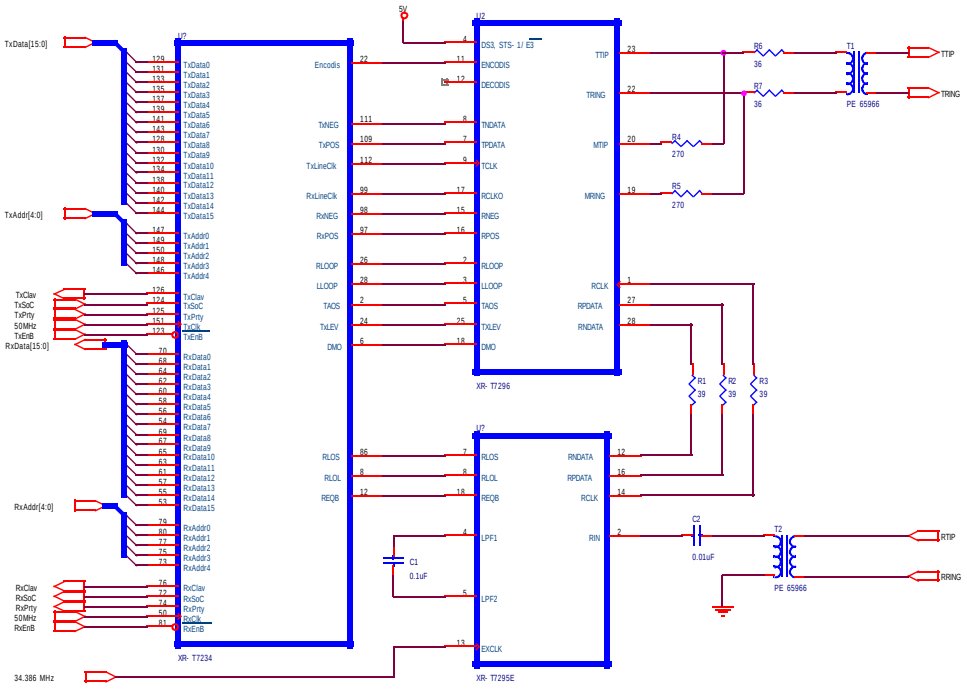

Figure 55. Illustration on how the Receive E3 Framer interfaces to the Line Interface Unit, while the UNI is operating in the Bipolar Mode.

The Bipolar mode is the "mode of choice" if the user intends to transmit and receive E3 data over transformer- coupled copper medium (e.g., coaxial cable). This is because the Bipolar line codes (AMI and HDB3) contain no dc components; thereby eliminating dc distortion in the line. The role of the RxPOS, RxNEG, and RxLineClk input pins, for the Bipolar Mode are discussed below.

RxPOS-Receive "Positive- Polarity" pulse

The external LIU will assert this input pin, when it is receiving a "positive- polarity" pulse from the line.

RxNEG-Receive "Negative- Polarity" pulse

The external LIU will assert this input pin, when it is receiving a "negative- polarity" pulse from the line.

RxLineClk-Receive Line Clock

This signal is the "recovered clock" output signal, which the LIU has extracted from the incoming E3 line signal. The UNI IC uses this signal, to sample the state of the RxPOS and RxNEG input pins.

7.1.2.1.2.1 AMI Decoding

In the AMI or Alternate Mark Inversion line code, consecutive "one's" (or mark) pulses will be of opposite polarity with respect to each other. This line code involves the use of three different amplitude levels: "+1", "0", and "- 1". The "+1" and "- 1" amplitude signals are used to represent the one's (or mark) pulses in the E3 data- stream. The "0" amplitude pulses (or the absence of a pulse) are used to represent zeros (or space) pulses, in the E3 data- stream. The general rule for AMI is: if a given "mark" pulse is of positive polarity, then the very next "mark" pulse will be of negative polarity, and vice versa. This alternating- polarity relationship exists between two consecutive mark pulses, independent of the number of zeros that exist between these two pulses. The Line Interface Unit has the task of receiving the line signal and converting it into a "dual- rail" format (e.g., positive- polarity data and negative- polarity data). The two signals that comprise the "dual- rail" format are applied to the RxPOS and RxNEG input pins of the UNI. Figure 56 presents an illustration of the AMI Line Code as would appear at the RxPOS and RxNEG pins of the UNI, as well as the incoming line signal.

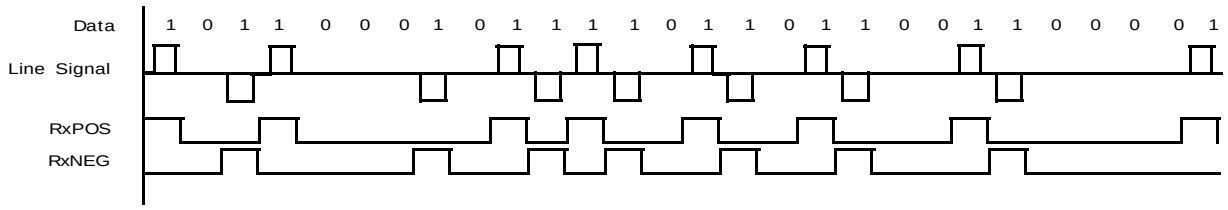

Figure 56. Illustration of the AMI Line Code

7.1.2.1.2.2 HDB3 Decoding

The Transmit E3 Framer, and the associated LIU embed and combine the E3 frame data and clocking information into the line signal that is transmitted to the "Far- End" terminal. The Line Interface Unit and Receive E3 Framer at the "Far- End" terminal, has the task of recovering this data and timing information from the incoming E3 data stream. Most E3 LIUs, on the market today, employ clock and data recovery schemes that rely on the use of Phase- Locked- Loop (PLL) circuitry. The PLL circuitry, within the LIU, assists in the clock and data recovery process by "locking" onto the transitions in the E3 line signal. Therefore, these clock recovery schemes are vulnerable to the occurrence of a long stream of consecutive zeros (e.g., no transitions in the line). This scenario can cause the PLL to lose "lock" with the incoming E3 data, thereby causing the "clock" and data recovery process of the LIU device to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One standard technique that is used in the E3 Transport Medium is "HDB3 encoding". In general, the HDB3 line code behaves just like AMI; with the exception of the case when a long string of consecutive zeros occurs on the line. In HDB3 line coding, any string of 4 consecutive zeros will be replaced with either a "000V" or a "B00V" where "B" refers to a Bipolar pulse (e.g., a pulse with a polarity that is compliant with the AMI coding rule). And "V" refers to a Bipolar Violation pulse (e.g., a pulse with a polarity that violates the alternating polarity scheme of AMI.) The decision between inserting an "000V" or a "B00V" is made to insure that an odd number of Bipolar (B) pulses exist between any two Bipolar Violation (V) pulses. The Receive E3 Framer, when operating with the HDB3 line code is responsible for decoding

the HDB3- encoded data (e.g., substituting the "000V" or "B00V" pattern with 4 consecutive "0s".) in order to restore the data (transmitted over the E3 transport medium) into its original format. Figure 57 presents a timing diagram that illustrates the following.

- The "bipolar" pulses of the incoming E3 line signal.
- The resulting "dual- rail" input signals applied to the RxPOS and RxNEG input pins of the UNI (Note this signal is still in the HDB3- format).
- The "HDB3 decoded" E3 data stream (in the "Data" row of this figure).

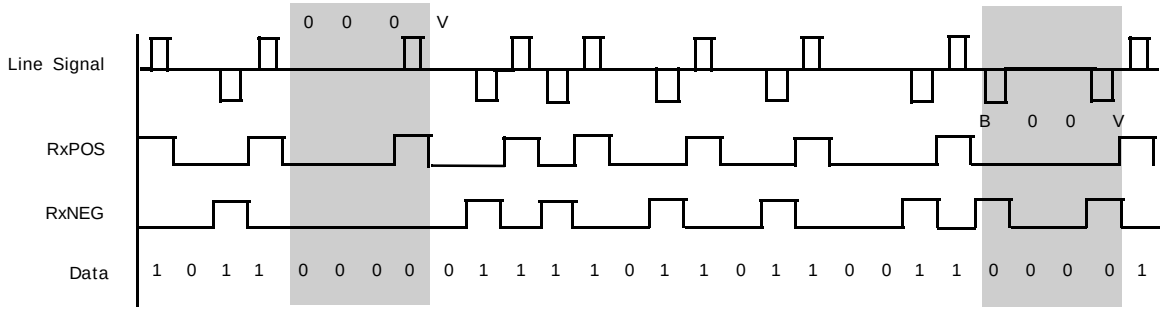

Figure 57. Illustration of two examples of HDB3 Decoding

Note: Figure 57 presents two examples of HDB3- decoding, which is accomplished by the Receive E3 Framer block (within the UNI device). In the first example, a "000V" pattern is decoded back into a string of "4" consecutive zeros. In the second example, a "B00V" pattern is decoded back into a string of "4" consecutive zeros.

7.1.2.1.2.3 Line Code Violations

The Receive E3 Framer will also check the incoming E3 data stream for line code violations. For example, if the Receive E3 Framer detects the occurrence of a bipolar violation pulse that is determined to be invalid, (e.g., the Bipolar Violation pulse is not involved in the substitution of an "000V" or a "B00V" pattern in the place of 4 consecutive "0s".) then an LCV (Line Code Violation) is flagged and the Receive E3 Framer will increment the "PMON LCV Event Count- MSB/ LSB" Registers (Address = 40h and 41h). Additionally, the LCV- One Second Accumulation Registers will also be incremented. Another example of a Line Code Violation would be: if the incoming E3 data is HDB3 encoded, and if four (or more) consecutive zeros are received in the E3 line signal. Such an event would also cause the Receive E3 Framer to increment these PMON LCV Registers.

The user can determine the number of line code violation events that have been detected by the Receive E3 Framer, by reading the PMON LCV Event Count Registers. The bit- format for these registers is presented below

PMON LCV Event Count Register-MSB (Address = 40h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
LCV Event Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON LCV Event Count Register-LSB (Address = 4h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LCV Event Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

These registers contain a 16- bit expression on the number of line code violation events that have been detected since the last read of these registers. These registers are reset upon read.

7.1.2.1.3 RxLineClk Clock Edge Selection

The incoming Unipolar or Bipolar data, applied to the RxPOS and the RxNEG input pins are clocked into the Receive E3 Framer via the RxLineClk signal. The UNI allows the user to specify which edge (e.g., rising or falling) of the RxLineClk signal will sample and latch the signals at the RxPOS and RxNEG input pins into the UNI. The user can make this selection by writing the appropriate data to bit 1 (RxLineClk Inv) of the "UNI I/O Control" Register, as depicted below.

UNI I/O Control Register (Address = 01h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LOC Rx	LOC Tx	Interrupt Enable Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxLine Clk Inv	RxLine Clk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	x	1	0	x	x	0

The following table relates the contents of this bit- field to the sampling clock edge of the RxLineClk signal (e.g., the edge of the RxLineClk signal that the E3 data, input at the RxPOS and RxNEG pins, are sampled and latched into the Receive E3 Framer).

Table 27. The Relationship between the contents of Bit 1 (RxLineClk Inv) within the "UNI I/O Control" Register, and the sampling edge of the RxLineClk signal.

RxLineClk Inv (Bit 1)	Sampling Edge of RxLineClk
0	**Rising Edge:** Inputs at the RxPOS and/ or RxNEG pins are sampled and latched into the Receive E3 Framer circuitry on the rising edge of the RxLineClk signal. See Figure 58 for the timing relationship between RxLineClk, RxPOS, and RxNEG, for this selection.
1	**Falling Edge:** Inputs at the RxPOS and/ or RxNEG pins are sampled and latched into the Receive E3 Framer circuitry on the falling edge of the RxLineClk signal. See Figure 59 for the timing relationship between RxLineClk, RxPOS and RxNEG, for this selection.

Figures 58 and 59 present the Waveform and Timing Relationships between RxLineClk, RxPOS, and RxNEG for each of these configurations.

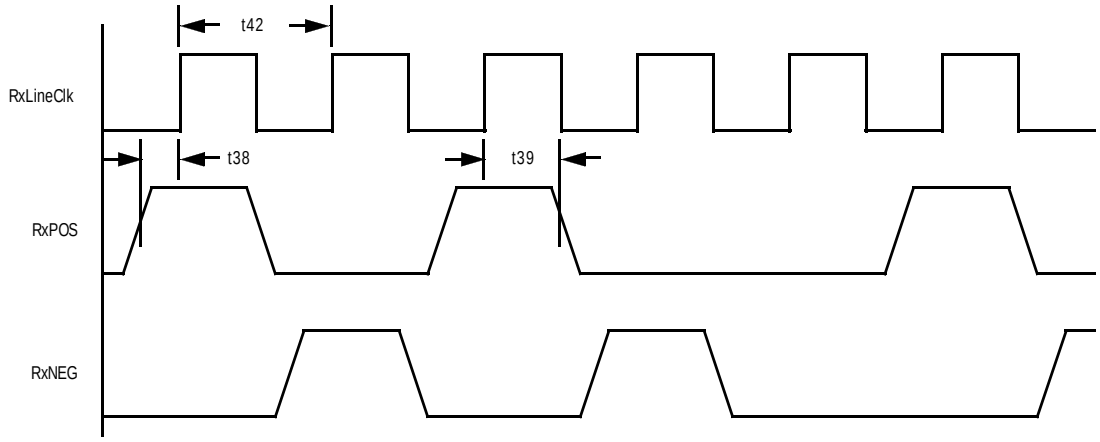

Figure 58. Waveform/ Timing Relationship between RxLineClk, RxPOS and RxNEG-When RxPOS and RxNEG are to be sampled on the rising edge of RxLineClk

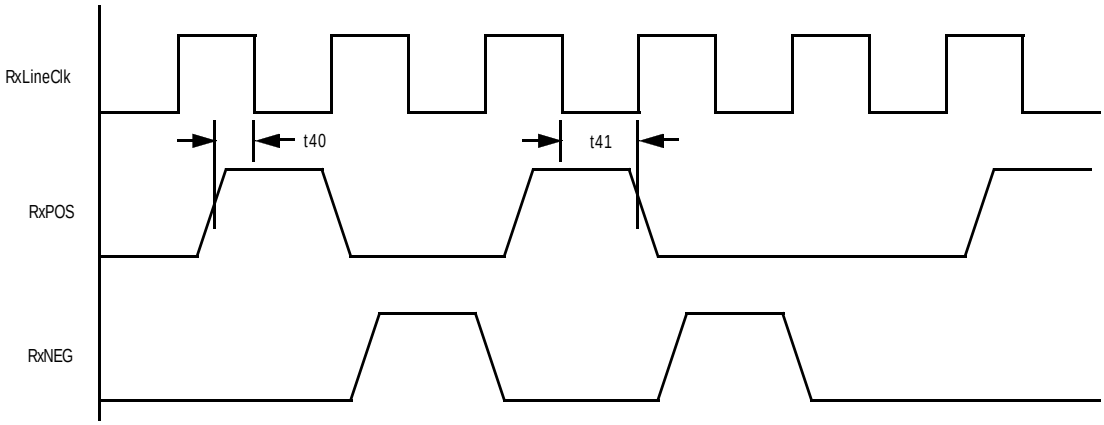

Figure 59. Waveform/ Timing Relationship between RxLineClk, RxPOS and RxNEG - When RxPOS and RxNEG are to be sampled on the rising edge of RxLineClk.

7.1.2.2 E3 Frame Synchronization

Once the incoming HDB3 (or AMI) encoded data has been decoded into a binary data- stream, the "Receive Frame Synchronizer" section of the Receive E3 Framer will use portions of this data- stream in order to synchronize itself to the "Far- End" Transmit E3 Framer. At any given time, the "Receive Frame Synchronizer" section of the Receive E3 Framer will be operating in one of two framing modes.

- The Frame Acquisition Mode: In this mode, the Receive E3 Framer is trying to acquire synchronization with the incoming E3 Frame, or
- The Frame Maintenance Mode: In this mode, the Receive E3 Framer is trying to maintain frame synchronization with the incoming E3 frame.

Each of these two framing modes are discussed in detail below. The Receive E3 Framer will transition between the "Frame Acquisition" Mode and the "Frame Maintenance" Mode, in accordance with the "E3 Frame Acquisition/ Maintenance" algorithm. Throughout the discussion of the E3 Framing Acquisition/ Maintenance Algorithm, the reader will be referred to Figure 60. Figure 60 presents a State Machine diagram that depicts the Receive E3 Framer's "E3 Frame Acquisition/ Maintenance" Algorithm.

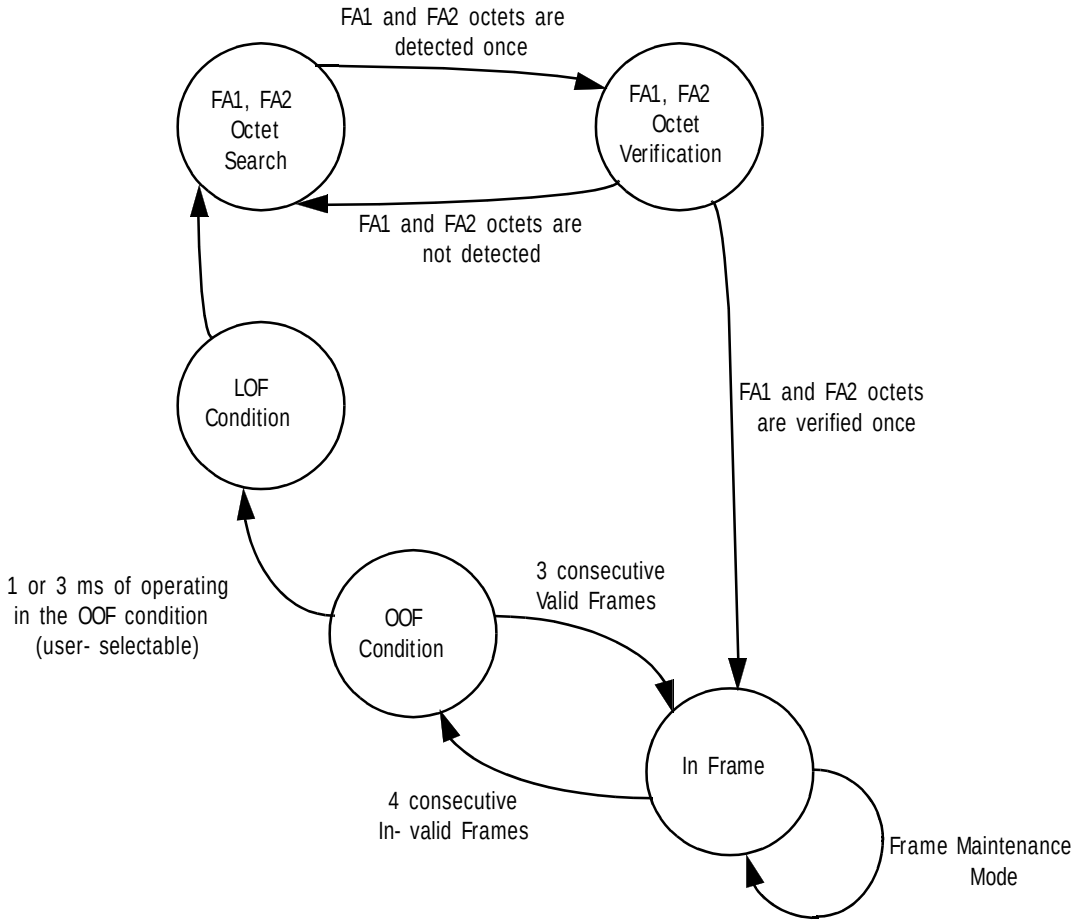

Figure 60. The State Machine Diagram for the Receive E3 Framer's "E3 Frame Acquisition/ Maintenance" Algorithm

7.1.2.2.1 The Framing Acquisition Mode

The Receive E3 Framer is considered to be operating in the "Frame Acquisition Mode, if it is operating in any one of the following states within the "E3 Frame Acquisition/ Maintenance" Algorithm, per Figure 60.

- FA1, FA2 Octet Search State
- FA1, FA2 Octet Verification State
- OOF Condition State
- LOF Condition State

Each of these "Framing Acquisition" States, within the Receive E3 Framer "Framing Acquisition/ Maintenance" State Machine are discussed below.

The 'FA1, FA2 Octet Search' State

When the Receive E3 Framer is first powered up, it will be operating in the "FA1, FA2 Octet Search" state. While the Receive E3 Framer is operating in this state, it will be performing a bit- by- bit search for the FA1 and FA2 Framing Alignment octets. FA1 is assigned the value of F6h; and FA2 is assigned the value of 28h. Figure 61, which presents an illustration of the ITU- T G.832 Compliant E3 Frame Format, indicates that these two octets will occur at the beginning of each E3 frame, and that the FA2 octet will appear immediately after the FA1 octet.

FA1	FA2
EM	530 Octet Payload
TR	
MA	
NR	
GC	
1 byte	59 bytes

Figure 61. Illustration of the ITU- T G.832 Complaint E3 Frame Format

When the Receive E3 Framer detects the FA1 octet, and determines that this octet is immediately followed by FA2 octet, then it will transition to the "FA1, FA2 Octet Verification" State, per Figure 60.

The "FA1, FA2 Octet Verification" State

Once the Receive E3 Framer has detected an "F628h" pattern (e.g., the concatenation of the FA1 and the FA2 octets), it must verify that this pattern is indeed the FA1 and FA2 octets, and not some other set of bytes, within the E3 frame, mimicking the Frame Alignment bytes. Hence, the purpose of the "FA1, FA2 Octet Verification" state.

When the Receive E3 Framer enters this state, it will then quit performing its "bit- by- bit" search for the Frame Alignment bytes. Instead, the Receive E3 Framer will read in the two octets that occur 537 bytes (e.g., one E3 frame period later) after the "candidate" Frame Alignment patterns were first detected. If these two bytes match the assigned values for the FA1 and FA2 octets, then the Receive E3 Framer will conclude that it has found the Frame Alignment bytes and will then transition to the "In- Frame" State. However, if these two bytes do not match the assigned values for the FA1 and FA2 octets, then the Receive E3 Framer will concluded that it has been "fooled" by data mimicking the Frame Alignment bytes, and will transition back to the "FA1, FA2 Octet Search" state.

In- Frame State

Once the Receive E3 Framer enters the "In- Frame" State, then it will cease performing "Frame Acquisition" functions, and will proceed to perform "Framing Maintenance" functions. Therefore, the operation of the Receive E3 Framer, while operating in the "In- Frame" state, can be found in Section 7.1.2.2.2 (The Framing Maintenance Mode).

OOF (Out of Frame) Condition State

If the Receive E3 Framer, while operating in the "In- Frame" state detects 4 consecutive frames, which do not have the valid Frame Alignment (FA1 and FA2 octet) patterns, then it will transition into the "OOF Condition State". The Receive E3 Framer's operation, while in the "OOF Condition" state is a unique mix of Framing Maintenance and Framing Acquisition operation. The Receive E3 Framer will exhibit some Framing Acquisition characteristics by attempting to locate (once again) the Frame Alignment octets. However, the Receive E3 Framer will also exhibit some Frame Maintenance behavior by still using the most recent frame synchronization for its OH byte and payload byte processing.

The Receive E3 Framer will inform the local µP of its transition from the "In- Frame" state to the "OOF Condition" state, by generating a "Change in OOF Condition" interrupt. When this occurs, Bit 3 (OOF Interrupt Status), within the "Rx E3 Interrupt Status Register- 1", will be set to "1", as depicted below.

Address = 12h, Rx E3 Interrupt Status Register- 1

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			OOFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	1	0	0	0

The Receive E3 Framer will also inform the external circuitry of its transition into the "OOF Condition" state, by toggling the RxOOF output pin "high".

If the Receive E3 Framer is capable of finding the Frame Alignment octets within a "user- selectable" number of E3 frame periods, then it will transition back into the "In- Frame" state. The Receive E3 Framer will inform the local µP of its transition back into the "In- Frame" state by generating the "Change in OOF Condition" interrupt.

However, if the Receive E3 Framer resides in the "OOF Condition" state for more than this "user- selectable" number of E3 frame periods, then it will automatically transition to the "LOF Condition" state.

The user can select this "user- selectable" number of E3 frame periods that the Receive E3 Framer will remain in the "OOF Condition" state by writing the appropriate value to Bit 7 (RxLOF Algo) within the "Rx E3 Configuration & Status Register, as depicted below.

Rx E3 Configuration & Status Register (Address = 0Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TiMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
x	0	0	0	0	x	x	x

Writing a "0" into this bit- field causes the Receive E3 Framer to reside in the "OOF Condition" state for at most 24 E3 frame periods (3 ms). Writing a "1" into this bit- field causes the Receive E3 Framer to reside in the "OOF Condition" state for at most 8 E3 frame periods (1 ms).

LOF (Loss of Framing) Condition state

If the Receive E3 Framer enters the LOF Condition state, then the following things will happen.

- The Receive E3 Framer will discard the most recent frame synchronization; and
- The Receive E3 Framer will make an unconditional transition to the "FA1, FA2 Octet Search" state.
- The Receive E3 Framer will notify the local µP of its transition to the "LOF Condition" state, by generating the "Change in LOF Condition" interrupt. When this occurs, Bit 2 (LOF Interrupt Status) within the "Rx E3 Interrupt Status Register- 1" will be set to "1", as depicted below.

Address = 12h, Rx E3 Interrupt Status Register- 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			OOFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	1	0	0

Finally, the Receive E3 Framer will also inform the external circuitry of its transition to the "LOF Condition" state by toggling the "RxLOF" output pin "high".

7.1.2.2.2 The Framing Maintenance Mode

Once the Receive E3 Framer enters the "In- Frame" state, then it will notify the local µP of this fact by generating both the "Change in OOF Condition" and the "Change in LOF Condition" interrupts. When this happens, bit 2 and 3 (LOF Interrupt Status and OOF Interrupt Status) will be set to "1", as depicted below:

Address = 12h, Rx E3 Interrupt Status Register- 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	1	1	0	0

Additionally, the Receive E3 Framer will inform the external circuitry of its transition to the "In- Frame" state by toggling both the RxOOF and the RxLOF output pins low.

Finally, the Receive E3 Framer will negate both the "RxOOF" and the "RxLOF" bit- fields within the "Rx E3 Configuration & Status Register, as depicted below:

Rx E3 Configuration & Status Register (Address = 0Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TiMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
x	0	0	0	0	x	x	x

When the Receive E3 Framer is operating in the "In- Frame" state, it will then begin to perform "Frame Maintenance" operations; where it will continue to verify that the Frame Alignment octets (FA1, FA2) are present, at their proper locations. While the Receive E3 Framer is operating in the "Frame Maintenance" mode, it will declare an "Out- of-Frame" (OOF) Condition if it detects invalid Framing Alignment bytes in four consecutive frames.

Since the Receive E3 Framer requires the detection of invalid Frame Alignment bytes in four consecutive frames, in order for it to transition to the "OOF Condition" state, it can tolerate some errors in the Frame Alignment bytes, and still remain in the "In- Frame" state. However, each time the Receive E3 Framer detects an error in the "Frame Alignment" bytes, it will increment the PMON Framing Error Event Count Registers (Address = 42h and 43h). The bit- format for these two registers are depicted below:

PMON Framing Error Event Count-MSB (Address = 42h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Alignment Error Count-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO

PMON Framing Error Event Count-LSB (Address = 42h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Alignment Error Count-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO

7.1.2.2.3 Forcing a Reframe via Software Command

The UNI allows the user to command a reframe procedure with the Receive E3 Framer via software command. If the user writes a "1" into Bit 0 (Reframe) within the UNI I/O Control Register, as depicted below, then the Receive E3 Framer will be forced into the "FA1, FA2 Octet Search" State, per Figure 60, and will begin its search for the FA1 and FA2 octets.

UNI I/O Control Register (Address = 01h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LOC Rx	LOC Tx	Int En Reset	AMI/ HDB3*	Unipolar/ Bipolar*	TxClk Inv	RxClk Inv	Reframe
RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

The UNI will also respond to this command by doing the following.

1. Asserting both the RxOOF and RxLOF output pins.
2. Generating both the "Change in OOF Status" and the "Change in LOF Status" interrupts to the local µP.
3. Asserting both the RxLOF and the RxOOF bit- fields within the Rx E3 Configuration & Status Register, as depicted below.

Rx E3 Configuration & Status Register (Address = 0Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
x	1	1	0	0	x	x	x

7.1.2.2.4 Performance Monitoring of the Frame Synchronization Section of the Receive E3 Framer

The user can monitor the number of framing bytes (FA1 and FA2 bytes) errors that have been detected by the Receive E3 Framer. This is accomplished by periodically reading the PMON Framing Error Event Count Registers (Address = 42h and 43h). The byte format of these registers are presented below.

PMON Framing Error Event Count Register-MSB (Address = 42h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Frame Alignment Byte Error Count–High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

PMON Framing Error Event Count Register-LSB (Address = 43h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Frame Alignment Byte Error Count –Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

When the local µP/µC reads these register, it will read in the number of errors, that have been detected in the Frame Alignment octets (FA1 and FA2), since the last read of these two registers. These registers are reset upon read.

7.1.2.2.5 The RxOOF and RxLOF output pins

The user can roughly determine the current framing state that the Receive E3 Framer is operating in by reading the logic states of the RxOOF and the RxLOF output pins. Table 28, presents this relationship between the state of the RxOOF and RxLOF output pins, and the Framing State of the Receive E3 Framer.

