4 CHANNEL, ATM UNI/PPP DS3/E3 FRAMING CONTROLLER

OCTOBER 2003 REV. P1.1.1

GENERAL DESCRIPTION

The XRT74L74 4 Channel, ATM UNI/PPP Physical Layer Processor with integrated DS3/E3 framing controller is designed to support ATM direct mapping and cell delineation as well as PPP mapping and Frame processing. For ATM UNI applications, this device provides the ATM Physical Layer (Physical Medium Dependent and Transmission Convergence sub-layers) interface for the public and private networks at DS3/E3 rates. For Clear-Channel Framer applications, this device supports the transmission and reception of "user data" via the DS3/E3 payload.

The XRT74L74 DS3 ATM UNI/Clear-Channel Framer incorporates Receive, Transmit, Microprocessor Interface, Performance Monitor, Test and Diagnostic and Line Interface Unit Scan Drive functional sections.

FEATURES

- Compliant with 8/16 bit UTOPIA Level I and II and PPP Multi-PHY Interface Specifications and supports UTOPIA Bus operating at 25, 33 or 50 MHz
- Contains on-chip 16 cell FIFO (configurable in depths of 4, 8, 12 or 16 cells), in both the Transmit (TxFIFO) and Receive Directions (RxFIFO)
- Contains on-chip 54 byte Transmit and Receive OAM Cell Buffer for transmission, reception and processing of OAM Cells
- Supports ATM cell or PPP Packet Mapping
- Supports M13 and C-Bit Parity Framing Formats
- Supports DS3/E3 Clear-Channel Framing Applications
- Includes PRBS Generator and Receiver
- Supports Line, Cell, and PLCP Loop-backs
- Interfaces to 8 Bit wide Intel, Motorola, PowerPC, and Mips µPs
- HDLC controller per channel for Tx and Rx
- Low power 3.3V, 5V Input Tolerant, CMOS
- Available in 388 pin PBGA Package

APPLICATIONS

- Digital Access and Cross Connect Systems
- Digital, ATM, WAN and LAN Switches
- Network Interface Service Units

FIGURE 1. BLOCK DIAGRAM OF THE XRT74L74 ATM UNI/PPP DS3/E3 FRAMING CONTROLLER

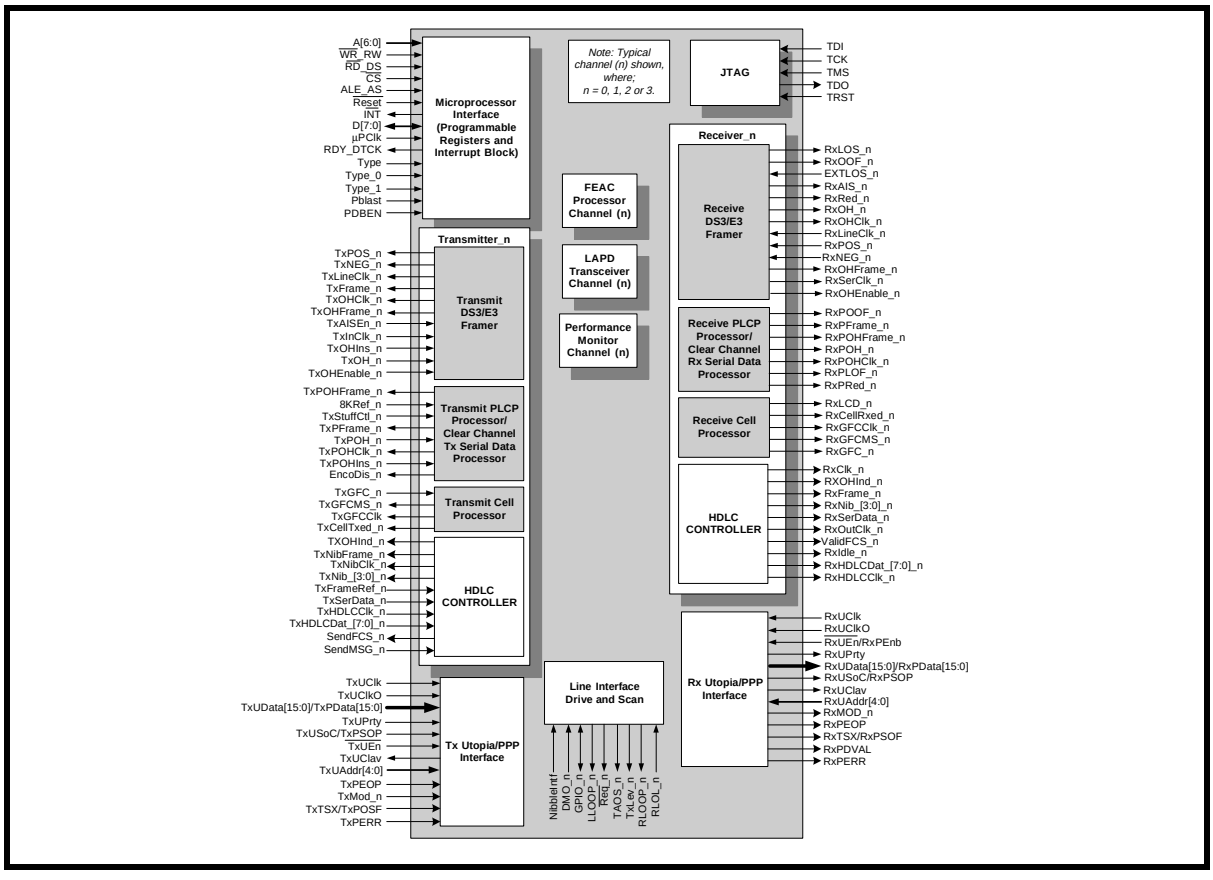
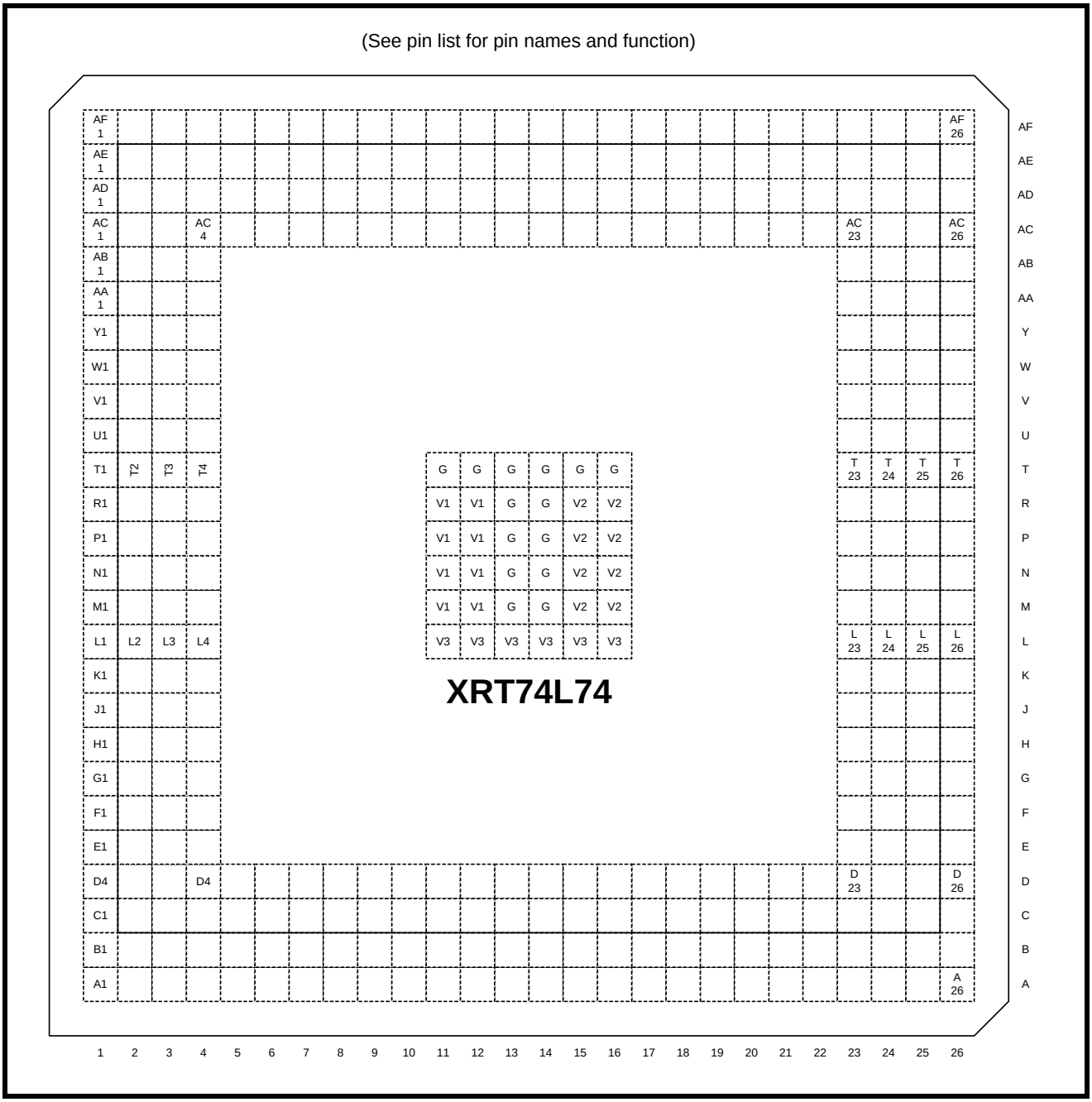


FIGURE 2. PIN OUT OF THE XRT74L74 DS3/E3 ATM UNI/PPP (388 BALL PBGA)



ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE
XRT74L74IB	35 x 35 mm, 388 Plastic Ball Grid Array	-40°C to +85°C

TABLE OF CONTENTS

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Microprocessor Interface			
AB3 AA4 AA1 AA2 Y1 Y2 Y3	A0 A1 A2 A3 A4 A5 A6	I	**Address Bus Input (Microprocessor Interface):** These pins are used to select the on-chip Framer/UNI registers and RAM space for READ/WRITE operations with the “local” microprocessor.
V4	NibbleIntf	I	**Nibble Interface Select Input pin:** This input pin permits the user to configure the Transmit Payload Data Input Interface and the Receive Payload Data Output Interface blocks to operate in either the "Serial" or the "Nibble-Parallel" Mode. Setting this input pin "high" configures each of these blocks to operate in the Nibble-Parallel Mode. In this mode, the "Transmit Payload Data Input Interface" block will accept the "outbound" payload data (from the local terminal equipment) in a "nibble-parallel" manner via the "TxNib[3:0]" input pins. Further, the Receive Payload Data Output Interface block will output "inbound" payload data (to the local terminal equipment) in a "nibble-parallel" via the "RxNib[3:0] output pins. Setting this input pin "low" configures each of these blocks to operate in the Serial Mode. In this mode, the Transmit Payload Data Input Interface block will accept the "outbound" payload data (from the local terminal equipment) in a "serial" manner via the "TxSer_n" input pin. Further, the Receive Payload Data Output Interface block will output the "inbound" payload data (to the local terminal equipment) in a serial manner, via the "RxSer_n" output pin. ***NOTE:*** *This input pin is only active if the XRT74L74 device has been configured to operate in the Clear-Channel Framer Mode.*
AC7 AD6 AE5 AF4 AC5 AD4 AE3 AF2	D0 D1 D2 D3 D4 D5 D6 D7	I/O	**Bi-Directional Data Bus (Microprocessor Interface Section):** These pins 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.
AE2	ALE_AS	I	**Address Latch Enable/Address Strobe:** This input is used to latch the address (present at the Microprocessor Interface Address Bus, A[6:0]) into the Framer/UNI Microprocessor Interface circuitry and to indicate the start of a READ or WRITE cycle. This input is active-"High" in the Intel Mode (MOTO = “Low”) and active-”Low” in the Motorola Mode (MOTO = “High”).
AB1	$\overline{CS}$	I	**Chip Select Input:** This active-”Low” input signal selects the Microprocessor Interface Section of the UNI/Framer and enables Read/Write operations between the “local” microprocessor and the UNI/Framer on-chip registers and RAM locations.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AE4	$\overline{INT}$	O	**Interrupt Request Output:** This open-drain, active-"Low" output signal will be asserted when the UNI/Framer is requesting interrupt service from the local microprocessor. This output pin should typically be connected to the "Interrupt Request" input of the local microprocessor.
AC4 AD3 AF3	PTYPE0 PTYPE1 PTYPE2	I	**Microprocessor Type Select Input:** These three input pins permit the user to configure the Microprocessor Interface block to readily support a wide variety of Microprocessor Interfaces. The relationship between the settings of these input pins and the corresponding Microprocessor Interface configuration is presented below. PTYPE[2:0] Microprocessor Interface Mode 000 Asynchronous Intel 001 Asynchronous Motorola 010 Intel X86 011 Intel I960, Motorola MPC860 100 IDT3051/52 (MIPS) 101 IBM Power PC
AF1	$\overline{RD}$_$\overline{DS}$	I	**Read Data Strobe (Intel Mode):** If the microprocessor interface is operating in the Intel Mode, then this input will function as the $\overline{RD}$ (READ Strobe) input signal from the local µP. Once this active-"Low" signal is asserted, then the UNI/Framer will place the contents of the addressed registers (within the UNI/Framer IC) on the Microprocessor Data Bus (D[7:0]). 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" $\overline{DS}$ (DATA Strobe) signal.
AD1	RDY_$\overline{DTACK}$	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 Framer will assert this pin in order to inform the local microprocessor that the present READ or WRITE cycle is nearly complete. If the UNI/Framer requires that the current READ or WRITE cycle be extended, then the UNI/Framer 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.
V2	$\overline{Reset}$	I	**Reset Input:** When this active-"Low" signal is asserted, the UNI/Framer will be asynchronously reset. When this occurs, all outputs will be "tri-stated" and all on-chip registers will be reset to their default values.
AD5	µPClk	I	**Microprocessor Interface Clock Input** This clock input signal is used for synchronous/burst/DMA data transfer operations. This clock can be running up to 33MHz.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AC2	$\overline{WR}$_R/$\overline{W}$	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 $\overline{WR}$ (Write Strobe) input signal from the µP. Once this active-"Low" signal is asserted, then the UNI/Framer will latch the contents of the µP Data Bus ([D:[7:0]) into the addressed register (or RAM location) within the UNI/Framer 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/$\overline{W}$ pin. In the Motorola Mode, a "READ" operation occurs if this pin is at a logic "1". A WRITE operation occurs if this pin is at a logic "0".
P4	$\overline{BLAST}$	I	**Last Burst Transfer Indicator input pin:** If the Microprocessor Interface is operating in the Intel-I960 Mode, then this input pin is used to indicate (to the Microprocessor Interface block) that the current data transfer is the last data transfer within the current burst operation.The Microprocessor should assert this input pin (by toggling it "Low") in order to denote that the current READ or WRITE operation (within a BURST operation) is the last operation of this BURST operation.
U4	$\overline{DBEN}$	I	**Bi-directional Data Bus Enable Input pin:** If the Microprocessor Interface is operating in the Intel-I960 Mode, then this input pin is used to enable the Bi-directional Data Bus. Setting this input pin "Low" enables the Bi-directional Data bus. Setting this input "High" tri-states the Bi-directional Data Bus.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Test and Diagnostic			
U2	TCK	I	**Test Clock:** Boundary Scan clock input. ***NOTE:*** *This input pin should be pulled "Low" for normal operation.*
U1	TDI	I	**Test Data In:** Boundary Scan Test data input. ***NOTE:*** *This input pin should be pulled "Low" for normal operation.*
V3	TDO	O	**Test Data Output:** Boundary Scan test data output.
V1	TestMode	***	**Factory Test Mode Pin:** *The user should tie this pin to ground.*
T4	TMS	I	**Test Mode Select:** Boundary Scan Test Mode Select input pin. *This input pin should be pulled "Low" for normal operation.*
U3	TRST	I	**Test Mode Reset:** Boundary Scan Mode Reset input pin. ***NOTE:*** *This input pin should be pulled "low" for normal operation.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
General Purpose Input and Output Pins			
C13 B13 A13 B12	DMO_0 DMO_1 DMO_2 DMO_3	I	**"Drive Monitor Output" Input (from the XRT73L04 LIU IC):** This input pin is intended to be tied to the DMO output pin of the XRT73L04 E3/DS3/STS-1 LIU IC. The user can determine the state of this input pin by reading Bit 2 (DMO) within the Line Interface Scan Register (Address = 0x73). If this input signal is "High", then it means that the drive monitor circuitry (within the XRT73L04 LIU 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 XRT73L04 device. ***NOTE:*** *If the designer is not using the XRT73L04 E3/DS3/STS-1 LIU IC, then this input pin can be used for other purposes.*
R4 T3 T2 T1	GPIO_0 GPIO_1 GPIO_2 GPIO_3	I/O	**General Purpose Input/Output pins:** Each of these pin can be configured to function as either input or output pins. If a given pin is configured to function as an Input pin, then the state of this input pin can be monitored by reading Bit X within the "XXX" Register (Address Location = 0xXX, 0xXX). If a given pin is configured to function as a Output pin, then the state of this output pin can be controlled by writing the appropriate value into Bit X within the "XXX" Register.
A5 D5 C4 A4	LLOOP_0 LLOOP_1 LLOOP_2 LLOOP_3	O	**Local Loop-back Output Pin (to the XRT73L04 E3/DS3/STS-1 LIU IC):** This output pin is intended to be connected to the LLOOP input pin of the XRT73L04 LIU IC. This input pin, along with "RLOOP" permits the user to configure the XRT73L04 LIU IC to operate in either of the following three (3) loop-back modes. • Analog Local Loop-Back Mode • Digital Local Loop-Back Mode • Remote Loop-Back Mode. For a detailed description on how to configure the XRT73L04 to operate in each of these loop-back modes, please see Section _. Writing a "1" to bit 1 of the "Line Interface Drive Register" (Address = 0xXX, 0xXX) 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 XRT73L04 DS3/E3/STS-1 LIU IC, then this output pin can be used for other purposes.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
D20 B19 A19 C18	REQ_EN_0 REQ_EN_1 REQ_EN_2 REQ_EN_3	O	**Receive Equalization Bypass Control Output Pin—(to be connected to the XRT73L04 E3/DS3/STS-1 LIU IC):** This output pin is intended to be connected to the REQEN input pin of the XRT73L04 E3/DS3/STS-1 LIU IC. The user can control the state of this output pin by writing a '0' or '1' to Bit 5 (REQEN) of the Line Interface Driver Register (Address = 0xXX, 0xXX). If the user commands this signal to toggle "High" then it will cause the incoming DS3 line signal to "by-pass" equalization circuitry, within the XRT73L04 Device. Conversely, if the user commands this output signal to toggle "Low", then the incoming DS3 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 "XRT73L04 E3/DS3/STS-1 LIU IC" data sheet. Writing a "1" to Bit 5 of the Line Interface Drive Register 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 designer is not using the XRT73L04 E3/DS3/STS-1 LIU IC, then this output pin can be used for other purposes.*
P23 N24 N25 N26	RLOL_0 RLOL_1 RLOL_2 RLOL_3	I	**Receive Loss of Lock Indicator—from the XRT73L04 E3/DS3/STS-1 LIU IC:** This input pin is intended to be connected to the RLOL (Receive Loss of Lock) output pin of the XRT73L04 LIU 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 = 0xXX, 0xXX). If this input pin is "low" it means that the Clock Recovery PLL (within the corresponding channel of the XRT73L04 device) is properly locked onto and is performing clock and data recovery on the "incoming" DS3 or E3 data stream. If this input pin is "high" then it means the Clock Recovery PLL is NOT properly locked onto the incoming DS3 or E3 line signal. Further, this indicates that the Clock Recovery PLL is NOT performing clock and data recovery on this incoming DS3 or E3 data stream. For more information on the operation of the XRT73L04 E3/DS3/STS-1 LIU IC, please consult the "XRT73L04 E3/DS3/STS-1 LIU IC" data sheet. ***NOTE:*** *If the designer is not using the XRT73L04 DS3/E3/STS-1 LIU IC, this input pin can be used for other purposes.*
A15 B14 A14 D14	RLOOP_0 RLOOP_1 RLOOP_2 RLOOP_3	O	**Remote Loop-back Output Pin (to the XRT73L04 DS3/E3/STS-1 LIU IC):** This output pin is intended to be connected to the RLOOP input pin of the XRT73L04 LIU IC. This output pin, along with the LLOOP input pin permits the user to configure the XRT73L04 to operate in either of the following three (3) loop-back modes. • Analog Local Loop-Back Mode • Digital Local Loop-Back Mode • Remote Loop-Back Mode. For a detailed description on how to configure the XRT73L04 to operate in each of these loop-back modes, please see Section _. Writing a "1" to bit 1 of the "Line Interface Drive Register (Address = 0x72) 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 XRT73L04 DS3/E3/STS-1 IC, then this output pin can be used for other purposes.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
A22 C21 B21 A21	TAOS_0 TAOS_1 TAOS_2 TAOS_3	O	**Transmit All Ones Signal (TAOS) Command (for the XRT73L04 LIU IC).** These output pins are intended to be connected to each of the TAOS input pins of the XRT73L04 device. The user can control the state of these output pins by writing a "0" or "1" into Bit 4 (TAOS) within the corresponding Line Interface Drive Register (Address = 0xXX, 0xXX). If the user commands this signal to toggle "high", then it will force the corresponding channel (within the XRT73L04 device) to transmit an "All Ones" pattern onto the line. Conversely, if the user commands this output signal to toggle "low", then the corresponding channel will proceed to transmit data based upon the data that it receives via the TxPOS and TxNEG output pins (of the XRT74L74 device). Writing a "1" to Bit 4 of the Line Interface Drive Register 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 designer is not using the XRT73L04 DS3/E3/STS-1 LIU IC, then this output pin can be used for other purposes*
B9 A9 D9 B8	ENDECDIS_0 ENDECDIS_1 ENDECDIS_2 ENDECDIS_3	O	**Encoder/Decoder (B3ZS & HDB3) Disable Output pin (intended to be connected to the XRT73L04 DS3/E3/STS-1 LIU IC):** These output pins are intended to be connected to each of the ENDECDIS input pins of the XRT73L04 LIU IC. The user can control the state of this output pin by writing a "0" or "1" into Bit 3 (ENDECDIS) within the Line Interface Drive Register (Address = 0xXX, 0xXX). If the user commands this signal to toggle "high", then it will disable the B3ZS/HDB3 Encoder and Decoder circuitry within the corresponding channel (within the XRT73L04 device). Conversely, if the user commands this output signal to toggle "low", then the B3ZS/HDB3 Encoder and Decoder circuitry, within the corresponding LIU channel will be enabled. Writing a "1" to Bit 3 of the Line Interface Drive Register 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 B3ZS/HDB3 Encoder and Decoder (within the XRT73L04 LIU IC) if the B3ZS/HDB3 Encoder and Decoder blocks (within the XRT74L74 device) are enabled.* 2. *If the designer is not using the XRT73L04 DS3/E3/STS-1 LIU IC, then this output pin can be used as a General Purpose Output pin*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
B16 A16 C15 B15	TxLev_0 TxLev_1 TxLev_2 TxLev_3	O	**Transmit Line Build Enable/Disable Select (to be connected to the TxLev input pin of the XRT73L04 E3/DS3/STS-1 LIU IC):** These output pins are intended to be connected to the TxLEV input pins of the XRT73L04 DS3/E3/STS-1 LIU 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 Drive Register (Address = 0xXX, 0xXX).If the user commands this signal to toggle "high", then it will disable the "Transmit Line Build-Out" circuit within the corresponding channel (of the XRT73L04 LIU IC). In this case, the LIU channel will output unshaped pulses onto the "Transmit Line signal". In order to insure that the XRT73L04 LIU IC generates a line signal that is compliant with the Bellcore GR-499-CORE Pulse Template requirements (as the DSX-3 Cross-Connect) location, the user is advised to set this output pin "high" if the cable length (between the Transmit Output of the LIU Channel and the DSX-3 Cross-Connect) is greater than 225 feet. Conversely, if the user commands this signal to toggle "high", then it will enable the "Transmit Line Build-Out" circuit within the corresponding channel (of the XRT73L04 LIU IC). In this case, the LIU channel will output shaped pulses onto the "Transmit Line Signal". In order to ensure that the XRT73L04 LIU IC generates a line signal that is compliant with the Bellcore GR-499-CORE Pulse Template requirements (at the DSX-3 Cross Connect), the user is advised to set this output pin "low", if the cable length (between the Transmit Output of the XRT73L04 and the DSX-3 Cross-Connect) is less than 225 feet of cable. Writing a "1" to Bit 2 of the Line Interface Drive register 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 setting for TxLEV has no impact on the shape of the transmit output pulse if the LIU channel is configured to operate in the E3 Mode.* *2. If the designer is not using the XRT73L04 DS3/E3/STS-1 LIU IC, then this output pin can be used for other purposes.*
G26 G23 F24 F25	EXTLOS_0 EXTLOS_1 EXTLOS_2 EXTLOS_3	I	**Receive LOS (Loss of Signal) Indicator Input (from XRT73L04 E3/DS3/STS-1 Line Interface Unit).** This input pin is intended to be connected to each of the RLOS (Receive Loss of Signal) output pins of the XRT73L04 DS3/E3/STS-1 LIU IC. The user can monitor the state of this input pin by reading the state of Bit 0 (RLOS) within the Line Interface Scan Register (Address = 0xXX, 0xXX). If this input pin is "Low", then it means that the corresponding channel (within the XRT73L04 device) is currently NOT declaring an LOS condition. However, if this input pin is "high", then it means that this particular channel is currently declaring an LOS condition. For more information on the operation of the XRT73L04 E3/DS3/STS-1 Line Interface Unit IC, please consult the “XRT73L04 ” data sheet. ***NOTE:*** *Asserting the RLOS input pin will cause the XRT74L74 Framer/UNI to declare an “LOS” (Loss of Signal) condition. Therefore, this input pin should not be used as a general purpose input.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
E23 F26 H25 K23	RxNib_0_0/ RxHDLCDat_0_0 RxNib_0_1/ RxHDLCDat_0_1 RxNib_0_2/ RxHDLCDat_0_2 RxNib_0_3/ RxHDLCDat_0_3	O	**Receive Nibble Interface Output pin - Bit 0/Receive HDLC Controller Data Bus output pin - Bit 0:** The function of this output pin depends upon whether the channel has been configured to operate in the Clear-Channel/Nibble-Parallel Mode, the High-Speed HDLC Controller Mode, or in some other mode. **Clear-Channel/Nibble-Parallel Mode - RxNib_0_n:** The channel will output "Received data" (from the remote terminal equipment) to the local terminal equipment via this pin, along with RxNib_1_n through RxNib_3_n: This particular output pin functions as the LSB. The data at this pin is updated on the rising edge of the RxClk_n output signal. Hence, the user's local terminal equipment should sample this signal upon the falling edge of RxClk_n. **High-Speed HDLC Controller Mode - RxHDLCDat_0_n:** This output pin, along with RxHDLCDat_[7:1]_n function as the Receive HDLC Controller byte wide output data bus. This particular output pin functions as the LSB (Least Significant Bit) of the Receive HDLC Controller byte wide data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal. ***NOTE:*** *This output pin is only active if the channel is configured to operate in the "Clear-Channel/Nibble-Parallel" Mode or in the "High-Speed HDLC Controller" Mode. This output is inactive for all remaining modes.*
A26 E26 G25 U23	RxRed_0/ RxNib_3_0/ RxHDLCDat_3_0 RxRed_1/ RxNib_3_1/ RxHDLCDat_3_1 RxRed_2/ RxNib_3_2/ RxHDLCDat_3_2 RxRed_3/ RxNib_3_3/ RxHDLCDat_3_3	O	**Receive Section Red Alarm Indicator/Receive Nibble Interface Output pin - Bit 3/Receive HDLC Controller Data Bus output pin - Bit 3:** The exact function of this output pin depends upon whether the channel has been configured to operate in the "Clear-Channel Framer/Nibble-Parallel Mode, the High-Speed HDLC Controller Mode, or in some other mode. **Clear-Channel Framer/Nibble-Parallel Mode - RxNib_3_n:** The channel will output "Received data" (from the remote terminal equipment) to the local terminal equipment via this pin, along with RxNib_0_n through RxNib_2_n. This particular output pin functions as the LSB. The data at this pin is updated on the rising edge of the RxClk_n output signal. Hence, the user's local terminal equipment should sample this signal upon the falling edge of RxClk_n. **High-Speed HDLC Controller Mode - RxHDLCDat_3_n** This output pin, along with RxHDLCDat_[7:4]_n and RxHDLCDat_[2:0]_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal. **Other Modes - RxRED_n:** The Framer/UNI asserts this output pin to denote that one of the following events has been detected by the Receive DS3/E3 Framer block: • LOS - Loss of Signal Condition • OOF - Out of Frame Condition • AIS - Alarm Indication Signal Detection