Table 28, The Relationship between the Logic State of the RxOOF and RxLOF output pins and the Framing State of the Receive E3 Framer

RxLOF	RxOOF	Framing State
0	0	In Frame
0	1	OOF Condition (The Receive E3 Framer is operating in the 3 ms OOF period).
1	0	Invalid
1	1	LOF Condition

7.1.2.3 E3 RECEIVE ALARMS

7.1.2.3.1 The Loss of Signal (LOS) Alarm

Asserting the LOS Indicators

The Receive E3 Framer will declare a "Loss of Signal (LOS) condition, when it detects 32 consecutive incoming "0s" via the RxPOS and RxNEG input pins or if the RLOS input pin (from the XRT7295E E3 Line Receiver IC) is asserted. The Receive E3 Framer will indicate the occurrence of an LOS condition by:

- Asserting the RxLOS output pin (e.g., toggles it "high").
- Setting Bit 4 of the Rx E3 Configuration/ Status Register to "1", as depicted below.

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TMark	RxFERF
R/ W	RO	RO	RO	RO	RO	RO	RO
0	x	x	1	0	x	x	x

- The Receive E3 Framer will generate a "Change in LOS Status" interrupt request. Upon generating this interrupt request, the Receive E3 Framer will assert Bit 1 (LOS Interrupt Status) within the "Rx E3 Framer Interrupt Status Register- 1; as depicted below.

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	X	X	X	1	0

Negating the LOS Indicators

The Receive E3 Framer will negate the "LOS" condition when it encounters a stream of 32 bits that does not contain a string of 4 consecutive zeros.

When the Receive E3 Framer negates the LOS Status, then it will notify the local µP and the external circuitry of this occurrence by:

- Generating the "Change in LOS Status" interrupt to the local µP.

• Negating Bit 4 (RxLOS) within the "Rx E3 Configuration & Status" register, as depicted below:

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLCF Algo	RxLCF	RxCCF	RxLOS	RxAIS	RxPld UnStb	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	x	x	0	0	x	x	x

• Negate the RxLOS output pin (e.g., toggle it low).

7.1.2.3.2 Alarm Indication Signal (AIS) Condition

Asserting the AIS Condition

The Receive E3 Framer will identify and declare an "AIS" condition, if it detects an "all ones" pattern in the incoming E3 Data Stream. More specifically, the Receive E3 Framer will declare an AIS condition, if 7 or less "0s" are detected in each of 2 consecutive frames.

If the Receive E3 Framer declares an "AIS Condition", then it will

• Generating the "Change in AIS Condition" interrupt to the local µP. Hence, the Receive E3 Framer will assert Bit 1 (AIS Interrupt Status) within the "Rx E3 Framer Interrupt Status Register- 1", as depicted below:

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			CCFA Interrupt Status	CCF Interrupt Status	LCF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	X	X	X	0	1

• Assert the RxAIS output pin.

• Set Bit 3 (RxAIS)of the Rx E3 Configuration/ Status Register, as depicted below:

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLCF Algo	RxLCF	RxCCF	RxLOS	RxAIS	RxPld UnStb	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	x	x	0	1	x	x	x

Negating the AIS Condition

The Receive E3 Framer will negate the "AIS" Condition when it detects two consecutive E3 frames, with eight or more "zeros" in the incoming data stream. The Receive E3 Framer will inform the local µC/µP of this negation of the "AIS" Condition, by:

- Generating a "Change in AIS Condition" interrupt to the local µP. Hence, the Receive E3 Framer will assert Bit 1 (AIS Interrupt Status) within the "Rx E3 Framer Interrupt Status Register- 1".
- Negating the RxAIS output pin (e.g., toggling it "low")
- Setting the RxAIS bit- field, within the Rx E3 Configuration/ Status Register, to "0", as depicted below:

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCF Algo	RxLCF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	x	x	0	0	x	x	x

7.1.2.3.3 The Far- End Receive Failure (FERF) Condition

Asserting the FERF Condition

The Receive E3 Framer will declare a "Far- End- Receive- Failure" (FERF) condition if it detects a "user- selectable" number of consecutive incoming E3 Frames, with the "FERF" bit- field (Bit 7 within the MA byte) set to "1". The bit format of the MA Byte is presented below:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker
1	x	0	1	0	x	x	x

This "User- Selectable" number of E3 Frames is either 3 or 5 depending upon the value that has been written into Bit 4 (Rx FERF Algo) of the Rx E3 Configuration/ Status Register, as depicted below:

Rx E3 Configuration/ Status Register (Address = 0Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxPldType[2:0]			RxFERF Algo	Rx TMark Algo	RxPldExp[2:0]		
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	1	0

Writing a "0" to this bit- field causes the Receive E3 Framer to declare an "FERF" condition, if it detects 3 consecutive incoming E3 Frames, that have the "FERF" bit (within the MA byte) set to "1". Writing a "1" to this bit- field causes the Receive E3 Framer to declare an "FERF" condition, if it detects 5 consecutive incoming E3 Frames, that have the "FERF" bit (within the MA byte) set to "1".

Whenever the Receive E3 Framer declares a "FERF" Condition, then it will do the following

- Generate a "Change in FERF Condition" interrupt to the local µP. Hence, the Receive E3 Framer will assert Bit 3 (FERF Interrupt Status) within the "Rx E3 Framer Interrupt Status Register- 2"; as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

- Set the "RxFERF" bit- field, within the Rx E3 Configuration/ Status Register, to "1", as depicted below.

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TiMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	0	0	0	0	0	X	1

Negating the FERF Condition

The Receive E3 Framer will negate the FERF condition once it has received a "User- Selectable" number of consecutive incoming E3 frames with the "FERF" bit- field set to "0". This "User- Selectable" number of E3 Frames is either 3 or 5 depending upon the value that has been written into Bit 4 (Rx FERF Algo) of the "Rx E3 Configuration/ Status Register, as discussed above.

Whenever the Receive E3 Framer negates the FERF Status, then it will do the following.

1. Generate a "Change in the FERF Status" Interrupt to the local µP.
2. Negate Bit 0 (Rx FERF) within the Rx E3 Configuration & Status Register, as depicted below.

Rx E3 Configuration/ Status Register (Address = 0Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld UnStb	Rx TiMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	0	0	0	0	0	X	0

7.1.2.4 Error- Checking of the Incoming E3 Frames

The Receive E3 Framer performs error- checking on the incoming E3 Frame data that it receives from the "Far- End" Terminal. It performs this error checking by computing the BIP- 8 value of an incoming E3 Frame. Once the Receive E3 Framer has obtained this value, it will compare this value with that of the EM byte that it receives, within the very next E3 Frame. If the "locally computed" BIP- 8 value matches the EM byte of the corresponding E3 frame, then the Receive E3 Framer will conclude that this particular frame has been properly received. The Receive E3 Framer will then inform the "Far- End" Terminal of this fact by having the "Near- End" Transmit E3 Framer send the "Far- End" Terminal an E3 Frame, with the "FEBE" bit- field, within the MA byte, set to "0".

This procedure is illustrated in Figure 62 and 63, below.

Figure 62 illustrates the "Near- End" Receive E3 Framer receiving an "error- free" E3 frame. In this figure, the locally computed BIP- 8 value of "5Ah" matches that received from the "Far- End" Terminal, within the EM byte- field. Figure 63 illustrates the subsequent action of the "Near- End" Transmit E3 Framer, which will transmit an E3 frame, with the FEBE bit- field set to "0"; to the "Far- End" Terminal. This signaling indicates that the "Near- End" Receive E3 Framer has received an "error- free" E3 frame.

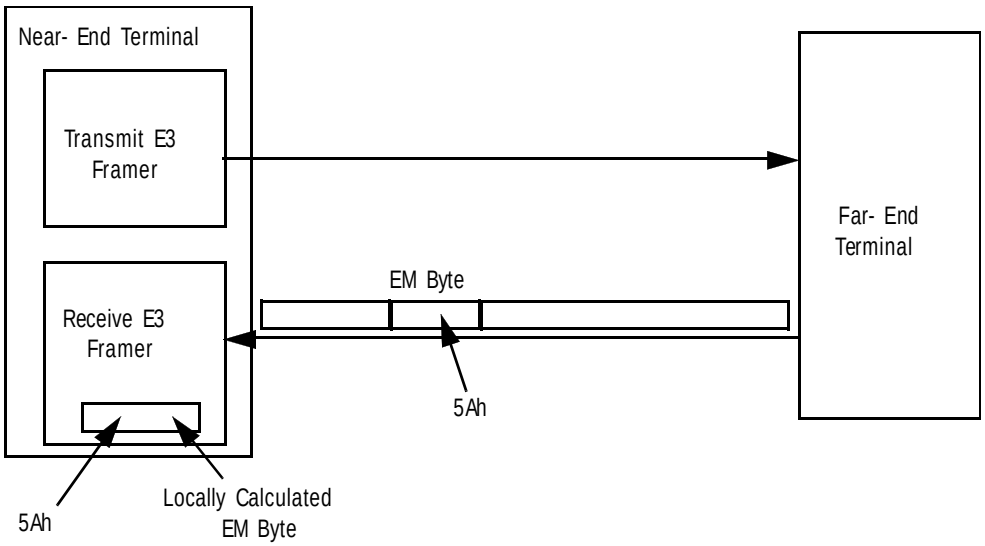

Figure 62. Illustration of the "Near- End" Receive E3 Framer, receiving an E3 Frame (from the "Far- End" Terminal) with a correct EM- byte.

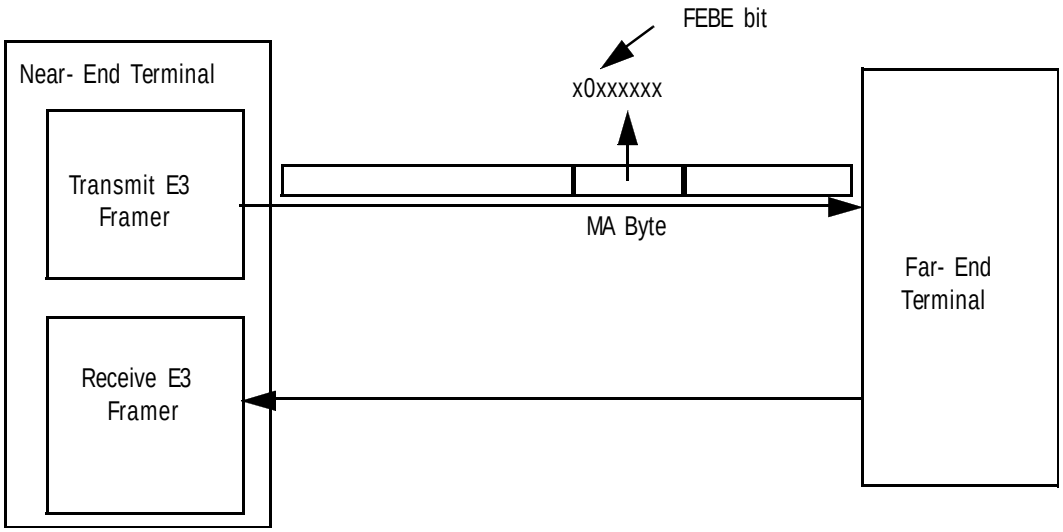

Figure 63. Illustration of the "Near- End" Transmit E3 Framer, transmitting an E3 Frame (to the "Far- End" Terminal) with the FEBE bit (within the MA byte- field) set to '0".

However, if the locally computed BIP- 8 value does not match the EM byte of the corresponding E3 frame, then the Receive E3 Framer will do the following.

- It will inform the "Far- End" Terminal of this fact by having the "Near- End" Transmit E3 Framer send the "Far- End" Terminal an E3 Frame, with the "FEBE" bit- field, within the MA byte, set to "1". This phenomenon is illustrated below in Figures 64 and 65.

Figure 64 illustrates the "Near- End" Receive E3 Framer receiving an "errored" E3 frame. In this figure, the "Near- End" Receive E3 Framer is receiving an E3 with an EM byte containing the value "5Ah". This value does not match the "locally computed" EM byte value of "5Bh". Consequently, there is an error in this E3 frame.

Figure 65 illustrates the subsequent action of the "Near- End" Transmit E3 Framer, which will transmit an E3 frame, with the FEBE bit- field set to "1" to the "Far- End" Terminal. This signaling indicates that the "Near- End" Receive E3 Framer has received an "errored" E3 frame.

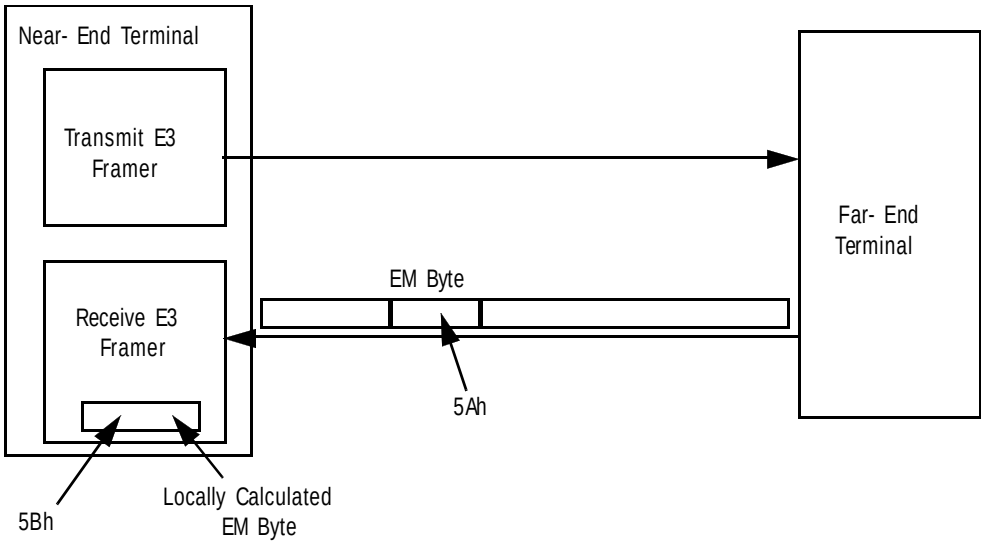

Figure 64. Illustration of the "Near- End" Receive E3 Framer receiving an E3 Frame (from the "Far- End" Terminal) with

an incorrect EM- byte.

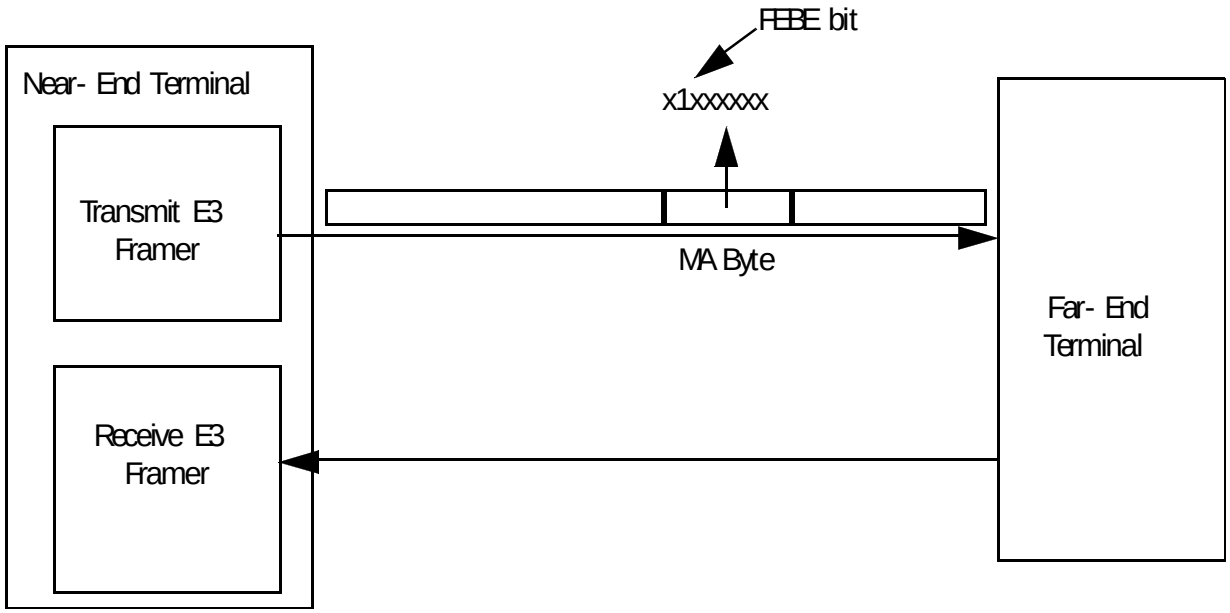

Figure 65. Illustration of the 'Near- End' Transmit E3 Framer, transmitting an E3 Frame (to the 'Far- End' Terminal) with the FEBE bit (within the MA byte- field) set to "1".

- In addition to the "FEBE" bit- field signaling; the Receive E3 Framer will generate the "BIP- 8 Error" Interrupt to the local µP. Hence, it will set bit 2 (BIP- 8 Error Interrupt Status) to "1" as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Mis Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

- Finally, the Receive E3 Framer will increment the "PMON Framing BIP- 8 Error Count" Register. The byte format of these registers are presented below.

PMON BIP- 8 Error Count Register-MSB (Address = 46h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
BIP- 8 Error Count-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO

PMON BIP- 8 Error Count Register-LSB (Address = 46h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
BIP- 8 Error Count-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO

The user can determine the number of BIP- 8 Errors that have been detected by the Receive E3 Framer, since the last read of these registers. These registers are reset upon read.

7.1.2.5 LAPD Receiver

The LAPD Transceiver uses either the GC byte- field or the NR byte- field, within each E3 Frame to transmit and receive performance monitor data link (PMDL) messages. The "Far- End" LAPD Transmitter will transmit a PMDL message, encapsulated in a LAPD Message frame to the "Near- End" LAPD Receiver, via one of these two byte- fields, within each incoming E3 Frame. The LAPD Receiver will receive and extract the PMDL message from the incoming LAPD Message frame. Afterwards, the LAPD Receiver will write the PMDL Message into the "Receive LAPD Message" buffer, which is located at addresses: DEh through 135h, within the on- chip RAM. The LAPD Receiver (within the "Near- End" Receive E3 Framer) has the following responsibilities.

- Receiving the incoming LAPD Message frame octets and reassembling these octets into LAPD Message frames.
- Filtering out stuffed "0s" (within the PMDL Message portion of the LAPD Message Frame).
- Extracting the PMDL Message from the incoming LAPD Message frame, and writing the PMDL Message into the "Receive LAPD Message" buffer.
- Performing Frame Check Sequence (FCS) Verification of the incoming LAPD Message frame.
- Provide status indicators for
  - End of Message (EOM)
  - Flag Sequence Byte detected
  - Abort Sequence detected
  - LAPD Message Type Received
  - C/ R Type
  - The detection of FCS Errors

The user can control and monitor the actions of the LAPD Receiver via the following two registers.

- Rx E3 LAPD Control Register
- Rx E3 LAPD Status Register

Operation of the LAPD Receiver

The LAPD Receiver, once enabled, will begin searching for the boundaries of the incoming LAPD Message. The LAPD Message Frame boundaries are delineated via the "Flag Sequence" octets (7Eh), as depicted below in Figure 66.

Flag Sequence (8 bits)		
SAPI (6- bits)	C/R	EA
TEI (7 bits)		EA
Control (8 bits)		
76 or 82 Bytes of Information (PMDL Message)		
FCS-MSB		
FCS-LSB		
Flag Sequence (8 bits)		

Where: Flag Sequence = 7Eh
SAPI + C/R + EA = 3Ch or 3Eh
TEI + EA = 01h
Control = 03h

The 16 bit FCS is calculated using CRC- 16, $x^{16} + x^{12} + x^5 + 1$

Figure 66. LAPD Message Frame Format

Enabling and Configuring the LAPD Receiver

Before the LAPD Receiver can begin to receive and process incoming LAPD Message frames, the user must do two things.

1. He/ she must specify which byte- field, within each E3 Frame, will be carrying the comprising octets of the LAPD Message frame, and
2. He/ she must enable the LAPD Receiver.

Each of these steps are discussed in detail, below.

1. Specifying which byte- field, within each E3 Frame, will be carrying the LAPD Message Frame.

The LAPD Receiver can receive the LAPD Message frame octets via either the GC byte- field or the NR byte- field, within each incoming E3 frame. The user makes this selection by writing the appropriate bit to Bit 1 (DL From NR) within the Rx E3 LAPD Control Register, as depicted below.

Rx E3 LAPD Control Register (Address = 26h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused						DL From NR	RxLAPD Enable
RO	RO	RO	RO	RO	RO	R/ W	R/ W
0	0	0	0	0	0	X	X

Writing a "0" to this bit- field causes the LAPD Receiver to read in the octets from the GC byte- field of each E3 frame, and with these octets; reassembling the LAPD Message frame. Writing a "1" to this bit- field causes the LAPD Receive to receive the LAPD Message frame octets from the NR byte- field of each E3 frame.

2. Enabling the LAPD Receiver

The LAPD Receiver must be enabled before it can begin receiving and processing any LAPD Message frames. The LAPD Receiver can be enabled by writing a "1" to bit 1 (Rx LAPD Enable) of the Rx E3 LAPD Control Register, as indicated below.

Rx E3 LAPD Control Register (Address = 26h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused						DL From NR	RxLAPD Enable
RO	RO	RO	RO	RO	RO	R/ W	R/ W
0	0	0	0	0	0	X	1

Once the LAPD Receiver has been enabled, it will begin searching for the Flag Sequence octets (7Eh), in either the GC or the NR byte- fields within each incoming E3 frame. When the LAPD Receiver finds the flag sequence byte, it

will assert the "Flag Present" bit (Bit 0) within the Rx E3 LAPD Status Register, as depicted below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	x	x	x	0	0	1

The receipt of the Flag Sequence octet can mean one of two things.

1. This Flag Sequence byte may be marking the beginning of an incoming LAPD Message frame.
2. The received Flag Sequence octet could be just one of many Flag Sequence octets that are transmitted via the E3 Transport Medium, during idle periods between the transmission of LAPD Message frames.

The LAPD Receiver will negate the "Flag Present" bit as soon as it has received an octet that is something other than the "Flag Sequence" octet. Once this happens, the LAPD Receiver should be receiving either octet #2 of the incoming LAPD Message, or an Abort Sequence (e.g., a string of seven or more consecutive "1s"). If this next set of data is an abort sequence, then the LAPD Receiver will assert the RxAbort bit (Bit 6) within the "Rx E3 LAPD Status" Register, as depicted below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	1	x	x	x	0	0	0

However, if this next octet is Octet #2 of an incoming LAPD Message frame, then the LAPD Receiver is beginning to receive a LAPD Message frame.

As the LAPD Receiver receives this LAPD Message frame, it is reading in the LAPD Message frame octets, from either the GC or the NR byte- fields within each incoming E3 frame. Secondly, it is reassembling these octets into a LAPD Message frame.

Once the LAPD Receiver has received the complete LAPD Message frame, then it will proceed to perform the following five (5) steps.

1. PMDL Message Extraction

The LAPD Receiver will extract out the PMDL Message, from the "newly received" LAPD Message frame. The LAPD Receiver will then write this PMDL Message into the "Receive LAPD Message" buffer within the UNI.

Note: As the LAPD Receiver is extracting the PMDL Message, from the "newly received" LAPD Message frame, the LAPD Receiver will also check the PMDL data for the occurrence of stuff bits (e.g., "0s" that were inserted into

the PMDL Message by the "Far- End" LAPD Transmitter, in order to prevent this data from mimicking the Flag Sequence byte or an Abort Sequence), and remove them prior to writing the PMDL Message into the "Receive LAPD Message" buffer. Specifically, the LAPD Receiver will search through the PMDL message data and will remove any "0" that immediately follows a string of 5 consecutive "1s".

For information on how the LAPD Transmitter inserted these stuff bits, please see Section 6.3.3.5.

1. FCS (Frame Check Sequence) Word Verification

The LAPD Receiver will compute the CRC- 16 value of the header octets and the PMDL Message octets, within this LAPD Message frame and will compare it with the value of the two octets, residing in the FCS word- field of this LAPD Message frame. If the FCS value of the "newly received" LAPD Message frame matches the "located computed" CRC- 16 value, then the LAPD Receiver will conclude that it has received this LAPD Message frame in an "error- free" fashion.

However, if the FCS value does not match the "locally- computed" CRC- 16 value, then the LAPD Receiver will conclude that this LAPD Message frame is errored.

The LAPD Receiver will indicate the results of this "FCS Verification process" by setting Bit 2 (Rx FCS Error), within the "Rx E3 LAPD Status" Register, to the appropriate value as tabulated below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	Rx FCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO

Table 28. The Relationship between the contents of Bit 2 (Rx FCS Error) within the "Rx E3 LAPD Status Register" and the results of FCS Verification.

Rx FCS Error (Bit 2)	Results of FCS Verification
0	LAPD Message frame was received in an "error- free" fashion
1	LAPD Message frame is errored

Note: The LAPD Receiver will extract and write the PMDL Message into the "Receive LAPD Message" buffer independent of the results of FCS Verification. Hence, the user is urged to validate each PMDL Message that he/ she reads in from the "Receive LAPD Message" buffer, by first checking the state of this bit- field.

2. Check and Report the State of the C/ R Bit- Field

After the receiving the LAPD Message frame, the LAPD Receiver will check the state of the "C/ R" bit- field, within

octet #2 of the LAPD Message frame header; and will reflect this value in Bit 3 (Rx CR Type) of the "Rx E3 LAPD Status Register, as depicted below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	Rx FCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO

When this bit- field is "0", it means that this LAPD Message frame is originating from a customer installation. When this bit- field is "1", it means that this LAPD Message frame is originating from a network terminal.

3. Identify the Type of LAPD Message Frame/ PMDL Message

Next the LAPD Receiver will check the value of the first octet, within the PMDL Message field, of the LAPD Message frame. Recall from Section 6.3.3.5.2, that when operating the LAPD Transmitter, the user is required to write in a byte of a specific value into the first octet position within the Transmit LAPD Message" buffer. The value of this byte corresponds to the type of LAPD Message frame/ PMDL Message that is to be transmitted to the "Far- End" LAPD Receiver. This "Message- type identification" octet is transported to the "Far- End" LAPD Receiver, along with the rest of the LAPD Message frame. From this "Message- type identification" octet, the LAPD Receiver will know the type and size of the "newly received" PMDL Message. The LAPD Receiver will then reflect this information in Bits 4 and 5 (RxLAPDType[1:0]) within the "Rx E3 LAPD Status" Register; as depicted below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	Rx FCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO

Table 30 presents the relationship between the contents of "RxLAPDType[1:0]" and the type of message received by the LAPD Receiver.

Table 29. The Relationship between the contents of RxLAPDType[1:0] bit- fields and the PMDL Message Type/ Size

RxLAPDType[1:0]	PMDL Message Type	PMDL Message Size
00	Test Signal Identification	76 Bytes
01	Idle Signal Identification	76 Bytes
10	CL Path Identification	76 Bytes
11	ITU- T Path Identification	82 Bytes

Note: Prior to reading in the PMDL Message from the "Receive LAPD Message" buffer the user is urged to read the state of the "RxLAPDType[1:0] bit- fields in order to determine the size of this message.

4. Inform the local µP/ External Circuitry of the receipt of the new LAPD Message frame

Finally, after the LAPD Receiver has received and processed the "newly received" LAPD Message frame (per steps

1 through 4, as described above), it will inform the local µP that a LAPD Message frame has been received and is ready for "user- system" handling. The LAPD Receiver will inform the local µP and external circuitry of this by:

- Generating a "LAPD Message Frame Received" interrupt to the local µP. The purpose of this interrupt is to let the local µP know that the "Receive LAPD Message" buffer contains a new PMDL message that needs to be read and processed. When the LAPD Receiver generates this interrupt, it will set bit 5 (LAPD Interrupt Status) within the "Rx E3 Framer Interrupt Status Register- 2" to "1", as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Mis Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

- Setting Bit 1 (End of Message) within the "Rx E3 LAPD Status" Register, to "1" as depicted below.

Rx E3 LAPD Status Register (Address = 27h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	RxAbort	RxLAPDType[1:0]		RxCR Type	Rx FCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO

In Summary, Figure 67 presents a flow chart depicting how the LAPD Receiver works.

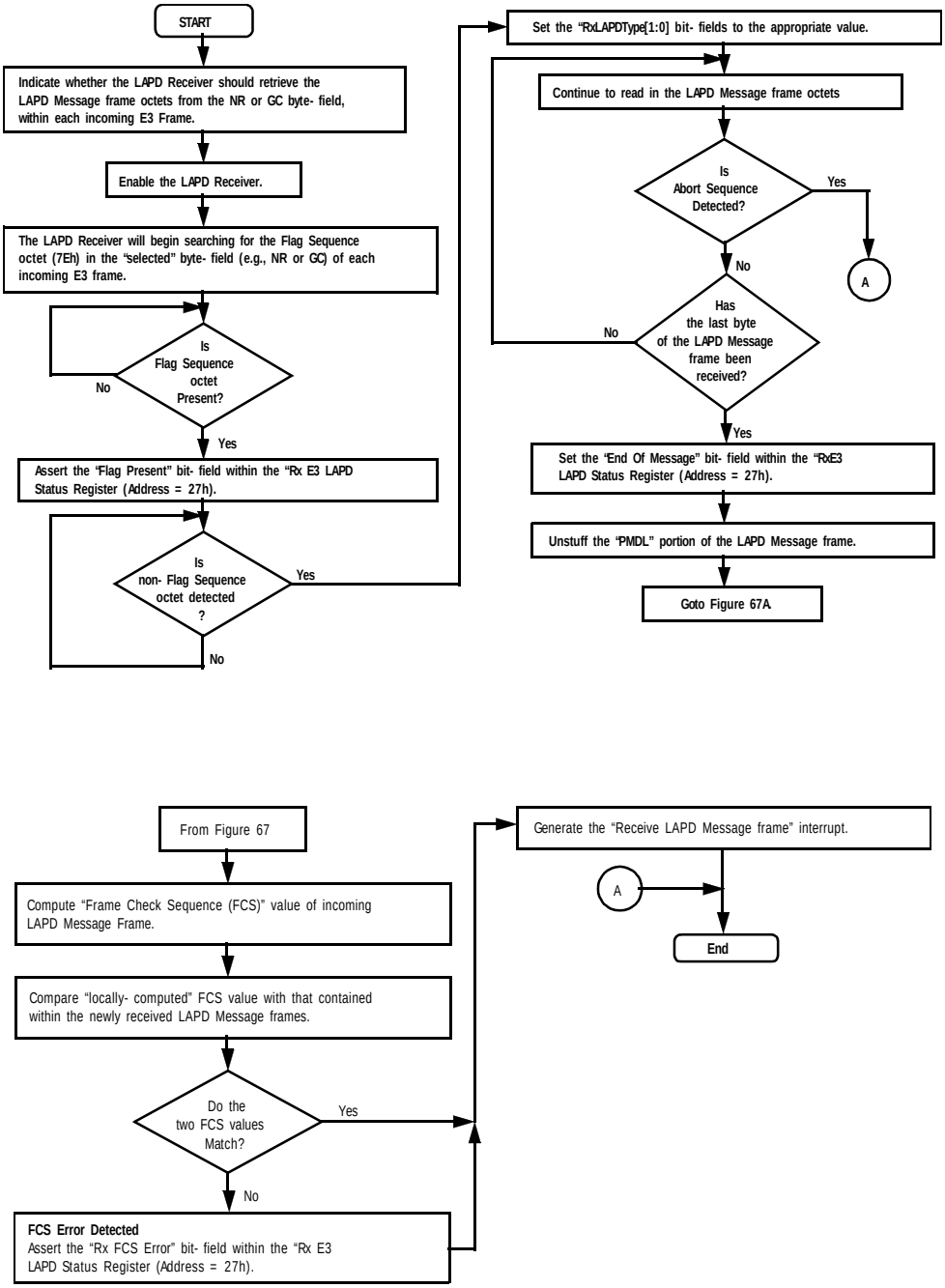

Figure 67. Flow Chart depicting the Functionality of the LAPD Receiver

7.1.2.6 Processing of the Far- End- Block Error (FEBE) Bit- Fields

Whenever the Receive E3 Framer detects an error in an incoming E3 frame, via EM byte verification, it will inform the "Near- End" Transmit E3 Framer of this fact. The "Near- End" Transmit E3 Framer will, in turn, notify the "Far- End" Terminal (e.g., the source of the errored E3 frame) by transmitting an E3 frame, with the FEBE bit- field (within the MA byte) set to "1".

If the Receive E3 Framer receives any E3 frame, with the FEBE bit-field set to "1", then it will do the following.

- It will generate a "FEBE Event" interrupt to the local µP. Hence the Receive E3 Framer will set bit 4 (FEBE Interrupt Status) within the "Rx E3 Framer Interrupt Status Register-2", as depicted below.

Rx E3 Framer Interrupt Status Register-2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

- Increment the "PMON Received FEBE Event Count Register-MSB/LSB, which is located at 44h and 45h in the UNI Address space. The byte-format of these registers are presented below.

PMON Received FEBE Event Count Register-MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Received FEBE Event Count-High Byte							
RO	RO	RO	RO	RO	RO	RO	RO

PMON Received FEBE Event Count Register-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Received FEBE Event Count-Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO

The user can determine the total number of FEBE Event (e.g., E3 frames that have been received with the FEBE bit-field set to "1") that have occurred since the last read of this register. This register is reset upon read.

7.1.2.7 Trail Trace Buffers

The XR-T7234 E3 UNI device contains 16 bytes worth of "Transmit" Trail Trace Buffers, and 16 bytes worth of "Receive" Trail Trace Buffer. The role of the "Receive" Trail Trace Buffers is described below. The role of the "Transmit" Trail Buffers are described in Section 6.3.3.4.

The XR-T7234 E3 UNI device contains 16 "Receive Trail Trace Buffer" Registers (e.g., Rx TTB-0 through Rx TTB-15). The purpose of these registers are to receive and store the incoming "Trail Access Point Identifier" from the "Far-End" Transmitting Terminal. The "Near-End" Receiving Terminal will use this information to verify that it is still receiving data from its intended transmitter. The specific use of these registers follows.

For "Trail Trace Buffer" purposes, the "Far-End" Transmit E3 Framer will group 16 consecutive E3 Frames into a "Trail Trace Buffer" super-frame. When the "Far-End" Transmit E3 Framer is generating the first E3 frame, within a

"Trail Trace Buffer" super- frame, it will insert the value [1, C6, C5, C4, C3, C2, C1, C0], into the TR byte- field of this "outbound" E3 Frame. The remaining 15 TR byte- fields (within this "Trail Trace Buffer" super- frame) will consists of ASCII characters that are required for the E.164 numbering format.

When the "Near- End" Receiver E3 Framer receives an E3 Frame, containing a value in the TR byte- field that has a "1" in the MSB position, then it (the Receive E3 Framer) will write this value into the Rx TTB- 0 Register (Address = 16h). Once this occurs, the Receive E3 Framer will notify the local µP of this "new" incoming Trail Trace Buffer message by generating the "Change in Trail Trace Buffer Message" interrupt. The Receive E3 Framer will also set bit 6 (TTB Change Interrupt Status) within the "Rx E3 Framer Interrupt Status-Register- 2"; as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Error Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

The contents of the TR byte- field, in the very next E3 frame will be written into the Rx TTB- 1 Register (Address = 17h), and so on until all 16 bytes have been received.

Note:

1. Anytime the Receive E3 Framer receives an E3 frame that contains an octet in the TR byte- field, with a "1" in the MSB (most significant bit) position; then the Receive E3 Framer will do the following.

- It will write the contents of the TR byte- field (in this E3 frame) into the Rx TTB- 0 Register.
- It will generate the "Change in Trail Trace Buffer Message" interrupt

The Receive E3 Framer will do these things independent of the number of E3 frames that have been received since the occurrence of the last "occurrence" of the "Change in Trail Trace Buffer Message" interrupt. Hence, the user, when writing data into the Tx TTB registers, must take care to insure that only Tx TTB- 0 contains an octet with a "1" in the MSB position. All remaining Tx TTB registers (e.g., Tx TTB- 1 through Tx TTB- 15) must contain octets with a "0" in the MSB position.

2. The UNI will not verify the CRC- 7 value that is written into the Rx TTB- 0 Register. It is up to the user's system hardware and/ or software to perform this verification.

7.1.2.8 Extracting the E3 Overhead Bytes via the Serial Output Port

The Receive E3 Framer block also consists of the "RxOH" Serial Output Port. This serial output port consists of the following output pins:

- RxOH
- RxOHClk

- RxOHFrame

The Receive E3 Framer will serially output the OH bytes (of the incoming E3 frame) via the "RxOH" output pin. This output signal will be updated on the rising edge of the RxOHClk clock signal. Finally, the RxOHFrame output pin will pulse "high" when the first bit of the FA1 octet, within an E3 frame is being output at the RxOH output pin. The order in which these OH bytes are output (via the RxOH pin) is in accordance with that in Figure 61.

Figure 68 presents a timing diagram that illustrates the behavior of the RxOH Serial Output port signals.

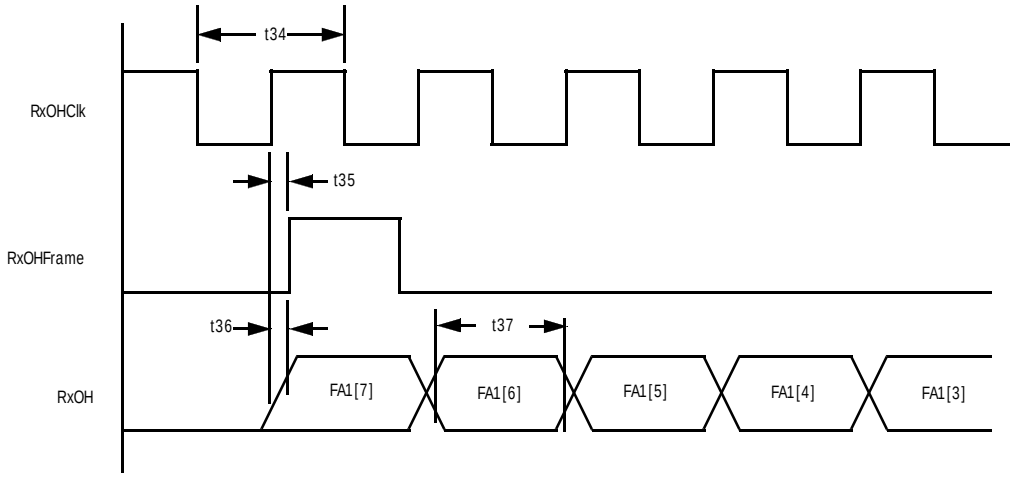

Figure 68. Illustration of the RxOH Serial Output Port Signals

7.1.2.9 Receive E3 Framer Interrupt Conditions

The Receive E3 Framer can generate an interrupt request upon any of the following conditions.

- Change in OOF (Out of Frame) Condition
- Change in LOF (Loss of Frame) Condition
- Change in LOS (Loss of Signal) Condition
- Change in AIS Status
- Detection of BIP-8 Error
- Change in FERF (Far-End-Receive Failure) Condition
- Change in Frame Alignment
- Receipt of new Trail Trace Buffer Message
- Payload Type Mismatch Change
- Receipt of a new LAPD Message frame

If one of these conditions occur, and if that particular condition has been enabled for interrupt generation, the when the local µP reads the UNI Interrupt Status Register, as shown below; it should read "0xx1xxxxb" (where the -b suf-

fix denotes a binary expression, and the "x" denotes a "don't care" value.)

UNI Interrupt Status Register (Address = 05h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	One Sec Interrupt Status	Tx E3 Framer Interrupt Status	Rx E3 Framer Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	1	x	x	x	x

At this point, the local µP will have determined that the Receive E3 Framer block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly. In order to accomplish this, the local µP should now proceed to read one or both of the following registers.

- Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)
- Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)

The role of the bits, within each of these register, to interrupt processing, are described below.

7.1.2.9.1 "Change in OOF Condition" Interrupt

The Receive E3 Framer will generate the "Change in OOF Condition" interrupt in response to either of the following two occurrences.

1. Whenever the Receive E3 Framer transitions from the "In- Frame" state to the "OOF Condition" state, within the "E3 Framing Acquisition/ Maintenance" algorithm (per Figure 60).
2. Whenever the Receive E3 Framer transitions from the "OOF Condition" state, back into the "In Frame" state, within the "E3 Framing Acquisition/ Maintenance" algorithm.

Recall from the discussion in Section 7.1.2.2, that the Receive E3 Framer will transition from the "In- Frame" state into the "OOF Condition" state, if the Receive E3 Framer detects errors in the framing alignment bits (FA1 and FA2) for four (4) consecutive E3 frames. Additionally, the Receive E3 Framer will transition from the "OOF Condition" state to the "In- Frame" state, if the Receive E3 Framer is able to properly locate the Frame Alignment bytes (FA1 and FA2) for two consecutive E3 frames.

When the Receive E3 Framer generates the "Change in OOF Condition" interrupt, then it will assert Bit 3 (OOF Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 1" (Address = 12h); as depicted below.

Rx E3 Framer Interrupt Status Register - 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	x	1	x	x	x

This "Reset upon Read" bit- field will be set to "1" if a "Change in OOF Condition" has occurred since the last read of

this register.

The user can enable or disable the "Change in OOF Condition" interrupt by writing the appropriate value to bit 3 (OOF Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 1", as depicted below.

Rx E3 Framer Interrupt Enable Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Enable	OOF Interrupt Enable	LOF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/Write" bit- field disables the "Change in OOF Condition" interrupt. Writing a "1" to this bit-field enables the "Change in OOF Condition" interrupt.

7.1.2.9.2 "Change in LOF Condition" Interrupt

The Receive E3 Framer will generate the "Change in LOF Condition" interrupt in response to either of the following two occurrences.

1. Whenever the Receive E3 Framer transitions from the "OOF Condition" state into the "LOF Condition" state, within the "E3 Framing Acquisition/ Maintenance" algorithm (per Figure 60).
2. Whenever the Receive E3 Framer transitions from the "FA1, FA2 Octet Verification" state to the "In- Frame" state, within the "E3 Framing Acquisition/ Maintenance" algorithm (per Figure 60).

Recall from the discussion in Section 7.1.2.2.1, that the Receive E3 Framer will transition from the "OOF Condition" state into the "LOF Condition" state, if the Receive E3 Framer remains in the "OOF Condition" state for a "user-selectable" amount of time (either 1 ms or 3 ms) and is unable to regain the "In- Frame" state. Additionally, the Receive E3 Framer will transition from the "FA1, FA2 Octet Verification" state to the "In- Frame" state, if the Receive E3 Framer is able to properly locate the Frame Alignment bytes (FA1 and FA2) for two consecutive E3 frames.

When the Receive E3 Framer generates the "Change in LOF Condition" interrupt, then it will assert Bit 2 (LOF Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 1" (Address = 12h); as depicted below.

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	x	x	1	x	x

This "Reset upon Read" bit- field will be set to "1" if a "Change in LOF Condition" interrupt has occurred since the last read of this register.

The user can enable or disable the "Change in LOF Condition" interrupt by writing the appropriate value to bit 2

(LCF Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 1" as depicted below:

Rx E3 Framer Interrupt Enable Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Enable	COF Interrupt Enable	LCF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Change in LCF Condition" interrupt. Writing a "1" to this bit- field enables the "Change in LCF Condition" interrupt.

7.1.2.9.3 "Change in LOS Condition" Interrupt

The Receive E3 Framer will generate the "Change in LOS Condition" interrupt in response to either of the following two occurrences.

1. Whenever the Receive E3 Framer detects a string of 32 consecutive "0s" in the incoming E3 frame data, via the RxPOS and RxNEG input pins.
2. Whenever the Receive E3 Framer negates the LOS Condition.

Recall, from Section 7.1.2.3.1, that the Receive E3 Framer will negate the LOS Condition upon the detection of a string a 32 bits that does not contain a string of 4 consecutive 0's.

When the Receive E3 Framer generates the "Change in LOS Condition" interrupt, then it will assert Bit 1 (LOS Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 1" (Address = 12h); as depicted below:

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			COFA Interrupt Status	COF Interrupt Status	LCF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	x	x	x	1	x

This "Reset upon Read" bit- field will be set to "1" if a "Change in LOS Condition" interrupt has occurred since the last read of this register.

The user can disable or enable the "Change in LOS Condition" interrupt by writing the appropriate value to bit 1

(LOS Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 1" as depicted below.

Rx E3 Framer Interrupt Enable Register- 1 (Address = 10h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			CCFA Interrupt Enable	COF Interrupt Enable	LOF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/Write" bit- field disables the "Change in LOS Condition" interrupt. Writing a "1" to this bit-field enables the "Change in LOS Condition" interrupt.

7.1.2.9.4 "Change in AIS Status" Interrupt

The Receive E3 Framer will generate the "Change in AIS Condition" interrupt in response to either of the following two occurrences.

1. Whenever the Receive E3 Framer detects 7 or fewer zeros in the incoming E3 frame datastream, for two consecutive E3 frame periods.
2. Whenever the Receive E3 Framer negates the "AIS Condition".

Recall, from Section 7.1.2.3.2, that the Receive E3 Framer will negate the AIS Condition upon the detection two consecutive E3 frames, that each contain 8 or more zeros.

When the Receive E3 Framer generates the "Change in AIS Condition" interrupt, then it will assert Bit 0 (AIS Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 1 (Address = 12h); as depicted below.

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			CCFA Interrupt Status	COF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	x	x	x	x	1

This "Reset upon Read" bit- field will be set to "1" if a "Change in AIS Condition" interrupt has occurred since the last read of this register.

The user can disable or enable the "Change in AIS Condition" interrupt by writing the appropriate value to bit 0 (AIS Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 1" as depicted below.

Rx E3 Framer Interrupt Enable Register- 1 (Address = 10h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused			CCFA Interrupt Enable	COF Interrupt Enable	LOF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/Write" bit-field disables the "Change in AIS Condition" interrupt. Writing a "1" to this bit-field enables the "Change in AIS Condition" interrupt.

7.1.2.9.5 The "BIP-8 Error" Interrupt

The Receive E3 Framer will generate the "BIP-8 Error" interrupt upon detection of an errored incoming E3 frame.

Recall from Section 7.1.2.4, that the Receive E3 Framer computes the BIP-8 value, based upon all 537 octets, within a given incoming E3 framer. Afterwards, the Receive E3 Framer will compare its "locally computed" BIP-8 value with the contents of the EM byte-field, in the very next incoming E3 frame. If the "locally computed" BIP-8 value match that received in the EM byte of incoming E3 frame, then the Receive E3 Framer will conclude that this E3 Frame was received in an "error-free" manner. However, if the "locally computed" BIP-8 value does not match that received in the EM byte of the incoming E3 frame, then the Receive E3 Framer will conclude the this newly received E3 frame is errored.

When the Receive E3 Framer generates the "BIP-8 Error" Interrupt, then it will assert Bit 2 (BIP-8 Error) within the Rx E3 Framer Interrupt Status Register-1" (Address = 13h); as depicted below.

Rx E3 Framer Interrupt Status Register-2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	x	x	1	x	x

This "Reset upon Read" bit-field will be set to "1" if the "BIP-8 Error" interrupt has occurred since the last read of this register.

The user can disable or enable the "BIP-8 Error" interrupt by writing the appropriate value to bit 2 (BIP-8 Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register-2", as depicted below.

Rx E3 Framer Interrupt Enable Register-2 (Address = 11h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/Write" bit-field disables the "BIP-8 Error" interrupt. Writing a "1" to this bit-field enables the "BIP-8 Error" interrupt.

7.1.2.9.6 "Change in FERF Condition" Interrupt

The Receive E3 Framer will generate the "Change in FERF Condition" interrupt in response to either of the following two occurrences.

1. Whenever the Receive E3 Framer declares a "FERF" condition

2. Whenever the Receive E3 Framer negates the "FERF" condition

Recall from Section 7.1.2.3.3, the Receive E3 Framer will declare a "FERF (Far- End- Receive Failure)" condition if it receives a "user- selectable" number of incoming E3 frames, with the "FERF" bit set to "1". This "user selectable" number of E3 frames is either 3 or 5.

Likewise, the Receive E3 Framer will negate the "FERF" condition, if it receives the "user- selectable" number of incoming E3 frames, with the "FERF" bit set to "0". Once again, this "user- selectable" number of E3 frames is either 3 or 5.

When the Receive E3 Framer generates the "Change in FERF Condition" interrupt, then it will assert Bit 3 (FERF Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below:

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	x	1	x	x	x

This "Reset upon Read" bit- field will be set to "1" if a "Change in FERF Condition" interrupt has occurred since the last read of this register.

The user can disable or enable the "Change in FERF Condition" interrupt by writing the appropriate value to bit 3 (FERF Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2" as depicted below:

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Change in FERF Condition" interrupt. Writing a "1" to this bit- field enables the "Change in FERF Condition" interrupt.

7.1.2.9.7 "Change of Frame Alignment" Interrupt

The Receive E3 Framer will generate the "Change of Frame Alignment" interrupt if it has detected a change in Frame Alignment, in the incoming E3 frames.

Note: This interrupt will typically be accompanied with the occurrences of the following types of Interrupts:

- Framing Bit Error
- Change in OOF Status
- Change in LOF Status

When the Receive E3 Framer generates the "Change in Frame Alignment" interrupt, then it will assert Bit 4 (COFA Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 1 (Address = 12h); as depicted below:

Rx E3 Framer Interrupt Status Register- 1 (Address = 12h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			COFA Interrupt Status	OOF Interrupt Status	LOF Interrupt Status	LOS Interrupt Status	AIS Interrupt Status
RO	RO	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	1	x	x	x	x

This "Reset upon Read" bit- field will be set to "1" if the "Change of Frame Alignment" interrupt has occurred since the last read of this register.

The user can disable or enable the "Change of Frame Alignment" interrupt by writing the appropriate value to Bit 4 (COFA Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 1" as depicted below:

Rx E3 Framer Interrupt Enable Register- 1 (Address = 10h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused			COFA Interrupt Enable	OOF Interrupt Enable	LOF Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Change of Framing Alignment" interrupt. Writing a "1" to this bit- field enables the "Change of Framing Alignment" interrupt.

7.1.2.9.8 "Receipt of new Trail Trace Buffer Message" Interrupt

The Receive E3 Framer will generate the "Receipt of New Trail Trace Buffer Message" interrupt upon detection of a TR byte, (within an incoming E3 frame) that contains a "0" in the MSB position.

Recall from Section 7.1.2.7, that when the Receive E3 Framer receives an E3 frame, containing a TR byte of the form "0xxxxxxx", then the Receive E3 Framer will write this value into the Rx TTB- 0 Register; and will generate the "Receipt of New Trail Trace Buffer Message" interrupt. The purpose of this interrupt is to notify the local µP that the Receive E3 Framer is in the process of receiving a new Trail Trace Buffer Message, from the "Far- End" Transmitter.

When the Receive E3 Framer generates the "New Trail Trace Buffer Message" interrupt, then it will assert Bit 6 (TTB Change Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	x	x	x	x	x	x

This "Reset upon Read" bit- field will be set to "1" if a "Receipt of New Trail Trace Buffer Message" interrupt has occurred since the last read of this register.

The user can disable or enable the "Receipt of New Trail Trace Buffer Message" interrupt by writing the appropriate value to Bit 6 (TTB Change Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2" as depicted below.

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Receipt of New Trail Trace Buffer Message" interrupt. Writing a "1" to this bit- field enables this interrupt.

7.1.2.9.9 "Payload Type Mismatch" Interrupt

The Receive E3 Framer will generate the "Payload Type Msmatch" interrupt, when it detects that the values, within the Payload Type bit- fields of the incoming E3 frame, are different from that which is expected by the Receive E3 Framer.

Recall from Section 6.3.2.1.4, that the MA byte (within the E3 frame) contains three bit- fields that are used to identify the "kind of data" that is being transported in the 530 bytes of E3 frame payload data. In this particular device, the E3 frames will typically be carrying ATM cell data, in their payload bytes. Hence, the Payload Type value, for these E3 frame will typically be "010". The user can specify the "payload type" value that the Receive E3 Framer should expect from the incoming E3 frames, by writing this into bits 2-0, within the "Rx E3 Configuration & Status" Register (Address = 0Eh); as depicted below.

Rx E3 Configuration and Status Register- 1 (Address = 0Eh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxPLDType[2:0]			RxFERF Algo	Rx TMark Algo	RxPLDExp[2:0]		
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

If the Receive E3 Framer starts to receive E3 frames that contain a different "Payload Type" than what is written into the "Rx E3 Configuration and Status" Register- 1; then the Receive E3 Framer will generate the "Payload Type Mismatch" Interrupt. The purpose of this interrupt is to let the local µP know that the "Near- End" terminal is now receiving E3 frames that are carrying a different type of data (in its payload bytes), than that of previous E3 frames.

When the Receive E3 Framer generates the "Payload Type Mismatch" interrupt, then it will assert Bit 0 (Rx Pld Ms. Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	x	x	x	x	1

This "Reset upon Read" bit- field will be set to "1" if a "Payload Type Mismatch" interrupt has occurred since the last read of this register.

The user can disable or enable the "Payload Type Mismatch" interrupt by writing the appropriate value to Bit 0 (Rx Pld Ms. Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2" as depicted below.

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Payload Type Mismatch" interrupt. Writing a "1" to this bit- field enables this interrupt.

7.1.2.9.10 "Receipt of New LAPD Message frame" Interrupt

The Receive E3 Framer will generate the "Receipt of New LAPD Message frame" interrupt, when the LAPD Receiver has received a complete LAPD Message frame, from the "Far- End" LAPD Transmitter. The purpose of this interrupt is to inform the local µP that the LAPD Receiver has received a LAPD Message, from the "Far- End" LAPD Transmitter, and that the "Receive LAPD Message Buffer" contains a PMDL Message that is ready to be read and processed by the µP.

When the Receive E3 Framer generates the "Receipt of New LAPD Message" interrupt, then it will assert Bit 5 (LAPD Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below:

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	1	x	x	x	x	x

This "Reset upon Read" bit- field will be set to "1" if a "Receipt of New LAPD Message" interrupt has occurred since the last read of this register.

The user can disable or enable the "Receipt of New LAPD Message" interrupt by writing the appropriate value to Bit 5 (LAPD Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2", as depicted below:

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Receipt of New LAPD Message" interrupt. Writing a "1" to this bit- field enables this interrupt.

7.1.2.9.11 FEBE Interrupt

The Receive E3 Framer will generate the "Far- End- Block- Error" (FEBE) interrupt, anytime it detects a "1" in the FEBE bit- field, within an incoming E3 frame. The purpose of this interrupt is to let the local µP know that the "Far- End" Receive E3 Framer is receiving errored E3 frames, from the "Near- End" Transmit E3 Framer.

When the Receive E3 Framer generates the "FEBE" interrupt, then it will assert Bit 4 (FEBE Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Bit Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	1	x	x	x	x

This "Reset upon Read" bit- field will be set to "1" if the "FEBE" interrupt has occurred since the last read of this register.

The user can disable or enable the "FEBE" interrupt by writing the appropriate value to bit 4 (FEBE Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2" as depicted below.

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Bit Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/Write" bit- field disables the "FEBE" interrupt. Writing a "1" to this bit- field enables the "FEBE" interrupt.

7.1.2.9.12 Framing Byte Error Interrupt

The Receive E3 Framer will generate the "Framing Byte Error" interrupt upon detection of an error in either the FA1 or FA2 framing octets, within an incoming E3 frame.

Recall from Section 7.1.2.2, that the purpose of the FA1 and FA2 octets are to allow the Receive E3 Framer to locate the boundaries of each incoming E3 frames. The FA1 and FA2 octets are assigned the value of F6h and 28h, respectively.

The purpose of this interrupt is to let the local µP know that the "Near- End" Receive E3 Framer may be experiencing the early stages of an "OOF (Out- of- Frame)" condition.

When the Receive E3 Framer generates the "Framing Byte Error" interrupt, then it will assert Bit 1 (Framing Byte Error Interrupt Status) within the Rx E3 Framer Interrupt Status Register- 2 (Address = 13h); as depicted below.

Rx E3 Framer Interrupt Status Register- 2 (Address = 13h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Status	LAPD Interrupt Status	FEBE Interrupt Status	FERF Interrupt Status	BIP- 8 Interrupt Status	Framing Byte Error Interrupt Status	Rx Pld Ms Interrupt Status
RO	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	x	x	x	x	x	1	x

This "Reset upon Read" bit- field will be set to "1" if the "Framing Byte Error" Interrupt has occurred since the last read of this register.

The user can disable or enable the "Framing Byte Error" interrupt by writing the appropriate value to bit 1 (Framing Byte Error Interrupt Enable) within the "Rx E3 Framer Interrupt Enable Register- 2" as depicted below.

Rx E3 Framer Interrupt Enable Register- 2 (Address = 11h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	TTB Change Interrupt Enable	LAPD Interrupt Enable	FEBE Interrupt Enable	FERF Interrupt Enable	BIP- 8 Interrupt Enable	Framing Byte Error Interrupt Enable	Rx Pld Ms Interrupt Enable
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Writing a "0" to this "Read/ Write" bit- field disables the "Framing Byte Error" interrupt. Writing a "1" to this bit- field enables the "Framing Byte Error" interrupt.

7.2 RECEIVE CELL PROCESSOR

7.2.1 Brief Description of the Receive Cell Processor

The Receive Cell Processor receives unframed ATM cell data from the Receive E3 Framer. The Receive Cell Processor will then perform the following operations on this data.

- Cell Delineation
- HEC Byte Verification
- Idle Cell Filtering (optional)
- User/ OAM Cell Filtering (optional)
- Cell- payload de- scrambling (optional)

The Receive Cell Processor will also output the GFC Nibble value of each incoming cell, via the "GFC Nibble- Field" serial output port.

Figure 69 presents a simple block diagram of the Receive Cell Processor block along with its external pins.

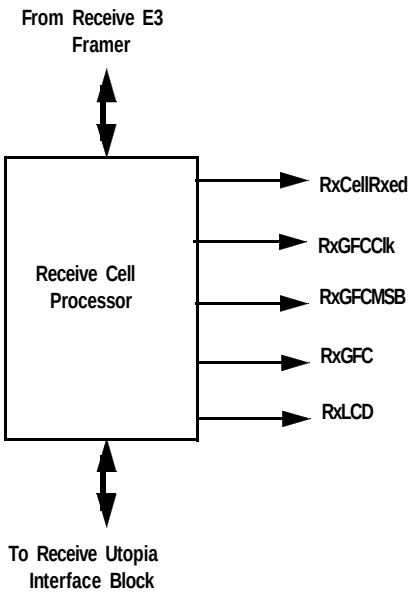

Figure 69. Simple Illustration of the Receive Cell Processor, with associated Pins

7.2.2 Functional Description of Receive Cell Processor

The Receive Cell Processor receives unframed ATM Cell data from the Receive E3 Framer. Once the Receive Cell Processor receives this information then it will proceed to perform the following functions.

- Cell Delineation
- HEC Byte Verification (Header Error Detection/ Correction)
- Idle Cell Filtering
- User Cell Filtering
- Cell Payload De- Scrambling

Each of these functions are discussed in detail below. Figure 70 presents a functional block diagram of the Receive Cell Processor.