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Transmit System Side Interface Pins			
A12 D12 C11 B11	TxAISEn_0 TxAISEn_1 TxAISEn_2 TxAISEn_3	I	**Transmit AIS Pattern input:** This input pin permits the user to command the Transmit DS3/E3 Framer block to transmit an AIS pattern to the remote terminal equipment. Setting this input pin "High" configures the Transmit DS3/E3 Framer block to transmit an AIS pattern to the remote terminal equipment. Setting this input pin "Low" configures the Transmit DS3/E3 Framer block to NOT transmit an AIS pattern to the remote terminal equipment. ***NOTE:*** *For normal operation, or if the user wishes to control the "Transmit AIS" function, via Software Control; the user should tie these input pins to GND.*
A11 B10 A10 C9	TxFrame_0 TxFrame_1 TxFrame_2 TxFrame_3	O	**Transmit End of DS3 Frame Indicator:** This output pin is pulse "high" for one DS3 or E3 clock period, when the Transmit Section of the channel is processing the last bit of a given DS3 or E3 frame. The implications of these output pins, for each mode of operation. **ATM UNI/PPP/High-Speed HDLC Controller Mode** This output pin simply serves as an "end-of-frame" indication to the local terminal equipment. **Clear-Channel Framer Mode** If the XRT74L74 device is configured to operate in the Clear-Channel Framer mode, then this output pin serves to alert the Local Terminal Equipment that it needs to begin transmission of a new DS3 or E3 frame. Hence, the Local Terminal Equipment uses this output signal to maintain "Framing Alignment" with the XRT74L74 device.
A8 C7 B7 A7	TxFrameRef_0 TxFrameRef_1 TxFrameRef_2 TxFrameRef_3	I	**Transmit DS3 Framer—Frame Reference Input Pin:** If the Transmit Section of the Channel is configured to operate in the "Local-Timing/Frame-Slave" Mode, then the Transmit DS3/E3 Framer block will use this input signal as the Framing Reference. When a given channel is configured to operate in this mode, then any rising edge at this input pin will cause the Transmit DS3/E3 Framer block to begin its creation of a new DS3 or E3 frame. Consequently, the user must supply a clock signal that is equivalent to the DS3 or E3 frame rates (to this input pin). Further, it is imperative that this clock signal be synchronized with the 44.736MHz or 34.368MHz clock signal applied to the TxInClk input pin. ***NOTE:*** *This input pin should be tied to "GND" if it is not used as the Transmit DS3E3 Framer, frameing reference signal.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
E4 G4 F2 E1	TxInClk_0 TxInClk_1 TxInClk_2 TxInClk_3	I	**Transmit DS3 Framer Block—Clock Signal:** If the Transmit Section of a given channel (within the XRT74L74 device) is configured to operate in the Local-Timing Mode, then it will use this signal as the Timing Reference. If the user is operating a channel in the DS3 Mode, then user is expected to apply a high-quality 44.736MHz clock signal to this input pin. Likewise, if the user is operating a channel in the E3 Mode, then the user is expected to apply a high-quality 34.368MHz clock signal to this input pin.A **Note for Clear-Channel Framer Operation:** If the user is operating the XRT74L74 device in the Clear-Channel Framer mode, then the user should design the local terminal equipment circuitry, such that "outbound" DS3 or E3 data will be output, upon the falling edge of TxInClk. The Transmit Payload Data Input Interface (within the Transmit Section of the XRT74L74 device) will sample the data, applied to the "TxSer" input pin, upon the rising edge of TxInClk. ***NOTE:*** *This input pin should be tied to GND if the XRT74L74 device is configured to operate in the "Loop-Timing" Mode.*
F3 F1 G3 G2	TxOH_0/ TxHDLCDat_5_0 TxOH_1/ TxHDLCDat_5_1 TxOH_2/ TxHDLCDat_5_2 TxOH_3/ TxHDLCDat_5_3	I	**Transmit Overhead Input Pin/Transmit HDLC Controller Data Bit 5:** The function of these input pins depends upon whether the channel has been configured to operate in the High Speed HDLC Controller Mode or not. **Non-High Speed HDLC Controller Mode - TxOH_n:** The Transmit Overhead Data Input Interface accepts overhead via these input pins, and insert this data into the "overhead" bit positions within the outbound DS3 or E3 frames. If the "TxOHIns_n" input pin is pulled "high", then the Transmit Overhead Data Input Interface will sample the overhead data, via this input pin, upon the falling edge of the TxOHClk_n output signal. Conversely, if the TxOHIns_n input pin is NOT pulled "high", then the Transmit Overhead Data Input Interface block will be inactive and will not accept any overhead data via the TxOH_n input pin. **High Speed HDLC Controller Mode - TxHDLCDat_5_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 5" within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.
B6 A6 C5 B5	TxOHClk_0 TxOHClk_1 TxOHClk_2 TxOHClk_3	O	**Transmit Overhead Clock:** This output pin functions as the "Transmit Overhead Data Input Interface clock signal. If the user enables the "Transmit Overhead Data Input Interface" block by asserting the "TxOHIns" input pin, then the Transmit Overhead Data Input Interface block will sample and latch the data (residing on the "TxOH_n" input pin) upon the falling edge of this signal. ***NOTE:*** *The Transmit Overhead Data Input Interface block is disabled if the user has configured the channel to operate in the "High-Speed HDLC Controller" Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
D3 D2 E3 E2	TxOHFrame_0/ TxHDLCClk_0 TxOHFrame_1/ TxHDLCClk_1 TxOHFrame_2/ TxHDLCClk_2 TxOHFrame_3/ TxHDLCClk_3	O	**Transmit Overhead Framing Pulse/Transmit HDLC Controller Clock Output pin:** The function of this output pin depends upon whether the channel (within the XRT74L74 device) has been configured to operate in the "High-Speed HDLC Controller" Mode or not. **Non-High-Speed HDLC Controller Mode - TxOHFrame_n:** This output pin pulses high for one TxOHClk_n period coincident with the instant the Transmit Overhead Data Input Interface would be accepting the first overhead bit within an outbound DS3 or E3 frame. **High Speed HDLC Controller Mode - TxHDLCClk_n:** This output pin functions as the "demand" clock output signal for the "Transmit HDLC Controller" byte-wide input interface. This clock signal is ultimately derived from either the TxInClk_n or the RxOutClk_n signal. Hence, the frequency of this clock signal is nominally one-eight of that of the TxInClk_n or the RxOutClk_n signals. The Transmit HDLC Controller block will sample the contents of the "Transmit HDLC Controller" byte-wide input interface upon the rising edge of this clock output signal. Therefore, the local terminal equipment should be designed to output data (onto the TxHDLCDat[7:0] bus) upon the falling edge of this clock output signal.
A3 A1 B2 C1	TxOHIns_0/ TxHDLCDat_4_0 TxOHIns_1/ TxHDLCDat_4_1 TxOHIns_2/ TxHDLCDat_4_2 TxOHIns_3/ TxHDLCDat_4_3	I	**Transmit Overhead Data Insert Input:** Transmit Overhead Data Insert Input/Transmit HDLC Controller Data Bit 4 input pin:The function of these input pins depends upon whether the channel (within the XRT74L74 device) has been configured to operate in the "High-Speed HDLC Controller" Mode or not. **Non-High Speed HDLC Controller Mode - TxOHIns_n:** This input pin permits the user to either enable or disable the "Transmit Overhead Data Input Interface" block. If the Transmit Overhead Data Input Interface block is enabled, then it will accept overhead data (from the local terminal equipment) via the "TxOH_n" input pin; and insert this data into the overhead bit positions within the outbound DS3 or E3 data stream. Conversely, if the Transmit Overhead Data Input Interface block is disabled, then it will NOT accept overhead data from the local terminal equipment .Pulling this input pin "high" enables the "Transmit Overhead Data Input Interface" block. Pulling this input pin "low" disables the "Transmit Overhead Data Input Interface" block. **High-Speed HDLC Controller Mode - TxHDLCDat_4_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 4" within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Transmit Line Side Signals			
B1 C6 C16 D22	TxPOS_0 TxPOS_1 TxPOS_2 TxPOS_3	O	**Transmit Positive Polarity Pulse Output:** The exact role of this output pin depends upon whether the Framer is operating in the Single-Rail or Dual-Rail mode. **Single-Rail Mode:** This output pin functions as the "Single-Rail" (e.g., binary data stream) output signal for the "outbound" DS3 or E3 data stream. The signal at this output pin, will be updated on the "user-selected" edge of the "TxLineClk_n" signal. **Dual-Rail Mode:** This output pin functions as one of the two dual-rail output signal that command the sequence of bipolar pulses, which are to be driven onto the line. TxNEG_n is the other output pin. This output pin is typically connected to the TPDATA/TPOS input of the external DS3/E3 LIU IC. When this output pulses "high", and latched into the LIU, the LIU will then proceed to generate a positive polarity pulse on to line.
D1 C8 C14 C23	TxNEG_0 TxNEG_1 TxNEG_2 TxNEG_3	O	**Transmit Negative Polarity Pulse Output:** The exact function of this output pin depends upon whether the Framer/UNI has been configured to operate in the Single-Rail or Dual-Rail Mode. **Single-Rail Mode:** This output signal pulses "high" for one (DS3 or E3) bit period, at the end of each "outbound" DS3 or E3 frame. This output signal is pulled "low" for all of the remaining bit periods within the "outbound" DS3 or E3 frames. **Dual-Rail 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 output pin is typically connected to the TPDATA/TPOS input pin of the external DS3/E3 LIU IC. When this output pin is pulse "high" and latched into the LIU IC, the LIU IC will then proceed to generate a negative-polarity pulse, on to the line.
C2 D7 D16 B24	TxLineClk_0 TxLineClk_1 TxLineClk_2 TxLineClk_3	O	**Transmit Line Interface Clock:** This clock signal is output to the Line Interface Unit, along with the TxPOS_n and TxNEG_n signal. The purpose of this clock signal is two-fold. **1.** To provide the LIU with timing information that it can use to generate the bipolar pulses and deliver them over the transmission medium to the remote terminal equipment. **2.** To provide the LIU with a clock signal, that it can use to sample the data on the "TxPOS_n" and "TxNEG_n" input pins. The user can configure the source of this clock signal to be either the "RxLineClk_n" signal (from the Receive Section of the channel) or the TxInClk_n input. The nominal frequency for this clock signal is 44.736MHz (for DS3 applications) and 34.368MHz (for E3 applications).