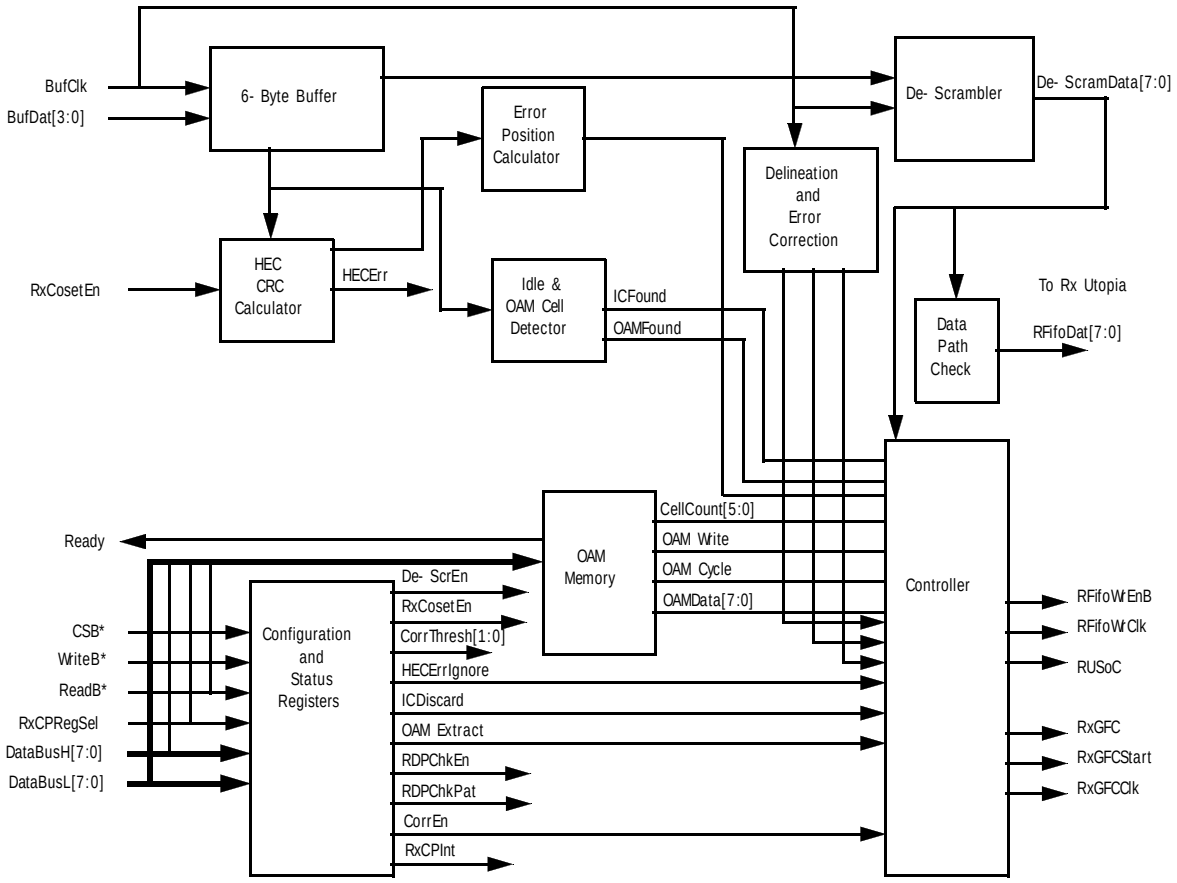

Figure 70. Functional Block Diagram of the Receive Cell Processor

7.2.2.1 Cell Delineation

The Receive Cell Processor is receiving unframed cell data from the Receive E3 Framer. Therefore, the Receive Cell Processor 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 71. Each of these states are discussed below.

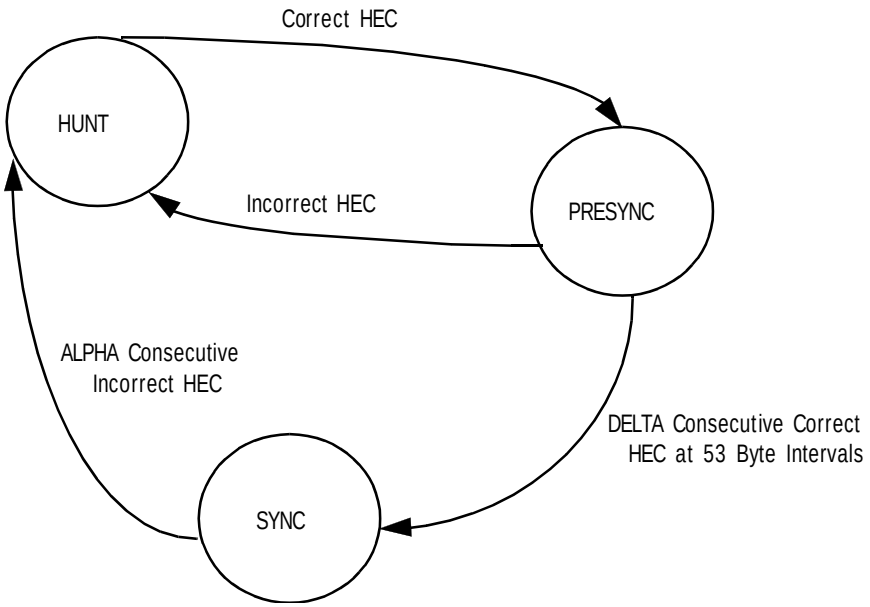

Figure 71. Cell Delineation Algorithm Employed by the Receive Cell Processor

The HUNT State

When the UNI chip is first powered up, the Receive Cell Processor will initially be operating in the "HUNT" state. While the Receive Cell Processor 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 Receive Cell Processor 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 Receive Cell Processor will read in five octets of the data that it receives from the Receive E3 framer. The Receive Cell Processor will then compute a "HEC byte" value based upon the first four of these five octets. The Receive Cell Processor will then compare this computed value with that of the 5th "read- in" octet. If the two values are not the same, then the Receive Cell Processor will increment its sampling set (of the 5 bytes) by one bit, and repeat the above- process with this new set of "candidate" header bytes. In other words, the Receive Cell Processor make its next selection of the five "sample" octets, 53 bytes and 1 bit later.

If the Receive Cell Processor 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 Receive Cell Processor will transition to the PRESYNC state.

The PRE- SYNC State

The Receive Cell Processor 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 Receive Cell Processor is being "fooled" by user data that mimics the cell header byte pattern. Therefore, further evaluation is required in order to confirm that this set of octets are truly valid cell header bytes. The purpose of the "PRE- SYNC" state is to facilitate this "further evaluation".

When the Receive Cell Processor is operating in the PRE- SYNC state, it will then begin to sample 5 "candidate header bytes" at 53 byte intervals. During this sampling process, the Receive Cell Processor will compute and compare its newly computed "HEC byte value" with that of the fifth (read- in) octet. If the Receive Cell Processor, while operating in the PRE- SYNC state, comes across a single invalid cell header byte pattern, then the Receive Cell Processor will transition back to the "HUNT" state. However, if the Receive Cell Processor detects "DELTA" consecutive valid cell byte headers, then it will transition into the SYNC state.

The SYNC State

The Receive Cell Processor will notify the local mP (and external circuitry) of its transition to the SYNC state by

- Generating a "Change of LCD (Loss of Cell Delineation) State" interrupt. When the Receive Cell Processor generates the "Change in LCD Condition" interrupt, it will also set Bit 1 (LCD Interrupt Status) within the "Rx CP Interrupt Status" Register, as depicted below.

Rx CP Interrupt Status Register (Address = 61h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused					Received OAM Cell Interrupt Status	LCD Interrupt Status	HEC Error Interrupt Status
RO	RO	RO	RO	RO	RUR	RUR	RUR
0	0	0	0	0	0	1	x

- Negating the RxLCD output pin (e.g., toggling it "low"); and
- Setting bit 7 (Rx LCD) within the Rx CP Configuration Register to "0".

When the Receive Cell Processor is operating in the SYNC state, it is assumed to be properly delineating the ATM Cells.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	De-Scramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	x	x	x	x	x	x	x

Once the Receive Cell Processor is running the "SYNC" state, it will tolerate some sporadic errors in the cell header bytes and, in some cases, even attempt to correct them. However, the occurrence of "ALPHA" consecutive cells with header byte errors (single or multi- bit), will cause the Receive Cell Processor to return to the "HUNT" state. The Receive Cell Processor will notify the external circuitry that it is not properly delineating cells by doing the following.

- Generating a "Change in LCD State" interrupt
- Asserting the RxLCD output pin (e.g., toggling it "high").

- Setting bit 7 (Rx LCD) within the Rx CP Configuration Register to "0", as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	De-Scramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	x	x

The remaining discussion of the Receive Cell Processor, within this data sheet, presumes that it (the Receive Cell Processor) is operating in the "SYNC" state and is properly delineating cells.

The Overall Cell Filtering/ Processing Approach within the Receive Cell Processor block

Once the Receive 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 written to the RxFIFO within the Receive Utopia Interface block.

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
- Inserting of the "Data Path Integrity Check" pattern into the 5th octet of each cell.

This sequence of processing; (within the Receive Cell Processor) is also illustrated in Figure 72.

Each of these "Filtering/ Processing" steps (within the Receive Cell Processor) are discussed in detail below.

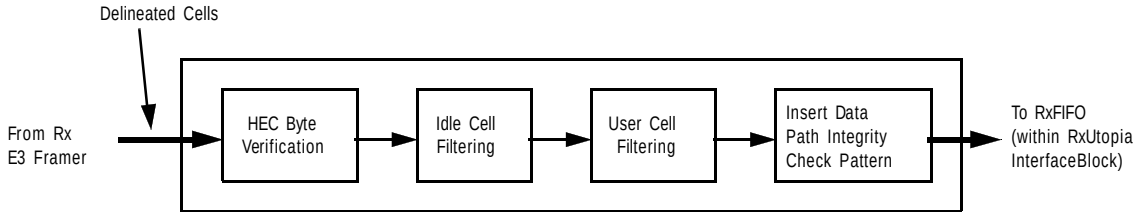

Figure 72. Illustration of Overall Cell Filtering/ Processing procedure that occurs within the Receive Cell Processor Block

7.2.2.2 HEC Byte Verification

Once the Receive Cell Processor is properly delineating cells, the Receive Cell Processor will perform "HEC Byte Verification" of incoming cell data from the Receive E3 Framer, in order to protect against mis- routed or mis- inserted cells. In performing HEC Byte Verification the Receive Cell Processor will take the first four bytes of each cell (e.g., the header bytes) and independently compute its own value for the HEC byte. Afterwards, the Receive Cell Processor will compare its value of the HEC byte with the fifth octet that it has received from the Receive E3 Framer. If the two HEC byte values match then the Receive Cell Processor will retain this cell for further processing. However, if the Receive Cell Processor detects errors in the header bytes of a cell, then the Receive Cell Processor will call up and employ the "HEC Byte Error Correction/ Detection" Algorithm (see below).

The Receive Cell Processor 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 Cell Processor must be configured to account for this by writing a "1" to Bit 1 (Rx Coset Enable) of the Rx CP Configuration Register; as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	De-Scramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	1	x

The "HEC Byte Error Correction/ Detection" Algorithm

If the Receive Cell Processor detects one or more errors in the header bytes of a given cell, then the "HEC Byte Error Correction/ Detection" algorithm will be employed. The "HEC Byte Error Correction/ Detection" Algorithm has two states: Detection and Correction. Figure 73 presents a State Machine Diagram of the "HEC Byte Error Correction/ Detection" Algorithm. Each of these states are discussed below.

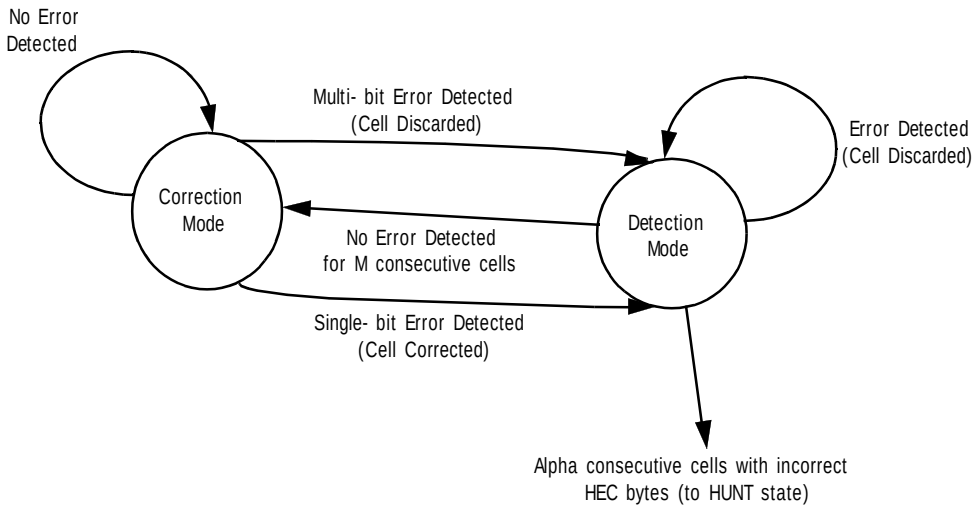

Figure 73. State Machine Diagram of the HEC Byte Error Correction/ Detection Algorithm

The "Correction" State:

When the "HEC Byte Correction/ Detection" Algorithm is operating in the "Correction Mode", cells with single bit errors (within the header bytes) will be corrected. However, cells with multiple bit errors are discarded, unless configured by the user. The user can configure the Receive Cell Processor to retain these cells with multi- bit errors, by writing to bit 0 (HEC Error Ignore) of the Rx CP Configuration Register, as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pat	RDPChk PatEn	Idle Cell Discard	OAM Chk Bit	DeScrEn	RxCoset En	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	x	x

Writing a "1" into this bit- field causes the Receive Cell Processor to retain errored cells for further processing. Writing a "0" to this bit- field causes the Receive Cell Processor to discard those cells with multi- bit errors.

Note: The occurrence of any cells with header byte errors (single- bit or multi- bit errors) will cause the Receive Cell Processor to transition from the "Correction" state to the "Detection" state.

Monitoring of Single- Bit Errors, during HEC Byte Verification.

The user can monitor the number of Single Bit Errors that have been detected by the Receive Cell Processor during HEC Byte Verification. Each time the Receive Cell Processor detects a Single- Bit error, the "PMON Received Single- Bit HEC Error Count" registers are incremented. These registers are located at addresses 48h and 49h and their bit- formats are presented below.

Address = 48h, PMON Received Single HEC Error Count - MSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
S- HEC Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Address = 49h, PMON Received Single HEC Error Count-LSB							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
S- HEC Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The contents of these registers reflect the total number of Single- Bit Errors that have been detected by the Receive Cell Processor since the last read of this register. These registers are reset upon read.

Monitoring of Multi- Bit Errors, during HEC Byte Verification

The user can also monitor the number of Multiple Bit Errors that have been detected by the Receive Cell Processor, during HEC Byte Verification by reading the PMON Received Multiple- Bit HEC Error Count Registers (Addresses = 4Ah and 4Bh). These registers are incremented once for each incoming cell that contains multiple (e.g., more than 1) bit- errors. The bit format of these two registers follow.

PMON Received Multiple- Bit HEC Error-MSB (Address = 4Ah)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
M- HEC Error Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Received Multiple- Bit HEC Error-LSB (Address = 4Bh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
M- HEC Error Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The contents of these registers reflect the number of cells with Multiple- Bit Errors that have been detected by the Receive Cell Processor, during HEC Byte Verification, since the last read of this register. These registers are reset upon read.

The "Detection" State:

When the "HEC Byte Error Detection/ Correction" algorithm is operating in the Detection mode, then all errored cells (e.g., those cells with single- bit errors and multi- bit errors) will be discarded, unless configured otherwise by the user. The user can configure the Receive Cell Processor to retain errored cells by writing to bit 0 (HEC Error Ignore) of the Rx CP Configuration register (Address = 5Eh), as described above.

The "HEC Byte Error Correction/ Detection" Algorithm will transition back into the "Correction" state once the Receive Cell Processor has detected "M" consecutive cells with the correct HEC byte values. The user has the option to use the following values for "M": 0, 1, 3, and 7. The user can configure the UNI to use any of these values for M by writing the appropriate values to the "Rx CP Additional Configuration" Register (Address = 5Fh), as depicted below.

Rx CP Additional Configuration Register (Address = 5Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		User Cell Filter Discard	User Cell Filter Enable	Correction Threshold [1, 0]		Correct Enable	Unused
RO	RO	R/W	R/W	R/W	R/W	R/W	RO

The definition of the bits relevant to the "HEC Byte Error Correction/ Detection" algorithm follow.

Bit 1-Correction (Mode) Enable

This "Read/ Write" bit field allows the user to enable/ disable the "Correction Mode" portion of the "HEC Byte Error Correction/ Detection" algorithm. If the user writes a "0" to this bit- field, this the "HEC Byte Error Correction/ Detection" algorithm be disabled from entry/ operation in the "Correction" mode. Therefore, the Receive Cell Processor will only operate in the "Detection" state. If the user writes a "1" to this bit field then the "HEC Byte Error Correction/ Detection" algorithm will transition into and out of the "Correction" mode as dictated by the selected "Correction Threshold" value.

Bits 2 and 3-Correction Threshold [1, 0]

These "Read/ Write" bit- fields allow the user to select the "Correction" Threshold for the "HEC Byte Error Correction/ Detection" algorithm. The following table relates the content of these bit- fields to the Correction Threshold Value (M). Once again, M is the number of consecutive "Error- Free" cells that the Receive Cell Processor must detect before the "HEC Byte Correction/ Detection" algorithm will allow a transition back into the "Correction" Mode.

Table 30. The Relationship between CorrThreshold[1:0] and the "Correction Threshold" Value (M)

Bit 3	Bit 2	Correction Threshold Value (M)
0	0	M = 0
0	1	M = 1
1	0	M = 3
1	1	M = 7

7.2.2.3 Cell Filtering

As mentioned earlier, the Receive Cell Processor will filter (e.g., discard) incoming cells based upon the following criteria.

- HEC Byte Errors (via the "HEC Byte Correction/ Detection" algorithm, as described in 7.2.2.2.)
- Idle Cells
- Header Byte Patterns-User Cells
- Segment OAM Cells

Each of these cell filtering approaches are presented below.

Filtering of Cells with HEC Byte Errors

Please see the "HEC Byte Correction/ Detection" algorithm in Section 7.2.2.2.

7.2.2.3.1 Idle Cell Filtering

The user can configure the Receive Cell Processor to either discard or retain Idle cells by writing to bit 4 (Idle Cell Discard) within the "Rx CP Configuration" Register, as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pattern	RDPChk PatEn	Idle Cell Discard	OAM Chk Bit	DeScrEn	RxCosetEn	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	x	x

If the user writes a "0" to this bit- field, then the Idle Cells will be retained and will ultimately be sent on to the User Cell Filter, within the Receive Cell Processor block. However, if the user writes a "1" to this bit- field, then the Receive Cell Processor will discard all detected Idle- cells.

If the user wishes to have the Receive Cell Processor discard the Idle Cells then he/she must specify the header byte patterns of these Idle cells. The Idle Cell header byte pattern is defined based upon the content of 8 read/write registers. These eight registers are the four "Rx CP Idle Cell Pattern Header byte" registers, and the four "Rx CP Idle Cell Mask Header-Byte" Registers. In short, when a cell reaches the "Idle Cell Filter" portion of the Receive Cell Processor, the contents of each header byte of this cell (bytes 1 through 4), will be compared against the contents of the corresponding "Rx CP Idle Cell Pattern Header Byte" registers; based upon constraints specified by the contents within the "Rx CP Idle Cell Mask Header Byte" registers. The user of these registers in "Idle Cell Identification" and filtering is illustrated in the example below.

Example-Idle Cell Filtering

For example, header byte 1 of a given incoming cell (which may be an Idle cell or a User cell) will be subjected to a bit- by- bit comparison to the contents of the "Rx CP Idle Cell Pattern Header Byte- 1" register (Address = 62h). The purpose of having the Receive Cell Processor perform this comparison is to determine if this incoming cell is an Idle Cell or not. The contents of the "Rx CP Idle Cell Mask Header Byte- 1" register (Address = 66h) also plays a role in this comparison process. For instances, if bit- field "0" within the "Rx CP Idle Cell Mask Header Byte- 1" register contains a "1", then the Receive Cell Processor will perform the comparison operation between bit- field "0" within the "Rx CP Idle Cell Pattern Header Byte- 1" register; and bit- field "0" within header byte 1 of the newly received cell. Conversely, if bit- field "0" within the "Rx CP Idle Cell Mask Header Byte- 1" register contains a "0", then this comparison will not be made and bit- field "0" will be treated as a "don't care". The role of these two read/write registers, in these comparison operations is more clearly defined in Figure 74, below.

Content of Header Byte- 1 (of Incoming Cell)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	0	1	0	0	1	0	1

Content of 'Rx CP Idle Cell Mask Header Byte- 1 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	1	1	1	0	0	0	0

Content of 'Rx CP Idle Cell Header Byte- 1 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	0	1	0	1	1	0	1

Comments							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Comparison is Forced (by the "1s" in the Rx CP Idle Cell Mask Header Byte- 1 Register)				Don't Care	Don't Care	Don't Care	Don't Care

Header Byte- 1 Identification of Idle Cell							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	0	1	0	x	x	x	x

Figure 74. Illustration of the Role of the 'Rx CP Idle Cell Pattern Header Byte" Register, and the 'Rx CP Idle Cell Mask Header Byte" Register

Based upon these register settings, any cell containing values in the range of A0h-AFh are considered be matching the "Idle Cell Pattern", at the first byte. This incoming cell will be subjected to three (3) more tests (e.g., one for each of the remaining header bytes.); before it is identified as an Idle Cell or not.

Consequently, if the user opts to "discard" Idle Cells, then any cells, passing the above- described tests, will be identified as an Idle Cell and will be discarded by the Receive Cell Processor.

The bit format for each of these eight "Idle Cell" identification registers are listed below.

Rx CP Idle Cell Pattern Header byte- 1 Register (Address = 62h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Pattern-Header byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP Idle Cell Pattern Header byte- 2 Register (Address = 63h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Pattern-Header byte 2							
R/W	***R/W***	***R/W***	***R/W***	***R/W***	***R/W***	***R/W***	***R/W***
0	***0***	***0***	***0***	***0***	***0***	***0***	***0***

Rx CP Idle Cell Pattern Header byte- 3 Register (Address = 64h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Pattern-Header byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP Idle Cell Pattern Header byte- 4 Register (Address = 65h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Idle Cell Pattern-Header byte							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP Idle Cell Mask Header- Byte 1 Register (Address = 66h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Mask Header-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP Idle Cell Mask Header-Byte 2 Register (Address = 67h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Mask Header-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP Idle Cell Mask Header-Byte 3 Register (Address = 68h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Mask Header-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP Idle Cell Mask Header-Byte 4 Register (Address = 69h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Mask Header-Byte 1							
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 user can periodically monitor the number of Idle Cells that have been detected by the Receive Cell Processor, by reading the "PMON Received Idle Cell Count" Register (Addresses = 4Ch, 4Dh). The bit- format of these registers are presented below.

"PMON Received Idle Cell Count-MSB" Register (Address = 4Ch)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Count–High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

"PMON Received Idle Cell Count-LSB" Register (Address = 4Dh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx Idle Cell Count–Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The content of these registers are the number of Idle Cells that have been detected, by the Receive Cell Processor, since the last read of these registers. These registers are reset upon read.

7.2.2.3.2 User Cell Filtering

The user can configure the Receive Cell Processor to filter incoming user or OAM cells based upon the value of their header bytes. In all, the UNI provides the user with three (3) options.

- Disable the User Cell Filter.
- Pass only those cells with header byte patterns matching the settings of the User Cell Filter.
- Discard only those cells with header byte patterns matching the settings of the User Cell Filter.

Each of these User- Cell Filtering Options are discussed below.

Disable the User- Cell Filter

If the user disables the User- Cell Filter, within the Receive Cell Processor, then all user cells (independent of their header byte patterns) will be written into the Rx FIFO, within the Receive Utopia Interface block.

Rx CP Additional Configuration Register (Address = 5Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused		User Cell Filter Discard	User Cell Filter Enable	Correct Threshold [1, 0]		Correction Enable	Unused
RO	RO	R/W	R/W	R/W	R/W	R/W	RO

Writing a "1" to Bit 4 (User Cell Filter Enable) enables the User Cell Filter. Whereas, writing a '0" to this bit- field disables the User Cell Filter.

Enable the User Cell Filter

If the User Cell Filter is enabled, then the Receive Cell Processor will be filtering user cells in one of two possible manners.

1. Pass Only those cells with header bytes patterns matching the User Cell Filter settings (e.g., the contents of the "Rx CP User Cell Filter Pattern Header Byte" registers), or
2. Discard only those cells with header byte patterns matching the User Cell Filter settings.

The User (or Assigned) cell filtering criteria is defined based upon the contents of 8 read/write registers. These eight registers are the four "Rx CP User Cell Filter Pattern Header byte" registers and the four "Rx CP User Cell Filter Mask Header Byte" registers. In short, when a user cell reaches the Receive Cell Processor, the contents of each header byte of this cell (bytes 1 through 4), will be compared against the contents of the corresponding "Rx CP User Cell Filter Pattern Header Byte" registers based upon constraints specified by the contents of the "Rx CP User Cell Filter Mask Header Byte" registers. The role of these registers in "User Cell Filtering" is illustrated in the example below.

Example-User Cell Filtering

Header byte 1 of a given incoming user cell will be subjected to a bit- by- bit comparison to the contents of the "Rx CP User Cell Filter Pattern Header Byte- 1" register (Address = 6Ah). However, the contents of the "Rx CP User Cell Filter Mask Header Byte- 1" register (Address = 6Eh) also plays a role in this comparison process. For example, if bit- field "0" within the "Rx CP User Cell Filter Mask Header Byte- 1" register contains a "1", then the Receive Cell Processor will perform the comparison operation between bit- field "0" within the "Rx CP User Cell Filter Pattern Header Byte- 1" register; and bit- field "0" within header byte 1 of the newly received user cell. Conversely, if bit- field '0' within the "Rx CP User Cell Filter Mask Header Byte- 1" register contains a '0', then this comparison will not be made and bit- field '0' will be treated as a 'don't care'. The role of these two read/write registers in these comparison operations is more clearly illustrated in Figure 75, below.

Content of Header Byte- 1 (of Incoming User Cell)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	0	1	0	0	1	0	1

Content of 'Rx CP User Cell Filter Mask Header Byte- 1 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	1	1	1	0	0	0	0

Content of 'Rx CP User Cell Filter Pattern Header Byte- 1 Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
1	0	1	0	1	1	0	1

Comments							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Comparison is Forced (by the "1s" in the Rx CP User Cell Filter Mask Header Byte- 1 Register)				Don't Care	Don't Care	Don't Care	Don't Care

Resulting "User Cell Filter" Pattern for Header Byte- 1							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
1	0	1	0	x	x	x	x

Figure 75. Illustration of the Role of the 'Rx CP User Cell Filter Pattern Header Byte" register and the 'Rx CP User Cell Filter Mask Header Byte" register.

Based upon these register settings, any cell containing values in the range of A0h-AFh are considered to be matching, at the first byte. This cell will be subjected to three (3) more tests (e.g., one for each of the remaining header bytes.)

After all of these comparison tests have been performed, a given user cell will be deemed either "matching" or "not matching" the settings of the User Cell Filter. Once the cell has been classified into one of these two categories, its disposition (or fate) is dependent upon the content of bit- field 5 (User Cell Filter Discard) within the "Rx CP Additional Configuration Register (Address = 5Fh). If this bit- field is '0', then only- matching cells will be retained, and are written into the RxFIFO. All remaining User Cells will be discarded. Conversely, it this bit- field is '1', then only 'non-matching' User Cells will be retained and written to the RxFIFO. All 'matching' User Cells will be discarded.

The bit- formats of the 8 registers that define the User Cell Filtering criteria are presented below.

User Cell Filter Header Byte Pattern Registers

Rx CP User Cell Filter Pattern Header-Byte 1 (Address = 6Ah)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Header Pattern-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP User Cell Filter Pattern Header-Byte 2 (Address = 6Bh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Header Pattern-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP User Cell Filter Pattern Header-Byte 3 (Address = 6Ch)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx User Cell Header Pattern-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Rx CP User Cell Filter Pattern Header-Byte 4 (Address = 6Dh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Header Pattern-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

User Cell Filter Mask Registers

Rx CP User Cell Filter Mask Header-Byte 1 (Address = 6Eh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Mask Header-Byte 1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP User Cell Filter Mask Header-Byte 2 (Address = 6Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Mask Header-Byte 2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP User Cell Filter Mask Header-Byte 3 (Address = 70h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Mask Header-Byte 3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Rx CP User Cell Filter Mask Header-Byte 4 (Address = 71h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Rx User Cell Mask Header-Byte 4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

7.2.2.4 OAM Cell Processing

OAM (Operation Administration and Maintenance) cells, are special cells that are generated by the "Layer Management" entity (within the BISDN Reference Model), and are typically used to carry maintenance- related information such as:

- Virtual Path Connection (VPC)/ Virtual Circuit Connection (VCC) failure reporting
- VPC/ VCC continuity check information
- VPC/ VCC continuity verification: OAM Cell Loopback Testing
- VPC/ VCC Performance Monitoring

OAM cells are identified and distinguished from user cells by their specific cell header byte patterns. ATM layer entities can typically use one of four types of OAM cells. These types of OAM cells are listed below.

- F4-Segment
- F4-End to End
- F5-Segment
- F5-End to End

F4 type OAM cells usually carry maintenance related information regarding a specific Virtual Path Connection (VPC). Whereas F5 type OAM cells usually carry maintenance related regarding a specific Virtual Circuit Connection (VCC). The header byte patterns of each of these types of OAM cells is tabulated below.

Table 31. The Header Byte Pattern formats for the Various Types of OAM Cells

OAM Cell	Octet 1	Octet 2	Octet 3	Octet 4
F4 End- to- End	0000aaaa	aaaa0000	00000000	01000a0a
F4 Segment	0000aaaa	aaaa0000	00000000	00110a0a
F5 End- to- End	0000aaaa	aaaazzzz	zzzzzzzz	zzzz101a
F5 Segment	0000aaaa	aaaazzzz	zzzzzzzz	zzzz100a

where: a-bit is available for use by the ATM layer entity
z-Any VCI value other than 0

As far as the XR-T7234 E3 UNI IC is concerned, whether an OAM cell is an F4 or F5 type OAM cell, is rather unimportant. The Receive Cell Processor circuitry has been designed to recognize both types of OAM cells, based upon their header byte pattern. However, whether an OAM cell is a "Segment type" or an "End-to-End type" is more important in regards to UNI IC operation. The manner in which the Receive Cell Processor handles "Segment" and "End-to-End" OAM cells is described below.

7.2.2.4.1 Segment Type OAM Cells

"Segment" type OAM cells are only intended for "point-to-point" transmission. In other words, a segment type OAM cell will be created at a source node, transmitted across a single link, to a destination node; and then terminated at this destination node. This Segment OAM cell is not intended to be read or processed by any other nodes, within the ATM Network

How the Receive Cell Processor handles Segment Type OAM Cells

The Receive Cell Processor has been designed to recognize incoming OAM cells, based upon their header byte pattern. Further, the Receive Cell Processor is also capable of reading the header byte patterns, in order to determine if the OAM cell is a "Segment" type or an "End-to-End" type OAM cell. If the incoming OAM cell is a "Segment" type OAM cell, then the Receive Cell Processor will not write this cell to the Rx FIFO, within the Receive Utopia Interface block and will discard this cell. This act of discarding the OAM cell terminates it and prevents it from propagating to other nodes in the network.

Note: If the user configures the User Cell Filter to pass cells with header bytes pattern ranges that includes that of the "Segment"-type OAM Cell, then the User Cell Filter settings will take precedence and will allow the "Segment"-type OAM Cell to be written to the Rx FIFO, within the Receive Utopia Interface block.

Although the Receive Cell Processor will discard this "Segment" OAM cell, the user can configure the Receive Cell Processor to have the contents of this cell written into the Receive OAM Cell Buffer, where it can be read out and processed by the local µP/µC.

If the user writes a "1" to bit 3 (OAM Check Bit) within the "Rx CP Configuration" register (Address = 5Eh), then all OAM cells that are received by the Receive Cell Processor will be written into the Receive OAM Cell buffer (located at 161h through 1A1h, in the UNI chip address space).

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	De-Scramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	x	x

Once the Receive Cell Processor has written the OAM cell into the "Receive OAM Cell" buffer, then the Receive Cell Processor will alert the local µP/µC of this fact, by generating the "Received OAM Cell" interrupt. If the user write a "0" to bit 3 of the "Rx CP Configuration" register, then the Receive Cell Processor will not write the contents of the OAM cells that it receives, to the "Receive OAM Cell" buffer.

Figure 76 presents an illustration, depicted how the Receive Cell Processor handles incoming Segment- type OAM cells, if the user has written a "1" to bit 3 (OAM Check Bit) of the "Rx CP Configuration" register.

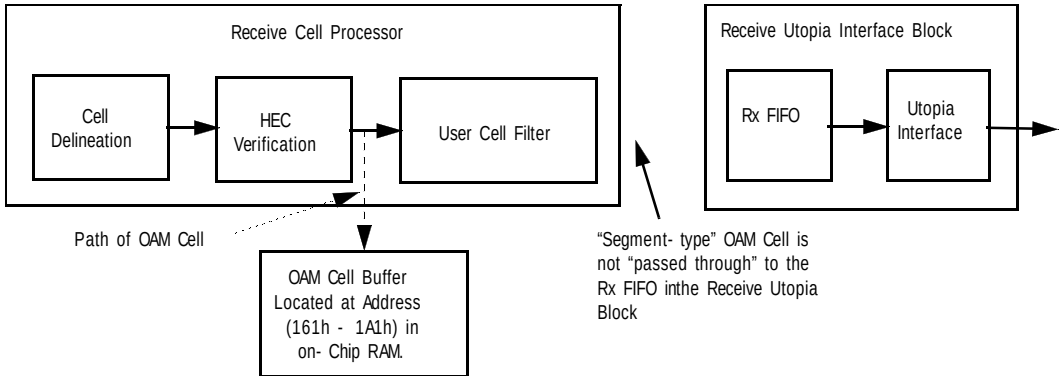

Figure 76. An Approach to Processing Segment OAM cells, via the Receive Cell Processor.

7.2.2.4.2 End- to- End Type OAM Cells

'End- to- End' type OAM cells, as the name implies, are intended for something more than a point- to- point transmission. In other words, an "End- to- End" type OAM cell will be created at a source node, transmitted across a single link, to a destination node. However, in this case, the "end- to- end" OAM cell is not terminated at this destination node; but is rather transmitted across other links to other nodes within the network.

How the Receive Cell Processor Handles End- to- End type OAM Cells

If the Receive Cell Processor determines that the incoming OAM Cell is an "End- to- End" type then it will be written into the Rx FIFO, within the Receive Utopia Interface block. This act will allow the ATM Layer processor to read in this OAM cell, from the UNI and propagate this cell to other nodes in the network.

Note: The Receive Cell Processor will write the "End- to- End" OAM cell to the Rx FIFO, independent of the User Cell Filter settings.

The user can also configure the Receive Cell Processor to write the contents of the "End- to- End" OAM cell into the Receive OAM Cell Buffer. For details on how this can be done, please see Section 7.2.2.4.1.

Figure 77 presents an illustration, which depicts how the Receive Cell Processor handles incoming End- to- End type OAM Cells, if the user has written a "1" to bit 3 (OAM Check Bit) of the "Rx CP Configuration" register.

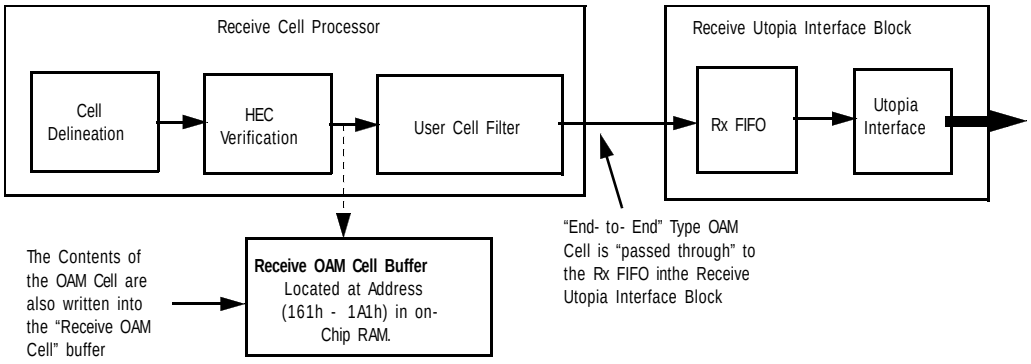

Figure 77. Approach to Processing 'End- to- End' OAM Cells

Monitoring the Number of User/ OAM Cells

The user can monitor the number of Valid cells (User and OAM) that have been received by the Receive Cell Processor, by reading the PMON Received Valid Cell Count Registers (Address = 4Eh, and 4Fh). The bit- format of these registers are presented below.

PMON Received Valid Cell Count-MSB (Address = 4Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Valid Cell Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Received Valid Cell Count-LSB (Address = 4Fh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Rx Valid Cell Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The contents of this register reflects the total number of valid cells that the Receive Cell Processor has received since the last reading of this register. This register is reset upon read.

Finally, the user can also monitor the total number of cells that have been discarded (either due to HEC byte errors, Idle Cell removal, or User cell filtering) by reading the PMON Discarded Cell Count Registers (Address = 50h, 51h). The bit- format of this register is presented below.

PMON Discarded Cell Count-MSB (Address = 50h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Cell Drop Count-High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Discarded Cell Count-LSB (Address = 51h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Cell Drop Count-Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The contents of these registers reflect the number of cells that have been discarded since the last read of these registers. These registers are reset upon read.

7.3.2.5 Cell Payload De- Scrambling

In numerous applications the payload portion of the incoming cells will scrambled by the Transmit Cell Processor, at the "Far End" Transmitting terminal. These cells are scrambled in order to prevent the user data from mimicking framing or control bytes. Therefore, the Receive 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 writing a "1" to Bit 2 (De- Scramble Enable) within the Rx CP Configuration Register, as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	De-Scramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	x	x	x	x	x	x	x

7.2.2.6 Data Path Integrity Check

The "Data Path Integrity" check is a test that is continually ran in order to verify that the connections, throughout the "ATM Layer" entity (e.g., from the Receive Utopia Interface of the "source" UNI to the Transmit Utopia Interface of the "destination" UNI) is functioning properly.

The manner in which the "Data Path Integrity Check" is employed is as follows. After an incoming cell has passed through the cell delineation, HEC byte verification, idle cell filtering and user cell filtering processes, it will be written to the Rx FIFO, within the Receive Utopia Interface Block. However, prior to being written into the Rx FIFO, the "Data Path Integrity Test" pattern will be written into the 5th octet (overwriting the HEC byte) of the "outbound" cell. This "Data Path Integrity Test" pattern is typically of the value "55h", for each outbound cell. However, it can also be configured to be an alternating pattern of "55h" and AAh" (alternating values with each cell).

The Transmit Cell Processor, within the "destination" UNI will perform a check of the 5th byte of all cells that it reads from the Tx FIFO, prior to computing and overwriting this byte with the HEC byte. For more information on how the Transmit Cell Processor handles the "Data Path Integrity Check" test patterns, please see section 6.2.2.6.

The Receive Cell Processor's Handling of the Data Path Integrity Test pattern

The user has a variety of options, when it comes to configuring the Receive Cell Processor to support the Data Path Integrity Test. First of all, the user can decide whether or not he/ she wishes to even transmit a Data Path Integrity Test pattern, via the outbound cell; or just allow the outbound cell, with the HEC byte to be written to the Rx FIFO. The user can configure the Receive Cell Processor per his/ her choice, by writing the appropriate value into bit 5 (RDPChk Pattern Enable) within the "Rx CP Configuration Register (Address = 5Eh) as depicted below.

Rx CP Configuration Register (Address = 5Eh)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
RxLCD	RDPChk Pattern	RDPChk Pattern Enable	Idle Cell Discard	OAM Check Bit	DeScramble Enable	RxCoset Enable	HEC Error Ignore
RO	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W
0	x	x	x	x	x	x	x

Writing a "1" to this bit- field configures the Receive Cell Processor to write the "Data Path Integrity Test" pattern into the 5th octet of each "outbound" cell, prior to transmittal to the RxFIFO. Conversely, writing a "0" to this bit- field configures the Receive Cell Processor to write the cell, with the HEC byte, into the RxFIFO.

Next, the Receive Cell Processor also allows the user to choose between two possible Data Path Integrity Test patterns. The user can configure his/ her selection by writing the appropriate value to Bit 6 (RDPChk Pattern) within the "Rx CP Configuration" Register (Address = 5Eh). Writing a "1" to this bit- field configures the Receive Cell Processor to write a "55h" into the 5th octet of each "outbound" cell, prior to it being written into the Rx FIFO. Conversely, writing a "0" to this bit- field configures the Receive Cell Processor to write an alternating pattern of "55h" or "AAh", into the 5th octet of each "outbound" cell, prior to it being written into the RxFIFO. The Receive Cell Processor will alternate between each of these two patterns with each "outbound" cell.

Note: The contents of Bit 6, within the Rx CP Configuration Register, is ignored if Bit 5 is set to "0".

7.2.2.7 GFC Nibble Extraction-via the RxGFC 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 valid (e.g., user or OAM) 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 valid 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 78 presents an illustration depicting the behavior of the RxGFC Serial Output Port signals.

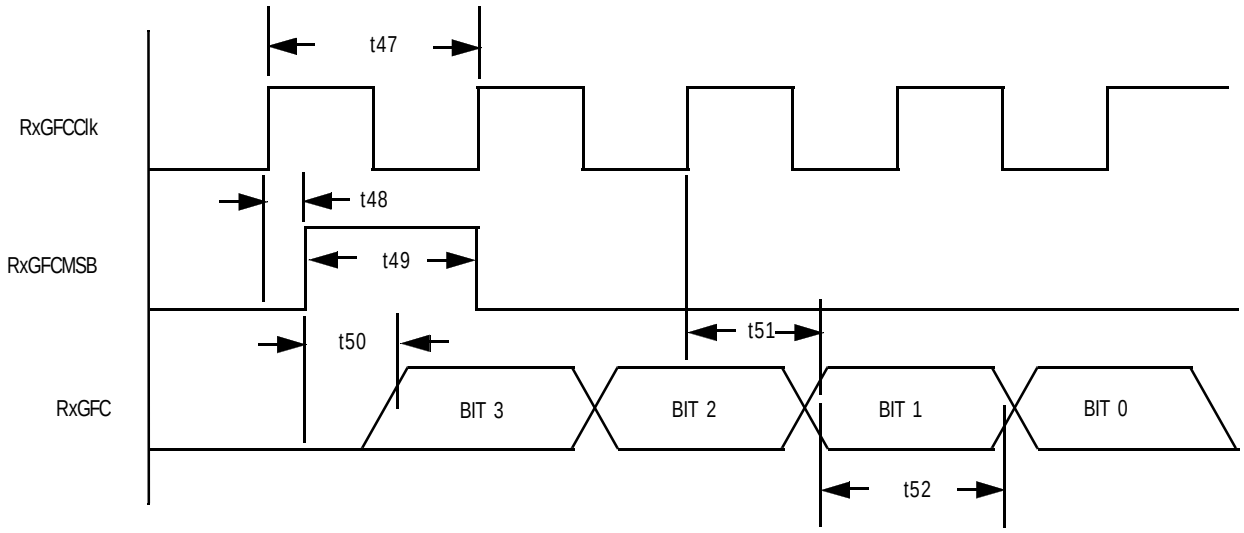

Figure 78. Illustration of the Behavior of the RxGFC Serial Output Port signals

7.3.2.8 Receive Cell Processor Interrupts

The Receive Cell Processor will generate interrupts upon

- HEC Byte 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 mC/ mP reads the UNI Interrupt Status Register, as shown below, it should read '0xxxx1xxb' (where the - b suffix denotes a binary expression, and 'x' denotes a "don't care" value).

UNI Interrupt Status Register (Address = 05h)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	One Sec Interrupt Status	Tx E3 Interrupt Status	Rx E3 Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO
x	x	x	x	x	1	x	x

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 = 61h). The bit format of this register is presented below.

Rx CP Interrupt Status Register (Address = 61h)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Unused					Received OAM Cell Interrupt Status	LCD Interrupt Status	HEC Error Interrupt Status
RO	RO	RO	RO	RO	RUR	RUR	RUR

The bit format of the Rx CP Interrupt Status Register indicates that only three (3) bit- fields within this register, are active. The role of each of these bit fields follows.

Bit 0–HEC Byte Error Interrupt Status

A "1" in this "Reset- upon- Read" bit- field indicates that the Receive Cell Processor has detected a HEC Byte error in an incoming cell, and has requested a "HEC Byte Error" Interrupt, since the last read of this register.

Bit 1–"Change in LCD (Loss of Cell Delineation) State" Interrupt Status

A "1" in this "Reset- upon- Read" bit- field indicates that the Receive Cell Processor has changed its "LCD" (Loss of Cell Delineation) state and has issued the "Change in LCD State" interrupt, since the last read of this register.

Note: This type of interrupt could occur due to a transition from the SYNC state to the HUNT state, in the "HEC Byte Cell Delineation Algorithm; during which the RxLCD pin will toggle "high". Additionally, this type of interrupt could also occur due to the transition from the PRE- SYNC state into the SYNC state. The user can distinguish between these two possibilities by reading the LCD bit- field (bit 7) within the "Rx CP Configuration" Register (Address = 5Eh).

Bit 2-Received OAM Cell Interrupt Status

A "1" in this "Reset- upon- Read" bit- field indicates that the Receive Cell Processor has detected an OAM Cell in the path of "incoming cells"; and has stored the contents of this OAM cell in the "Receive OAM Cell Buffer", since the last read of this register. The purpose of this interrupt is to alert the local µP/µC that the "Receive OAM Cell Buffer" (within the UNI) contains an OAM cell that needs to be read and processed.

7.3 Receive Utopia Interface Block

7.3.1 Brief Description of the Receive Utopia Interface Block

The Receive Utopia Interface Block provides a "Utopia Level 2" compliant interface to interconnect the UNI chip to ATM layer or ATM Adaptation Layer processors, operating up to 800 Mbps. 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 RxClav pin "high". Figure 79 presents a simple illustration of the Receive Utopia Interface block and the associated pins.

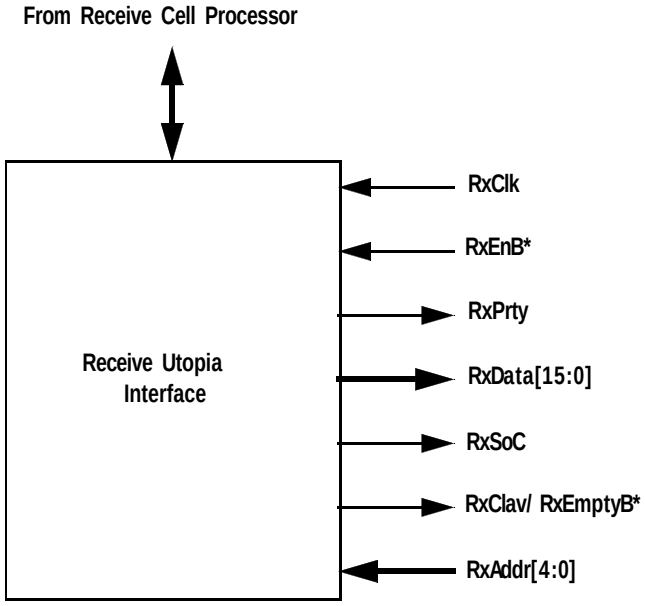

Figure 79. Simple Block Diagram of Receive Utopia Block of UNI.

7.3.2 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 RxSoC (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 RxSoC (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 mP, discard this "runt" cell, and resume normal operation.

The physical size of the Rx FIFO is four 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 RxClav output pin. Figure 74 presents a Functional Block Diagram of the Receive Utopia Interface Block.

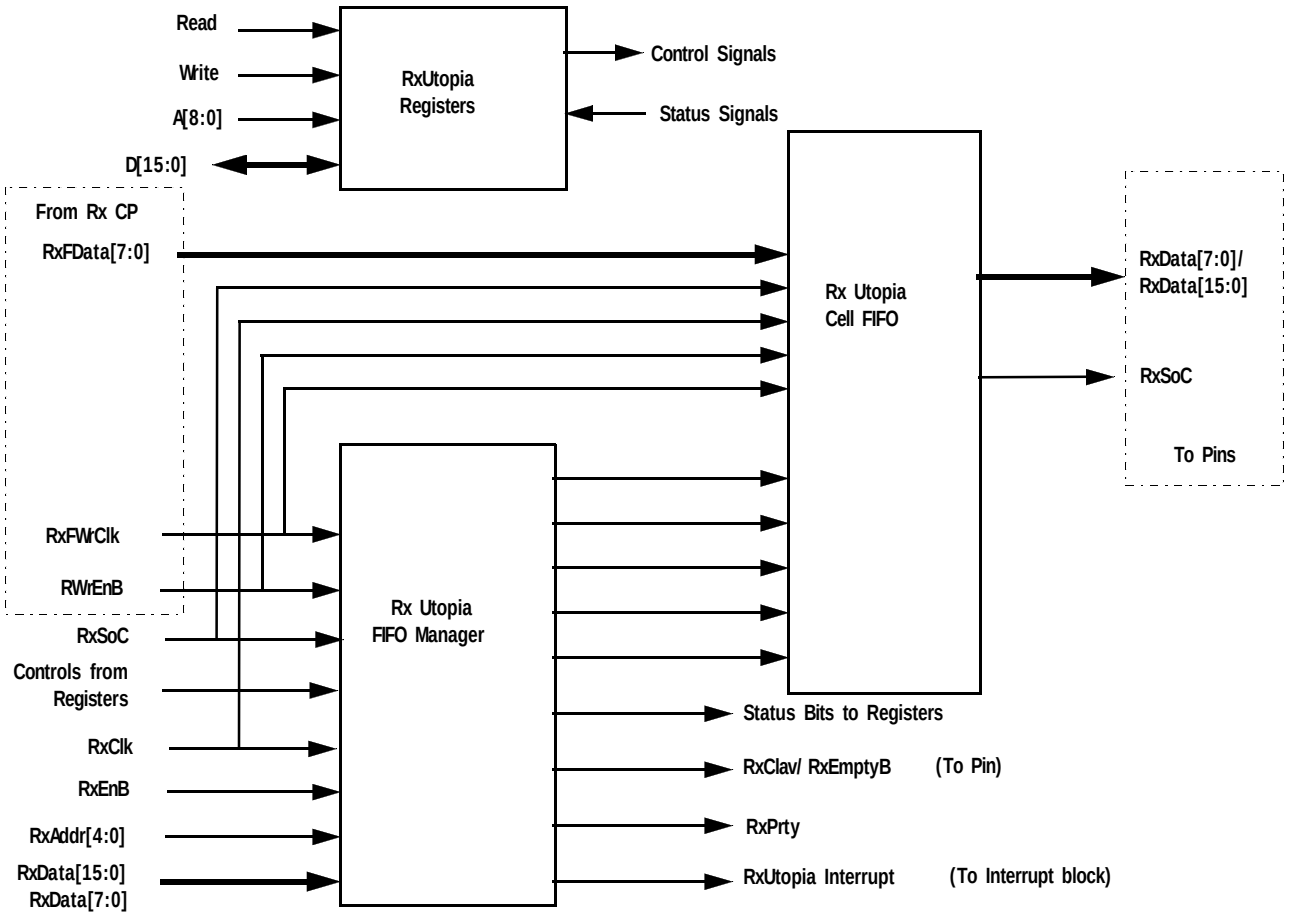

Figure 80. Functional Block Diagram of the Receive Utopia Interface Block

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.

7.3.2.1 Receive Utopia Bus Output Interface

The Receive Utopia output interface complies with the Utopia Level 2 standard interface (e.g., the Receive Utopia can support both Single- PHY and Multi- PHY operations). Additionally, the UNI provides the user will the option of varying the following features associated with the Receive Utopia Bus interface.

- 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)

A discussion of the operation of the Receive Utopia Bus Interface along with each of these options will be presented below.

7.3.2.1.1 The Pins of the Receive Utopia Bus Interface

The ATM Layer processor will interface to the Receive Utopia Interface block via the following pins.

- RxData[15:0]-Receive Utopia Data bus output pins
- RxAddr[4:0]-Receive Utopia Address bus input pins
- RxClk-Receive Utopia Interface Block clock input pin
- RxSoC-Receive "Start of Cell" Indicator" output pin
- RxPrty-Receive Utopia-Odd Parity output pin
- RxEnB*-Receive Utopia Data bus-Output Enable input pin
- RxClav/ RxFullB*-RxFIFO Cell Available output pin

Each of these signals are discussed below.

RxData[15:0]-Receive Utopia Data Bus output pin

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 7.3.2.1.2). If the "8- bit wide" mode is selected, then only the RxData[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., RxData[15:0]) will be active. The Receive Utopia Data bus is tri- stated while the active low RxEnB* (Receive Utopia Bus - Output Enable) input signal is "high". Therefore, the ATM Layer Processor must assert this signal (e.g., toggle RxEnB* 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, RxClk.

Note: The 100 pin version of the XR- T7234 device allocates only 8 pins for the Receive Utopia Data Bus: RxData[7:0]. The Receive Utopia Data Bus pins: RxData[15:8] are not available.

RxAddr[4:0]-Receive Utopia Address Bus input pins

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 7.3.2.2.2.2.

RxClk-Receive Utopia Interface block Clock Signal input pin

The Receive Utopia 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.

RxEnB-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.

RxPrty-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 RxPrty 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.

RxSoC-Receive Utopia-"Start of Cell" Indicator output pin

The Receive Utopia will pulse this output signal "high", for one clock period of RxClk, when the first byte (or word) of a new cell is present on the Receive Utopia Data Bus. This signal will be "low" at all other times.

RxClav/ 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 7.3.2.2.1 for more information regarding this signal.

7.3.2.1.2 Selecting the Utopia Data Bus Width (Applies to the 160 pin Version only)

The Utopia data bus width can be selected to be either 8 or 16 bits by writing the appropriate data into the UtWidth16 bit (bit 0) within the Utopia Configuration Register, as shown below.

Utopia Configuration Register: (Address = 7Ch)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		Hand-shake Mode	M- PHY	CellOf52 Bytes	TFIFODepth[1, 0]		UtWidth16
RO	RO	R/ W	R/ W	R/ W	R/ W		R/ W

If the user chooses a Utopia Data Bus width of 8 bits, then only the Receive Utopia Data output pins: RxData[7:0] will be active. (The output pins: RxData[15:8] will not be active). If the user chooses a Utopia Data bus width of 16 bits, then all of the Receive Utopia Data outputs: RxData[15:0] will be active. The following table relates the value of Bit 0 (UtWidth16) within the Utopia Configuration Register, to the corresponding width of the Utopia Data bus.

Table 32. The Relationship between the contents within Bit 0 (UtWidth16) within the Utopia Configuration Register, and the operating width of the Utopia Data Bus

Value for UtWidth16	Width of Utopia Data Bus
0	8 bit wide Data Bus
1	16 bit wide Data Bus

Note:

1. The selection of this bit also effects the width of the Transmit Utopia Data bus.
2. The Utopia Data Bus width will be 8 bits, upon power up or reset. Therefore, the user must write a "1" to this bit in order to set the width of the Receive Utopia (and the Transmit Utopia) data bus to 16 bits.

3. This option is only available for the 160 pin packaged version of the XR-T7234 device. The 100 pin device only contains 8 Receive Utopia Data Bus pins: RxData[7:0]. RxData[15:8] are not available in the 100 pin version.

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

The UNI allows 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 to bit 3 (CellOf52Bytes) within the Utopia Configuration Register, as depicted below.

Utopia Configuration Register: (Address = 7Ch)							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused		Hand-shake Mode	M-PHY	CellOf52 Bytes	TRFCDepth[1, 0]		UtWidth16
RO	RO	R/W	R/W	R/W	R/W	R/W	

The following table specifies the relationship between the value of this bit and the number of octets/cell that the Receive Utopia Interface block will process.

Table 33. The Relationship between the value of Bit 3 (CellOf52Bytes) within the Utopia Configuration Register, and the number of octets per cell that will be processed by the Transmit and Receive Utopia Interface blocks

CellOf52 Bytes	Number of Bytes/Cells
0	53 bytes when the Utopia Data Bus width is 8 bits. 54 bytes when the Utopia Data Bus width is 16 bits.
1	52 bytes, regardless of the configured width of the Utopia Data Bus

Note: This selection applies to both the Transmit Utopia and Receive Utopia interface blocks. Additionally, the shaded selection reflects the default condition upon power up or reset.

An Advisory to Users

The user must insure that the ATM Layer processor only reads in (from the Receive Utopia Interface block) the "configured" number of octets per cell, following the latest assertion of the RxSoC output pin. If the ATM Layer processor continues to try to read-in more octets, it will end up reading in in-valid data.

7.3.2.1.4 Parity Checking Handling of Errored Cell Data received from the Receive Utopia Interface Block

The Receive Utopia Interface block will compute the odd parity of each byte (or word) of ATM cell data it places on the Receive Utopia Data bus. The Receive Utopia Interface block will also output the value of this parity bit via the RxPrty pin. The RxPrty pin will contain the odd parity value of the byte or word that is residing on the Receive Utopia Data bus.

The user has the option to configure the ATM Layer processor hardware and or software to use this feature.

7.3.2.2 Receive Utopia FIFO Manager

The Rx FIFO Manager has the following responsibilities.

- Monitoring the fill level of the RxFIFO, and alerting the ATM Layer processor anytime the RxFIFO contains cell data that needs to be read.
- Detecting and discarding "Runt" cells and insuring that the RxFIFO can resume normal operation following the removal of the "Runt" cell.
- Insuring that the RxFIFO can respond properly to an "Overrun" condition, by generating the "RxFIFO Overrun Condition" interrupt, discarding the resulting "Runt" or errored cell, and resuming proper operation afterwards.

Receive Utopia FIFO Manager Features and Options

This section discusses the numerous features that are provided by the Receive Utopia FIFO Manager. Additionally, this section discusses how the user can optimize these features to suit his/her application needs.

The Receive Utopia FIFO Manager provides the user with the following options.

- Handshaking Mode (Octet Level vs Cell Level)
- Resetting the Rx FIFO
- Monitoring the Rx FIFO

7.3.2.2.1 Selecting the Handshaking Mode (Octet Level vs Cell Level)

The Receive Utopia Interface block offers two different data flow control modes for data transmission between the ATM Layer processor and the UNI IC. These two modes are: "Octet- Level" Handshaking and "Cell- Level" Handshaking; as specified by the Utopia Level 2, Version 8 Specifications, and are discussed below.