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Rx DS3 Framer			
C26 E25 G24 R23	RxAIS_0/ RxNib_2_0/ RxHDLCDat_2_0 RxAIS_1/ RxNib_2_1/ RxHDLCDat_2_1 RxAIS_2/ RxNib_2_2/ RxHDLCDat_2_2 RxAIS_3/ RxNib_2_3/ RxHDLCDat_2_3	O	**Receive AIS Pattern Indicator/Receive Nibble Output Interface - Bit 2/ Receive HDLC Controller Data Bus - Bit 2 output pin:** The exact function of this output pin depend upon whether the XRT74L74 device has been configured to operate in the Clear-Channel Framer/Nibble-Parallel Interface Mode, the High-Speed HDLC Controller Mode, or in the other modes. **Other Modes - RxAIS_n:** This output pin is driven "high" whenever the Receive Section of the channel has detected and is currently declaring an "AIS" (Alarm Indicator Signal) condition. **Clear-Channel Framer/Nibble-Parallel Interface Mode - RxNib_2_n:** If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then this output pin will function as the bit 2 output from the "Receive Nibble-Parallel" output interface. The Receive Payload Data Output Interface block will output this signal (along with RxNib_0_n, RxNib_1_n, and RxNib_3_n) upon the rising edge of the RxClk_n output signal. **High-Speed HDLC Controller Mode - RxHDLCDat_2_n:** This output pin, along with RxHDLCDat_[7:3]_n and RxHDLCDat_[1:0]_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal.
B18 A18 B17 A17	RxFrame_0 RxFrame_1 RxFrame_2 RxFrame_3	O	**Receive Boundary of DS3 or E3 Frame Output indicator:** The exact function of this output pin depends upon whether the channel is operating in the Clear-Channel Framer/Nibble-Parallel Mode or not. **Clear-Channel Framer/Nibble-Parallel Mode** The Receive Section of the channel will pulse this output pin "high" for one nibble period, when the Receive Payload Data Output interface block is driving the very first nibble of a given DS3 or E3 frame, on the "RxNib_n[3:0]" output pins. **Clear-Channel Framer/Serial Mode** The Receive Section of the channel will pulse this output pin "high" for one bit period, when the Receive Payload Data Output interface block is driving the very first nibble of a given DS3 or E3 frame, on the "RxSer_n" output pin. **All Other Modes:** The Receive Section of the channel will pulse this output pin "high" when the Receive DS3/E3 Framer block is processing the first bit within a new DS3 or E3 frame.
A2 D11 D18 C25	RxLineClk_0 RxLineClk_1 RxLineClk_2 RxLineClk_3	I	**Receive (Recovered LIU) Line Clock:** This input signal serves three purposes. **1.** The Receive Section of the XRT74L74 device use it to sample and latch the signals at the "RxPOS_n" and "RxNEG_n" input pins (into the Receive Framer/UNI circuitry). **2.** This input signal functions ass the timing reference for the Receive Sections of the XRT74L74 device. **3.** The Transmit Sections can be configured to use this input signal at its timing reference. ***NOTE:*** *This signal is the recovered clock from the external DS3/E3 LIU IC, which is derived from the incoming DS3 or E3 line signal.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
J25 J26 J23 H24	RxLOS_0 RxLOS_1 RxLOS_2 RxLOS_3	O	**Framer/UNI - Loss of Signal Output Indicator:** This pin is asserted when the Receive Section of the channel encounters 180 consecutive 0's (for DS3 applications) or 32 consecutive 0's (for E3 applications) via the RxPOS_n and RxNEG pins. This pin will be negated once the Receive DS3/E3 Framer has detected at least 60 "1s" out of 180 consecutive bits (for DS3 applications) or has detected at least four consecutive 32 bit strings of data that contain at least 8 "1s" in the receive path.
B4 C12 C17 D26	RxNEG_0 RxNEG_1 RxNEG_2 RxNEG_3	I	**Receive Negative Polarity Data Input:** The exact function of these input pins depend upon whether the XRT74L74 device is operating in the Single-Rail or Dual-Rail Mode. **Single-Rail Mode:** This input pin is inactive and should be pulled to GND, whenever the XRT74L74 device is operating in this mode. **Dual-Rail Mode:** This input pin functions as one of the dual-rail inputs for the incoming B3ZS/ HDB3 encoded DS3 or E3 data, which has been received from an external LIU IC. RxPOS_n as functions as the other dual-rail input for the Framer/UNI IC. When this input pin is pulsed "high", it means that the LIU IC has received a "negative-polarity" pulse from the line.This input signal will be sampled (by the XRT74L74 device) upon the "user-selected" edge of the RxLineClk_n signal.
B26 A25 B25 A24	RxOH_0/ RxHDLCDat_6_0 RxOH_1/ RxHDLCDat_6_1 RxOH_2/ RxHDLCDat_6_2 RxOH_3/ RxHDLCDat_6_3	O	**Receive Overhead Data Output Interface - output/Receive HDLC Controller Data Bus - Bit 6 output:** The exact function of this output pin depends upon whether the channel has been configured to operate in the "Clear-Channel Framer" mode or in the "High-Speed HDLC Controller" Mode. **Clear-Channel Framer Mode - RxOH_n:** All overhead bits, which are received via the "Receive Section" of the channel will be output via this output pin, upon the rising edge of "RxOHClk_n". **High-Speed HDLC Controller Mode - RxHDLCDat_6_n:** This output pin, along with RxHDLCDat_[5:0]_n and RxHDLCDat_7_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
B23 A23 C22 B22	RxOHClk_0/ RxHDLCClk_0 RxOHClk_1/ RxHDLCClk_1 RxOHClk_2/ RxHDLCClk_2 RxOHClk_3/ RxHDLCClk_3	O	**Receive Overhead Data Output Interface-Clock/Receive HDLC Controller - Clock output:** The exact function of this output pin depends upon whether the channel has been configured to operate in the "Clear-Channel Framer" mode or in the "High-Speed HDLC Controller" Mode. **Clear-Channel Framer Mode - RxOHClk_n:** The channel will output the overhead bits (within the incoming DS3 or E3 frames) via the RxOH_n output pin, upon the falling edge of this clock signal.As a consequence, the user's local terminal equipment should use the rising edge of this clock signal to sample the data on both the "RxOH" and "RxOHFrame" output pins. ***NOTE:*** *This clock signal is always active.* **High-Speed HDLC Controller Mode - RxHDLCClk_n:** This output pin functions as the "Receive HDLC Controller" Data bus clock output. The Receive HDLC Controller block outputs the contents of all received HDLC frames via the "Receive HDLC Controller Data bus (RxHDLCDat_[7:0]_n) upon the rising edge of this clock signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of this clock signal.
D21 C20 B20 A20	RxOHFrame_0/ RxHDLCDat_4_0 RxOHFrame_1/ RxHDLCDat_4_1 RxOHFrame_2/ RxHDLCDat_4_2 RxOHFrame_3/ RxHDLCDat_4_3	O	**Receive Overhead Data Interface - Framing Pulse indicator/Receive HDLC Controller Data Bus - Bit 4 output:** The exact function of this output pins depends upon whether the channel has been configured to operate in the "Clear-Channel Framer" Mode or in the "High-Speed HDLC Controller" Mode. **Clear-Channel Framer Mode - RxOHFrame_n:** This output pin pulses "high" whenever the Receive Overhead Data Output Interface block outputs the first overhead bit of a new DS3 or E3 frame. **High-Speed HDLC Controller Mode - RxHDLCDat_4_n:** This output pin, along with RxHDLCDat_[3:0]_n and RxHDLCDat_[7:5]_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
D25 E24 H26 N23	RxOOF_0/ RxNib_1_0/ RxHDLCDat_1_0 RxOOF_1/ RxNib_1_1/ RxHDLCDat_1_1 RxOOF_2/ RxNib_1_2/ RxHDLCDat_1_2 RxOOF_3/ RxNib_1_3/ RxHDLCDat_1_3	O	**Receive Out of Frame Indicator/Receive Nibble Interface Output pin - Bit 1/ Receive HDLC Controller Data Bus Output pin - Bit 1:** The exact function of this output pin depends upon whether the channel has been configured to operate in the Clear-Channel Framer/Nibble-Parallel Mode, the High-Speed HDLC Controller Mode, or not. **Clear-Channel Framer/Nibble-Parallel Mode - RxNib_1_n:** The channel will output "Received data" (from the remote terminal equipment) to the local terminal equipment via this pin, along with RxNib_0_n, RxNib_2_n and RxNib_3_n: This particular output pin functions as the LSB. The data at this pin is updated on the rising edge of the RxClk_n output signal. Hence, the user's local terminal equipment should sample this signal upon the falling edge of RxClk_n. **High-Speed HDLC Controller Mode - RxHDLCDat_1_n:** This output pin, along with RxHDLCDat_[7:2]_n and RxHDLCDat_0_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal. **All other Modes - RxOOF_n:** The UNI Receive DS3 Framer will assert this output signal whenever it has declared an "Out of Frame" (OOF) condition with the incoming DS3 frames. This signal is negated when the framer correctly locates the F- and M-bits and regains synchronization with the DS3 frame.
R26 P24 P25 P26	RxPOH_0/ RxSer_0 RxPOH_1/ RxSer_1 RxPOH_2/ RxSer_2 RxPOH_3/ RxSer_3	O	**Receive PLCP Path Overhead Output pin/Receive Serial Output pin:** The exact function of this output depends upon whether the channel has been configured to operate in the ATM/PLCP Mode, the Clear-Channel Framer Mode or not. **ATM/PLCP Mode - RxPOH_n:** This output pin, along with the RxPOHClk_n, RxPOHFrame_n and RxPOHIns_n pins comprise the "Receive PLCP Frame POH Byte" serial output port. For each PLCP frame, that is received by the Receive PLCP Processor, this serial output port will output the contents of all 12 POH (Path Overhead) bytes. The data that is output via this pin, is updated on the rising edge of the "RxPOHClk_n" output clock signal. The "RxPOHFrame_n" pin will pulse "high" whenever the first bit of the Z6 byte is being output via this output pin. **Clear-Channel Framer Mode - RxSer_n:** If the user opts to operate this channel in the "Clear-Channel Framer/Serial" Mode, then the chip will output all received data, via this output pin. This output signal will be updated upon the rising edge of RxClk. ***NOTE:*** *The user should either configure the channel to operate in the "Gapped-Clock" Mode, or validate the sampling of each bit (from the RxSer_n output) with the state of "RxOHInd_n' output pin, in order to prevent the local terminal equipment from sampling overhead bits.This output pin is only active if the channel has been configured to operate in the "ATM/PLCP" or the Clear-Channel Framer/ Serial Mode. This pin is inactive for all remaining modes of operation.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
M24 M25 M26 M23	RxPOHClk_0/ RxClk_0/ RxNibClk_0 RxPOHClk_1/ RxClk_1/ RxNibClk_1 RxPOHClk_2/ RxClk_2/ RxNibClk_2 RxPOHClk_3/ RxClk_3/ RxNibClk_3	O	**Receive PLCP Path Overhead Serial and Nibble-Parallel Output port clock/ Receive Serial Clock output:** The exact function of this output pin depends upon whether the channel has been configured to operate in the ATM/PLCP Mode, the Clear-Channel Framer Mode, or not. **ATM/PLCP Mode - RxPOH_Clk_n:** This output clock pin, along with "RxPOH_n", "RxPOHFrame_n" and "RxPOHIns_n" pins comprise the "Receive PLCP Frame POH Byte" serial output port. All POH (Path Overhead) data that is output via the "RxPOH_n" output pin is updated on the rising edge of this clock signal. ***NOTE:*** *This output signal is inactive if the XRT74L74 device has been configured to operate in the Direct-Mapped ATM Mode.* **Clear-Channel Framer Mode - RxClk_n:** This output pin is active whenever the channel has been configured to operate in either the Serial or Nibble Parallel Mode. **Clear-Channel Framer/Serial Mode - RxClk_n:** In this "serial" mode, this output is a 44.736MHz clock output signal (for DS3 applications) or 34.368MHz clock output signal (for E3 applications). The Receive Payload Data Output Interface will update the data via the RxSer_n output pin, upon the rising edge of this clock signal.The user is advised to design (or configure) the local terminal equipment to sample the "RxSer_n" data, upon the falling edge of this clock signal. **Clear-Channel Framer/Nibble-Parallel Mode - RxSer_n:** In the Nibble-Parallel Mode, the XRT74L74 device will derive this clock signal from the "RxLineClk_n" signal. The XRT74L74 device will pulse this clock signal 1176 times for each "inbound" DS3 frame (or 1074 times for each inbound E3/ ITU-T G.832 frame or 384 times for each inbound E3/ITU-T G.751 frame). The Receive Payload Data Output Interface block will update the data (on the RxNib_n[3:0] output) upon the falling edge of this clock signal.The user is advised to design (or configure) the local terminal equipment to sample the data on the "RxNib[3:0]" output pins, upon the rising edge of this clock signal.
B3 C10 C19 D24	RxPOS_0 RxPOS_1 RxPOS_2 RxPOS_3	I	**Receive Positive Polarity Data Input:** The exact function of these input pins depend upon whether the XRT74L74 device is operating in the Single-Rail or Dual-Rail Mode. **Single-Rail Mode:** This input pin functions as the "Single-Rail" (e.g., binary data stream) input for the incoming DS3 or E3 data stream. This signal at this input pin will be sampled and latched upon the "user-selected" edge of the RxLineClk_n signal. **Dual-Rail Mode:** This input pin functions as one of the dual-rail inputs for the incoming B3ZS/ HDB3 encoded DS3 or E3 data, which as been received the external LIU IC. RxNEG_n functions as the other dual-rail input for the Framer/UNI IC. When this input pin is pulse "high", it means that the LIU IC has received a positive polarity pulse from the line.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Tx PLCP Processor			
G1 H3 H2 H1	TxNib_1_0/ Tx8KRef_0/ TxHDLCDat_1_0 TxNib_1_1/ Tx8KRef_1/ TxHDLCDat_1_1 TxNib_1_2/ Tx8KRef_2/ TxHDLCDat_1_2 TxNib_1_3/ Tx8KRef_3/ TxHDLCDat_1_3	I	**Transmit Nibble Input Interface - Bit 1/Transmit PLCP Framing 8kHz Reference Input/Transmit HDLC Controller Data Bus - Bit 1 Input:** The exact function of this input pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High-Speed HDLC Controller Mode, or in the ATM/PLCP Mode. **Clear-Channel Framer Mode - TxNib_1_n:** If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then this input pin will function as the bit 1 input to the "Transmit Nibble-Parallel" input interface. The Transmit Payload Data Input Interface block will sample this signal (along with TxNib_0_n, TxNib_2_n and TxNib_3_n) upon the falling edge of TxNibClk_n ***NOTE:*** *This input pin is inactive if the Channel is configured to operate in the "Serial" Mode.* **ATM/PLCP Mode - Tx8KREF_n:** If the XRT74L74 is configured to operate in the ATM/PLCP Mode, then the Transmit PLCP Processor can be configured to synchronize its PLCP frame generation to this input clock signal. The Transmit PLCP Processor will also use this input signal to compute the nibble-trailer stuff opportunities. ***NOTE:*** *This input pin is inactive if the use has configured the XRT74L74 device to operate in the "Direct-Mapped" ATM Mode.* **High-Speed HDLC Controller Mode - TxHDLCDat_1_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 1" within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
K3 K2 K1 L3	TxNib_2_0/ TxStuffCtl_0/ TxHDLCDat_2_0 TxNib_2_1/ TxStuffCtl_1/ TxHDLCDat_2_1 TxNib_2_2/ TxStuffCtl_2/ TxHDLCDat_2_2 TxNib_2_3/ TxStuffCtl_3/ TxHDLCDat_2_3	I	**Transmit Nibble Input Interface - Bit 2/Transmit PLCP Stuff Control Input/ Transmit HDLC Controller Data Bus - Bit 2 Input:** The exact function of this input pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High-Speed HDLC Controller Mode, or in the ATM/PLCP Mode. **Clear-Channel Framer Mode - TxNib_2_n**: If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then this input pin will function as the bit 1 input to the "Transmit Nibble-Parallel" input interface. The Transmit Payload Data Input Interface block will sample this signal (along with TxNib_0_n, TxNib_2_n and TxNib_3_n) upon the falling edge of TxNibClk_n ***NOTE:*** *This input pin is inactive if the Channel is configured to operate in the "Serial" Mode.* **ATM/PLCP Mode - TxStuffCtl_n:** This input pin permits the user to externally exercise or forego trailer nibble stuffing opportunities by the Transmit PLCP Processor. PLCP trailer nibble stuff opportunities occur in periods of three PLCP frames (375 us). The first PLCP frame (first, within a "stuff opportunity period) will have 13 trailer nibbles appended to it. The second PLCP frame (second within a "stuff opportunity" period) will have 14 trailer nibbles appended to it. The third PLCP frame (the location of the stuff opportunity) will contain 13 trailer nibbles if this input pin is pulled "low", and 14 trailer nibbles if this input pin is pulled "high". ***NOTE:*** *This input pin is inactive if the XRT74L74 device is configured to operate in the Direct-Mapped ATM Mode.* **High-Speed HDLC Controller Mode - TxHDLCDat_2_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 1" within this byte wide interface. Data residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
J3 J2 J1 J4	TxOHInd_0/ TxPFrame_0/ TxHDLCDat_6_0 TxOHInd_1/ TxPFrame_1/ TxHDLCDat_6_1 TxOHInd_2/ TxPFrame_2/ TxHDLCDat_6_2 TxOHInd_3/ TxPFrame_3/ TxHDLCDat_6_3	O	**Transmit Overhead Data Indicator Output/Transmit PLCP Frame Boundary Indicator Output/Transmit HDLC Controller Data Bit 6 input pin:** The function of these input/output pins depends upon whether the channel (within the XRT74L74 device) has been configured to operate in the "Clear-Channel Framer" Mode, the "ATM/PLCP" Mode or the "High-Speed HDLC Mode. **Clear-Channel Framer Mode - TxOHInd_n:** In the Clear-Channel Framer Mode, this output pin functions as the transmit overhead data indicator for the local terminal equipment. This output pin is pulsed "high" for one DS3 or E3 bit period in order to indicate (to the local terminal equipment) that the Transmit Section of the Framer is going to be processing an overhead bit, upon the next rising edge of TxInClk_n., and will NOT latch the data that is applied to the TxSer_n input pin. Therefore, when the local terminal equipment samples the "TxOHInd_n" output pin "high", then it must not apply the next payload bit to TxSer_n input pin. This output pins serves as a warning that this particular payload bit is going to be ignored by the Transmit Section of the Framer, and will not be inserted into payload bits, within the outbound DS3 or E3 data stream. **ATM/PLCP Mode - TxPFrame_n:** If the XRT74L74 device is configured to operate in the ATM UNI/PLCP Mode, then this output pin will denote the boundaries of "outbound" PLCP frames, as they are being processed by the Transmit PLCP Processor block. This output pulses "high" when the last nibble (of a given PLCP frame) is being routed to the Transmit DS3/E3 Framer block. ***NOTE:*** *This output pin is inactive if the XRT74L74 is operating in the "Direct-Mapped" ATM Mode.* **High-Speed HDLC Controller Mode - TxHDLCDat_6_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 6" within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
M2 M1 N3 N2	TxSer_0/ TxPOH_0/ SendMSG_0 TxSer_1/ TxPOH_1/ SendMSG_1 TxSer_2/ TxPOH_2/ SendMSG_2 TxSer_3/ TxPOH_3/ SendMSG_3	I	**Transmit Payload Data Serial Input/Transmit PLCP Path Overhead Input/ Send HDLC Message Request Input:** The function of this input pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High-Speed HDLC Controller Mode or in the ATM/PLCP Mode. **Clear-Channel Framer Mode - TxSer_n:** If the XRT74L74 device is configured to operate in the Clear-Channel Framer mode, then this input pin functions as the "Transmit Payload Data Serial Input" pin. In this case, the local terminal equipment is expected to apply all outbound data (which is intended to be carried via the DS3 or E3 payload bits) to this input pin.The Transmit Payload Data Input Interface will sample the data, residing at the "TxSer_n" input pin, upon the rising edge of TxInClk. **ATM/PLCP Mode - TxPOH_n:** If the XRT74L74 device is configured to operate in the ATM Mode, and if (within the ATM Mode, the chip is also configured to operate in the PLCP Mode), then this input pin functions as the "Transmit PLCP Path Overhead Input Pin". In this mode, the user can externally insert "desired" path overhead byte values into the "outbound" PLCP frames.The Transmit PLCP Path Overhead Input Pin (and Port) become active whenever the user asserts the "TxPOHIns" input pin (by pulling it "high"). In this case, the data, residing upon the "TxPOH_n" input pin will be sampled upon the rising edge of the "TxPOHClk" signal. ***NOTE:*** *This input pin is inactive if the XRT74L74 device is configured to operate in the Direct-Mapped ATM Mode.* **High-Speed HDLC Controller Mode - SendMSG_n:** If the XRT74L74 device is configured to operate in the "High-Speed HDLC Controller" Mode, then this input pin functions as the "Transmit HDLC Controller Input Interface" enable input pin. If the user asserts this input pin (by pulling it "high") then the "Transmit HDLC Controller Input Interface" will proceed to latch the data, residing on the "TxHDLCDat[7:0]" input pins, upon each rising edge of the "TxHDLCClk_n signal. All data that is latched into the "Transmit HDLC Controller Input Interface" (for the duration that the "SendMSG_n" input pin is "high") will be encapsulated into an HDLC frame and ultimately transported via the payload bits of the outbound DS3 or E3 data stream. If the user pulling this input pin "low", then the Transmit HDLC Controller Input Interface will cease latching the data, residing on the TxHDLCDat[7:0] bus. ***NOTE:*** *This input pin is inactive if the XRT74L74 device has been configured to operate in the PPP Mode.*
N1 N4 P3 P2	TxPOHClk_0 TxPOHClk_1 TxPOHClk_2 TxPOHClk_3	O	**Transmit PLCP Frame POH Byte Insertion Clock:** This pin, along with the TxPOH_n and the TxPOHMSB_n input pins, function as the "Transmit PLCP Frame POH Byte" serial input port. This output pin functions as a clock output signal that is be used to sample the user's POH data at the TxPOH_n input pin. This output pin is always active, independent of the state of the "TxPOHIns" pin. ***NOTE:*** *This pin is only active if the XRT74L74 device has been configured to operate in the ATM/PLCP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
L2 L1 L4 M3	TxPOHFrame_0 TxPOHFrame_1 TxPOHFrame_2 TxPOHFrame_3	O	**Transmit PLCP Frame Path Overhead Byte Serial Input Port - Beginning of Frame indicator:** This output pin, along with the TxPOH_n, TxPOHClk_n, and the TxPOHIns_n pins comprise the "Transmit PLCP Frame POH Byte Insertion" serial input port. This particular pin will pulse "high" when the "Transmit PLCP POH Byte Insertion" serial input port is expecting the first bit of the Z6 byte at the TxPOH_n input pin. ***NOTE:*** *This pin is only active if the XRT74L74 device has been configured to operate in the ATM/PLCP Mode.*
P1 R3 R2 R1	TxNib_3_0/ TxPOHIns_0/ TxHDLCDat_3_0 TxNib_3_1/ TxPOHIns_1/ TxHDLCDat_3_1 TxNib_3_2/ TxPOHIns_2/ TxHDLCDat_3_2 TxNib_3_3/ TxPOHIns_3/ TxHDLCDat_3_3	I	**Transmit Nibble Interface - Bit 3/Transmit PLCP Path Overhead Insert enable/Transmit HDLC Controller Data Bus - Bit 3 input:** The exact function of this input pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High-Speed HDLC Controller Mode or in the ATM/PLCP Mode. **Clear-Channel Framer Mode - TxNib_3_n:** If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then this input pin will function as the bit 3 (MSB) input to the "Transmit Nibble-Parallel" input interface. The Transmit Payload Data Input Interface block will sample this signal (along with TxNib_0_n through TxNib_2_n) upon the falling edge of TxNibClk_n. ***NOTE:*** *This input pin is inactive if the Channel is configured to operate in the "Serial" Mode.* **ATM/PLCP Mode - TxPOHIns_n:** If the XRT74L74 device is configured to operate in the ATM Mode, and if within the ATM Mode, the chip is also configured to operate in the PLCP Mode, then this input pin functions as the "Transmit PLCP Path Overhead Port - Enable input pin". In this mode, the user can externally insert "desired" path overhead byte values into the "outbound" PLCP frames.The Transmit PLCP Path Overhead Input port becomes active whenever the user asserts this input pin (by pulling it "high"). Once this occurs, the data, residing upon the "TxPOH_n" input pin will be sampled upon the rising edge of the "TxPOHClk" signal. ***NOTE:*** *This input pin is inactive if the XRT74L74 device is configured to operate in the Direct-Mapped ATM Mode.* **High-Speed HDLC Controller Mode - TxHDLCDat_3_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 3" within this byte wide interface. Data residing on the "Transmit HDLC Controller" byte wide input interface will be sampled upon the rising edge of the TxHDLCClk_n output signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Rx PLCP Processor			
U25 U26 T24 T25	RxPFrame_0/ RxOHInd_0 RxPFrame_1/ RxOHInd_1 RxPFrame_2/ RxOHInd_2 RxPFrame_3/ RxOHInd_3	O	**Receive PLCP Frame Indicator/Receive Overhead Indicator Output:** The exact function of this output pin depends upon whether the channel has been configured to operate in the ATM/PLCP, the Clear-Channel Framer/Serial or the Clear-Channel Framer/Nibble-Parallel Modes. **ATM/PLCP Mode - RxPFrame_n:** This output pin pulses "high" when the Receive PLCP Processor is receiving the last bit of a PLCP frame. ***NOTE:*** *This output pin is inactive if the XRT74L74 is configured to operate in the Direct-Mapped ATM Mode.* **Clear-Channel Framer/Serial Mode - RxOHInd_n:** This output pin pulse "high" (for one bit-period) whenever an "overhead" bit is being output via the "RxSer_n" output pin, by the Receive Payload Data Output Interface block. ***NOTE:*** *If the user configures the channel to operate in the "Gapped-Clock" Mode, then this output pin will provide a demand clock to the local terminal equipment. In the "Gapped-Clock" Mode, this output pin will only provide a clock pulse, whenever a payload bit is being output via the "RxSer_n" output pin. This output pin will NOT generate a clock pulse, whenever an overhead is being output via the "RxSer_n" output pin.* **Clear-Channel Framer/Nibble-Parallel - RxOHInd_n:** This output pin pulse "high" (for one nibble-period) whenever an overhead nibble is being output via the "RxNib_n[3:0] output pins, by the Receive Payload Data Output Interface block. ***NOTE:*** *The purpose of this output pin is to alert the local terminal equipment that an overhead bit (or nibble) is being output via the "RxSer_n" or "RxNib_n[3:0]" output pins and that this data should be ignored.*
L24 L25 L26 L23	RxPLOF_0 RxPLOF_1 RxPLOF_2 RxPLOF_3	O	**Receive PLCP - "Loss of Frame" Output Indicator:** The Receive PLCP Processor will assert this pin, when it declares a "Loss of Frame" condition. This output will be negated when the Receive PLCP Processor reaches the "In Frame" Condition. ***NOTE:*** *This output pin is only active is the XRT74L74 device has been configured to operate in the ATM/PLCP Mode.*
T26 T23 R24 R25	RxPOHFrame_0 RxPOHFrame_1 RxPOHFrame_2 RxPOHFrame_3	O	**Receive PLCP Frame POH Serial Output Port - Frame Indicator:** This output pin, along with the "RxPOH_n" "RxPOHClk_n" and "RxPOHIns_n" pins comprise the "Receive PLCP Frame POH Byte" serial output port. This output pin provides framing information to external circuitry receiving and processing this POH (Path Overhead) data, by pulsing "high" whenever the first bit of the Z6 byte is being output via the "RxPOH_n" output pin. This pin is "low" at all other times during this PLCP POH Framing cycle. ***NOTE:*** *This output pin is only active if the XRT74L74 device has been configured to operate in the ATM/PLCP Modes.*
V25 V26 V23 U24	RxPOOF_0 RxPOOF_1 RxPOOF_2 RxPOOF_3	O	**Receive PLCP "Out of Frame" Indicator:** The Receive PLCP Processor will assert this pin, when it declares an "Out of Frame" condition. This output will be negated when the Receive PLCP Processor reaches the "In Frame" Condition. ***NOTE:*** *This output pin is only valid if the XRT74L74 device has been configured to operate in the ATM/PLCP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
K24 K25 K26 J24	RxPRed_0 RxPRed_1 RxPRed_2 RxPRed_3	O	**Receiver Red Alarm Indicator - Receive PLCP Processor:** The Framer/UNI asserts this output pin to denote that one of the following events has been detected by the Receive PLCP Processor: • OOF - Out of Frame Condition • LOF - Loss of Frame Condition ***NOTE:*** *This output pin is only valid if the XRT74L74 device has been configured to operate in the ATM/PLCP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Tx Cell Processor			
AC16 AE17 AF17 AF18	TxCellTxed_0/ TxNibFrame_0/ ValidFCS_0 TxCellTxed_1/ TxNibFrame_0/ ValidFCS_1 TxCellTxed_2/ TxNibFrame_0/ ValidFCS_2 TxCellTxed_3/ TxNibFrame_0/ ValidFCS_3	O	**Transmit Cell Generator Indicator/Transmit Nibble Frame Indicator/Valid FCS Indicator output:** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM Mode, the Clear-Channel Framer Mode or in the High-Speed HDLC Controller Mode. **ATM Mode - TxCellTxed_n:** This output pin pulses "high" each time the Transmit Cell Processor transmits a cell to either the Transmit PLCP Processor or the Transmit DS3/E3 Framer block. **Clear-Channel Framer Mode - TxNibFrame_n:** This output pin pulses "high" when the last nibble of a given DS3 or E3 frame is expected at the TxNib_n[3:0] input pins. The purpose of this output pin is to alert the local terminal equipment that it needs to begin the transmission of a new DS3 or E3 frame to the XRT74L74 device. ***NOTE:*** *This output pin is not active if the channel is configured to operate in the "Serial-Mode".* **High-Speed HDLC Controller Mode - ValidFCS_n:** The combination of the RxIdle_n and ValidFCS_n output signals are used to convey information about data that is being output via the Receive HDLC Controller output Data bus (RxHDLCDat_[7:0]_n). *If RxIdle = HIGH;* The Receive HDLC Controller block with drive this output pin "high" anytime the flag sequence octet (0x7E) is present on the "RxHDLCDat_n[7:0]" output data bus. *If RxIdle_n and ValidFCS_n are both "high"* The Receive HDLC Controller block has received a complete HDLC frame, and has determined that the FCS value (within this HDLC frame) are valid. *If RxIdle_n is "high" and "ValidFCS_n" is "low"* The Receive HDLC Controller block has received a complete HDLC frame, and has determined that the FCS value (within this HDLC frame) is invalid. *If "RxIdle_n" is "high" and "ValidFCS_n" is "low"* The Receive HDLC Controller block has received an ABORT sequence.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AD7 AE7 AF7 AE8	TxNib_0_0/ TxGFC_0/ TxHDLCDat_0_0 TxNib_0_1/ TxGFC_1/ TxHDLCDat_0_1 TxNib_0_2/ TxGFC_2/ TxHDLCDat_0_2 TxNib_0_3/ TxGFC_3/ TxHDLCDat_0_3	I	**Transmit Nibble Interface - Bit 0/Transmit GFC Input pin/Transmit HDLC Controller Data Bus - Bit 0 Input:** The exact function of this input pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High Speed HDLC Controller Mode or in the ATM Mode. Clear-Channel Framer Mode - TxNib_0_n: If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then this input pin will function as the bit 0 (LSB) input to the "Transmit Nibble-Parallel" input interface. The Transmit Payload Data Input Interface block will sample this signal (along with TxNib_1_n through TxNib_3_n) upon the falling edge of TxNibClk_n. ***NOTE:*** *This input pin is inactive if the Channel is configured to operate in the "Serial" Mode.* **ATM Mode - TxGFC_n:** This signal, along with TxGFCMSB_n and TxGFCClk_n 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 ATM cell, by serially transmitting its four bit-value into this input pin. Each of these four bits will be clocked into the port upon the rising edge of the TxGFCClk_n output signal. **High-Speed HDLC Controller Mode - TxHDLCDat_0_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 0" (the LSB) within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.
AF5 AE6 AF6 AC6	TxGFCClk_0 TxGFCClk_1 TxGFCClk_2 TxGFCClk_3	O	**Transmit GFC Nibble-Field Serial Input port - Clock Output signal:** This signal along with TxGFC_n and TxGFCMSB_n combine to function as the "Transmit GFC Nibble-field" serial input port. This output signal functions as the demand clock signal for this port. The user will specify the value of the GFC field, within a given ATM cell, by serially transmitting its four bit-value into the "TxGFC_n" input pin. The Transmit GFC Nibble-Field" serial input port will latch the contents of "TxGFC_n" upon the rising edge of this clock signal. Hence, the local terminal equipment should be designed to place its "outbound" GFC bits on to the "TxGFC_n" line, upon the falling edge of this clock signal. ***NOTE:*** *This output pin is only active if the XRT74L74 device has been configure to operate in the ATM Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AF8 AE9 AF9 AE10	TxNibClk_0/ TxGFCMSB_0/ SendFCS_0 TxNibClk_1/ TxGFCMSB_1/ SendFCS_1 TxNibClk_2/ TxGFCMSB_2/ SendFCS_2 TxNibClk_3/ TxGFCMSB_3/ SendFCS_3	O	**Transmit Nibble Clock Output pin/Transmit GFC Byte - MSB Indicator Output/Send FCS Value Request Input:** The exact function of this input/output pin depends upon whether the XRT74L74 device is configured to operate in the Clear-Channel Framer Mode, the High-Speed HDLC Controller Mode or in the ATM Mode. **Clear-Channel Framer Mode - TxNibClk_n** If the user opts to operate the XRT74L74 device in the Nibble-Parallel Mode, then the XRT74L74 device will derive this clock signal from either the "TxInClk" or the "RxLineClk" signal (depending upon whether the chip is operating in the "Local-Timing" or "Loop-Timing" Mode). The user is advised to configure the Terminal Equipment to output the "outbound" payload data (to the XRT74L74 device) onto the "TxNib_[3:0]_n" input pins, upon the rising edge of this clock signal. The Transmit Payload Data Input Interface block will sample the data, residing on the "TxNib_[3:0]_n line, upon the falling edge this clock signal. ***NOTES:*** *1. For DS3 applications, the XRT74L74 device will output 1176 clock pulses (to the local terminal equipment) for each "outbound" DS3 frame.* *2. For E3, ITU-T G.832 applications, the XRT74L74 device will output 1074 clock pulses (to the local terminal equipment) for each "outbound" E3 frame.* *3. For E3, ITU-T G.751 applications, the XRT74L74 device will output 384 clock pulses (to the local terminal equipment) for each "outbound" E3 frame.* **ATM Mode - TxGFCMSB_n:** 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 "outbound" cell) is expected at the TxGFC_n input pin. **High-Speed HDLC Controller Mode - SendFCS_n:** The local terminal equipment is expected to control both this input pin along with the SendMSG input pin during the construction and transmission of each outbound HDLC frame.This input pin permits the user to command the Transmit HDLC Controller block to compute and insert the computed FCS (Frame-Check Sequence) value into the back-end of the outbound HDLC frame as a trailer.If the user has configured the Transmit HDLC Controller block to compute and insert a CRC-16 value into the outbound HDLC frame, then the local terminal equipment is expected to hold this input pin "high" for two periods of "TxHDLCClk_n". Conversely, if the user has configured the Transmit HDLC Controller block to compute and insert a CRC-32 value into the outbound HDLC frame, then the local terminal equipment is expected to hold this input pin "high" for four (4) periods of "TxHDLCClk_n". ***NOTES:*** *1. This input/output pin is inactive if the XRT74L74 device has been configured to operate in the PPP Mode.* *2. This input/output pin is inactive if the XRT74L74 device has been configured to operate in the "Clear-Channel Framer/Serial mode".*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Rx Cell Processor			
AD19 AE19 AF19 AD20	RxCellRxed_0 RxCellRxed_1 RxCellRxed_2 RxCellRxed_3	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 PLCP Processor or the Receive DS3/E3 Framer block. ***NOTE:*** *This output pin is only active if the XRT74L74 device has been configured to operate in the ATM UNI Mode.*
AF23 AF25 AD26 AC25	RxGFC_0/ RxIdle_0 RxGFC_1/ RxIdle_0 RxGFC_2/ RxIdle_0 RxGFC_3/ RxIdle_0	O	**Receive GFC Nibble Field - Output Pin/Receive Idle Sequence Indicator:** The exact function of this output pin depends upon whether the XRT74L74 device is operating in the ATM Mode or in the High-Speed HDLC Controller Mode. **ATM Mode - RxGFC_n:** 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 cell that is processed via the Receive Cell Processor. This data is serially clocked out of this pin on the rising edge of the RxGFCClk signal. The MSB of each GFC value is designated by a pulse at the "RxGFCMSB_n" output pin. **High-Speed HDLC Controller Mode - RxIdle_n:** The combination of the RxIdle_n and ValidFCS_n output signals are used to convey information about data that is being output via the Receive HDLC Controller output Data bus (RxHDLCDat_[7:0]_n). *If RxIdle = HIGH;* The Receive HDLC Controller block with drive this output pin "high" anytime the flag sequence octet (0x7E) is present on the "RxHDLCDat_[7:0]_n" output data bus. *If RxIdle_n and ValidFCS_n are both "high"* The Receive HDLC Controller block has received a complete HDLC frame, and has determined that the FCS value (within this HDLC frame) are valid. *If RxIdle_n is "high" and "ValidFCS_n" is "low"* The Receive HDLC Controller block has received a complete HDLC frame, and has determined that the FCS value (within this HDLC frame) is invalid. *If "RxIdle_n" is "high" and "ValidFCS_n" is "low"* The Receive HDLC Controller block has received an ABORT sequence.
Y24 Y25 Y26 Y23	RxGFCClk_0 RxGFCClk_1 RxGFCClk_2 RxGFCClk_3	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_n and RxGFCMSB_n pins. This pin provides a clock pulse which allows external circuitry to latch in the GFC Nibble-Data via the RxGFC_n output pin. ***NOTE:*** *This output pin is only active if the XRT74L74 device is operating in the ATM UNI Mode.*
AB24 AB26 AA25 AA26	RxGFCMSB_0 RxGFCMSB_1 RxGFCMSB_2 RxGFCMSB_3	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. ***NOTE:*** *This output pin is only active if the XRT74L74 has been configured to operate in the “ATM UNI” Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
W24 W25 W26 V24	RxLCD_0/ RxOutClk_0/ RxHDLCDat_7_0 RxLCD_1/ RxOutClk_1/ RxHDLCDat_7_1 RxLCD_2/ RxOutClk_2/ RxHDLCDat_7_2 RxLCD_3/ RxOutClk_3/ RxHDLCDat_7_3	O	**Receive Loss of Cell Delineation indicator/Receive Output Clock signal/ Receive HDLC Controller Data Bus - Bit 7 Output:** The exact function of output pin depends upon whether the channel has been configured to operate in the ATM, Clear-Channel Framer or High Speed HDLC Controller Mode. **ATM Mode - RxLCD_n:** 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. **Clear-Channel Framer Mode - RxOutClk_n:** This clock signal functions as the Transmit Payload Data Input Interface clock source, if the channel has been configured to operate in the "local-timing" mode.In this mode, the local terminal equipment is expected to input data to the TxSer_n input pin, upon the rising edge of this clock signal. The channel will use the rising edge of this signal to sample the data on the TxSer_n input. **High-Speed HDLC Controller Mode - RxHDLCDat_7_n:** This output pin, along with RxHDLCDat_[6:0]_n function as the Receive HDLC Controller byte wide output data bus. This particular output pin functions ass the MSB (Most Significant Bit) of the Receive HDLC Controller byte wide data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Tx UTOPIA Interface			
AC11 AF11 AE11 AD11 AF10	TxUAddr0 TxUAddr1 TxUAddr2 TxUAddr3 TxUAddr4	I	**Transmit UTOPIA Address Bus:** These input pins comprise the Transmit UTOPIA Address Bus input pins. The Transmit UTOPIA Address Bus is only in use when the XRT74L74 is operating in the Multi-PHY mode. When the ATM Layer processor wishes to write data to a particular UNI (PHY-Layer) 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 TxUClk. The UNI will compare the data on the Transmit UTOPIA Address Bus with the pre-programmed contents of the TxUT Address Register (Address = 70h). If these two values are identical and the $\overline{\text{TxUEn}}$ pin is asserted, then the TxUClav pin will be driven to the appropriate state (based upon the TxFIFO fill level) for the Cell Level handshake mode of operation.
AD8	TxUClav/ TxPPA	O	**Transmit UTOPIA Interface - Cell Available Output Pin/Transmit POS-PHY Interface - Packet Data Available Output pin:** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or PPP Mode. **ATM UNI Mode - TxUClav** This output pin supports data flow control between the ATM Layer processor and the Transmit UTOPIA Interface block. 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 TxUClk 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. **PPP Mode - TxPPA** The XRT74L74 device will drive this output pin "high" whenever a (programmable) number of bytes of empty space is available (for writing more packet data) into the TxFIFO.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AD9	$\overline{\text{TxUEn}}$/ $\overline{\text{TxPEn}}$	I	**Transmit UTOPIA Interface Block - Write Enable/Transmit POS-PHY Interface - Write Enable:** The exact function of this input pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or PPP Mode. **ATM UNI Mode Operation - $\overline{\text{TxUEn}}$** 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 TxUClk. 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 TxUClk. When this signal is negated, then the Transmit UTOPIA Data bus inputs will be tri-stated. **PPP Mode Operation - $\overline{\text{TxPEn}}$** This active-low signal, from the Link Layer processor enables the data on the Transmit POS-PHY Data Bus to be written into the TxFIFO on the rising edge of TxPClk. When this signal is asserted, then the contents of the byte or word that is present, on the Transmit POS-PHY Data Bus, will be latched into the Transmit POS-PHY Interface block, on the rising edge of TxPClk.When this signal is negated, then the Transmit POS-PHY Data bus inputs will be tri-stated.
AD10	TxUClk/ TxPClk	I	**Transmit UTOPIA Interface Clock/Transmit POS-PHY Interface Clock Input:** The exact function of this input pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or in the PPP Mode. **ATM UNI Mode - TxUClk** 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 sampled on the rising edge of TxUClk. **PPP Mode - TxPClk** The Transmit POS-PHY Interface clock is used to latch the data on the Transmit POS-PHY Data bus, into the Transmit POS-PHY Interface block. This clock signal is also used as the timing source for circuitry used to process the Packet data into and through the TxFIFO.
AA10	TxUClkO/ TxPClkO	O	**Transmit UTOPIA Interface Clock/Transmit POS-PHY Interface Clock Output:** This output pin is derived from an internal PLL.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AF16 AE16 AD16 AF15 AE15 AD15 AC14 AF14 AD14 AF13 AE13 AD13 AC12 AF12 AE12 AD12	TxUData0/ TxPData0 TxUData1/ TxPData1 TxUData2/ TxPData2 TxUData3/ TxPData3 TxUData4/ TxPData4 TxUData5/ TxPData5 TxUData6/ TxPData6 TxUData7/ TxPData7 TxUData8/ TxPData8 TxUData9/ TxPData9 TxUData10/ TxPData10 TxUData11/ TxPData11 TxUData12/ TxPData12 TxUData13/ TxPData13 TxUData14/ TxPData14 TxUData15/ TxPData15	I	**Transmit UTOPIA Data Bus Inputs/Transmit POS-PHY Data Bus Inputs:** The exact function of these input pins depends upon whether the XRT74L74 is operating in the ATM UNI Mode or in the PPP Mode. **ATM UNI Operation - TxUData[15:0]** These input pins comprise the Transmit UTOPIA Data Bus input pins. When the ATM Layer Processor wishes to transmit ATM cell data through the XRT72L74 ATM 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 upon the rising edge of TxUClk. **PPP Operation - TxPDATA[15:0]** These input pins comprise the Transmit POS-PHY Data Bus input pins. When a Network Processor wishes to transmit PPP data through the XRT74L74 Framer/UNI IC, it must place this data on these pins. The data, on the Transmit POS-PHY Data Bus is latched into the Transmit POS-PHY Interface block upon the rising edge of TxPClk.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AE14	TxUPrty/TxPPrty	I	**Transmit UTOPIA Data Bus - Parity Input/Transmit POS-PHY Interface - Parity Input:** The exact function of this input pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or PPP Mode. **ATM UNI Mode - TxUPrty:** 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., TxUData[7:0] or TxUData[15:0]) inputs of the XRT74L74, respectively. ***NOTE:*** *This parity value should be computed based upon the odd-parity of the data applied at the Transmit UTOPIA Data Bus. The Transmit UTOPIA Interface block (within the XRT74L74 device) will independently compute an 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.* **PPP Mode - TxPPrty:** The Link Layer Processor will apply the parity value of the byte or word which is being applied to the Transmit POS-PHY Data Bus (e.g., TxPData[7:0] or TxPData[15:0]) inputs of the XRT74L74, respectively. ***NOTE:*** *This parity value should be computed based upon the odd-parity of the data applied to the Transmit POS-PHY Data Bus. The Transmit POS-PHY Interface block (within the XRT74L74 device) will independently compute an odd-parity value of each byte (or word) that it receives from the Link Layer processor and will compare it will the logic level of this input pin.*
AC9	TxUSoC/TxPSOP	I	**Transmit UTOPIA - Start of Cell Input/Transmit POS-PHY - Start of Packet Input:** The exact function of this input signal depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or in the PPP Mode. **ATM UNI Mode Operation - TxUSoC** 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 Processor. This input pin must be pulsed "high" whenever the first byte (or word) of a new cell is present on the Transmit UTOPIA Data Bus (TxUData[15:0]). This input pin must remain "low" at all other times. **PPP Mode Operation - TxPSoP/TxPSoC** If the XRT74L74 device has been configured to operate in the "Packet-Mode", then this input pin is pulsed "high" to denote that the first byte (or word) of a given packet is placed on the "TxPData[15:0]" input pins.If the XRT74L74 device has been configured to operate in the Cell-Chunk Mode, then this input pin is pulsed "high" to denote that the first byte of a packet chunk, if placed on the "TxPData[15:0]" input pins. ***NOTE:*** *This input pin is only valid if the XRT74L74 device has been configured to operate in the PPP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
Rx UTOPIA Interface			
AD22 AF21 AE21 AC20 AF20	RxUAddr0 RxUAddr1 RxUAddr2 RxUAddr3 RxUAddr4	I	**Receive UTOPIA Address Bus input (MSB):** These input pins functions as the Receive UTOPIA Address bus inputs. These input pins are only active when the Framer/UNI device is operating in the ATM UNI 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 = 0x6C). 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.
AE18	RxUClav/ RxPPA	O	**Receive UTOPIA - Cell Available/Receive POS-PHY Interface - Packet Available:** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or PPP Mode. **ATM UNI Mode - RxUClav** 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. 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 handshake mode. **PPP Mode - RxPPA** The XRT74L74 device will pulse this output pin "high" whenever a (programmable) number of bytes are available to be read from the RxFIFO.
AD17	RxUClk/ RxPClk	I	**Receive UTOPIA Interface Clock Input/Receive POS-PHY Interface Clock Input:** The exact function of this input pin depends upon whether the XRT74L74 device is operating in the ATM UNI or PPP Mode. **ATM UNI Mode - RxUClk** The byte (or word) data, on the Receive UTOPIA Data bus (RxUData[15:0]) is updated on the rising edge of this signal. The Receive UTOPIA Interface can be clocked at rates up to 50 MHz. **PPP Mode - RxPClk** This byte (or word) data, on the Receive POS-PHY Data Bus (RxPData[15:0]) is updated on the rising edge of this signal. The Receive POS-PHY Interface can be clocked at rates up to 50MHz.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AF26 AC26 AB25 AA24 AE26 AD25 AC24 AB23 AE24 AD23 AF24 AE22 AC22 AF22 AE20 AD21	RxUData0/ RxPData0 RxUData1/ RxPData1 RxUData2/ RxPData2 RxUData3/ RxPData3 RxUData4/ RxPData3 RxUData5/ RxPData4 RxUData6/ RxPData5 RxUData7/ RxPData7 RxUData8/ RxPData8 RxUData9/ RxPData9 RxUData10/ RxPData10 RxUData11/ RxPData11 RxUData12/ RxPData12 RxUData13/ RxPData13 RxUData14/ RxPData14 RxUData15/ RxPData15	O	**Receive UTOPIA Data Bus Input/Receive POS-PHY Data Bus Output pins:** The exact function of these output pins depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or in the PPP Mode. **ATM UNI Mode - RxUData[15:0]** These output pins function as the Receive UTOPIA Data Bus. ATM cell data that has been received from the Remote Terminal Equipment is output on the Receive UTOPIA Data Bus, where it can be read and processed by the ATM Layer Processor. **PPP Mode - RxPData[15:0]** These output pins function as the Receive POS-PHY Data Bus output pins. PPP Packet data that has been received from the Remote Terminal Equipment is output on the Receive POS-PHY Data Bus, where it can be reads and processed by the Link Layer Processor.
AC18	$\overline{\text{RxUEn}}$/$\overline{\text{RxPEn}}$	I	**Receive UTOPIA Interface - Output Enable/Receive POS-PHY Interface - Output Enable** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or PPP mode. **ATM UNI Mode - $\overline{\text{RxUEn}}$:** 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 RxUClk. **PPP Mode - $\overline{\text{RxPEn}}$** This active-low input signal is used to control the drivers of the Receive POS-PHY Data Bus. When this signal is "high" (negated) then the Receive POS-PHY 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 POS-PHY Data bus on the very next rising edge of RxPClk.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AE23	RxUPrty/ RxPPrty	O	**Receive UTOPIA Interface - Parity Output pin/Receive POS-PHY Interface - Parity Output:** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or the PPP Modes. **ATM UNI Mode - RxUPrty** The Receive UTOPIA interface block will compute the odd-parity value of each byte (or word) that it will place in 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. **PPP Mode - RxPPrty** The Receive POS-PHY Interface block will compute the odd-parity value of each byte (or word) that it will place in the Receive POS-PHY Data Bus. This odd parity value will be output on this pin, which the corresponding byte (or word) is present on the Receive POS-PHY Data Bus.
AD18	RxUSoC/ RxPSOP	O	**Receive UTOPIA Interface - Start of Cell Indicator/Receive POS-PHY Interface - Start of Packet Indicator:** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the ATM UNI or in the PPP Mode. **ATM UNI Mode - RxUSoC** 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; RxUData[15:0]. **PPP Mode - RxPSOP** This output pin allows the Link Layer Processor to determine the boundaries of the PPP packets that are output via the Receive POS-PHY Data Bus. The Receive POS-PHY Interface block will assert this signal when the first byte (or word) of a new packet is present on the Receive POS-PHY Data Bus, RxP-Data[15:0].

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AC15	RxMod_0	O	**Receive PPP Data Bus - Modulus Indicator:** The XRT74L74 device will indicate the number of valid packet octets that are being read out of the RxPData[15:0] output pins.The XRT74L74 device will drive this output pin "low" when both bytes (of the RxPData[15:0] data bus) consists of valid packet data. Conversely, the XRT74L74 device will drive this output pin "high" when only the upper byte (of the RxPData[15:0] data bus) consists of valid packet data. The Link Layer Processor is expected to validate all packet data (that it reads out of the RxPData[15:0] output pins) by also reading the state of this output pin. ***NOTES:****This output pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.*
AC19	RxPEOP	O	**Receive POS-PHY Interface - End of Packet:** The XRT74L74 device drives this output pin "high" whenever the last byte of a given Packet is being output via the "RxPData[15:0] data bus. ***NOTES:*** *1. This output pin is only valid when the XRT74L74 device is configured to operate in the PPP Mode.* *2. This output pin is only valid when the "Receive POS-PHY Interface - Read Enable Output pin".*
AC8	TxPEOP	I	**Transmit POS-PHY Interface - End of Packet:** The link layer processor toggles this output pin "high" whenever the Link Layer Processor is writing the last byte (or word) of a given Packet into the TxP-Data[15:0] data bus. ***NOTES:*** *1. This input pin is only valid when the XRT74L74 device is configured to operate in the PPP Mode.* *2. This input pin is only valid when the "Transmit POS-PHY Interface - Write Enable Input pin (TxPEnb*) is asserted.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
C3 D4 D6 D8	TxOHENable_0/ TxHDLCDat_7_0 TxOHENable_1/ TxHDLCDat_7_1 TxOHENable_2/ TxHDLCDat_7_2 TxOHENable_3/ TxHDLCDat_7_3	I	**Transmit Overhead Enable Output indicator/Transmit HDLC Controller Data Bit 7 Input:** The function of this input pin depends upon whether the XRT74L74 device is configured to operate in the "High Speed HDLC Controller Mode or not. **Non-High Speed HDLC Controller Mode - TxOHEnable_n:** The Channel will assert this output pin for one "TxInClk" period just prior to the instant that the Transmit Overhead Data Input Interface will be sampling and processing an overhead bit. If the local terminal equipment intends to insert its own value for an overhead bit into the outbound DS3 or E3 data stream, then it is expected to sample the state of this signal, upon the falling edge of "TxInClk". Upon sampling the "TxOHEnable_n" signal high, the local terminal equipment should (1) place the desired value of the overhead bit, onto the "TxOH_n" input pin and (2) assert the "TxOHIns_n" input pin. The Transmit Overhead Data Input Interface block will sample and latch the data on the "TxOH_n" signal, upon the rising edge of the very next "TxInClk_n" input signal. **High-Speed HDLC Controller Mode - TxHDLCDat_7_n:** If the channel is configured to operate in the High-Speed HDLC Controller mode, then the local terminal equipment will be provided with a "byte-wide" Transmit HDLC Controller byte-wide input interface. This input pin will function as "Bit 7" (the MSB) within this byte wide interface. Data, residing on the "Transmit HDLC Controller" byte wide input interface, will be sampled upon the rising edge of the TxHDLCClk_n output signal.
AC13	TxMod_0	I	**Transmit PPP Data Bus - Modulo Indicator:** This input pin permits the user to specify the number of valid packet octets are being placed on the TxPData[15:0] input pins.The Link Layer Processor is expected to set this input pin "low" when both bytes (on the TxPData[15:0] data bus) is valid packet data. Conversely, the Link Layer Processor is expected to set this input pin "high" when only the upper octet has valid packet data. ***NOTES:*** *1. This input pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.* *2. The Link Layer Processor is expected to set this input pin to the appropriate state, as each 16-bit word is being written into the TxPData[15:0] data bus.*
AC10	TxTSX/TxPSOF	I	**Transmit - Start of Transfer/Transmit - Start of PPP Packet (in Chunk Mode):** The exact function of this input pin depends upon whether the XRT74L74 device has been configured to operate in the Packet Mode or Cell-Chunk Mode. **Packet Mode - TxTSX** The Link-Layer processor pulses this input pin "high" when an "in-band" port address is present on the "TxPData[7:0]" bus. When this input pin and "TxPENB*" are both set "high" then the value of "TxPData[7:0]" is the address value of the TxFIFO to be selected. Subsequent write operations, into "TxPData[15:0]" will fill the TxFIFO corresponding to this "inband" address. **Chunk Mode - TxPSOF** The Link Layer processor pulses this input pin "high" in order to indicate that the first byte (or word) of a given Packet is placed on the "TxPData[15:0]" pins. ***NOTE:*** *This input pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
AC17	RxTSX/RxPSOF	O	**Receive - Start of Transfer/Receive - Start of PPP Packet (in Chunk Mode):** The exact function of this output pin depends upon whether the XRT74L74 device has been configured to operate in the Packet Mode or Cell-Chunk Mode. **Packet Mode - RxTSX** The XRT74L74 device pulses this output pin "high" when an inband port address is present on the "RxPData[7:0]" bus.When this output pin is "high", the value of "RxPData[7:0]" is the address value of the "RxFIFO" to be selected. Subsequent read operations, from "RxPData[15:0]" will be from the RxFIFO corresponding to this "inband" address. **Chunk Mode - RxPSOF** The XRT74L74 device pulses this output pin "high" in order to indicate that the first byte (or word) of a given Packet is placed on the "RxPData[15:0]" pins. ***NOTE:*** *This output pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.*
AE25	RxPDVAL	O	**Receive POS-PHY Interface Signal Valid Indicator:** This output signal indicates whether or not the Receive POS-PHY Interface signals (e.g., PRData[15:0], RxPSOP, RxPEOP, RxPPrty, RxPERR) are valid.This output pin will be driven "high", when these signals are valid. Conversely, this output pin will be driven "low" when these signals are NOT valid. ***NOTE:*** *This output pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.*
H23 F23 C24 D23	RxOHEnable_0/ RxHDLCDat_5_0 RxOHEnable_1/ RxHDLCDat_5_1 RxOHEnable_2/ RxHDLCDat_5_2 RxOHEnable_3/ RxHDLCDat_5_3	O	**Receive Overhead Data Output Interface - Enable Output/Receive HDLC Controller Data Bus - Bit 5 output:** The exact function of this output pin depends upon whether the channel has been configured to operate in the "Clear-Channel Framer" Mode or in the "High-Speed HDLC Controller" Mode. **Clear-Channel Framer Mode - RxOHEnable_n:** The channel will assert this output signal for one "RxOHClk_n" period when it is safe for the local terminal equipment to sample the data on the "RxOH_n" output pin. **High-Speed HDLC Controller Mode - RxHDLCDat_5_n:** This output pin, along with RxHDLCDat_[4:0]_n, RxHDLCDat_6_n and RxHDLCDat_7_n function as the Receive HDLC Controller byte wide output data bus. The Receive HDLC Controller will output the contents of all HDLC frames via this output data bus, upon the rising edge of the "RxHDLCClk_n" output signal. Hence, the user's local terminal equipment should be designed/configured to sample this data upon the falling edge of the "RxHDLCClk_n" output clock signal.
AD24	RxPERR	O	**Receive POS-PHY Interface - Error Indicator:** This output pin indicates whether or not the Receive POS-PHY Interface has detected an error in the inbound PPP Packet.This output pin toggles "high" if the Receive Section of the XRT74L74 device detects an FCS Error, an ABORT sequence, or a Runt Packet. ***NOTE:*** *This output pin is only valid if the XRT74L74 device has been configured to operate in the PPP Mode.*
W23	RxUClkO/ RxPClkO	O	**Receive UTOPIA Interface Clock/Receive POS-PHY Interface Clock Output:** This output pin is derived from an internal PLL.