7.3.2.2.1.1 Octet- Level Handshaking

The UNI will be operating in the Cell- Level Handshaking Mode following power up or reset. Therefore, the user will have to set bit 5 (Handshake Mode), within the Utopia Configuration Register to "0" in order to configure the UNI into the "Octet- Level" Handshake Mode. The main signal that is responsible for data- flow control between the ATM Layer processor and the Receive Utopia Interface block is the RxClav output pin.

When the UNI is operating in the Octet- Level Handshake mode, the Receive Utopia Interface block will assert the RxClav output pin, when the Rx FIFO contains at least one "read cycle's" worth of ATM Cell Data. In other words, if the Utopia Data bus width is configured to be 16 bits wide, then the RxClav signal will be asserted when the RxFIFO contains at least two bytes of cell data. Likewise, if the Utopia Data bus width is configured to be 8 bits wide, then the RxClav signal will be asserted when the RxFIFO contains at least one byte of ATM cell data. The

Receive Utopia Interface block will negate RxClav when the Rx FIFO has been depleted of any data. Therefore, the RxClav pin exhibits a role that is similar to a "Ready Ready" indicator in RS-232 based data transmission systems.

The ATM Layer processor is expected to monitor the state of the RxClav pin very closely (either in a tightly polled or interrupt driven approach). The ATM Layer processor is also expected to respond very quickly to the assertion of RxClav and read out the cell data in order to avoid an "Overrun Condition" in the Rx FIFO. Finally, the ATM Layer processor is expected to do one of two things, whenever RxClav toggles "low".

1. Quickly halting its reading of data from the Receive Utopia data bus.
2. Or, "validate" each byte or word of ATM cell data that it reads from the Receive Utopia Data bus; by checking the level of the RxClav signal. In this case, the ATM Layer processor must have the ability to internally remove any ATM cell data bytes or words that have been read in, after RxClav has toggled "low".

Figure 81 presents a timing diagram illustrating the behavior of the RxClav pin during reads from the Receive Utopia Interface block, while operating in the Octet-Level Handshaking Mode.

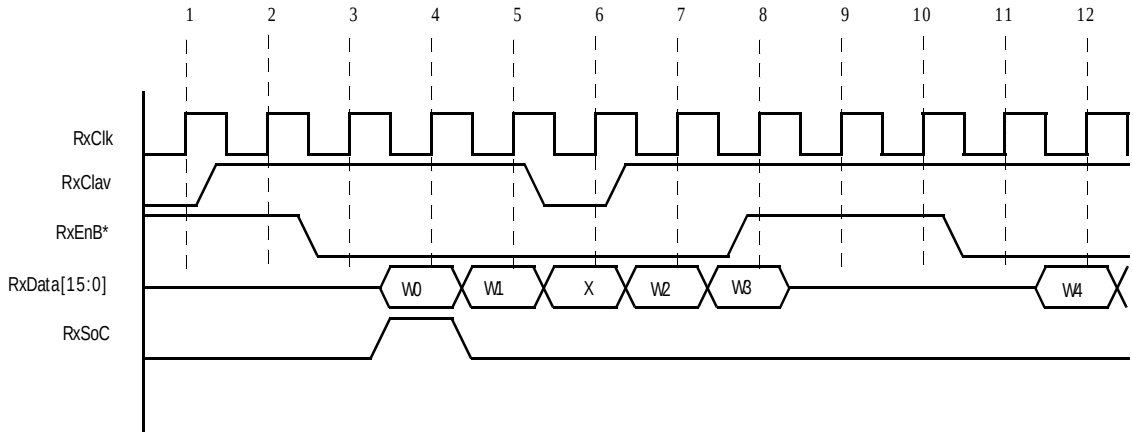

Figure 81. Timing Diagram of RxClav/ RxEmptyB and various other signals during reads from the Receive Utopia, while operating in the Octet-Level Handshaking Mode.

Notes regarding Figure 81

1. 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).
2. The Receive Utopia Interface block is configured to handle 54 bytes/cell. Hence, Figure 81 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.

In Figure 81, RxClav is initially "low" during clock edge #1. However, shortly after clock edge 1, the RxFIFO receives ATM cell data from the Receive Cell Processor block. At this point, the RxClav signal toggles "high" indicating that the RxFIFO contains at least one "read-cycle" worth of cell data. The ATM Layer processor will detect this "assertion of RxClav" during clock edge #2. Consequently, in order to begin reading this cell data, the ATM Layer processor will then assert the RxEnB* input pin. At clock edge #3, the Receive Utopia Interface block detects RxEnB* being

"low". Hence, the Receive Utopia Interface block then places word W0 on the Receive Utopia Data bus. The ATM Layer processor latches and reads in W0, upon clock edge #4. In this figure, shortly after the ATM Layer processor has read in word W1 (at clock edge #5), the RxFIFO is depleted which causes RxClav to toggle "low". In this figure, the ATM Layer processor will keep the RxEnB* signal asserted, and will read in an "invalid" word which is denoted by the "X" in Figure 81. Shortly thereafter, the RxFIFO receives some additional cell data from the Receive Cell Processor, which in turn causes RxClav to toggle "high". The ATM Layer processor then continues to read in words W2 and W3. Afterwards, the ATM Layer processor is unable to continue reading the ATM cell data from the Receive Utopia Interface block; and subsequently negates the RxEnB* signal; at clock edge #8. The Receive Utopia Interface block detects that RxEnB* is "high" at clock edge #8, and in turn, tri- states the Receive Utopia Data Bus at around clock edge # 9. Finally, prior to clock edge #11, the ATM Layer processor is able to resume reading in ATM cell data from the Receive Utopia Interface block, and indicates this fact by asserting the RxEnB* (e.g., toggling it "low"). The Receive Utopia Interface block detects this state change at clock edge #11 and subsequently places word W4 on the Receive Utopia Data bus.

7.3.2.2.1.2 Cell Level Handshaking

The UNI will be operating in the "Cell- Level" Handshaking mode following power up or reset. In the "Cell- Level" Handshaking mode, when the RxClav 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 RxClav toggles from "high" to "low", it indicates that Rx FIFO contains less than one complete ATM cell. As in the "Octet- Level" Handshake mode, the ATM Layer processor is expected to monitor the RxClav output, and quickly respond and read the Rx FIFO, whenever the RxClav output signal is asserted.

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 7.3.2.2.2.

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.

Utopia Configuration Register: (Address = 7Ch)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	Handshake Mode	M- PHY	CellOf52				
Bytes	TRFCDepth[1, 0]	UtWidth16					
RO	RO	R/W	R/W	R/W	R/W	R/W	

The following table specifies the relationship between this bit and the corresponding Handshaking Mode.

Table 34. The Relationship between the contents of Bit 5 (Handshake Mode) within the Utopia Configuration Register, and the resulting Utopia Interface Handshake Mode

Value	Resulting Handshake Mode
0	The Utopia Interfaces operate in the cell level handshake mode.
1	The Utopia Interfaces operate in the octet level handshake mode.

Note:

1. The Handshake Mode selection applies to both the Transmit Utopia and Receive Utopia Interface blocks.
2. 2. Since Multi- PHY mode operation requires the use of "Cell- Level" Handshaking; this bit is ignored if the UNI is

operating in the Multi- PHY mode.

3. Finally, the UNI will be operating in the "Cell- Level" Handshaking Mode upon power up or reset. Therefore, the user must write a "0" to this bit in order to configure "Octet- Level Handshaking" mode.

Figure 82 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" Handshake Mode.

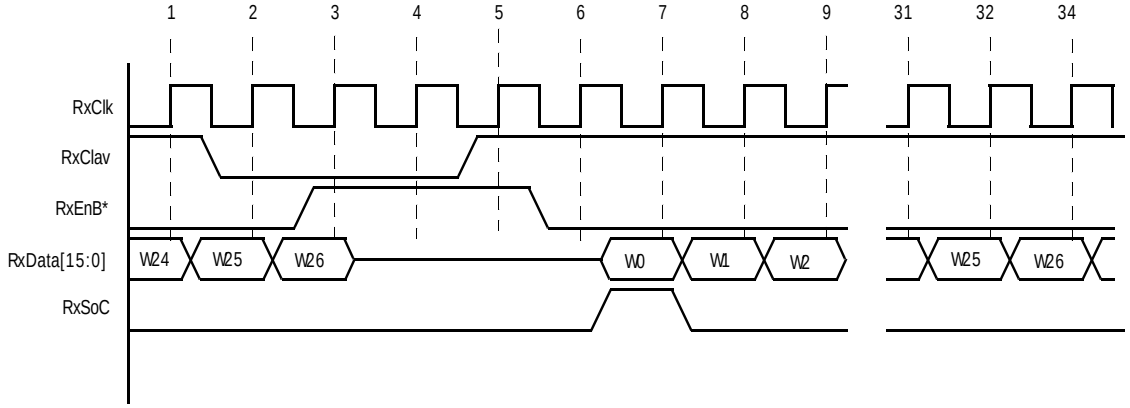

Figure 82. Timing Diagram of various Receive Utopia Interface block signals, when the Receive Utopia Interface block is operating in the "Cell Level" Handshake Mode.

Notes regarding Figure 82:

1. 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.
2. The Receive Utopia Interface block is configured to handle 54 bytes/ cell. Hence, Figure 82 illustrates the ATM Layer processor reading in 27 words (W0 through W26) for each ATM cell.

In Figure 82, 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 RxClav 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 UNI (e.g., words W25 and W26). Afterwards, the ATM Layer processor will negate the RxEnB* signal and will cease to read in anymore ATM cell data from the Receive Utopia Interface block; until RxClav 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 RxClav signal. The ATM Layer processor detects that the RxClav signal has toggled "high" at clock edge #5. Consequently, the ATM Layer processor then asserts the RxEnB* signal (e.g., toggles it "low") after clock edge #5. The Receive Utopia Interface block detects the fact that the RxEnB* 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.

7.3.2.2.1.3 Resetting the Rx FIFO via Software Command

The UNI allows the user to reset the Rx FIFO, via Software Command, without the need to implement a master reset of the entire UNI device. This can be accomplished by writing the appropriate data to bit 6 (Rx FIFO Reset) of

the Receive Utopia Interrupt Enable/ Status Register as depicted below.

Receive Utopia-Interrupt Enable/ Status Register (Address-7Dh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	Rx FIFO Reset	Rx FIFO Overrun Interrupt Enable	Unused	RCCCA Interrupt Enable	Rx FIFO Overrun Interrupt Enable	Unused	Rx FIFO CCCA Int. Status
R/O	R/W	R/W	RO	R/W	RUR	RO	RUR

Once the user has reset the RxFIFO, then the contents of the Rx FIFO will be "flushed" and the Receive FIFO Status register will reflect the "RxFIFO Empty" status.

7.3.2.2.1.4 Monitoring the Rx FIFO Status

The local µP has the ability to poll and monitor the status of the Rx FIFO via the Receive Utopia FIFO Status Register. The bit format of this register is presented below.

Rx UT FIFO Status Register (Address = 7Fh)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused						RxFIFO Full	RxFIFO Empty
RO	RO	RO	RO	RO	RO	RO	RO

The following tables define the values for Bits 1 and 0 and the corresponding meaning.

RxFIFO Full

RxFIFO Full (Bit 1)	Meaning
0	Rx FIFO is not full.
1	Rx FIFO is full, and if the next operation by the ATM Layer processor is not a read operation, then the Rx FIFO could be overrun.

Rx FIFO Empty

RxFIFO Empty (Bit 0)	Meaning
0	Rx FIFO is not empty
1	Rx FIFO is empty.

7.3.2.2.2 Utopia Modes of Operation (Single PHY and Multi- PHY operation)

The UNI chip can support both Single- PHY and Multi- PHY operation. Each of these operating modes are discussed below.

7.3.2.2.2.1 Single PHY Operation

The UNI chip will be operating in the Multi- PHY mode upon power up or reset. Therefore, the user must write a "1" into Bit 4 of the Utopia Configuration Register; as depicted below, in order to configure the UNI into the "Single- PHY" Mode.

Utopia Configuration Register: Address = 7Ch							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused		Handshake Mode	M- PHY*/ S- PHY	CellOf52 Bytes	TRFIFODepth [1, 0]		UtWidth16
RO		R/ W	R/ W	R/ W	R/ W		R/ W
x	x	x	1	x	xx	x	

Writing a "1" to this bit- field configures the UNI to operate in the Single- PHY mode. Writing a "0" configures the UNI to operate in the Multi- PHY mode.

In Single- PHY operation, the ATM layer processor is pumping data into and receiving data from only one UNI device, as depicted in Figure 83. ATM Cell data is read from the Rx FIFO, via the Receive Utopia Data Bus, provided that the Receive Utopia Output enable signal (RxEnB) is low. The data on the Receive Utopia Data bus is updated on the rising edge of the Receive Utopia clock (RxClk). The Receive Utopia Interface block will pulse the Receive start of cell signal (RxSoC) when the first byte (or word) of a new cell is present on the Receive Utopia Data bus. Odd parity of the output byte or word is calculated and output at RxPrty pin.

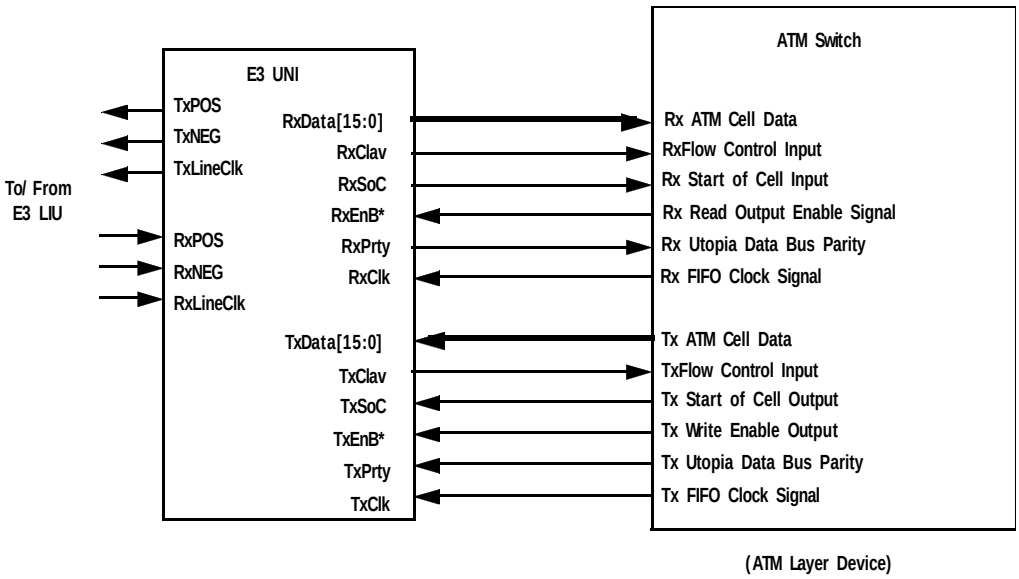

Figure 83. Simple Illustration of Single-PHY Operation

This section presents a detailed description of "Single- PHY" operation. Whenever the ATM Layer processor is responsible for receiving cell data from the Receive Utopia Interface block; it must do the following.

1. Check the level of the RxClav pin

If the RxClav 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 RxClav 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 RxClav toggles "high" before attempting to read any more cell data from the "Receive Utopia Interface block".

Note: The actual meaning associated with RxClav toggling "high" or "low" depends upon whether the UNI is operating in the "Cell Level" or "Octet Level" handshake modes.

2. Assert the RxEnB* 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 RxClav has toggled "high", then it should assert the RxEnB* input pin (e.g., toggling it "low"). Once the Receive Utopia Interface block has determined that the RxEnB* 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 RxSoC output pin (of the UNI IC) pulsing "high" for one clock period of RxClk.
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 RxPrty output pin.

This operation is optional, but should be done concurrently while checking for the assertion of the RxSoC 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.
- If the UNI is operating in the Octet- Level Handshake mode; then the ATM Layer processor should check the RxClav level prior to asserting the RxEnB* (Receive Utopia Interface-Output Enable) pin. The ATM Layer processor should only attempt to read the contents of the Receive Utopia Data Bus if the RxClav signal is "high".
- If the UNI is operating in the Cell- Level Handshake mode; then the ATM Layer processor should check the RxClav signal level just as it (the ATM Layer processor) is reading in the very last byte (or word) of a given cell. If the RxClav 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 RxClav 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 RxSoC 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 84, and in Timing Diagram form in

Figures 85 and 86.

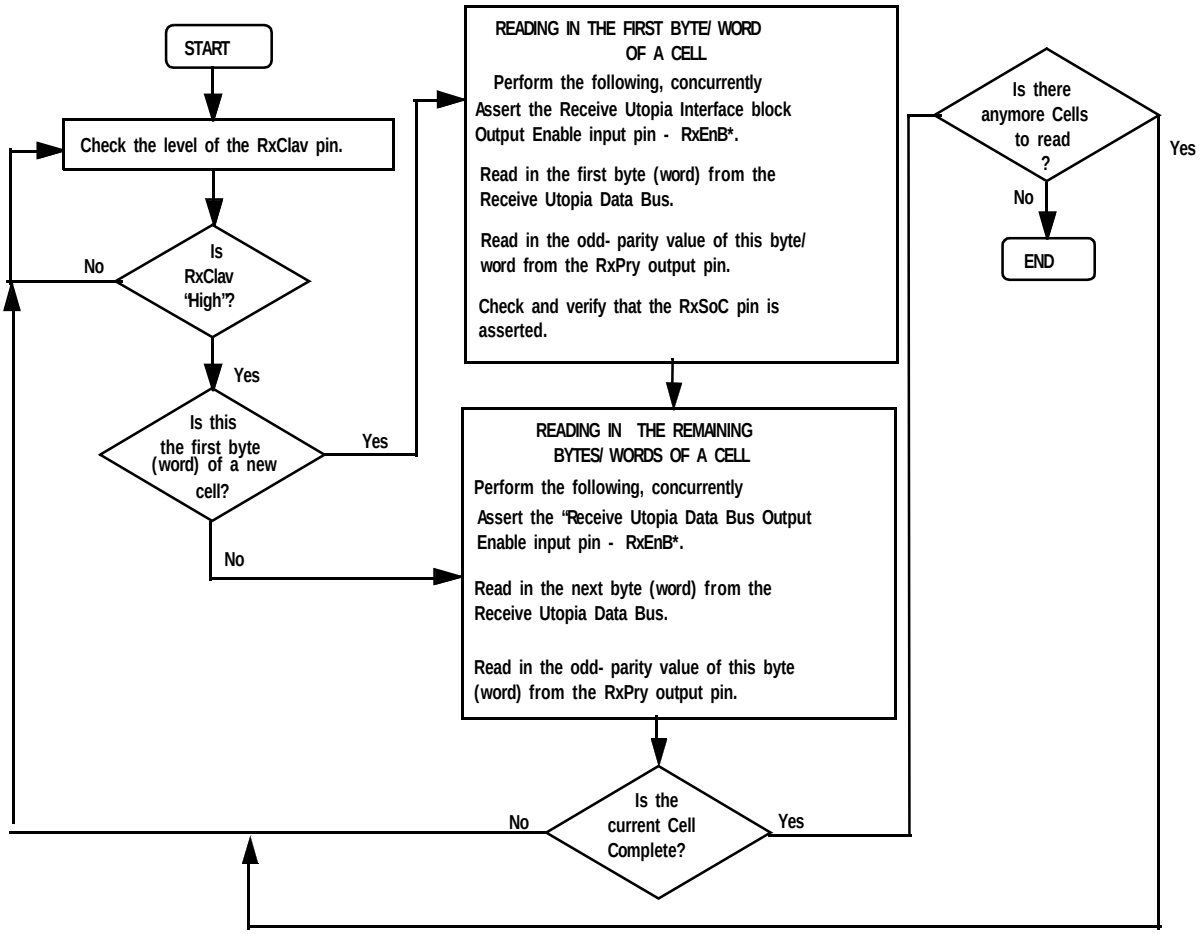

Figure 84. Flow Chart depicting the approach that the ATM Layer Processor should take when reading cell data from the Receive Utopia Interface, in the Single- PHY Mode.

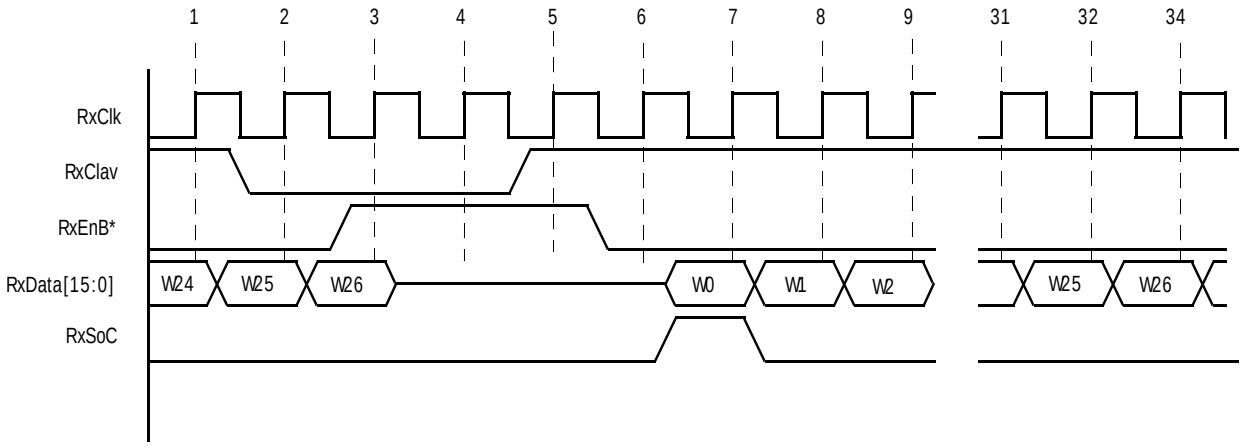

Figure 85. Timing Diagram of ATM Layer processor Receiving Data from the UNI over the Utopia Data Bus, (Single- PHY Mode/ Cell Level Handshaking).

Notes regarding Figure 85:

1. 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).
2. The Receive Utopia Data bus is configured to handle 54 bytes/ cell. Hence, Figure 85 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.
3. The Receive Utopia Interface block is configured to operate in the Cell Level Handshake mode.

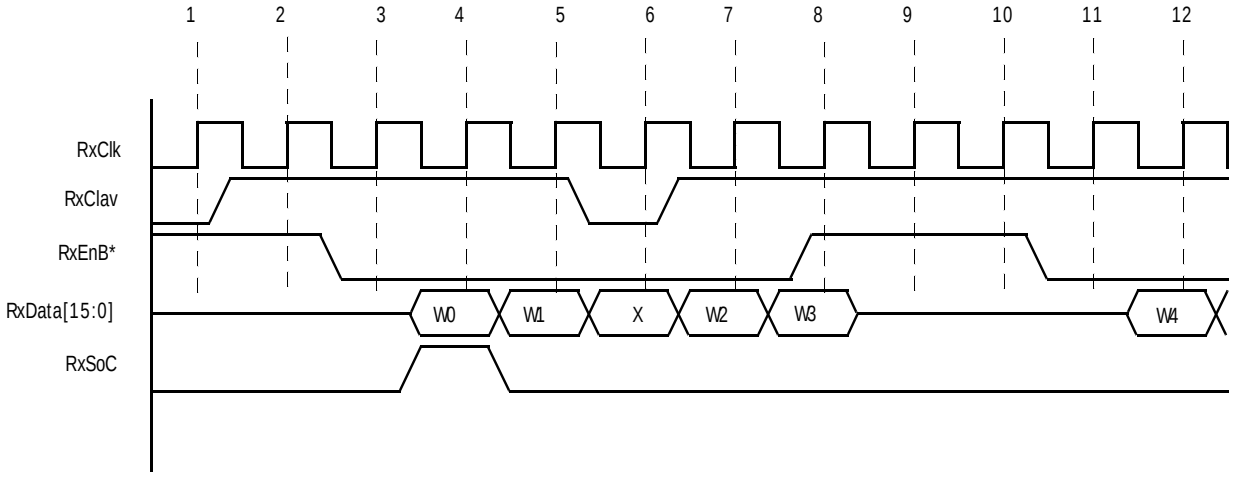

Figure 86. Timing Diagram of ATM Layer processor Receiving Data from the UNI over the Utopia Data Bus, (Single- PHY Mode/ Octet Level Handshaking).

Notes regarding Figure 86:

1. The Receive 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 Receive Utopia Interface block is configured to handle 54 bytes/cell. Hence, Figure 86 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.
3. The Receive Utopia Interface block is configured to operate in the Octet-Level Handshaking Mode.

Final Comments on Single-PHY Mode

The RxClav pin exhibits a role that is similar to the "Ready Ready" function in RS-232 based data communication. This pin is asserted when the RxFIFO contains ATM cell data that can be read by the ATM Layer processor. The RxClav pin will have a slightly different role when the UNI is operating in the Multi-PHY mode.

The UNI, while operating in Single-PHY mode, can be configured for either "Octet-Level" or "Cell Level" handshake modes. In either case, the ATM Layer Processor is expected to poll the RxClav pin before attempting to read in the next byte, word or cell from the RxFIFO.

7.3.2.2.2.2 Multi-PHY Operation

The UNI IC will be operating in the Multi-PHY mode upon power up or reset. In Multi PHY operating mode, the ATM layer processor may be pumping data into and reading data from several UNI devices in parallel. When the UNI is operating in Multi-PHY mode, the Receive Utopia Interface block will support two kinds of operations with the ATM Layer processor:

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

Each of these operations are 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 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 UNI device (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 that 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 "Utopia Interface block" is 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 "Rx Utopia Address Register" (Address = 7Eh) within its "host" UNI device. The bit-format of the "Rx Utopia Address Register" is presented below:

Rx Utopia Address Register (Address = 7Eh)							
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	0	0	0	0	0

Likewise, the user assigns a "Utopia address" value to a particular "Transmit Utopia Interface block", within one of the UNIs (in the "Multi- PHY" system) by writing this address value into the "Tx Utopia Address Register" (Address = 82h) within the "host" UNI device. The bit- format of the "Tx Utopia Address Register" is presented below.

Tx Utopia Address Register (Address = 82h)							
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	0	0	0	0	0

Note: The role of the Transmit Utopia Interface block, in Multi- PHY" operation is presented in Section 6.1.2.3.2.

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

When the UNI is operating in the "Multi- PHY" mode, the Receive Utopia Interface block will automatically be configured to support "polling". "Polling" allows an ATM Layer processor (which is interface to several UNI devices) to determine which UNIs contain ATM cell data that needs to be read, at any given time. The manner in which the ATM Layer processor "polls" its UNI devices follows.

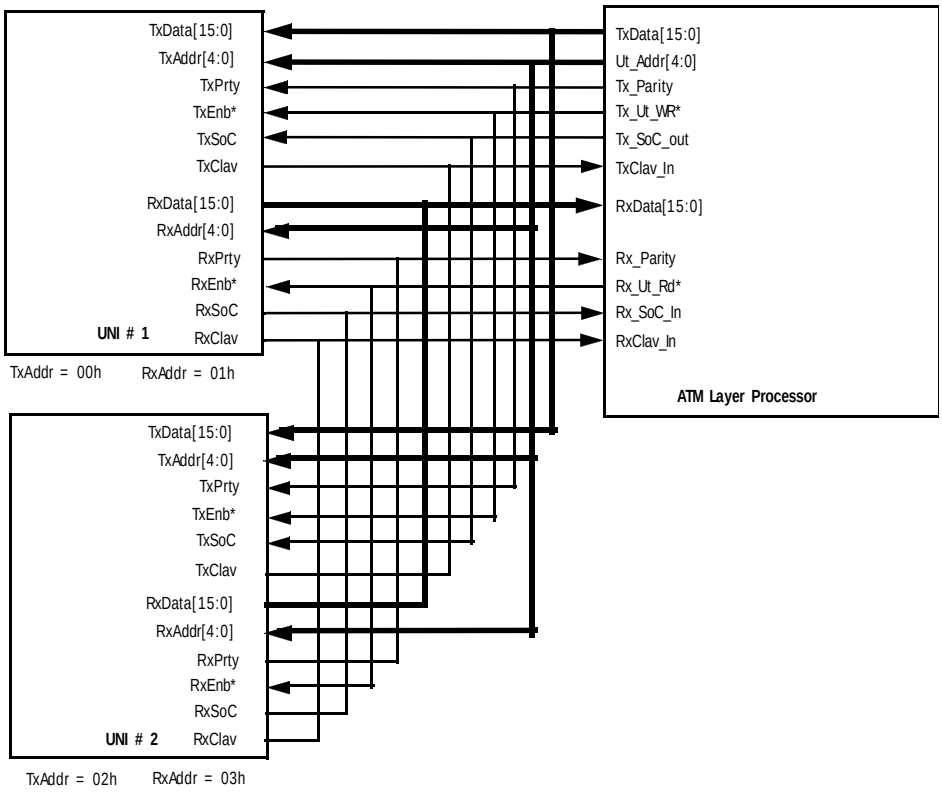

Figure 87. An Illustration of Multi- PHY Operation with UNI devices #1 and #2

Figure 87 depicts a "Multi- PHY" system consisting of an ATM Layer processor and two (2) UNI devices, 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, "Receive Utopia" Data Bus, a common TxClav line, a common RxClav line, as well as common TxEnB*, RxEnB*, TxSoC and RxSoC lines. The ATM Layer processor will also be

addressing the Transmit and Receive Utopia Interface block via a common "Utopia" address bus (Ut_Addr[4:0]). Therefore, the Transmit and Receive Utopia Blocks, of a given UNI must have different addresses; as depicted in Figure 87.