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
W4	TxPERR	I	**Transmit Error Indicator from Link Layer:** This input signal is used to indicate that the current packet is ABORTED and must be discarded. This input pin should only be asserted when the last byte (or word) is be written onto the "TxPData[15:0]" input pins. ***NOTE:*** *This input pin is only active if the XRT74L74 device has been configured to operate in the PPP Mode.*

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
NC, Power and Ground			
D10 D13 D15 D17 D19 F4 H4 M4 K4 W1 W2 W3 AA3 AB2 AB4 AC1 AC3 AC4 AC21 AC23 AD2 AE1 Y4	NC		**No Connection**

PIN DESCRIPTION

PIN#	NAME	TYPE	DESCRIPTION
L11 L12 L15 L16 M11 M12 M15 M16 N11 N12 N15 N16 P11 P12 P15 P16 R11 R12 R15 R16	VDD	***	**3.3VPower Supply Pins**
L13 L14 M13 M14 N13 N14 P13 P14 R13 R14 T11 T12 T13 T14 T15 T16	GND	***	**Ground**

1.0 REGISTER MAP OF THE XRT74L74

COMMONCONTROL REGISTERS OF THE XRT74L74

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
COMMON CONTROL REGISTERS			
0x0100	Operation Control Register - Byte 3	R/W	0x00
0x0101	Operation Control Register - Byte 2	R/W	0x00
0x0102	Operation Control Register - Byte 1	R/W	0x00
0x0103	Operation Control Register - Byte 0	R/W	0x00
0x0104	Device ID Register	R/W	0x7A
0x0105	Revision ID Register	R/W	0x01
0x0106 - 0x0111	Reserved		
0x0112	Operation Block Interrupt Status Register - Byte 1	RO	0x00
0x0113	Operation Block Interrupt Status Register - Byte 0	RO	0x00
0x0114 - 0x0115	Reserved		
0x0116	Operation Block Interrupt Enable Register - Byte 1	R/W	0x00
0x0117	Operation Block Interrupt Enable Register - Byte 0	R/W	0x00
0x0118	Reserved		
0x0119	Channel Interrupt Indicator - Receive Cell Processor/PPP Processor Block	R/O	0x00
0x011A - 0x011C	Reserved		
0x011D	Channel Interrupt Indicator - LIU/Jitter Attenuator Block	R/O	0x00
0x011E - 0x0120	Reserved		
0x0121	Channel Interrupt Indicator - Transmit Cell Processor/PPP Processor Block	R/O	0x00
0x0122 - 0x0126	Reserved		
0x0127	Channel Interrupt Indicator - DS3/E3 Framer Block - Byte 0	R/O	0x00
0x0128 - 0x0146	Reserved		
0x0147	Operation General Purpose Input/Output Register	R/W	0x00
0x0148 - 0x014A	Reserved		
0x014B	Operation General Purpose Input/Output Direction Register	R/W	0x00
0x014C - 0x04FF	Reserved		
0x0501	Receive POS-PHY Control Register - Byte 1	R/W	0x00
0x0502	Receive POS-PHY Control Register - Byte 0	R/W	0x00
0x0503	Receive UTOPIA Control Register	R/W	0x00
0x0504 - 0x0512	Reserved		

COMMONCONTROL REGISTERS OF THE XRT74L74

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
COMMON CONTROL REGISTERS			
0x0513	Receive UTOPIA Port Address Register		
0x0514 - 0x0516	Reserved		
0x0517	Receive UTOPIA Port Number Register	R/W	0x00
0x0518 - 0x0580	Reserved		
0x0581	Transmit POS-PHY Control Register - Byte 1	R/W	0x00
0x0582	Transmit POS-PHY Control Register - Byte 0	R/W	0x00
0x0583	Transmit UTOPIA Control Register	R/W	0x00
0x0584 - 0x0592	Reserved		
0x0593	Transmit UTOPIA Port Address Register	R/W	0x00
0x0594 - 0x0596	Reserved		
0x0597	Transmit UTOPIA Port Number Register	R/W	0x00

CLEAR-CHANNEL FRAMER BLOCK REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
CLEAR-CHANNEL FRAMER BLOCK REGISTERS			
0xn100	Operating Mode Register	R/W	0x2B
0xn101	I/O Control Register	R/W	0xC0
0xn102 - 0xn103	Reserved		
0xn104	Block Interrupt Enable Register	R/W	0x00
0xn105	Block Interrupt Status Register	R/O	0x00
0xn106 - 0xn10B	Reserved		
0xn10C	DS3 Test Register	R/W	0x00
0xn10D	Payload HDLC Control Register	R/W	0x00
0xn10E - 0xn10F	Reserved		
0xn110	RxDS3 Configuration and Status RegisterRxE3 Configuration and Status Register # 1 (G.832 & G.751)	R/O	0x12
0xn111	RxDS3 Status RegisterRxE3 Configuration and Status Register # 2 (G.832 & G.751)	R/O	0x00
0xn112	RxDS3 Interrupt Enable RegisterRxE3 Interrupt Enable Register 1 (G.832 & G751)	R/W	0x00

CLEAR-CHANNEL FRAMER BLOCK REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
CLEAR-CHANNEL FRAMER BLOCK REGISTERS			
0xn113	RxDS3 Interrupt Status RegisterRxE3 Interrupt Enable Register # 2 (G.832 & G.751)	RUR	0x00
0xn114	RxDS3 Sync Detect RegisterRxE3 Interrupt Status Register # 1 (G.832 & G.751)	R/W & RUR	0x00
0xn115	RxE3 Interrupt Status Register # 2 (G.832 & G.751)	RUR	0x00
0xn116	Reserved		
0xn117	RxDS3 FEAC Interrupt Enable and Status Register	R/W & RUR	0x00
0xn118	RxE3 LAPD Control Register	R/W & RUR	0x00
0xn119	RxLAPD Status Register	R/O	0x00
0xn11A	RxE3 NR Byte Register (G.832)RxE3 Service Bits Register (G.751)	R/O	0x00
0xn11B	RxE3 GC Byte Register (G.832)	R/O	0x00
0xn11C	RxE3 TTB Register # 0 (G.832)	R/O	0x00
0xn11D	RxE3 TTB Register # 1 (G.832)	R/O	0x00
0xn11E	RxE3 TTB Register # 2 (G.832)	R/O	0x00
0xn11F	RxE3 TTB Register # 3 (G.832)	R/O	0x00
0xn120	RxE3 TTB Register # 4 (G.832)	R/O	0x00
0xn121	RxE3 TTB Register # 5 (G.832)	R/O	0x00
0xn122	RxE3 TTB Register # 6 (G.832)	R/O	0x00
0xn123	RxE3 TTB Register # 7 (G.832)	R/O	0x00
0xn124	RxE3 TTB Register # 8 (G.832)	R/O	0x00
0xn125	RxE3 TTB Register # 9 (G.832)	R/O	0x00
0xn126	RxE3 TTB Register # 10 (G.832)	R/O	0x00
0xn127	RxE3 TTB Register # 11 (G.832)	R/O	0x00
0xn128	RxE3 TTB Register # 12 (G.832)	R/O	0x00
0xn129	RxE3 TTB Register # 13 (G.832)	R/O	0x00
0xn12A	RxE3 TTB Register # 14 (G.832)	R/O	0x00
0xn12B	RxE3 TTB Register # 15 (G.832)	R/O	0x00
0xn12C	RxE3 SSM Register (G.832)	R/O	0x00
0xn12D - 0xn12F	Reserved		

CLEAR-CHANNEL FRAMER BLOCK REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
CLEAR-CHANNEL FRAMER BLOCK REGISTERS			
0xn130	Transmit DS3 Configuration RegisterTransmit E3 Configuration Register	R/W	0x07
0xn131	TxDS3 FEAC Configuration and Status Register	RUR & R/W	0x00
0xn132	TxDS3 FEAC Register	R/W	0x7E
0xn133	TxLAPD Configuration Register	R/O & R/W	0x08
0xn134	TxLAPD Status and Interrupt Register	RUR & R/W	0x00
0xn135	TxDS3 M-Bit Mask RegisterTxE3 GC Byte Register (G.832)TxE3 Service Bits Register (G.751)	R/W	0x00
0xn136	TxDS3 F-Bit Mask Register # 1TxE3 MA Byte Register (G.832)	R/W	0x00
0xn137	TxDS3 F-Bit Mask Register # 2TxE3 NR Byte Register (G.832)	R/W	0x00
0xn138	TxDS3 F-Bit Mask Register # 3TxTTB Register # 0 (G.832)	R/W	0x00
0xn139	TxTTB Register # 1 (G.832)	R/W	0x00
0xn13A	TxTTB Register # 2 (G.832)	R/W	0x00
0xn13B	TxTTB Register # 3 (G.832)	R/W	0x00
0xn13C	TxTTB Register # 4 (G.832)	R/W	0x00
0xn13D	TxTTB Register # 5 (G.832)	R/W	0x00
0xn13E	TxTTB Register # 6 (G.832)	R/W	0x00
0xn13F	TxTTB Register # 7 (G.832)	R/W	0x00
0xn140	TxTTB Register # 8 (G.832)	R/W	0x00
0xn141	TxTTB Register # 9 (G.832)	R/W	0x00
0xn142	TxTTB Register # 10 (G.832)	R/W	0x00
0xn143	TxTTB Register # 11 (G.832)	R/W	0x00
0xn144	TxTTB Register # 12 (G.832)	R/W	0x00
0xn145	TxTTB Register # 13 (G.832)	R/W	0x00
0xn146	TxTTB Register # 14 (G.832)	R/W	0x00
0xn147	TxTTB Register # 15 (G.832)	R/W	0x00
0xn148	TxE3 FA1 Error Mask Register (G.832)TxE3 FAS Error Mask Register # 1 (G.751)	R/W	0x00
0xn149	TxE3 FA2 Error Mask Register (G.832)TxE3 FAS Error Mask Register # 2 (G.751)	R/W	0x00

CLEAR-CHANNEL FRAMER BLOCK REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
CLEAR-CHANNEL FRAMER BLOCK REGISTERS			
0xn14A	TxE3 BIP-8 Error Mask Register (G.832)TxE3 BIP-4 Error Mask Register (G.751)	R/W	0x00
0xn14B	TxE3 SSM Register	R/W	0x00
0xn14C - 0xn14F	Reserved	R/O	0x00
0xn150	PMON Line Code Violation Count Register - MSB	RUR	0x00
0xn151	PMON Line Code Violation Count Register - LSB	RUR	0x00
0xn152	PMON Framing Bit/Byte Error Count Register - MSB	RUR	0x00
0xn153	PMON Framing Bit/Byte Error Count Register - LSB	RUR	0x00
0xn154	PMON P-Bit/BIP-8/BIP-4 Error Count Register - MSB	RUR	0x00
0xn155	PMON P-Bit/BIP-8/BIP-4 Error Count Register - LSB	RUR	0x00
0xn156	PMON FEBE Event Count Register - MSB	RUR	0x00
0xn157	PMON FEBE Event Count Register - LSB	RUR	0x00
0xn158	PMON CP-Bit Error Count Register - MSB	RUR	0x00
0xn159	PMON CP-Bit Error Count Register - LSB	RUR	0x00
0xn15A	PMON PLCP BIP-8 Error Count Register - MSB	RUR	0x00
0xn15B	PMON PLCP BIP-8 Error Count Register - LSB	RUR	0x00
0xn15C	PMON PLCP Framing Byte Error Count Register - MSB	RUR	0x00
0xn15D	PMON PLCP Framing Byte Error Count Register - LSB	RUR	0x00
0xn15E	PMON PLCP FEBE Event Count Register - MSB	RUR	0x00
0xn15F	PMON PLCP FEBE Event Count Register - LSB	RUR	0x00
0xn160 - 0xn167	Reserved		
0xn168	PRBS Error Count Register - MSB	RUR	0x00
0xn169	PRBS Error Count Register - LSB	RUR	0x00
0xn16A - 0xn16C	Reserved		
0xn16D	One Second Error Status Register	R/O	0x00
0xn16E	One Second Accumulator - LCV Count Register - MSB	R/O	0x00
0xn16F	One Second Accumulator - LCV Count Register - LSB	R/O	0x00
0xn170	One Second Accumulator - P-Bit/BIP-8/BIP-4 Error Count Register - MSB	R/O	0x00
0xn171	One Second Accumulator - P-Bit/BIP-8/BIP-4 Error Count Register - LSB	R/O	0x00
0xn172	One Second Accumulator - CP Bit Error Count Register - MSB	R/O	0x00

CLEAR-CHANNEL FRAMER BLOCK REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
CLEAR-CHANNEL FRAMER BLOCK REGISTERS			
0xn173	One Second Accumulator - CP Bit Error Count Register - LSB	R/O	0x00
0xn174 - 0xn17F	Reserved		
0xn180	Line Interface Drive Register	R/W	0x08
0xn181	Line Interface Scan Register	R/O	0x00
0xn182 - 0xn18F	Reserved		
0xn190	RxPLCP Configuration & Status Register	R/O & R/W	0x06
0xn191	RxPLCP Interrupt Enable Register	R/W	0x00
0xn192	RxPLCP Interrupt Status Register	RUR	0x00
0xn193 - 0xn197	Reserved		
0xn198	TxPLCP A1 Byte Error Mask Register	R/W	0x00
0xn199	TxPLCP A2 Byte Error Mask Register	R/W	0x00
0xn19A	TxPLCP BIP-8 Byte Error Mask Register	R/W	0x00
0xn19B	TxPLCP G1 Byte Register	R/W	0x00
0xn19C - 0xn2FF	Reserved		

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xn700	Receive ATM Control - Byte 3	R/W	0x00
0xn701	Receive ATM Control - Byte 2	R/W	0x00
0xn702	Receive ATM Control - Byte 1	R/W	0x00
0xn703	Receive ATM Control - Byte 0Receive PPP Control Register	R/W	0x00
0xn704 - 0xn706	Reserved		
0xn707	Receive ATM Status Register	R/O	0x00
0xn708 - 0xn709	Reserved		
0xn70A	Receive ATM Interrupt Status Register -Byte 1	RUR	0x00
0xn70B	Receive ATM Interrupt Status Register - Byte 0Receive PPP Interrupt Status Register	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xn70C - 0xn70D	Reserved		
0xn70E	Receive ATM Interrupt Enable Register - Byte 1	R/W	0x00
0xn70F	Receive ATM Interrupt Enable Register - Byte 0Receive PPP Interrupt Enable Register	R/W	0x00
0xn710	Receive PPP Good Packet Count Register - Byte 3	RUR	0x00
0xn711	Receive PPP Good Packet Count Register - Byte 2	RUR	0x00
0xn712	Receive PPP Good Packet Count Register - Byte 1	RUR	0x00
0xn713	Receive ATM Cell Insertion/Extraction Memory Control RegisterReceive PPP Good Packet Count Register - Byte 0	R/O & R/W	0x00
0xn714	Receive ATM Cell Insertion/Extraction Memory Data Register - Byte 3Receive PPP FCS Error Count Register - Byte 3	R/O & R/W	0x00
0xn715	Receive ATM Cell Insertion/Extraction Memory Data Register - Byte 2Receive PPP FCS Error Count Register - Byte 2	R/O & R/W	0x00
0xn716	Receive ATM Cell Insertion/Extraction Memory Data Register - Byte 1Receive PPP FCS Error Count Register - Byte 1	R/O & R/W	0x00
0xn717	Receive ATM Cell Insertion/Extraction Memory Data Register - Byte 0Receive PPP FCS Error Count Register - Byte 0	R/O & R/W	0x00
0xn718	Receive ATM Cell UDF Data Register - Byte 3Receive PPP Abort Count Register - Byte 3	R/W & RUR	0x00
0xn719	Receive ATM Cell UDF Data Register - Byte 2Receive PPP Abort Count Register - Byte 2	R/W & RUR	0x00
0xn71A	Receive ATM Cell UDF Data Register - Byte 1Receive PPP Abort Count Register - Byte 1	R/W & RUR	0x00
0xn71B	Receive ATM Cell UDF Data Register - Byte 0Receive PPP Abort Count Register - Byte 0	R/W & RUR	0x00
0xn71C	Receive PPP Runt Frame Count Register - Byte 3	RUR	0x00
0xn71D	Receive PPP Runt Frame Count Register - Byte 2	RUR	0x00
0xn71E	Receive PPP Runt Frame Count Register - Byte 1	RUR	0x00
0xn71F	Receive PPP Runt Frame Count Register - Byte 0	RUR	0x00
0xn720	Receive ATM - Test Cell Header Byte Register - Byte 0	R/W	0x00
0xn721	Receive ATM - Test Cell Header Byte Register - Byte 1	R/W	0x00
0xn722	Receive ATM - Test Cell Header Byte Register - Byte 2	R/W	0x00
0xn723	Receive ATM - Test Cell Header Byte Register - Byte 3	R/W	0x00
0xn724	Receive ATM - Test Cell Error Count Register - Byte 3	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xn725	Receive ATM - Test Cell Error Count Register - Byte 2	RUR	0x00
0xn726	Receive ATM - Test Cell Error Count Register - Byte 1	RUR	0x00
0xn727	Receive ATM - Test Cell Error Count Register - Byte 0	RUR	0x00
0xn728	Receive ATM Cell Count Register - Byte 3	RUR	0x00
0xn729	Receive ATM Cell Count Register - Byte 2	RUR	0x00
0xn72A	Receive ATM Cell Count Register - Byte 1	RUR	0x00
0xn72B	Receive ATM Cell Count Register - Byte 0	RUR	0x00
0xn72C	Receive ATM Cell - Discard Cell Count Register - Byte 3	RUR	0x00
0xn72D	Receive ATM Cell - Discard Cell Count Register - Byte 2	RUR	0x00
0xn72E	Receive ATM Cell - Discard Cell Count Register - Byte 1	RUR	0x00
0xn72F	Receive ATM Cell - Discard Cell Count Register - Byte 0	RUR	0x00
0xn730	Receive ATM Correctable HEC Byte Error Count Register - Byte 3	RUR	0x00
0xn731	Receive ATM Correctable HEC Byte Error Count Register - Byte 2	RUR	0x00
0xn732	Receive ATM Correctable HEC Byte Error Count Register - Byte 1	RUR	0x00
0xn733	Receive ATM Correctable HEC Byte Error Count Register - Byte 0	RUR	0x00
0xn734	Receive ATM Uncorrectable HEC Byte Error Count Register - Byte 3	RUR	0x00
0xn735	Receive ATM Uncorrectable HEC Byte Error Count Register - Byte 2	RUR	0x00
0xn736	Receive ATM Uncorrectable HEC Byte Error Count Register - Byte 1	RUR	0x00
0xn737	Receive ATM Uncorrectable HEC Byte Error Count Register - Byte 0	RUR	0x00
0xn738 - 0xn742	Reserved		
0xn743	Receive ATM - User Cell Filter # 0 - Filter Control Register	R/W	0x00
0xn744	Receive ATM - User Cell Filter # 0 - Header Byte # 1 Pattern Register	R/W	0x00
0xn745	Receive ATM - User Cell Filter # 0 - Header Byte # 2 Pattern Register	R/W	0x00
0xn746	Receive ATM - User Cell Filter # 0 - Header Byte # 3 Pattern Register	R/W	0x00
0xn747	Receive ATM - User Cell Filter # 0 - Header Byte # 4 Pattern Register	R/W	0x00
0xn748	Receive ATM - User Cell Filter # 0 - Header Byte # 1 Check Register	R/W	0x00
0xn749	Receive ATM - User Cell Filter # 0 - Header Byte # 2 Check Register	R/W	0x00
0xn74A	Receive ATM - User Cell Filter # 0 - Header Byte # 3 Check Register	R/W	0x00
0xn74B	Receive ATM - User Cell Filter # 0 - Header Byte # 4 Check Register	R/W	0x00
0xn74C	Receive ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 3	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xn74D	Receive ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xn74E	Receive ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xn74F	Receive ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xn750 - 0xn752	Reserved		
0xn753	Receive ATM - User Cell Filter # 1 - Filter Control Register	R/W	0x00
0xn754	Receive ATM - User Cell Filter # 1 - Header Byte # 1 Pattern Register	R/W	0x00
0xn755	Receive ATM - User Cell Filter # 1 - Header Byte # 2 Pattern Register	R/W	0x00
0xn756	Receive ATM - User Cell Filter # 1 - Header Byte # 3 Pattern Register	R/W	0x00
0xn757	Receive ATM - User Cell Filter # 1 - Header Byte # 4 Pattern Register	R/W	0x00
0xn758	Receive ATM - User Cell Filter # 1 - Header Byte # 1 Check Register	R/W	0x00
0xn759	Receive ATM - User Cell Filter # 1 - Header Byte # 2 Check Register	R/W	0x00
0xn75A	Receive ATM - User Cell Filter # 1 - Header Byte # 3 Check Register	R/W	0x00
0xn75B	Receive ATM - User Cell Filter # 1 - Header Byte # 4 Check Register	R/W	0x00
0xn75C	Receive ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xn75D	Receive ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xn75E	Receive ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xn75F	Receive ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xn760 - 0xn762	Reserved		
0xn763	Receive ATM - User Cell Filter # 2 - Filter Control Register	R/W	0x00
0xn764	Receive ATM - User Cell Filter # 2 - Header Byte # 1 Pattern Register	R/W	0x00
0xn765	Receive ATM - User Cell Filter # 2 - Header Byte # 2 Pattern Register	R/W	0x00
0xn766	Receive ATM - User Cell Filter # 2 - Header Byte # 3 Pattern Register	R/W	0x00
0xn767	Receive ATM - User Cell Filter # 2 - Header Byte # 4 Pattern Register	R/W	0x00
0xn768	Receive ATM - User Cell Filter # 2 - Header Byte # 1 Check Register	R/W	0x00
0xn769	Receive ATM - User Cell Filter # 2 - Header Byte # 2 Check Register	R/W	0x00
0xn76A	Receive ATM - User Cell Filter # 2 - Header Byte # 3 Check Register	R/W	0x00
0xn76B	Receive ATM - User Cell Filter # 2 - Header Byte # 4 Check Register	R/W	0x00
0xn76C	Receive ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xn76D	Receive ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xn76E	Receive ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 1	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xn76F	Receive ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xn770 - 0xn772	Reserved		
0xn773	Receive ATM - User Cell Filter # 3 - Filter Control Register	R/W	0x00
0xn774	Receive ATM - User Cell Filter # 3 - Header Byte # 1 Pattern Register	R/W	0x00
0xn775	Receive ATM - User Cell Filter # 3 - Header Byte # 2 Pattern Register	R/W	0x00
0xn776	Receive ATM - User Cell Filter # 3 - Header Byte # 3 Pattern Register	R/W	0x00
0xn777	Receive ATM - User Cell Filter # 3 - Header Byte # 4 Pattern Register	R/W	0x00
0xn778	Receive ATM - User Cell Filter # 3 - Header Byte # 1 Check Register	R/W	0x00
0xn779	Receive ATM - User Cell Filter # 3 - Header Byte # 2 Check Register	R/W	0x00
0xn77A	Receive ATM - User Cell Filter # 3 - Header Byte # 3 Check Register	R/W	0x00
0xn77B	Receive ATM - User Cell Filter # 3 - Header Byte # 4 Check Register	R/W	0x00
0xn77C	Receive ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xn77D	Receive ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xn77E	Receive ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xn77F	Receive ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xn780 - 0xnEFF	Reserved		
0xnF00	Transmit ATM Control Register - Byte 3	R/W	0x00
0xnF01	Transmit ATM Control Register - Byte 2	R/W	0x00
0xnF02	Transmit ATM Control Register - Byte 1	R/W	0x00
0xnF03	Transmit ATM Control Register - Byte 0Transmit PPP Control Register - Byte 2	R/W	0x00
0xnF04	Transmit ATM Status Register - Byte 3	R/O	0x00
0xnF05	Transmit ATM Status Register - Byte 2	R/O	0x00
0xnF06	Transmit ATM Status Register - Byte 1	R/O	0x00
0xnF07	Transmit ATM Status Register - Byte 0	R/O	0x00
0xnF08 - 0xnF0A	Reserved		
0xnF0B	Transmit ATM Cell Processor Interrupt Status RegisterTransmit PPP Interrupt Status Register	RUR	0x00
0xnF0C - 0xnF0E	Reserved		
0xnF0F	Transmit ATM Cell Processor Interrupt Enable Register Transmit PPP Interrupt Enable Register	R/W	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xnF10 - 0xnF12	Reserved		
0xnF13	Transmit ATM Cell Insertion/Extraction Memory Control Register	R/O & R/W	0x00
0xnF14	Transmit ATM Cell Insertion/Extraction Data Register - Byte 3	R/O & R/W	0x00
0xnF15	Transmit ATM Cell Insertion/Extraction Data Register - Byte 2	R/O & R/W	0x00
0xnF16	Transmit ATM Cell Insertion/Extraction Data Register - Byte 1	R/O & R/W	0x00
0xnF17	Transmit ATM Cell Insertion/Extraction Data Register - Byte 0	R/O & R/W	0x00
0xnF18	Transmit ATM - Idle Cell Header Byte # 1 Register	R/W	0x00
0xnF19	Transmit ATM - Idle Cell Header Byte # 2 Register	R/W	0x00
0xnF1A	Transmit ATM - Idle Cell Header Byte # 3 Register	R/W	0x00
0xnF1B	Transmit ATM - Idle Cell Header Byte # 4 Register	R/W	0x00
0xnF1C - 0xnF1E	Reserved		
0xnF1F	Transmit ATM - Idle Cell Payload Byte Register	R/W	0x00
0xnF20	Transmit ATM - Test Cell Header Byte # 1 Register	R/W	0x00
0xnF21	Transmit ATM - Test Cell Header Byte # 2 Register	R/W	0x00
0xnF22	Transmit ATM - Test Cell Header Byte # 3 Register	R/W	0x00
0xnF23	Transmit ATM - Test Cell Header Byte # 4 Register	R/W	0x00
0xnF24 - 0xnF27	Reserved		
0xnF28	Transmit ATM Cell Count Register - Byte 3	RUR	0x00
0xnF29	Transmit ATM Cell Count Register - Byte 2	RUR	0x00
0xnF2A	Transmit ATM Cell Count Register - Byte 1	RUR	0x00
0xnF2B	Transmit ATM Cell Count Register - Byte 0	RUR	0x00
0xnF2C	Transmit ATM - Discarded Cell Count Register - Byte 3	RUR	0x00
0xnF2D	Transmit ATM - Discarded Cell Count Register - Byte 2	RUR	0x00
0xnF2E	Transmit ATM - Discarded Cell Count Register - Byte 1	RUR	0x00
0xnF2F	Transmit ATM - Discarded Cell Count Register - Byte 0	RUR	0x00
0xnF30	Transmit ATM HEC Byte Error Count Register - Byte 3	RUR	0x00
0xnF31	Transmit ATM HEC Byte Error Count Register - Byte 2	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xnF32	Transmit ATM HEC Byte Error Count Register - Byte 1	RUR	0x00
0xnF33	Transmit ATM HEC Byte Error Count Register - Byte 0	RUR	0x00
0xnF34	Transmit ATM Cell Processor - Parity Error Count Register - Byte 3	RUR	0x00
0xnF35	Transmit ATM Cell Processor - Parity Error Count Register - Byte 2	RUR	0x00
0xnF36	Transmit ATM Cell Processor - Parity Error Count Register - Byte 1	RUR	0x00
0xnF37	Transmit ATM Cell Processor - Parity Error Count Register - Byte 0	RUR	0x00
0xnF38 - 0xnF42	Reserved		
0xnF43	Transmit ATM - User Cell Filter # 0 - Filter Control Register	R/W	0x00
0xnF44	Transmit ATM - User Cell Filter # 0 - Header Byte # 1 Pattern Register	R/W	0x00
0xnF45	Transmit ATM - User Cell Filter # 0 - Header Byte # 2 Pattern Register	R/W	0x00
0xnF46	Transmit ATM - User Cell Filter # 0 - Header Byte # 3 Pattern Register	R/W	0x00
0xnF47	Transmit ATM - User Cell Filter # 0 - Header Byte # 4 Pattern Register	R/W	0x00
0xnF48	Transmit ATM - User Cell Filter # 0 - Header Byte # 1 Check Register	R/W	0x00
0xnF49	Transmit ATM - User Cell Filter # 0 - Header Byte # 2 Check Register	R/W	0x00
0xnF4A	Transmit ATM - User Cell Filter # 0 - Header Byte # 3 Check Register	R/W	0x00
0xnF4B	Transmit ATM - User Cell Filter # 0 - Header Byte # 4 Check Register	R/W	0x00
0xnF4C	Transmit ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xnF4D	Transmit ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xnF4E	Transmit ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xnF4F	Transmit ATM - User Cell Filter # 0 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xnF50 - 0xnF52	Reserved		
0xnF53	Transmit ATM - User Cell Filter # 1 - Filter Control Register	R/W	0x00
0xnF54	Transmit ATM - User Cell Filter # 1 - Header Byte # 1 Pattern Register	R/W	0x00
0xnF55	Transmit ATM - User Cell Filter # 1 - Header Byte # 2 Pattern Register	R/W	0x00
0xnF56	Transmit ATM - User Cell Filter # 1 - Header Byte # 3 Pattern Register	R/W	0x00
0xnF57	Transmit ATM - User Cell Filter # 1 - Header Byte # 4 Pattern Register	R/W	0x00
0xnF58	Transmit ATM - User Cell Filter # 1 - Header Byte # 1 Check Register	R/W	0x00
0xnF59	Transmit ATM - User Cell Filter # 1 - Header Byte # 2 Check Register	R/W	0x00
0xnF5A	Transmit ATM - User Cell Filter # 1 - Header Byte # 3 Check Register	R/W	0x00
0xnF5B	Transmit ATM - User Cell Filter # 1 - Header Byte # 4 Check Register	R/W	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xnF5C	Transmit ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xnF5D	Transmit ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xnF5E	Transmit ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xnF5F	Transmit ATM - User Cell Filter # 1 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xnF60 - 0xnF62	Reserved		
0xnF63	Transmit ATM - User Cell Filter # 2 - Filter Control Register	R/W	0x00
0xnF64	Transmit ATM - User Cell Filter # 2 - Header Byte # 1 Pattern Register	R/W	0x00
0xnF65	Transmit ATM - User Cell Filter # 2 - Header Byte # 2 Pattern Register	R/W	0x00
0xnF66	Transmit ATM - User Cell Filter # 2 - Header Byte # 3 Pattern Register	R/W	0x00
0xnF67	Transmit ATM - User Cell Filter # 2 - Header Byte # 4 Pattern Register	R/W	0x00
0xnF68	Transmit ATM - User Cell Filter # 2 - Header Byte # 1 Check Register	R/W	0x00
0xnF69	Transmit ATM - User Cell Filter # 2 - Header Byte # 2 Check Register	R/W	0x00
0xnF6A	Transmit ATM - User Cell Filter # 2 - Header Byte # 3 Check Register	R/W	0x00
0xnF6B	Transmit ATM - User Cell Filter # 2 - Header Byte # 4 Check Register	R/W	0x00
0xnF6C	Transmit ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xnF6D	Transmit ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 2	RUR	0x00
0xnF6E	Transmit ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xnF6F	Transmit ATM - User Cell Filter # 2 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xnF70 - 0xnF72	Reserved		
0xnF73	Transmit ATM - User Cell Filter # 3 - Filter Control Register	R/W	0x00
0xnF74	Transmit ATM - User Cell Filter # 3 - Header Byte # 1 Pattern Register	R/W	0x00
0xnF75	Transmit ATM - User Cell Filter # 3 - Header Byte # 2 Pattern Register	R/W	0x00
0xnF76	Transmit ATM - User Cell Filter # 3 - Header Byte # 3 Pattern Register	R/W	0x00
0xnF77	Transmit ATM - User Cell Filter # 3 - Header Byte # 4 Pattern Register	R/W	0x00
0xnF78	Transmit ATM - User Cell Filter # 3 - Header Byte # 1 Check Register	R/W	0x00
0xnF79	Transmit ATM - User Cell Filter # 3 - Header Byte # 2 Check Register	R/W	0x00
0xnF7A	Transmit ATM - User Cell Filter # 3 - Header Byte # 3 Check Register	R/W	0x00
0xnF7B	Transmit ATM - User Cell Filter # 3 - Header Byte # 4 Check Register	R/W	0x00
0xnF7C	Transmit ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 3	RUR	0x00
0xnF7D	Transmit ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 2	RUR	0x00

RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
CHANNEL N (N=1, 2, 3) CONTROL REGISTERS			
RECEIVE ATM CELL PROCESSOR/PPP PROCESSOR BLOCK CONTROL REGISTERS			
0xnF7E	Transmit ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 1	RUR	0x00
0xnF7F	Transmit ATM - User Cell Filter # 3 - Filtered Cell Count Register - Byte 0	RUR	0x00
0xnF80 - 0xnFFF	Reserved		

OPERATION BLOCK INTERRUPT REGISTER BIT FORMATS

OPERATION CONTROL REGISTER - BYTE 3 (ADDRESS = 0X0100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Configuration Control
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 6	Unused	R/O	
0	Configuration Control	R/W	**Configuration Control:** This READ/WRITE bit-field permits the user to configure the XRT74L74 device to support any of the following configurations. • ATM/PPP • Clear Channel/HDLC The following table presents the relationship between the value written into these register bits and the corresponding Mode of operation.

Configuration Control	Mode
0	ATM/PPP
1	Clear Channel/HDLC

OPERATION CONTROL REGISTER - BYTE 2 (ADDRESS = 0X0101)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused					Interrupt WC/INT*	Enable Interrupt Auto-Clear	Interrupt Enable
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 3	Unused	R/O	Please set to "0" for normal operation.

BIT NUMBER	NAME	TYPE	DESCRIPTION
2	Interrupt Write to Clear/ RUR	R/W	**Interrupt - Write to Clear/RUR Select:** This READ/WRITE bit-field permits the user to configure all of the "Source-Level" Interrupt Status bits (within the XRT74L74 device) to either be "Write to Clear" (WTC) or "Reset-upon-Read" (RUR) bits. 0 - Configures all "Source-Level" Interrupt Status register bits to function as "Reset-upon-Read" (RUR). 1 - Configures all "Source-Level" Interrupt Status register bits to function as "Write-to-Clear" (WTC).
1	Enable Interrupt Clear	R/W	**Enable Auto-Clear of Interrupts Select:** This READ/WRITE bit-field permits the user to configure the XRT74L74 device to automatically disable all interrupts that are activated. 0 - Configures the chip to NOT automatically disable any Interrupts following their activation. 1 - Configures the chip to automatically disable all Interrupts following their activation.
0	Interrupt Enable	R/W	**Interrupt Enable:** This READ/WRITE bit-field permits the user to configure the XRT74L74 device to generate interrupt requests to the Microprocessor. 0 - Configures the chip to NOT generate interrupt to the Microprocessor. All interrupts are disabled and the Microprocessor must poll the register bits. 1 - Configures the chip to generate interrupts the Microprocessor.

OPERATION CONTROL - LOOP-BACK CONTROL REGISTER (ADDRESS = 0X0102)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Loop-back Control [3:0]			
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 4	Unused	R/O	

BIT NUMBER	NAME	TYPE	DESCRIPTION
3 - 0	Loop-back Control [3:0]	R/W	**Loop-back Mode Select:** These READ/WRITE bit-fields permit the user to configure the XRT74L74 to operate in any of the following loop-back modes. • Local Medium Loop-back • Remote Host Loop-back The following table presents the contents of these bit-fields and the corresponding Loop-back Modes.

Loop-back Control [3:0]	Resulting Loop-back Mode
0000 - 0011	Reserved
0100	Local Medium Loop-back Mode
0101	Remote Host Loop-back Mode
0110 - 1111	Reserved

OPERATION CONTROL REGISTER - BYTE 0 (ADDRESS = 0X0103)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit UTOPIA PLL OFF	Receive UTOPIA PLL OFF	Reserved			PPP/ATM*	Reserved	Software RESET*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7	Transmit UTOPIA PLL OFF	R/W	
6	Receive UTOPIA PLL OFF	R/W	
5 - 3	Unused	R/O	
2	PPP/ATM*	R/W	**PPP/ATM UNI Mode Select:** This READ-WRITE bit-field permits the user to configure the XRT74L74 device to operate in either the ATM UNI or PPP Mode. If Bit 3 (Dual Bus), within the "Operation Control Register - Byte 3" is set to "0", then this bit-field will then dictate the operating mode of the XRT74L74 device. 0 - Configures the "Dedicated" UTOPIA/POS-PHY bus to operate in the UTOPIA (ATM) Mode. 1 - Configures the "Dedicated" UTOPIA/POS-PHY Bus to operate in the POS-PHY Mode. ***NOTE:*** *This bit-field is ignored if Bit 3 (Dual-Bus) within the "Operation Control Register - Byte 3" is set to "1".*
1	Reserved	R/O	

BIT NUMBER	NAME	TYPE	DESCRIPTION
0	Software RESET	R/W	**Software RESET:** This READ-WRITE bit-field permits the user to reset the XRT74L74 device. 0 - Configure the XRT74L74 device into RESET mode. 1 - Normal operation.

DEVICE ID REGISTER (ADDRESS = 0X0104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DEVICE_ID_VALUE [7:0]							
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	1	1	1	1	0	1	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 0	Device ID Value	R/O	**Device ID Value:** This READ-ONLY bit-field is set to the value "0x7A" and permits the user's software code to uniquely identify this device as the XRT74L74 device.