The Utopia Address values, that have been assigned to each of the Transmit and Receive Utopia Interface blocks, within Figure 87, are listed below in Table 36.

Table 35. Utopia Address values of the Utopia Interface Blocks illustrated in Figure 87

Block	Utopia Address Value
Transmit Utopia Interface block-UNI #1	00h
Receive Utopia Interface block-UNI #1	01h
Transmit Utopia Interface block-UNI #2	02h
Receive Utopia Interface block -UNI #2	03h

Recall, that the Receive Utopia Interface blocks were assigned these addresses by writing these values into the "Rx Utopia Address Register" (Address = 7Fh) within their "host" UNI device. The discussion of the Transmit Utopia Interface blocks, within UNIs #1 and #2 is presented in Section 6.1.2.3.2.1.

Polling Operation

Consider that the ATM Layer processor is currently reading a continuous stream of cells from UNI #1. While reading this cell data from UNI #1, the ATM Layer processor can also "poll" UNI #2 for "availability" (e.g., tries to determine if the RxFIFO, within UNI #2, contains some ATM cell data that needs to be read).

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 RxEnB* input pin (if it not asserted already).

The UNI device (being "polled") will know that this is only a "polling" operation, if the RxEnB* input pin is asserted, prior to detecting its Utopia Address on the "Utopia Address" bus.

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 "RxClav_in" input pin (see Figure 87).

The UNI devices role in the "polling" operation

UNI #2 will sample the signal levels placed on its Rx Utopia Address input pins (RxAddr[4:0]) on the rising edge of its "Receive Utopia Interface block" clock input signal, RxClk. 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 = 7Fh).

If these values do not match (e.g., RxAddr[4:0] ¼03h) then UNI #2 will keep its "RxClav" output signal "tri- stated"; and will continue to sample its "Receive Utopia Address bus input" pins, with each rising edge of RxClk.

If these two values do match (e.g., RxAddr[4:0] = 03h) then UNI #2 will drive its "RxClav" output pin to the appropriate level, reflecting its RxFIFO "fill status". Since the UNI is automatically operating in the "Cell Level Handshaking" mode, while it is operating in the "Multi- PHY" mode, the UNI will drive the RxClav output signal "high" if it contains

at least one complete cell of data that needs to be read by the ATM Layer processor. Conversely, the UNI will drive the "RxClav" output signal "low" if its RxFIFO is depleted, or does not contain at least one full cell of data.

When UNI #2 has been selected for "polling", UNI #1 will continue to keeps its "RxClav" output signal "tri- stated". Therefore, when UNI #2 is driving its "RxClav" output pin to the appropriate level; it will be driving the entire "RxClav" line, within the "Multi- PHY" system. Consequently, UNI#1 will also be driving the "RxClav_in" input pin of the ATM Layer processor (see Figure 87).

If UNI #2 drives the "RxClav" line "low", upon the application of its address on the Utopia Address bus, then the ATM Layer processor will "learn" that UNI #2 does not contain any ATM cell data that is ready to be read. However, if UNI #2 drives the RxClav line "high" (during "polling"), then the ATM Layer processor will know that UNI#2 contain at least one cell of data that needs to be read.

Figure 88 presents a timing diagram, that depicts the behavior of the ATM Layer processor's and the UNI's signals during polling.

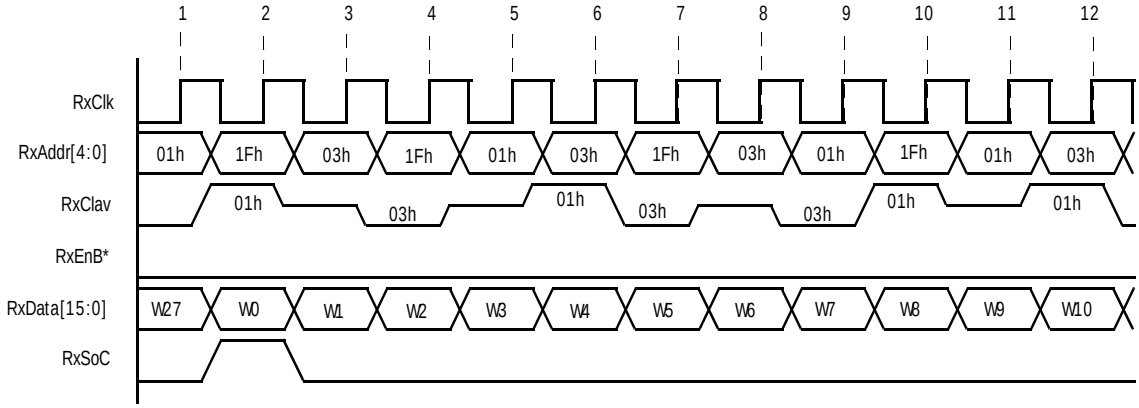

Figure 88. Timing Diagram illustrating the Behavior of various signals from the ATM Layer processor and the UNI, during Polling.

Notes regarding Figure 88:

1. 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).
2. The Receive Utopia Interface block is configured to handle 54 bytes/ cell. Hence, Figure 88 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.
3. The ATM Layer processor is currently reading ATM cell data from the Receive Utopia Interface block, within UNI #1 (RxAddr[4:0] = 01h) during this "polling process".
4. The RxFIFO, within UNI#2's Receive Utopia Interface block (RxAddr[4:0] = 03h) is either depleted or does not contain enough data to constitute a complete ATM cell. Hence, the RxClav line will be driven "low" whenever this particular Receive Utopia Interface block is "polled".
5. The Receive Utopia Address of 1Fh is not associated with any UNI device, within this "Multi- PHY" system. Hence, the RxClav line is tri- stated whenever this address is "polled".

Note: Although Figure 87 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 6.1.2.3.2.1 for a discussion on the Transmit Utopia Interface block during Multi- PHY operation.

7.3.2.2.2.2.2 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 RxEnB* 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 RxEnB* signal.
4. Check and insure that the RxSoC 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 89 presents a flow- chart that depicts the "UNI Device Selection and Read" process in Multi- PHY operation.

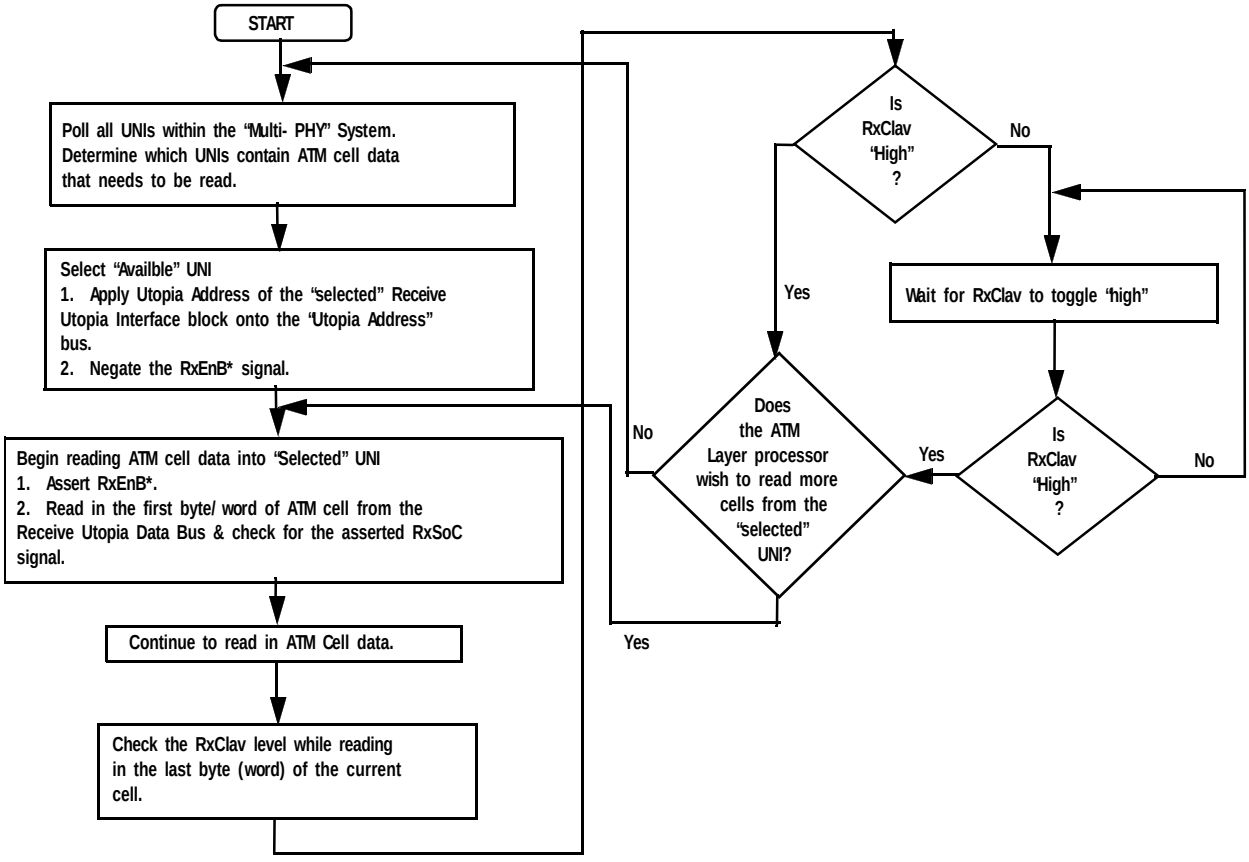

Figure 89. Flow- Chart of the "UNI Device Selection and Read Procedure" for the Multi- PHY Operation.

Figure 90 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.

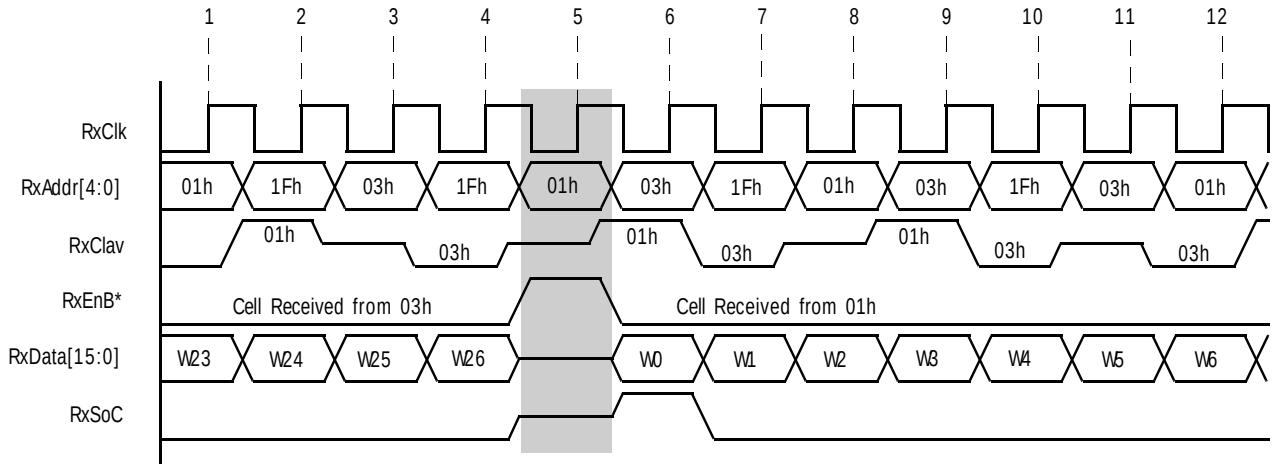

Figure 90. Timing Diagram of the Receive Utopia Data and Address Bus signals, during the 'Multi- PHY" UNI Device Selection and Write Operations.

Notes regarding Figure 90:

1. 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).
2. The Receive Utopia Interface Block is configured to handle 54 bytes/ cell. Hence, Figure 90 illustrates the ATM Layer processor reading 27 words (e.g., W0 through W26) for each ATM cell.

In Figure 90, the ATM Layer processor is initially reading ATM cell data from the Receive Utopia Interface within UNI #2 (RxAddr[4:0] = 03h). However, the ATM Layer processor is also polling the Receive Utopia Interface block within UNI #1 (RxAddr[4:0] = 01h) and some "non- existent" device at RxAddr[4:0] = 1Fh. 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 (RxAddr[4:0] = 03h). 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 90). 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 (RxAddr[4:0] = 01h).
2. The RxEnB* 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.

7.3.2.3 Receive Utopia Interrupt Servicing

The Receive Utopia Interface block will generate interrupts upon the following conditions:

- Change of Cell Alignment (e.g., the detection of "runt" cells)
- Rx FIFO Overrun

If one of these conditions occur and if that particular condition is enabled for interrupt generation, then when the

local mP/ mC reads the UNI Interrupt status register, as shown below, it should read "0xxxxxx1b" (where the - b suffix denotes a binary expression, and the - x denotes a "don't care" value).

UNI Interrupt Status Register (Address = 05h)							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	One Sec Interrupt Status	Tx E3 Interrupt Status	Rx E3 Interrupt Status	Tx CP Interrupt Status	Rx CP Interrupt Status	Tx Utopia Interrupt Status	Rx Utopia Interrupt Status
RO	RUR	RO	RO	RO	RO	RO	RO
0	x	x	x	x	x	x	1

At this point, the local µC/ µP has determined that the Receive Utopia Interface block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly.

The next step in the interrupt service routine should be to determine which of the two Receive Utopia Block interrupt conditions has occurred and is causing the Interrupt. In order to accomplish this, the local µP/ µC should now read the "Rx UT Interrupt Enable/ Status Register, which is located at address 7Dh in the UNI device. The bit format of this register is presented below.

Address = 7Dh, Rx UT Interrupt Enable/ Status Register							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Unused	RxFIFO Reset	RxFIFO Overflw Interrupt Enable	Unused	RCCCA Interrupt Enable	RxFIFO Overflw Interrupt Status	Unused	RCCCA Interrupt Status
RO	R/ W	R/ W	RO	R/ W	RO	RO	RUR

The Rx UT Interrupt Enable/ Status Register has eight bit- fields. However, only four of these bit- fields are relevant to interrupt processing. Bits 0-2 are the interrupt status bits and bits 3-5 are the interrupt enable bits; for the Receive Utopia Interface block. Each of these "interrupt processing relevant" bit- fields are defined below.

Bit 0-RCOCA Interrupt Status-Receive Utopia Change of Cell Alignment Condition

If the RxFIFO Manager detects a "runt" cell, then it will generate the "Receive Utopia Change of Cell Alignment Condition" interrupt, and the "runt" cell will be discarded. The Receive Utopia Interface block will indicate that it is generating this kind of interrupt by asserting Bit 0 (RCOCA Interrupt Status) of the Receive Utopia Interrupt Enable/ Status Register, as depicted below.

Address = 7Dh, Rx UT Interrupt Enable/ Status Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	RxFIFO Reset	RxFIFO Overflw Interrupt Enable	Unused	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	Unused	RCOCA Interrupt Status
RO	R/W	R/W	RO	R/W	RO	RO	RUR
0	0	x	0	1	x	0	1

Bit 2-Rx FIFO Overflw Interrupt Status-RxFIFO Overrun Condition

If the RxFIFO is filled to capacity, and if the ATM Layer processor is unable to begin reading its contents, before the Receive Cell Processor writes another cell into the RxFIFO, some of the data within the RxFIFO will be overwritten, and in turn lost. If the Receive Utopia Interface block detects this condition, and if this interrupt condition has been enabled then the UNI will assert the INT* pin to the local µP/µC. Additionally, the UNI will set bit 2, within the Receive Utopia Interrupt Enable/ Status Register to "1" as depicted below.

Address = 7Dh, Rx UT Interrupt Enable/ Status Register							
Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Unused	RxFIFO Reset	RxFIFO Overflw Interrupt Enable	Unused	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	Unused	RCOCA Interrupt Status
RO	R/W	R/W	RO	R/W	RO	RO	RUR
0	0	1	0	x	1	0	x

Note: The user (or ATM Layer processor) must deplete this register before this interrupt condition will be reset. The user can deplete this register by commanding a "RESET" of the RxFIFO (e.g., writing "01xxxxxxb" to this register).

Bit 3-RCOCA Interrupt Enable-Receive Utopia Change of Cell Alignment Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disables the generation of interrupts due to a detected "Change of Cell Alignment" condition, within the RxFIFO. The local µP/µC can enable this interrupt by writing a "1" to this bit- field. Upon power up or reset conditions, this bit- field will contain a "0". Therefore the default condition is for this interrupt to be disabled.

Bit 5-RxFIFO Overflw Interrupt Enable-Rx FIFO Overrun Condition Interrupt Enable

This "Read/ Write" bit- field allows the user to enable or disable the generation of interrupts due to an "Rx FIFO Overrun" condition. The local µP/µC can enable this interrupt by writing a "1" to this bit- field. Upon power up or reset conditions, this bit- field will contain a "0". Therefore, the default condition is for this interrupt to be disabled.

8.0 TIMING DIAGRAMS

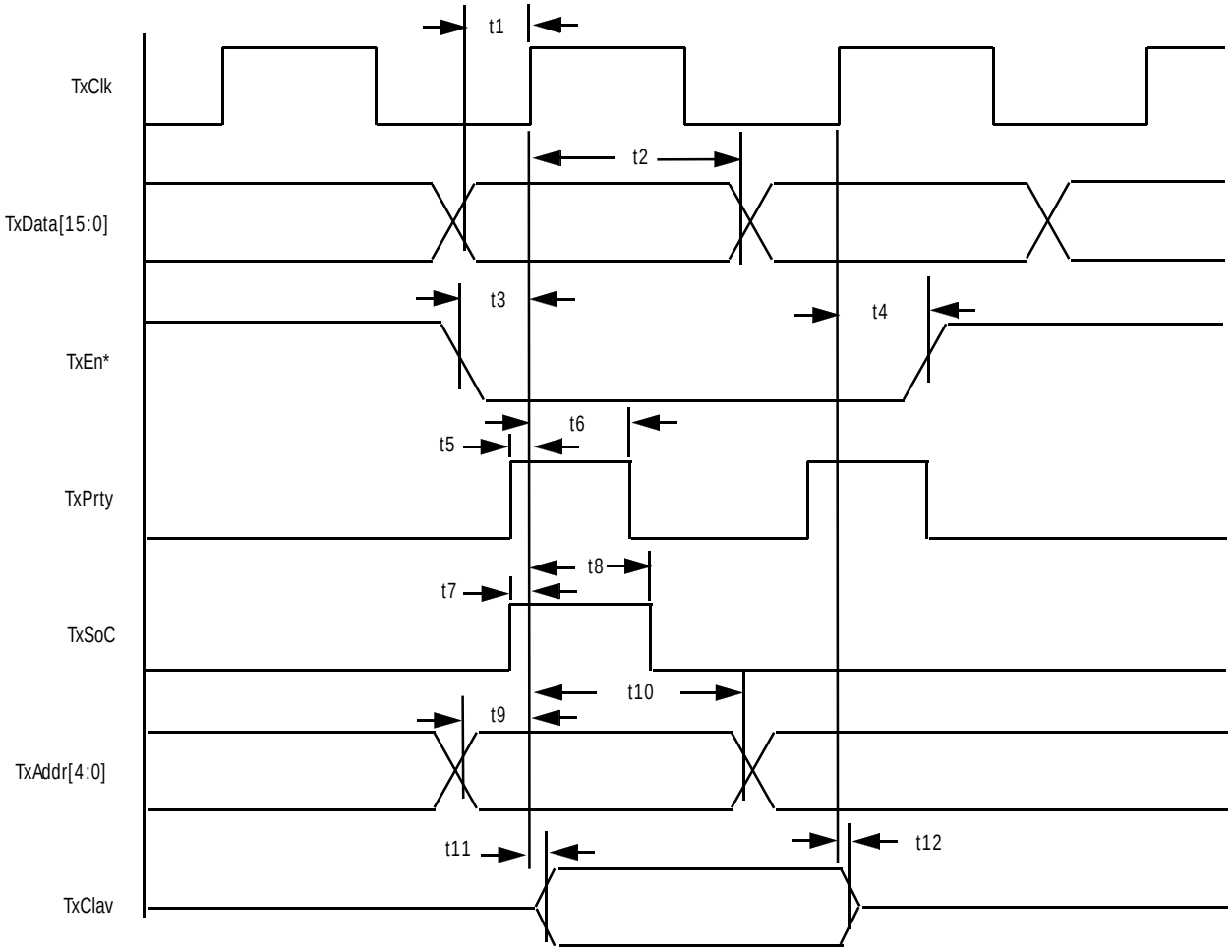

Figure 91. XRT7234 Transmit Utopia Interface Block Timing

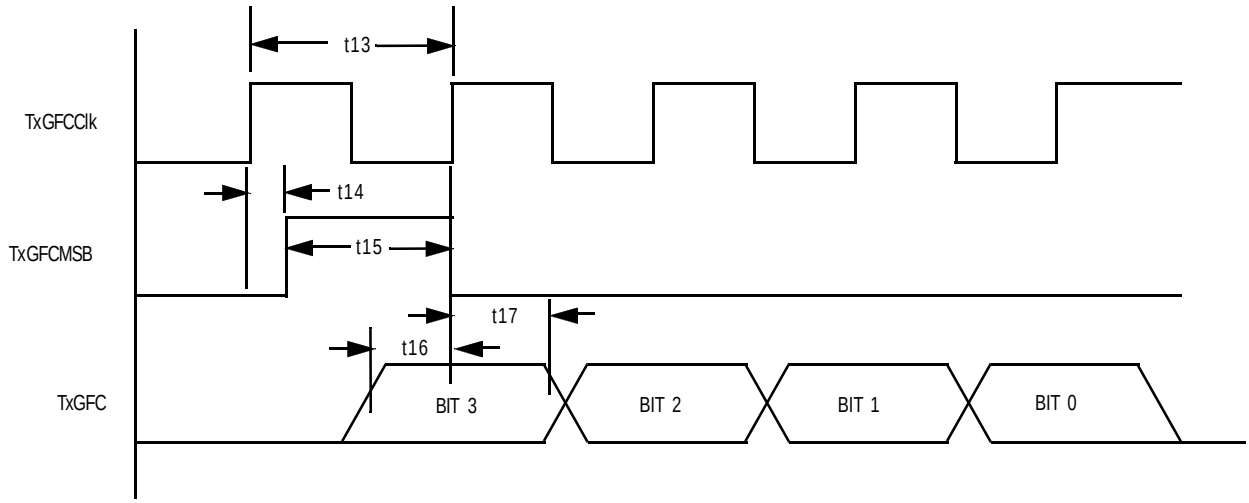

Figure 92. GFC Nibble- Field Serial Input Interface (at Transmit Cell Processor) Timing

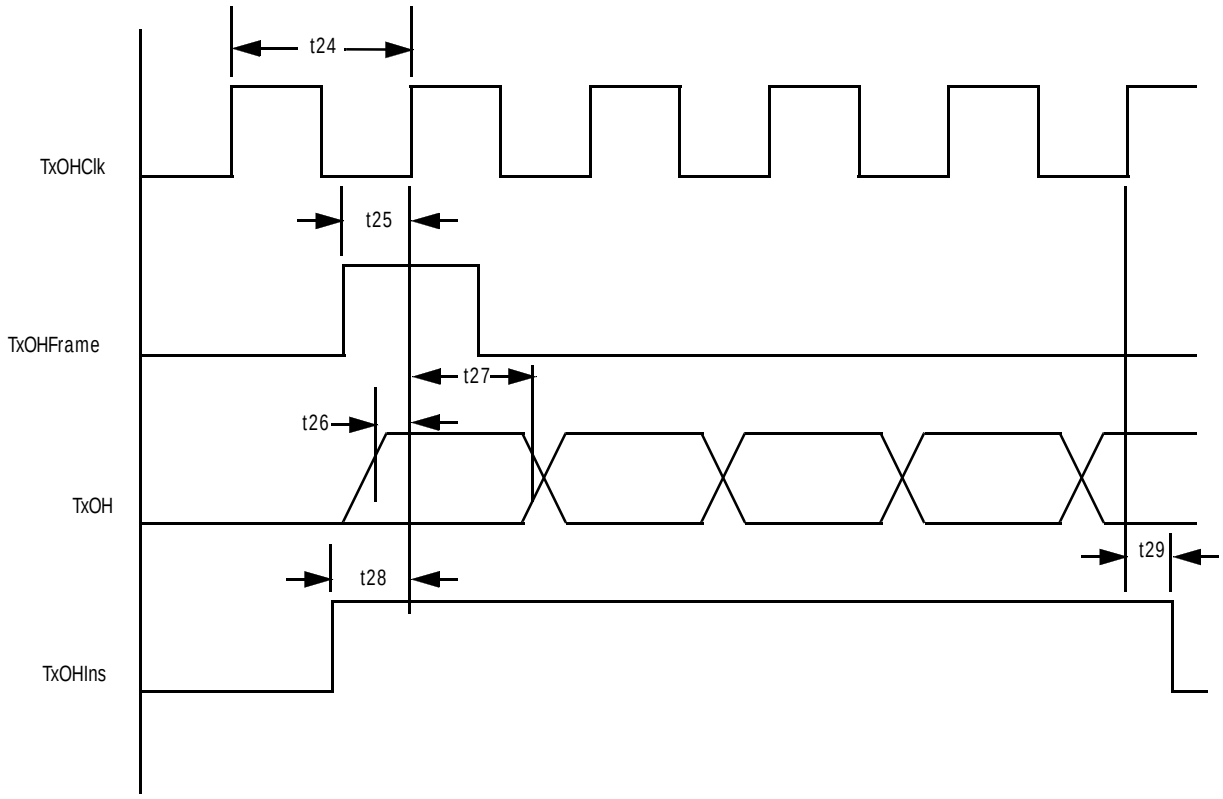

Figure 93. Transmit E3 Framer-GH Bit Serial Input Port Interface Timing

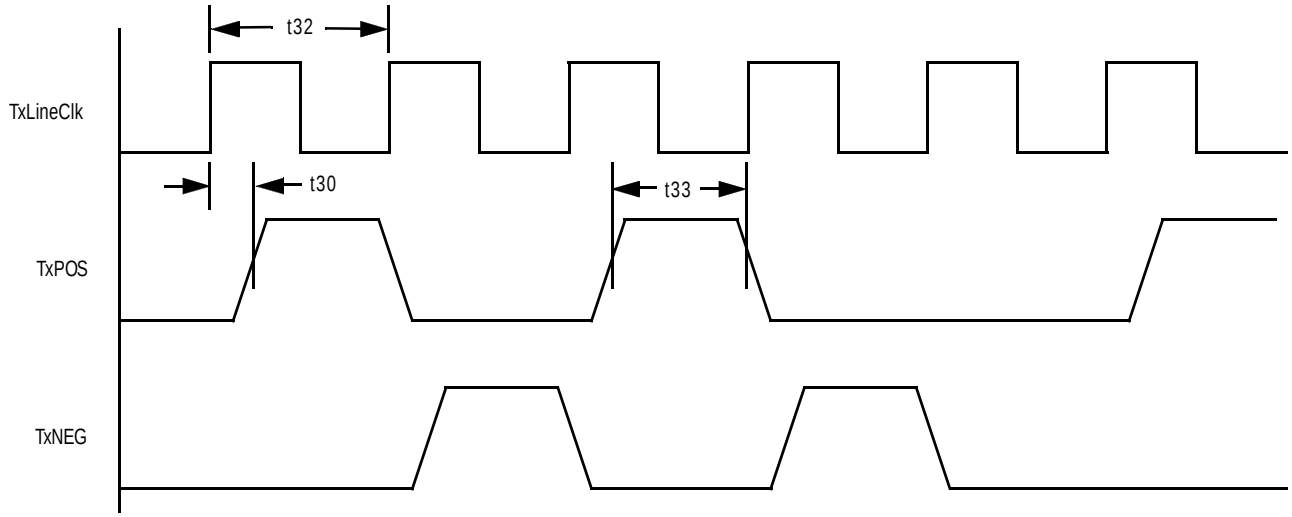

Figure 94. Transmit E3 Framer Line Interface Output Timing (TxPOS and TxNEG are updated on the rising edge of TxLineClk)

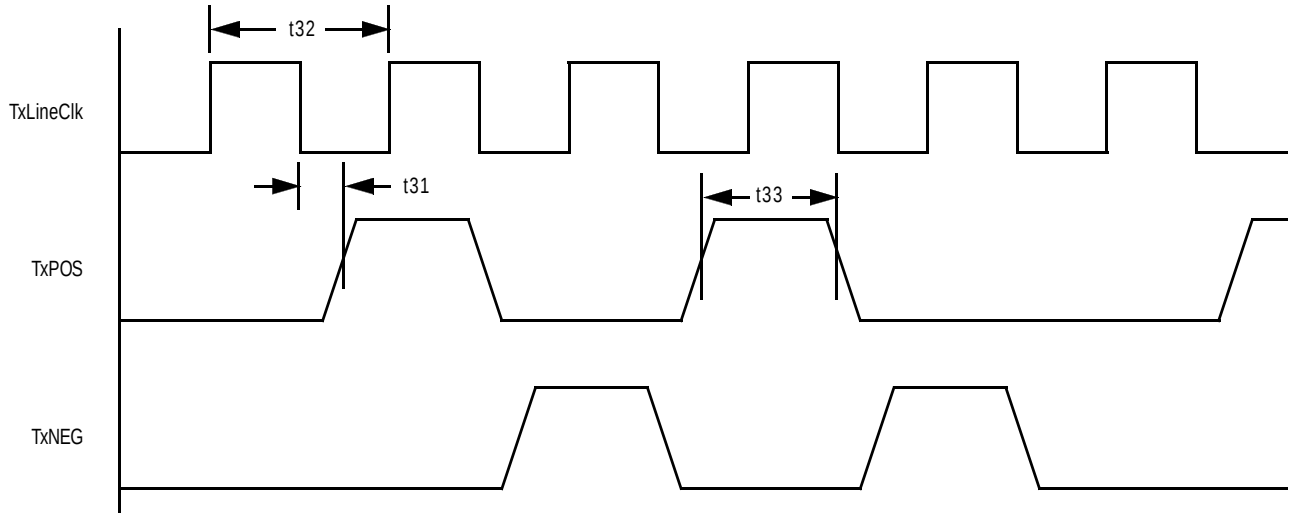

Figure 95. Transmit E3 Framer Line Interface Output Timing (TxPOS and TxNEG are updated on the falling edge of TxLineClk)