REVISION ID REGISTER (ADDRESS = 0X0105)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Revision Number Value							
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 0	Revision Number Value	R/O	**Revision Number Value:** This READ-ONLY bit-field is set to the value that corresponds to its revision number. Revision A silicon will be set to the value "0x01". This register permits the user's software code to uniquely identify the revision number of the XRT74L74 device.

OPERATION INTERRUPT STATUS REGISTER - BYTE 1 (ADDRESS = 0X0112)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				DS3/E3 LIU/JA Block Interrupt Status	DS3/E3 Framer Block Interrupt Status	Unused	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 4	Unused	R/O	
3	DS3/E3 LIU/JA Block Interrupt Status	R/O	**DS3/E3 LIU/JA Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "DS3/E3 LIU/JA Block" interrupt is awaiting service. 0 - No "DS3/E3 LIU/JA" block interrupt is awaiting service. 1 - At least one "DS3/E3 LIU/JA" block interrupt is awaiting service.
2	DS3/E3 Framer Block Interrupt Status	R/O	**DS3/E3 Framer Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "DS3/E3 Framer Block" interrupt is awaiting service. 0 - No "DS3/E3 Framer" block interrupt is awaiting service. 1 - At least one "DS3/E3 Framer" block interrupt is awaiting service.
1 - 0	Unused	R/O	

OPERATION INTERRUPT STATUS REGISTER - BYTE 0 (ADDRESS = 0X0113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive UTOPIA/ POS-PHY Interface Block Interrupt Status	Unused		Receive ATM Cell/PPP Processor Block Interrupt Status	Transmit UTOPIA/ POS-PHY Interface Block Interrupt Status	Unused		Transmit ATM Cell/PPP Processor Block Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7	Receive UTOPIA POS-PHY Interface Block Interrupt Status	R/O	**Receive UTOPIA/POS-PHY Interface Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "Receive UTOPIA/POS-PHY Interface" block interrupt is awaiting service. 0 - No "Receive UTOPIA/POS-PHY Interface" block interrupt is awaiting service. 1 - At least one "Receive UTOPIA/POS-PHY Interface" block interrupt is awaiting service.
6 -5	Unused	R/O	
4	Receive ATM Cell/PPP Processor Block Interrupt Status	R/O	**Receive ATM Cell/PPP Processor Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "Receive ATM Cell/PPP Processor Block" Interrupt is awaiting service. 0 - No "Receive ATM Cell/PPP Processor block" interrupt is awaiting service. 1 - At least one "Receive ATM Cell/PPP Processor" block interrupt is awaiting service.

BIT NUMBER	NAME	TYPE	DESCRIPTION
3	Transmit UTOPIA POS-PHY Interface Block Interrupt Status		**Transmit UTOPIA/POS-PHY Interface Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "Transmit UTOPIA/POS-PHY Interface" block interrupt is awaiting service. 0 - No "Transmit UTOPIA/POS-PHY Interface" block interrupt is awaiting service. 1 - At least one "Transmit UTOPIA/POS-PHY Interface" block interrupt is awaiting service.
2 - 1	Unused	R/O	
0	Transmit ATM Cell/PPP Processor Block Interrupt Status	R/O	**Receive ATM Cell/PPP Processor Block Interrupt Status:** This READ-ONLY bit-field indicates whether or not a "Receive ATM Cell/PPP Processor Block" Interrupt is awaiting service. 0 - No "Receive ATM Cell/PPP Processor block" interrupt is awaiting service. 1 - At least one "Receive ATM Cell/PPP Processor" block interrupt is awaiting service.

OPERATION INTERRUPT ENABLE REGISTER - BYTE 1 (ADDRESS = 0X0116)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				DS3/E3 LIU/JA Block Interrupt Enable	DS3/E3 Framer Block Interrupt Enable	Unused	
R/O	R/O	R/O	R/O	R/W	R/W	R/O	R/O
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 4	Unused		
3	DS3/E3 LIU/JA Block Interrupt Enable	R/W	**DS3/E3 LIU/JA Block Interrupt Enable:** This READ/WRITE bit permit the user to either enable or disable the DS3/E3 LIU/JA Block for interrupt generation. If the user writes a "0" to this register bit and disables the "DS3/E3 LIU/JA Block" (for interrupt generation), then all "DS3/E3 LIU/JA Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "DS3/E3 LIU/JA Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "DS3/E3 LIU/JA Block" interrupts within the device. 1 - Enables the "DS3/E3 LIU/JA Block" at the "Block-Level".

BIT NUMBER	NAME	TYPE	DESCRIPTION
2	DS3/E3 Framer Block Interrupt Enable	R/W	**DS3/E3 Framer Block Interrupt Enable:** This READ/WRITE bit permits the user to either enable or disable the DS3/E3 Framer Block for interrupt generation. If the user writes a "0" to this register bit and disables the "DS3/E3 Framer Block" (for interrupt generation), then all "DS3/E3 Framer Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "DS3/E3 Framer Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "DS3/E3 Framer Block" interrupts within the device. 1 - Enables the "DS3/E3 Framer Block" at the "Block-Level".
1 - 0	Unused		

OPERATION INTERRUPT ENABLE REGISTER - BYTE 0 (ADDRESS = 0X0117)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive UTOPIA/ POS-PHY Interface Block Interrupt Enable	Unused		Receive ATM Cell/PPP Processor Block Interrupt Enable	Transmit UTOPIA/ POS-PHY Interface Block Interrupt Enable	Unused		Transmit ATM Cell/PPP Processor Block Interrupt Enable
R/W	R/O	R/O	R/W	R/W	R/O	R/O	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7	Receive UTOPIA/POS-PHY Interface Block Interrupt Enable	R/W	**Receive UTOPIA/POS-PHY Interface Block Interrupt Enable:** This READ/WRITE bit permit the user to either enable or disable the Receive UTOPIA/POS-PHY Interface Block for interrupt generation. If the user writes a "0" to this register bit and disables the "Receive UTOPIA/POS-PHY Interface Block" (for interrupt generation), then all "Receive UTOPIA/POS-PHY Interface Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "Receive UTOPIA/POS-PHY Interface Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "Receive UTOPIA/POS-PHY Interface Block" interrupts within the device. 1 - Enables the "Receive UTOPIA/POS-PHY Interface Block" at the "Block-Level".
6 - 5	Unused	R/O	

BIT NUMBER	NAME	TYPE	DESCRIPTION
4	Receive ATM Cell/PPP Processor Block Interrupt Enable	R/W	**Receive ATM Cell/PPP Processor Block Interrupt Enable:** This READ/WRITE bit permit the user to either enable or disable the Receive ATM Cell/PPP Processor Block for interrupt generation. If the user writes a "0" to this register bit and disables the "Receive ATM Cell/PPP Processor Block" (for interrupt generation), then all "Receive ATM Cell/PPP Processor Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "Receive ATM Cell/PPP Processor Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "Receive ATM Cell/PPP Processor Block" interrupts within the device. 1 - Enables the "Receive ATM Cell/PPP Processor Block" at the "Block-Level".
3	Transmit UTOPIA/POS-PHY Interface Block Interrupt Enable	R/W	**Transmit UTOPIA/POS-PHY Interface Block Interrupt Enable:** This READ/WRITE bit permit the user to either enable or disable the Transmit UTOPIA/POS-PHY Interface Block for interrupt generation. If the user writes a "0" to this register bit and disables the "Transmit UTOPIA/POS-PHY Interface Block" (for interrupt generation), then all "Transmit UTOPIA/POS-PHY Interface Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "Transmit UTOPIA/POS-PHY Interface Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "Transmit UTOPIA/POS-PHY Interface Block" interrupts within the device. 1 - Enables the "Transmit UTOPIA/POS-PHY Interface Block" at the "Block-Level".
2 - 1	Unused	R/O	
0	Transmit ATM Cell/PPP Processor Block Interrupt Enable	R/W	**Transmit ATM Cell/PPP Processor Block Interrupt Enable:** This READ/WRITE bit permit the user to either enable or disable the Transmit ATM Cell/PPP Processor Block for interrupt generation. If the user writes a "0" to this register bit and disables the "Transmit ATM Cell/PPP Processor Block" (for interrupt generation), then all "Transmit ATM Cell/PPP Processor Block" interrupts will be disabled for interrupt generation. If the user writes a "1" to this register bit, he/she will still need to enable the individual "Transmit ATM Cell/PPP Processor Block" interrupt(s) at the "Source Level" in order to enable that particular interrupt. 0 - Disable all "Transmit ATM Cell/PPP Processor Block" interrupts within the device. 1 - Enables the "Transmit ATM Cell/PPP Processor Block" at the "Block-Level".

CHANNEL INTERRUPT INDICATION REGISTERS

CHANNEL INTERRUPT INDICATOR - RECEIVE CELL PROCESSOR/PPP PROCESSOR BLOCK (ADDRESS = 0X0119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Receive Cell Processor Block Interrupt
R/O							R/O
0							0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 1	Unused	R/O	
0	Receive Cell Processor Block Interrupt - XRT74L74	R/O	**Receive Cell Processor Block Interrupt - XRT74L74:** This READ/ONLY bit-field indicates whether or not the "Receive Cell Processor" block, associated with XRT74L74 is declaring an Interrupt, as described below. 0 - The Receive Cell Processor block, associated with XRT74L74 is NOT declaring an Interrupt. 1 - The Receive Cell Processor block, associated with XRT74L74 is currently declaring an interrupt.

CHANNEL INTERRUPT INDICATOR - LIU/JITTER ATTENUATOR BLOCK (ADDRESS = 0X011D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							LIU/JA Block Interrupt
R/O							R/O
0							0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 1	Unused	R/O	
0	LIU/JA Block Interrupt - XRT74L74	R/O	**LIU/JA Block Interrupt - XRT74L74:** This READ/ONLY bit-field indicates whether or not the "LIU/JA" block, associated with XRT74L74 is declaring an Interrupt, as described below. 0 - The LIU/JA block, associated with XRT74L74 is NOT declaring an Interrupt. 1 - The LIU/JA block, associated with XRT74L74 is currently declaring an interrupt.

CHANNEL INTERRUPT INDICATOR - TRANSMIT CELL PROCESSOR/PPP PROCESSOR BLOCK (ADDRESS = 0X0121)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Transmit Cell Processor Block Interrupt
R/O							R/O
0							

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 1	Unused	R/O	
0	Transmit Cell Processor Block Interrupt - XRT74L74	R/O	**Transmit Cell Processor Block Interrupt - XRT74L74:** This READ/ONLY bit-field indicates whether or not the "Transmit Cell Processor" block, associated with XRT74L74 is declaring an Interrupt, as described below. 0 - The Transmit Cell Processor block, associated with XRT74L74 is NOT declaring an Interrupt. 1 - The Transmit Cell Processor block, associated with XRT74L74 is currently declaring an interrupt.

CHANNEL INTERRUPT INDICATOR - DS3/E3 FRAMER BLOCK (ADDRESS = 0X0127)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							DS3/E3 Framer Block Interrupt
R/O							R/O
0							0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 1	Unused	R/O	
0	DS3/E3 Framer Block Interrupt - XRT74L74	R/O	**DS3/E3 Framer Block Interrupt - XRT74L74:** This READ/ONLY bit-field indicates whether or not the "DS3/E3 Framer" block, associated with XRT74L74 is declaring an Interrupt, as described below. 0 - The DS3/E3 Framer block, associated with XRT74L74 is NOT declaring an Interrupt. 1 - The DS3/E3 Framer block, associated with XRT74L74 is currently declaring an interrupt.

OPERATION GENERAL PURPOSE PIN DATA REGISTER (ADDRESS = 0X0147)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				General Purpose Data [3]	General Purpose Data [2]	General Purpose Data [1]	General Purpose Data [0]
R/O				R/W	R/W	R/W	R/W
0				0	0	0	0

OPERATION GENERAL PURPOSE PIN DIRECTION CONTROL REGISTER (ADDRESS = 0X014B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				General Purpose Pin Direction [3]	General Purpose Pin Direction [2]	General Purpose Pin Direction [1]	General Purpose Pin Direction [0]
R/O				R/W	R/W	R/W	R/W
0				0	0	0	0

RECEIVE UTOPIA INTERFACE BLOCK

This section presents the Register Description/Address Map of the control registers associated with the Receive UTOPIA/POS-PHY Interface block.

TABLE 1: RECEIVE UTOPIA/POS-PHY INTERFACE BLOCK - REGISTER/ADDRESS MAP

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
RECEIVE UTOPIA/POS-PHY- CONTROL REGISTERS			
0x0501	Receive UTOPIA/POS-PHY Control Register - Byte 2	R/W	0x00
0x0502	Receive UTOPIA/POS-PHY Control Register - Byte 1	R/W	0x00
0x0503	Receive UTOPIA/POS-PHY Control Register - Byte 0	R/W	0x00
0x0504 - 0x0512	Reserved	R/O	0x00
0x0513	Receive UTOPIA Port Address Register	R/W	0x00
0x0514 - 0x0516	Reserved	R/O	0x00
0x0517	Receive UTOPIA Port Number Register	R/W	0x00
0x0518 - 0x0580	Reserved	R/O	0x00

RECEIVE UTOPIA/POS-PHY CONTROL REGISTER - BYTE 0 (ADDRESS = 0X0503)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	0	0	1	1	1	1

BIT NUMBER	NAME	TYPE	DESCRIPTION
7	UTOPIA Level 3 Disable	R/W	

BIT NUMBER	NAME	TYPE	DESCRIPTION
6	Multi-PHY Polling Enable	R/W	**Multi-PHY Polling Enable:** This READ/WRITE bit-field permits the user to either enable or disable Multi-PHY Polling for the Receive UTOPIA Interface block. If the user implements this feature (and configures the XRT74L74 device to operate in the Multi-PHY Mode) then the RxUClav output pin will be driven (either "high" or "low") based upon the fill-status of the Receive FIFO within the Channel that corresponds to the "Receive UTOPIA Address" that is currently being applied to the "RxUAddr[4:0]" input pins. If the user does not implement this feature (and then configures the XRT74L74 device to operate in the Single-PHY Mode), then the "RxUClav" output pin will unconditionally reflect the "Receive FIFO fill-status" for Channel 0. No attention will be paid to the address values placed upon the "RxUAddr[4:0]" input pins. 0 - Configures the Receive UTOPIA Interface block to operate in the Single-PHY Mode. 1 - Configures the Receive UTOPIA Interface block to operate in the Multi-PHY Mode.
5	Back-to-Back Polling Enable	R/W	**Back-to-Back Polling Enable:** This READ/WRITE bit-field permits the user to configure the Receive UTOPIA Interface block to support "Back-to-Back Polling". Ordinarily, for Multi-PHY polling, the user is required to interleave all UTOPIA Address values (that are to be placed on the "RxUAddr[4:0]" input pins) with the NULL Address (e.g., 0x1F). However, if the user configures the Receive UTOPIA Interface block to operate in the "UTOPIA Level 3" Mode, and if the user also enables "Back-to-Back Polling", then he/she does not need interleave the UTOPIA Addresses with the NULL Address. In this case, the user can simply apply a "back-to-back" stream of "relevant" UTOPIA Addresses to the "RxUAddr[4:0]" input pins, and the XRT74L74 device will respond by driving the RxUClav output pins to the appropriate states (depending upon the Receive FIFO fill-status). 0 - Disables "Back-to-Back" Polling. In this mode, the user must interleave all UTOPIA Addresses (that are to be applied to the "RxUAddr[4:0]" input pins) with the NULL Address. 1 - Enables "Back-to-Back" Polling. In this mode, the user does not need to interleave all UTOPIA Addresses (that are to be applied to the "RxUAddr[4:0]" input pins) with the NULL Address. ***NOTE:*** *In order to configure the Receive UTOPIA Interface block to operate in the "Back-to-Back Polling" Mode, the user must also do the following.* **a.** Configure the Receive UTOPIA Interface to operate in the "UTOPIA Level 3" Mode. This is accomplished by setting Bit 7 (UTOPIA Level 3 Disable) within this Register to "0". **b.** Configure the Receive UTOPIA Interface to support "Multi-PHY" Polling. This is accomplished by setting Bit 6 (Multi-PHY Polling Enable) within this register to "1".
4	Direct Status Indication Enable	R/W	

<table>
<tr><th>BIT NUMBER</th><th>NAME</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>3 - 2</td><td>UTOPIA/POS-PHY Data Bus Width[1:0]</td><td>R/W</td><td>UTOPIA/POS-PHY Data Bus Width[1:0]:
These READ/WRITE bit-fields permit the user to select the width of the Receive UTOPIA and POS-PHY Data Buses. The relationship between the contents of these bit-fields and the corresponding widths of the Receive UTOPIA and POS-PHY Data Bus is tabulated below.
<table><tr><th colspan="2">UTOPIA/POS-PHY Data Bus Width[1:0]</th><th>Corresponding UTOPIA/POS-PHY Data Bus Width</th></tr><tr><td>0</td><td>0</td><td>Not Valid</td></tr><tr><td>0</td><td>1</td><td>8 bits</td></tr><tr><td>1</td><td>0</td><td>16 bits</td></tr><tr><td>1</td><td>1</td><td>Not Valid</td></tr></table></td></tr>
<tr><td>1 - 0</td><td>Cell Size[1:0]</td><td></td><td>Cell Size[1:0]:
These two READ/WRITE bit-fields permit the user to specify the size of the ATM cell that will be handled by the Receive UTOPIA Interface blocks. The relationship between the contents of these bit-fields and the corresponding Cell Sizes are tabulated below.
<table><tr><th colspan="2">Cell Size[1:0]</th><th>Resulting Cell Size (Bytes)</th></tr><tr><td>0</td><td>0</td><td>52 bytes</td></tr><tr><td>0</td><td>1</td><td>53 bytes (Only valid for UTOPIA Level 1, and if the UTOPIA Data Bus Width is set to 8 bits)</td></tr><tr><td>1</td><td>0</td><td>54 bytes (Only valid for UTOPIA Levels 1 and 2)</td></tr><tr><td>1</td><td>1</td><td>56 bytes</td></tr></table>
NOTE: The user must bear in mind the UTOPIA Level and the UTOPIA Data Bus width selected, when selecting the Cell Size.</td></tr>
</table>

RECEIVE UTOPIA PORT ADDRESS REGISTER (ADDRESS = 0X0513)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Receive UTOPIA Port Address[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 5	Unused	R/O	
4 - 0	Receive UTOPIA Port Address[4:0]	R/W	**Receive UTOPIA Port Address[4:0]:** These READ/WRITE register bits, along with the "Receive UTOPIA Port Number[4:0]" bits (within the "Receive UTOPIA Port Number" Register (Address = 0x0517) permit the user to assign a unique Receive UTOPIA address to each of the XRT74L74 device. For UTOPIA Level 2/3 applications, the user can write in any value, ranging from 0x00 through 0x1E into this register. **The Receive UTOPIA Address Assignment Procedure:** In order to assign a UTOPIA Address to a given Channel (or Port) within the XRT74L74 device, the user must do the following. **a.** Write the value corresponding to a given XRT74L74 Channel into the "Receive UTOPIA Port Number" Register (Address = 0x0517). **b.** Write the corresponding UTOPIA Address value into this register. Once this "two-step" procedure has been executed, then the XRT74L74 Channel (as specified during step "a") will be assigned the "Receive UTOPIA Address" value (as specified during step "b").

RECEIVE UTOPIA PORT NUMBER REGISTER (ADDRESS = 0X0517)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Receive UTOPIA Port Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 5	Unused	R/O	

BIT NUMBER	NAME	TYPE	DESCRIPTION
4 - 0	Receive UTOPIA Port Number[4:0]	R/W	**Receive UTOPIA Port Number[4:0]:** These READ/WRITE register bits, along with the "Receive UTOPIA Port Address[4:0]" bits (within the "Receive UTOPIA Port Address" Register (Address = 0x0513) permit the user to assign a unique Receive UTOPIA address to the XRT74L74 device. **The Receive UTOPIA Address Assignment Procedure:** In order to assign a UTOPIA Address to a given Channel (or Port) within the XRT74L74 device, the user must do the following. **a.** Write the value corresponding to a given XRT74L74 Channel into this register. **b.** Write the corresponding UTOPIA Address value into the "Receive UTOPIA Port Address" Register (Address = 0x0513). Once this "two-step" procedure has been executed, then the XRT74L74 Channel (as specified during step "a") will be assigned the "Receive UTOPIA Address" value (as specified during step "b").

TRANSMIT UTOPIA INTERFACE BLOCK

This section presents the Register Description/Address Map of the control registers associated with the Transmit UTOPIA/POS-PHY Interface blocks.