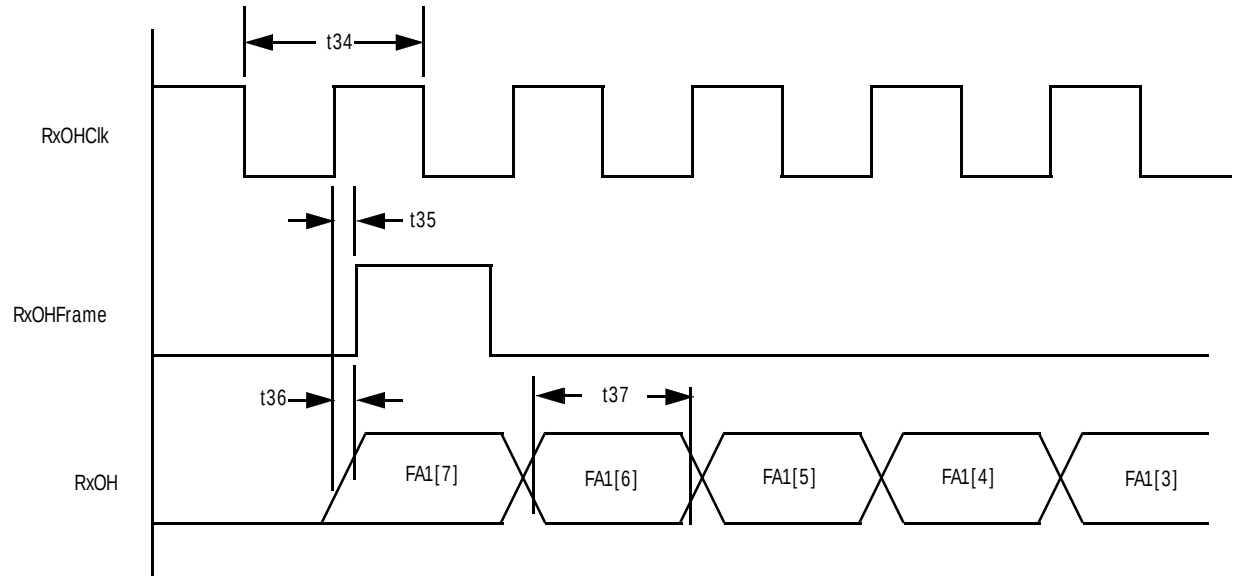

Figure 96. Receive E3 Framer-GH Bit Serial Output Port Interface Timing

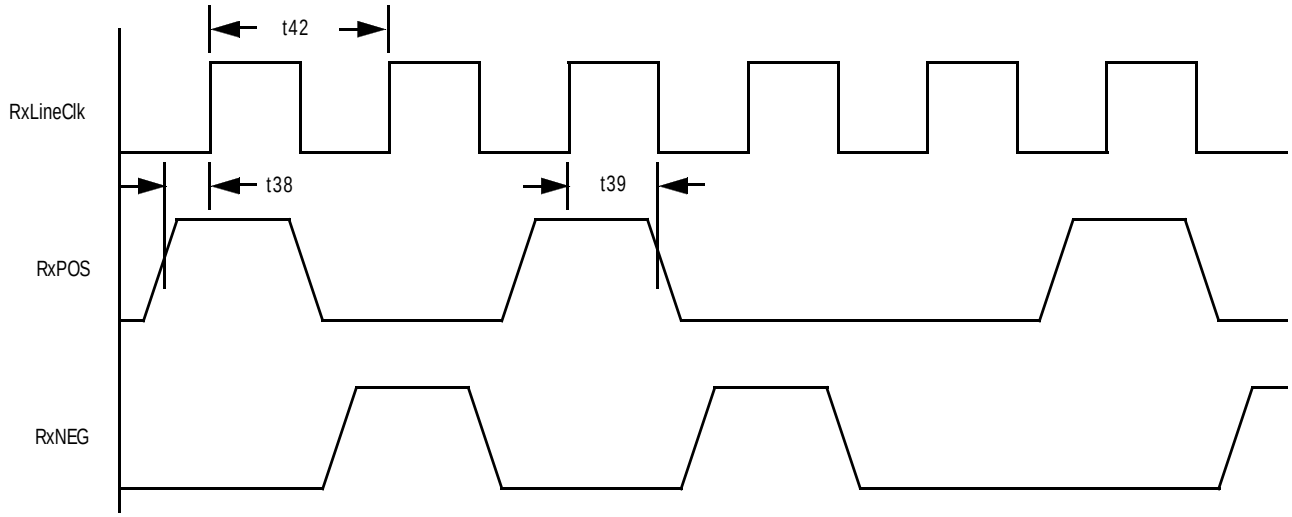

Figure 97. Receive E3 Framer Line Interface Input Signal Timing (RxPOS and RxNEG are sampled on rising edge of RxLineClk)

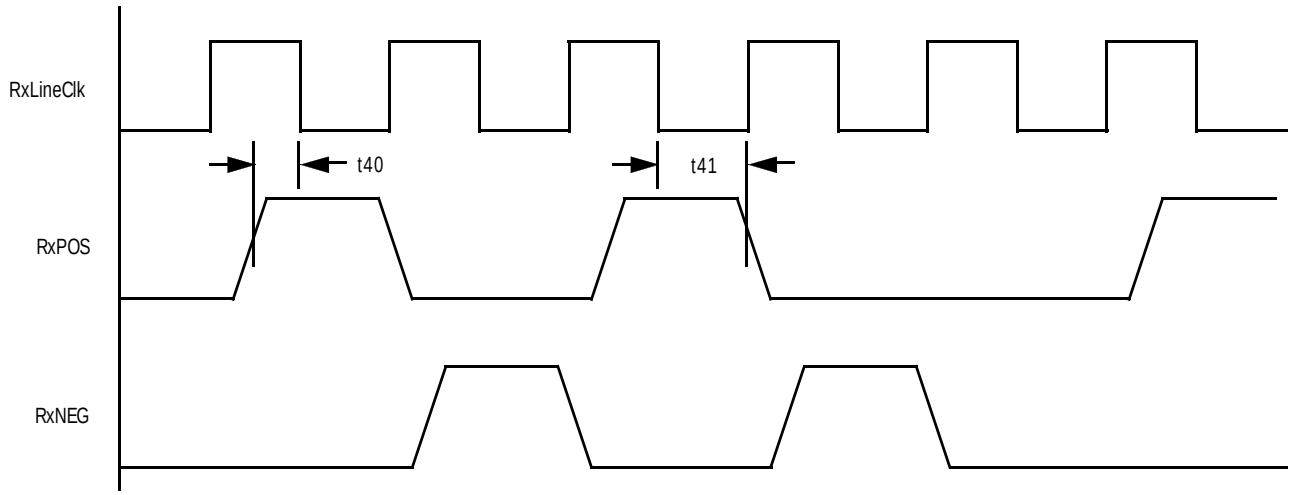

Figure 98. Receive E3 Framer Line Interface Input Signal Timing (RxPOS and RxNEG are sampled on the falling edge of RxLineClk)

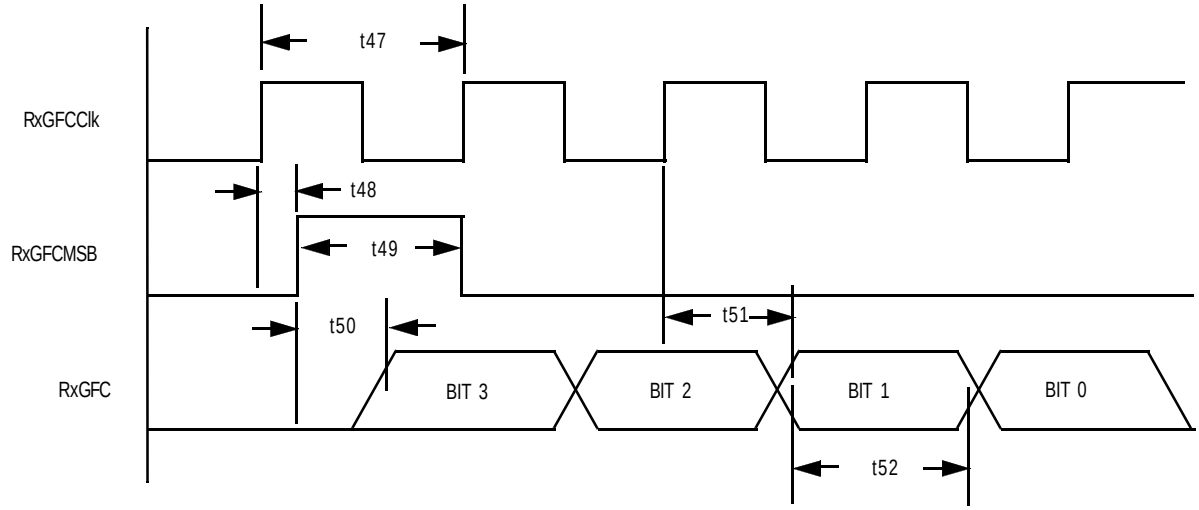

Figure 99. GFC Nibble- Field Serial Output Port Timing (Receive Cell Processor)

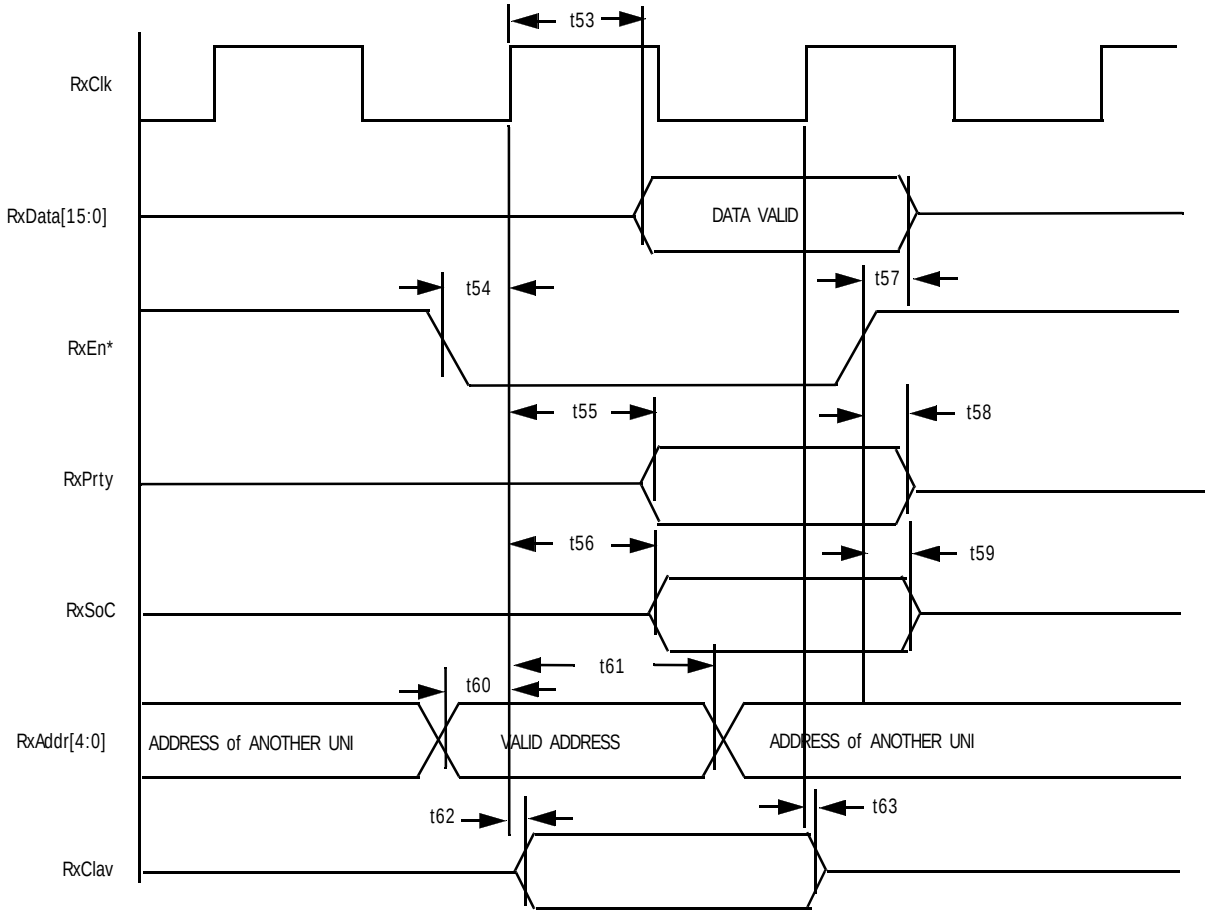

Figure 100. Receive Utopia Interface Block Timing

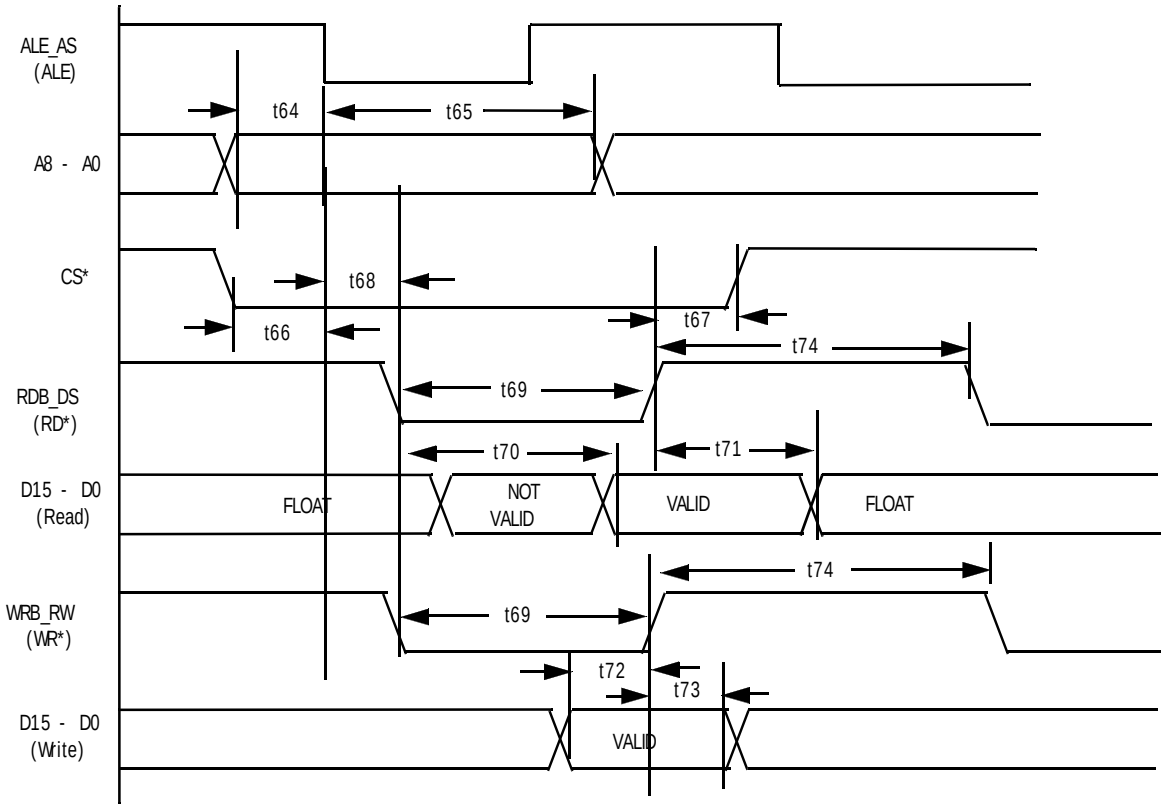

Figure 101. Microprocessor Interface Timing-Read and Write Operations

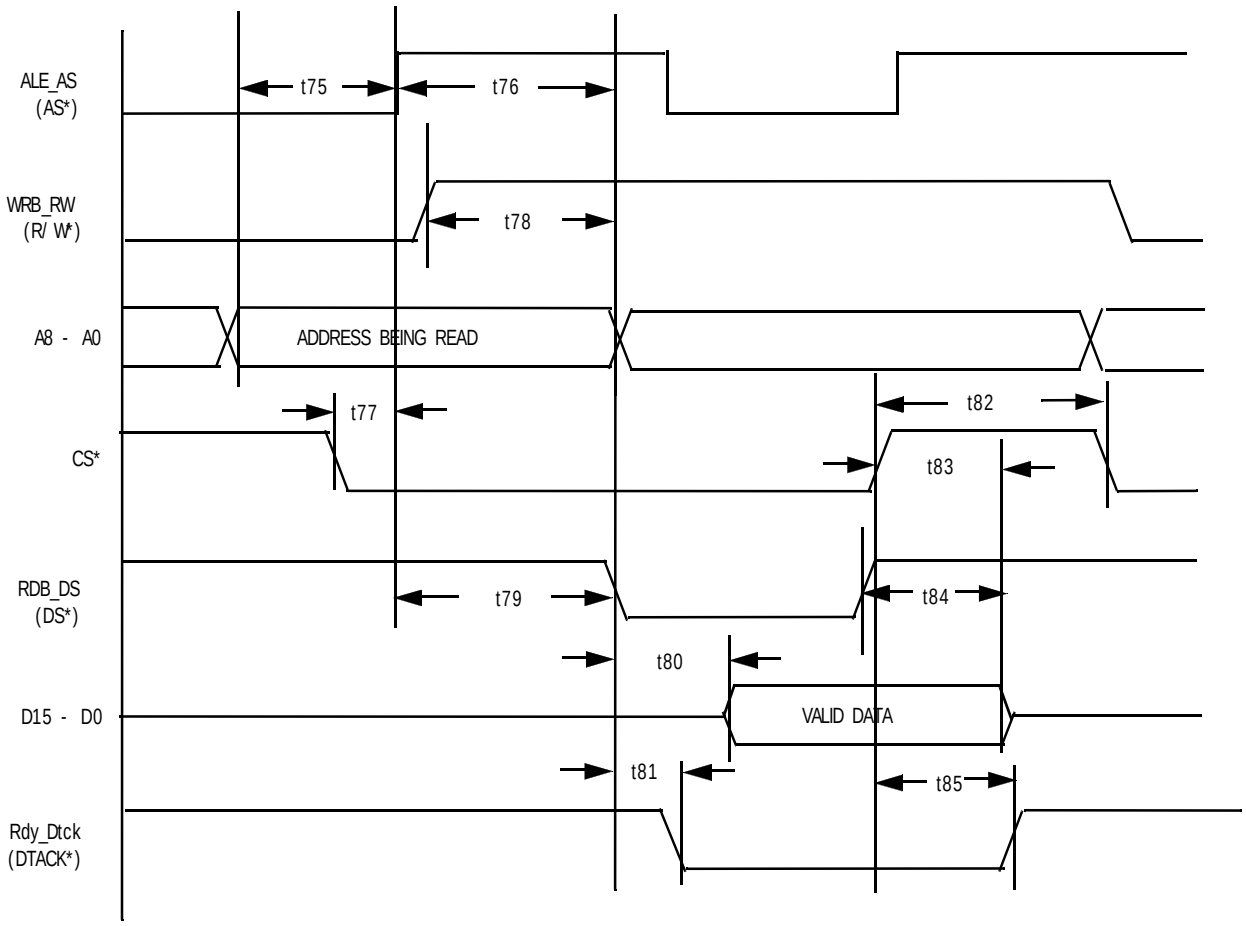

Figure 102. Intel Type Processors, Non- Burst Mode

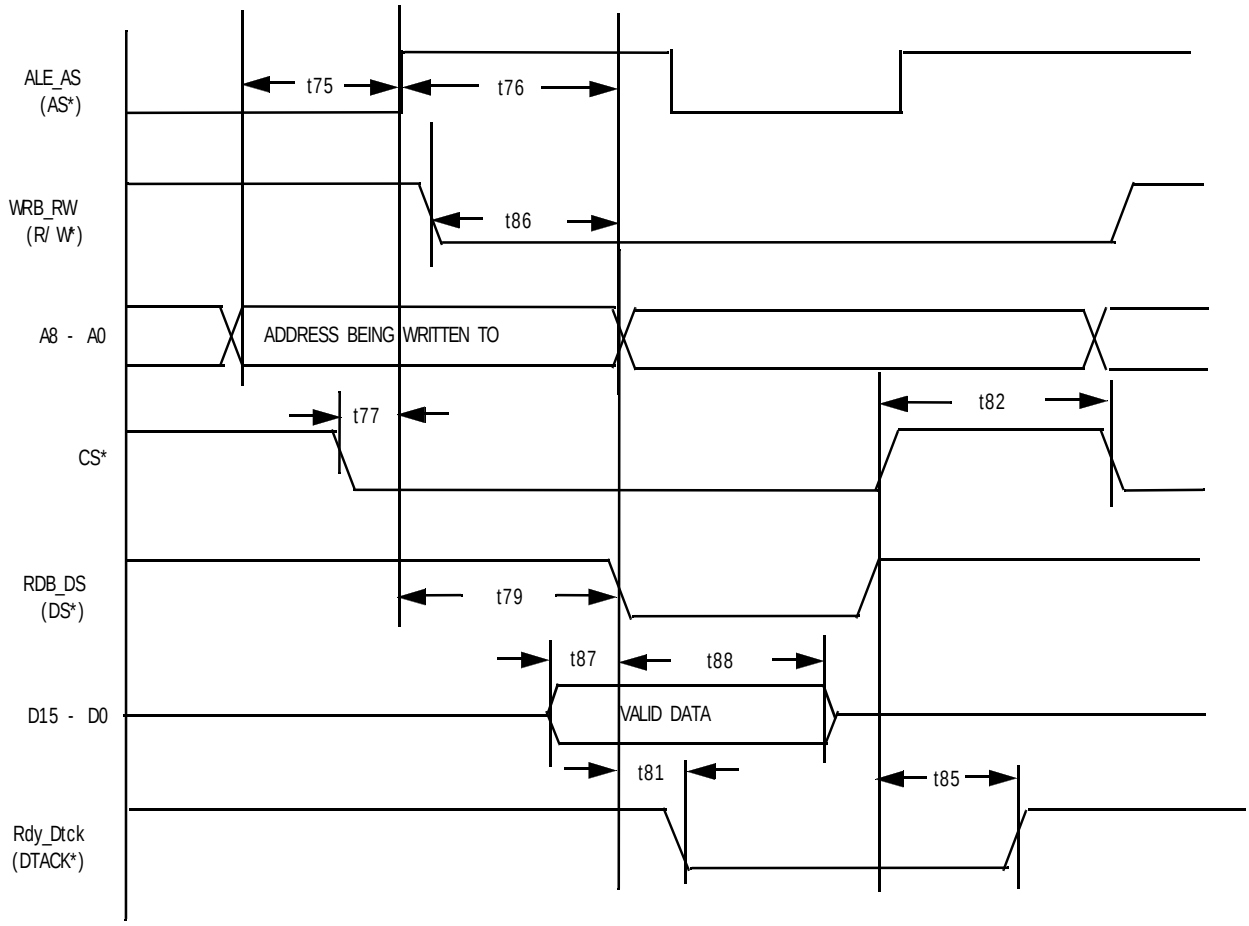

Figure 103. Microprocessor Interface Timing-Motorola Type Processors (Read Operations) Non- Burst Mode

Figure 104. Microprocessor Interface Timing-Motorola Type Processor (Write Operations) Non- Burst Mode

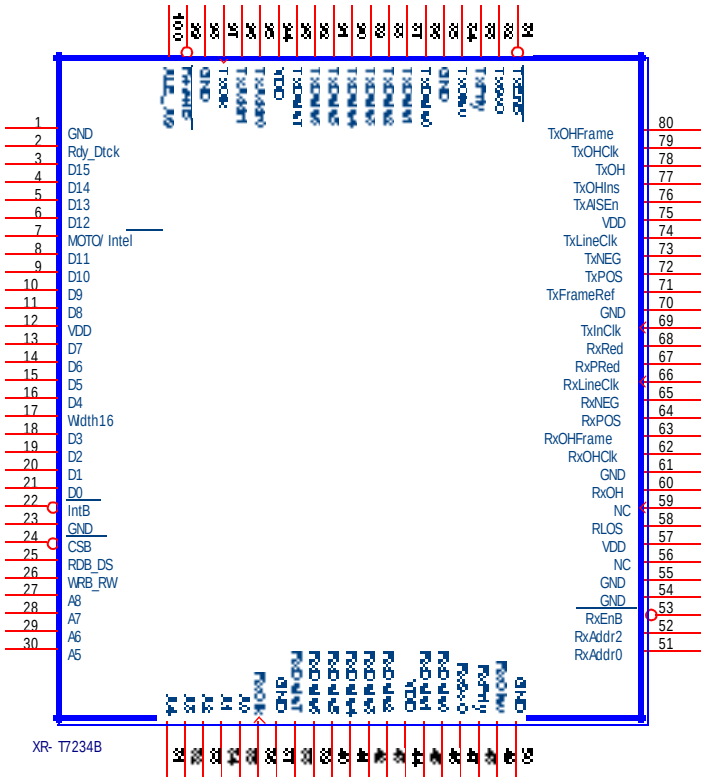

Figure 105. Pin out of the XR- T7234-100 Pin Package Version

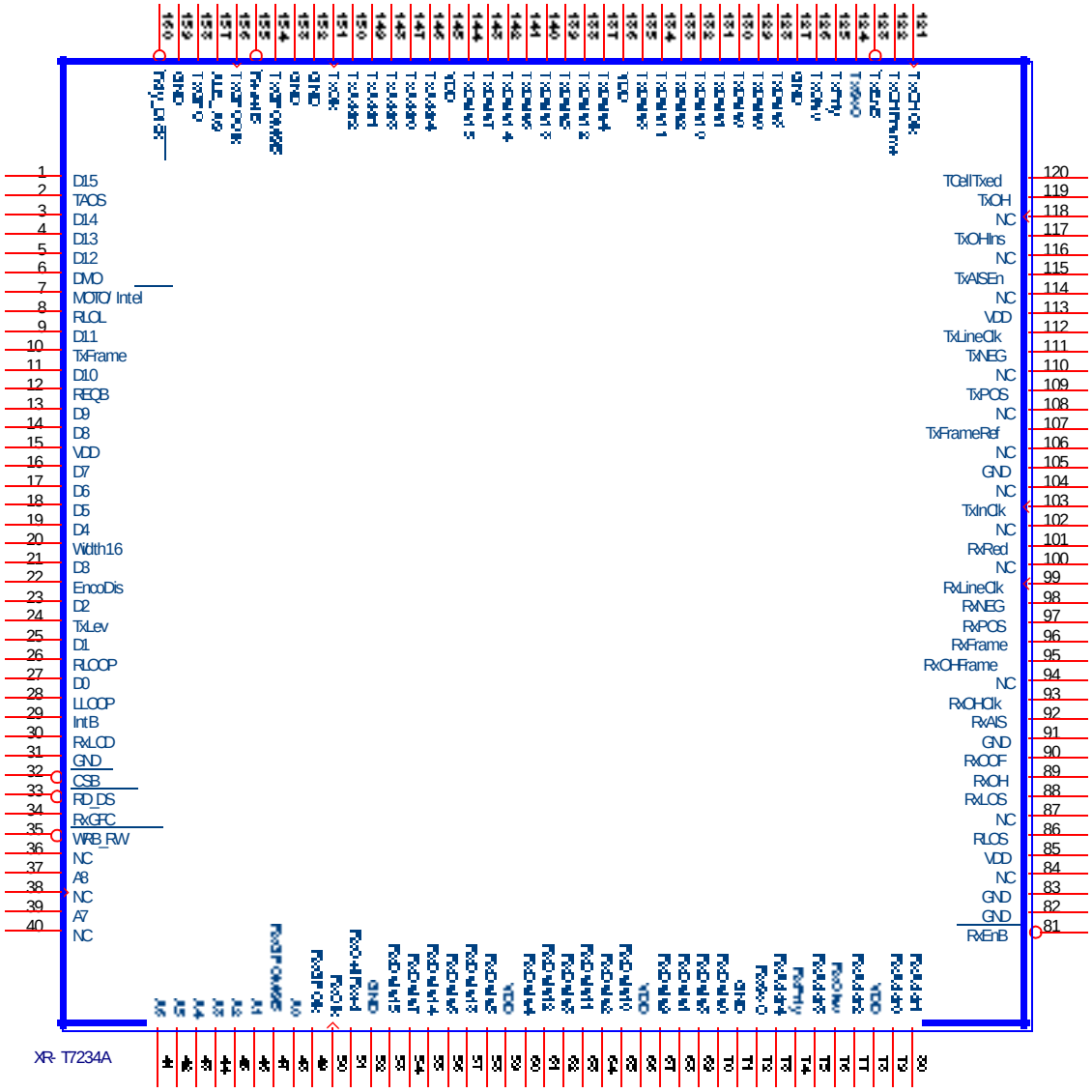

Figure 106. Pin out of the XR- T7234-160 Pin Package Version