TABLE 2: TRANSMIT UTOPIA INTERFACE BLOCK - REGISTER/ADDRESS MAP

ADDRESS LOCATION	REGISTER NAME	TYPE	DEFAULT VALUE
TRANSMIT UTOPIA/POS-PHY CONTROL REGISTERS			
0x0581	Transmit UTOPIA/POS-PHY Control Register - Byte 2	R/W	0x38
0x0582	Transmit UTOPIA/POS-PHY Control Register - Byte 1	R/W	0x00
0x0583	Transmit UTOPIA/POS-PHY Control Register - Byte 0	R/W	0x00
0x0584 - 0x0592	Reserved	R/O	0x00
0x0593	Transmit UTOPIA Port Address Register	R/W	0x00
0x0594 - 0x0596	Reserved	R/O	0x00
0x0597	Transmit UTOPIA Port Number Register	R/W	0x00
0x0598 - 0x10FF	Reserved	R/O	0x00

TRANSMIT UTOPIA/POS-PHY CONTROL REGISTER - BYTE 0 (ADDRESS = 0X0583)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UTOPIA Level 3 Disable	Multi-PHY Polling Enable	Back to Back Polling Enable	Direct Status Indication Enable	UTOPIA/POS-PHY Data Bus Width		Cell Size[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	0	0	1	1	1	1

BIT NUMBER	NAME	TYPE	DESCRIPTION
7	UTOPIA Level 3 Disable	R/W	

BIT NUMBER	NAME	TYPE	DESCRIPTION
6	Multi-PHY Polling Enable	R/W	**Multi-PHY Polling Enable:** This READ/WRITE bit-field permits the user to either enable or disable Multi-PHY Polling for the Transmit UTOPIA Interface block. If the user implements this feature (and configures the XRT74L74 device to operate in the Multi-PHY Mode) then the TxUClav output pin will be driven (either "high" or "low") based upon the fill-status of the Transmit FIFO within the Channel that corresponds to the "Transmit UTOPIA Address" that is currently being applied to the "TxUAddr[4:0]" input pins. If the user does not implement this feature (and then configures the XRT74L74 device to operate in the Single-PHY Mode), then the "TxUClav" output pin will unconditionally reflect the "Transmit FIFO fill-status" for Channel 0. No attention will be paid to the address values placed upon the "TxUAddr[4:0]" input pins. 0 - Configures the Transmit UTOPIA Interface block to operate in the Single-PHY Mode. 1 - Configures the Transmit UTOPIA Interface block to operate in the Multi-PHY Mode.
5	Back-to-Back Polling Enable	R/W	**Back-to-Back Polling Enable:** This READ/WRITE bit-field permits the user to configure the Transmit UTOPIA Interface block to support "Back-to-Back Polling". Ordinarily, for Multi-PHY polling, the user is required to interleave all UTOPIA Address values (that are to be placed on the "TxUAddr[4:0]" input pins) with the NULL Address (e.g., 0x1F). However, if the user configures the Transmit UTOPIA Interface block to operate in the "UTOPIA Level 3" Mode, and if the user also enables "Back-to-Back Polling", then he/she does not need interleave the UTOPIA Addresses with the NULL Address. In this case, the user can simply apply a "back-to-back" stream of "relevant" UTOPIA Addresses to the "TxUAddr[4:0]" input pins, and the XRT74L74 device will respond by driving the TxUClav output pins to the appropriate states (depending upon the Transmit FIFO fill-status). 0 - Disables "Back-to-Back" Polling. In this mode, the user must interleave all UTOPIA Addresses (that are to be applied to the "TxUAddr[4:0]" input pins) with the NULL Address. 1 - Enables "Back-to-Back" Polling. In this mode, the user does not need to interleave all UTOPIA Addresses (that are to be applied to the "TxUAddr[4:0]" input pins) with the NULL Address. ***NOTE:*** *In order to configure the Transmit UTOPIA Interface block to operate in the "Back-to-Back Polling" Mode, the user must also do the following.* **a.** Configure the Transmit UTOPIA Interface to operate in the "UTOPIA Level 3" Mode. This is accomplished by setting Bit 7 (UTOPIA Level 3 Disable) within this Register to "0". **b.** Configure the Transmit UTOPIA Interface to support "Multi-PHY" Polling. This is accomplished by setting Bit 6 (Multi-PHY Polling Enable) within this register to "1".
4	Direct Status Indication Enable	R/W	

<table>
<tr><th>BIT NUMBER</th><th>NAME</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>3 - 2</td><td>UTOPIA/POS-PHY Data Bus Width[1:0]</td><td>R/W</td><td>UTOPIA/POS-PHY Data Bus Width[1:0]:
These READ/WRITE bit-fields permit the user to select the width of the Transmit UTOPIA and POS-PHY Data Buses. The relationship between the contents of these bit-fields and the corresponding widths of the Transmit UTOPIA and POS-PHY Data Bus is tabulated below.
<table><tr><th colspan="2">UTOPIA/POS-PHY Data Bus Width[1:0]</th><th>Corresponding UTOPIA/POS-PHY Data Bus Width</th></tr><tr><td>0</td><td>0</td><td>Not Valid</td></tr><tr><td>0</td><td>1</td><td>8 bits</td></tr><tr><td>1</td><td>0</td><td>16 bits</td></tr><tr><td>1</td><td>1</td><td>Not Valid</td></tr></table></td></tr>
<tr><td>1 - 0</td><td>Cell Size[1:0]</td><td></td><td>Cell Size[1:0]:
These two READ/WRITE bit-fields permit the user to specify the size of the ATM cell that will be handled by the Transmit UTOPIA Interface blocks. The relationship between the contents of these bit-fields and the corresponding Cell Sizes are tabulated below.
<table><tr><th colspan="2">Cell Size[1:0]</th><th>Resulting Cell Size (Bytes)</th></tr><tr><td>0</td><td>0</td><td>52 bytes</td></tr><tr><td>0</td><td>1</td><td>53 bytes (Only valid for UTOPIA Level 1, and if the UTOPIA Data Bus Width is set to 8 bits)</td></tr><tr><td>1</td><td>0</td><td>54 bytes (Only valid for UTOPIA Levels 1 and 2)</td></tr><tr><td>1</td><td>1</td><td>56 bytes</td></tr></table>
NOTE: The user must bear in mind the UTOPIA Level and the UTOPIA Data Bus width selected, when selecting the Cell Size.</td></tr>
</table>

TRANSMIT UTOPIA PORT ADDRESS REGISTER (ADDRESS = 0X0593)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Transmit UTOPIA Port Address[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 5	Unused	R/O	
4 - 0	Transmit UTOPIA Port Address[4:0]	R/W	**Transmit UTOPIA Port Address[4:0]:** These READ/WRITE register bits, along with the "Transmit UTOPIA Port Number[4:0]" bits (within the "Trasnmit UTOPIA Port Number" Register (Address = 0x0597) permit the user to assign a unique Transmit UTOPIA address the XRT74L74 device. For UTOPIA Level 2/3 applications, the user can write in any value, ranging from 0x00 through 0x1E into this register. **The Transmit UTOPIA Address Assignment Procedure:** In order to assign a UTOPIA Address to a given Channel (or Port) within the XRT74L74 device, the user must do the following. **a.** Write the value corresponding to a given XRT74L74 Channel into the "Transmit UTOPIA Port Number" Register (Address = 0x0597). **b.** Write the corresponding UTOPIA Address value into this register. Once this "two-step" procedure has been executed, then the XRT74L74 Channel (as specified during step "a") will be assigned the "Transmit UTOPIA Address" value (as specified during step "b").

TRANSMIT UTOPIA PORT NUMBER REGISTER (ADDRESS = 0X0597)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Transmit UTOPIA Port Number[4:0]				
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

BIT NUMBER	NAME	TYPE	DESCRIPTION
7 - 5	Unused	R/O	

BIT NUMBER	NAME	TYPE	DESCRIPTION
4 - 0	Transmit UTOPIA Port Number[4:0]	R/W	**Transmit UTOPIA Port Number[4:0]:** These READ/WRITE register bits, along with the "Transmit UTOPIA Port Address[4:0]" bits (within the "Transmit UTOPIA Port Address" Register (Address = 0x0593) permit the user to assign a unique Transmit UTOPIA address to each XRT74L74 device. **The Transmit UTOPIA Address Assignment Procedure:** In order to assign a UTOPIA Address to a given Channel (or Port) within the XRT74L74 device, the user must do the following. **a.** Write the value corresponding to a given XRT74L74 Channel into this register. **b.** Write the corresponding UTOPIA Address value into the "Transmit UTOPIA Port Address" Register (Address = 0x0593). Once this "two-step" procedure has been executed, then the XRT74L74 Channel (as specified during step "a") will be assigned the "Transmit UTOPIA Address" value (as specified during step "b").

2.0 MICROPROCESSOR INFO

3.0 TRANSMIT SECTION

The purpose of the Transmit section of the XRT74L74 DS3/E3 ATM UNI 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 DS3 transport medium.

The Transmit section of the DS3/E3 UNI chip consists of the following blocks:

- Transmit UTOPIA Interface
- Transmit Cell Processor
- Transmit PLCP Processor
- Transmit DS3/E3 Framer

The ATM Layer processor will write ATM Cell Data to 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 TxFIFO 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 cell from the TxFIFO. 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 PLCP Processor (or Transmit DS3 Framer, if the “Direct Mapped” ATM option is selected). If the TxFIFO (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—44.736 Mbps in this case). The Transmit Cell Processor has provisions to allow the for the generation and transmission of 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 PLCP Processor block will take 12 ATM cells and pack them into a single PLCP frame. In addition to the ATM Cells, the PLCP frame will consists of numerous overhead bytes and either a 13 or 14 nibble trailer to frequency justify the PLCP frame to the specified 8 kHz frame rate. Once these PLCP frames have been formed they will be transferred to the Transmit DS3 Framer.

The Transmit DS3 Framer will take the PLCP frame (or ATM cells, if the Direct-Mapped ATM option was selected), and insert this data into the payload portions of the DS3 frame. The Transmit DS3 Framer will also generate and insert overhead bits 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 DS3 Framer supports both the C-bit Parity and M13 Framing Formats.

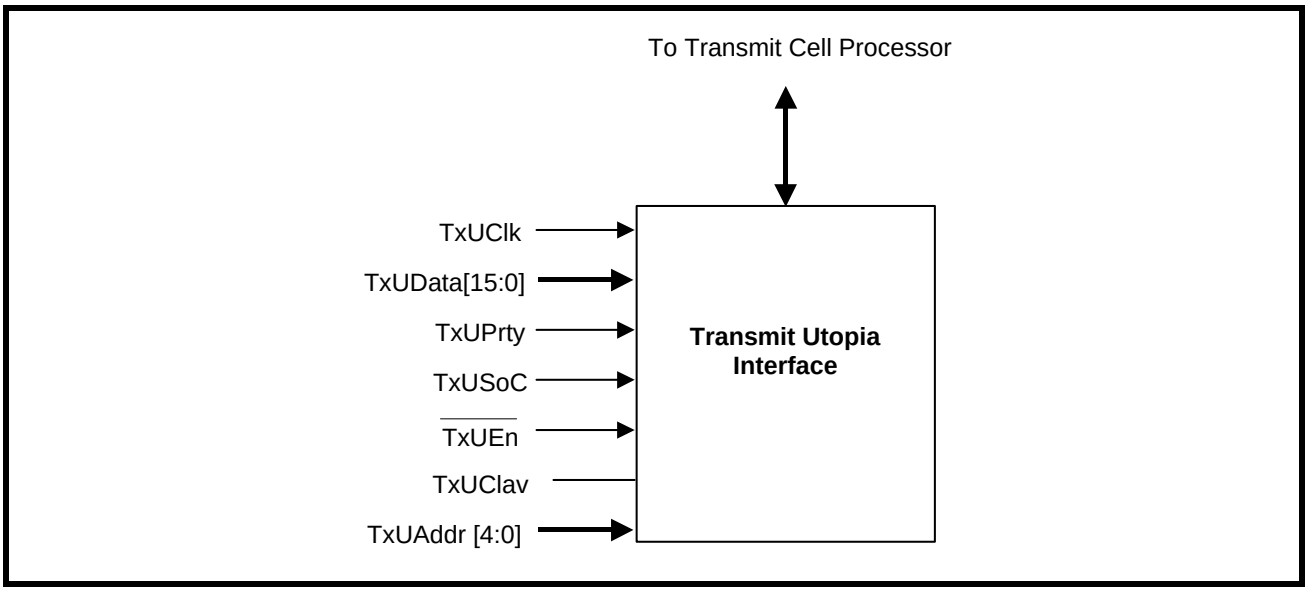
The following sections discuss the blocks comprising the Transmitter Portion of the DS3/E3 UNI in detail.

3.1 Transmit UTOPIA Interface Block

3.1.1 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, DS3), the received data (from the ATM layer processor) is written into an internal FIFO. This FIFO will be referred to as the TxFIFO throughout this document. The contents of the TxFIFO 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 TxUClav pin, Figure 3 presents a simple illustration of the Transmit UTOPIA interface block and the associated pins.

FIGURE 3. SIMPLE BLOCK DIAGRAM OF TRANSMIT UTOPIA INTERFACE



3.1.2 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 TxUClav 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.

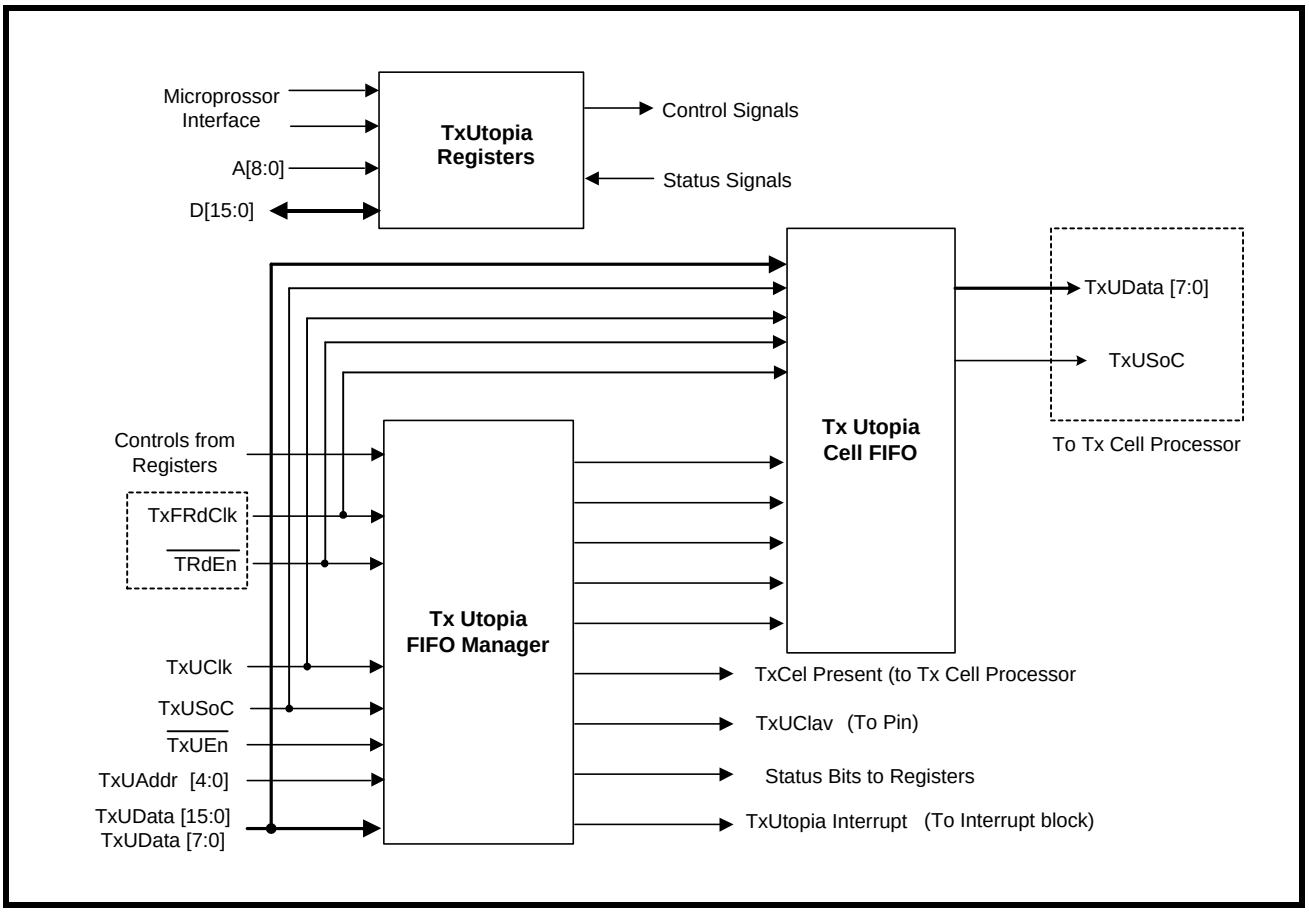
- Transmit UTOPIA Input Interface
- Transmit UTOPIA Configuration/Status Registers
- Transmit UTOPIA FIFO Manager
- Transmit UTOPIA Cell FIFO (TxFIFO)

The Transmit UTOPIA Interface block consists of an input interface which complies to the “UTOPIA Level 2 interface specifications”, and the TxFIFO. 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 TxUPrty input pin, while the corresponding data byte [word] is present at the TxUData[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 TxFIFO where it can be read-in and further processed by the Transmit Cell Processor. A FIFO manager maintains the TxFIFO and indicates FIFO empty, FIFO full, cell space available, etc. Figure 4 presents a functional block diagram of the Transmit UTOPIA Interface Block.

FIGURE 4. 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 to select/configure these features in order to suit particular application needs. Detailed discussion of Single-PHY and Multi-PHY operation will each be presented in its own section even though it involves the use of all of these functional blocks.

3.1.2.1 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 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.

3.1.2.1.1 The Pins of the Transmit UTOPIA Bus Interface

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

- TxUData[15:0]—Transmit UTOPIA Data Bus Input pins
- TxUAddr[4:0]—Transmit UTOPIA Address Bus Input pins
- TxUClk—Transmit UTOPIA Interface block clock input pin
- TxUSoC—Transmit “Start of Cell” indicator input pin
- TxUPrty—Transmit UTOPIA—Odd Parity Input pin
- $\overline{\text{TxUEn}}$—Transmit UTOPIA Data Bus—Write Enable input pin
- TxUClav—TxFIFO Cell Available

Each of these signals are briefly discussed below.

TxUData[15:0] Transmit UTOPIA Data Bus inputs

The ATM Layer Processor will write its ATM Cell Data into the Transmit UTOPIA Interface block, by placing it, in a byte-wide (or word-wide) manner on these input pins. The Transmit UTOPIA Data Bus can be configured to operate in the "8-bit wide" or "16-bit wide" mode (See Section 6.1.2.1.2). If the "8-bit wide" mode is selected, then only the TxUData[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., TxUData[15:0]) are active. The Transmit UTOPIA Data bus is tri-stated while the active-low TxUEn (Transmit UTOPIA Data Bus—Write Enable) input signal is "high". Therefore, the ATM Layer processor must assert this signal (e.g., toggling TxUEn "low") in order write the cell data, on the Transmit UTOPIA Data bus, into the Transmit UTOPIA Interface Block. The data on the Transmit UTOPIA Data Bus is sampled and latched into the Transmit UTOPIA Interface block, on the rising edge of the Transmit UTOPIA Interface Block Clock signal, TxUClk.

Additionally, the Transmit UTOPIA Interface block will only process one cell worth of data (e.g., 52, 53 or 54 bytes, as configured via the CellOf52Bytes option—See Section 6.1.2.1.3), following the latest assertion of the TxUSoC (Transmit-Start of Cell) pin. Afterwards, the Transmit UTOPIA Data bus will become tri-stated and will cease to process any more data from the ATM Layer Processor until the next assertion of the TxUSoC pin. Once the Transmit UTOPIA Interface block reaches this condition, it will ignore the assertions of the TxUEn pin, and will keep the Transmit UTOPIA Data bus input pins tri-stated until the ATM Layer Processor pulses the TxUSoC input pin, once again.

If the Transmit UTOPIA Interface block detects a "runt" cell (e.g., if the amount of data that is read into the TxFIFO is less than that configured via the "CellOf52Bytes" option), then the Transmit UTOPIA Interface block will discard this cell, and resume normal operation.

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

These input pins are 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.

TxUClk—Transmit UTOPIA Interface Block Clock signal input pin

The Transmit UTOPIA Interface block uses this signal to sample and latch the data on the Transmit UTOPIA Data bus into the Transmit UTOPIA Address block (for Multi-PHY operation) into the Transmit UTOPIA Interface block. This clock signal can run at frequencies of 25 MHz, 33 MHz, or 50 MHz.

$\overline{\text{TxUEn}}$ —Transmit UTOPIA Data Bus—Write Enable input

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

TxUPrty—Transmit UTOPIA—Odd Parity Bit Input Pin

The ATM Layer Processor is expected to compute the odd-parity value of each byte (or word) of ATM Cell data that it intends to place on the Transmit UTOPIA Data bus. The ATM Layer Processor is then expected to apply this parity value at the TxUPrty pin, while the corresponding byte (or word) is present on the Transmit UTOPIA Data Bus.

TxUSoC—Transmit UTOPIA—"Start of Cell" Indicator

The ATM Layer processor is expected to pulse this signal "high", for one clock period of TxUClk, 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 TxUSoC pin "high", the Transmit UTOPIA Interface Block will proceed to read in and process only one cell of data (e.g., 52, 53, or 54 bytes, as configured via the "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 TxUSoC 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 TxUSoC 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 Transmit UTOPIA Interface block is configured to process 53 bytes per cell, then following the assertion of the TxUSoC pin (which is coincident with the placement of the first byte of the cell on the Transmit UTOPIA Data bus), the Transmit UTOPIA Interface block will read in and process 52 more bytes of data via the Transmit UTOPIA data bus resulting in a total of 53 bytes being processed. After the Transmit UTOPIA Interface block has read in the 53rd byte, it will no longer read in any more data from the ATM Layer Processor, until the TxUSoC pin has been asserted.

Example-2

If the ATM Layer processor were to prematurely asserts the TxUSoC 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.

TxUClav/TFullB*—TxFIFO 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 1.1.2.2.1 for more information regarding this signal.

Selecting the UTOPIA Data Bus Width

The UTOPIA data bus width can be selected to be either 8 or 16 bits by writing the appropriate data to the UTOPIA Configuration Register, as shown below.

UTOPIA Configuration Register: Address = 6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[1, 0]		UtWidth16
RO		R/W	R/W	R/W	R/W		R/W

If a UTOPIA Data Bus width of 8 bits is chosen, then only the Transmit UTOPIA Data inputs: TxUData[7:0] will be active. (The input pins: TxData[15:8] will not be active). If a UTOPIA Data bus width of 16 bits is chosen, then all of the Transmit UTOPIA Data inputs: 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 3: 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 also affects the width of the Receive UTOPIA Data bus.*
2. *Upon power up or reset, the UTOPIA Data Bus width will be 8 bits. Therefore, a “1” must be written to this bit in order to set the width of the Transmit UTOPIA (and the Receive UTOPIA) to 16 bits.*

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

The UNI can be configured to select the number of octets per cell that the Transmit UTOPIA Interface block will process, following each assertion of the TxUSoC input pin. Specifically, the following cell size options are available.

- 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 selection is made by writing the appropriate data to bit 3 (CellOf52 Bytes) within the UTOPIA Configuration Register, as depicted below.

UTOPIA Configuration Register: Address = 6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[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 4: 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.

3.1.2.1.3 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 TxUPrty 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 TxUPrty 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 Transmit UTOPIA Interface block can be configured to discard or retain this "errored" cell by writing the appropriate data to the Transmit UTOPIA Interrupt/Status Register (Address = 6Eh) as depicted below.

Transmit UTOPIA Interrupt/Status Register (Address = 6Eh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TFIFO Reset	Discard Upon PErr	TPerr IntEn	TFIFO ErrIntEn	TCOCA IntEn	TPErr IntStat	TFIFO" OverInt Stat	TCOCA IntStat
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

If this bit is set to a "1", then the Transmit UTOPIA Input Interface block will discard the errored cell. If this bit-field to is set to "0", then the Transmit UTOPIA Interface block will not discard the errored cell and this cell will be written into the TxFIFO.

3.1.2.2 Transmit UTOPIA FIFO Manager

The TxFIFO Manager has the following responsibilities.

- Monitoring the fill level of the TxFIFO, 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 TxFIFO can resume normal operation following the removal of the runt cell.
- Insuring that the TxFIFO can respond properly to an “Overrun” condition, by generating the “TxFIFO 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 these features can be customized to suit particular application needs.

The Transmit UTOPIA FIFO Manager provides the following options.

- Handshaking Mode (Octet Level vs Cell Level)
- User selected Operating TxFIFO Depth
- Resetting the TxFIFO
- Monitoring the TxFIFO

3.1.2.3 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.

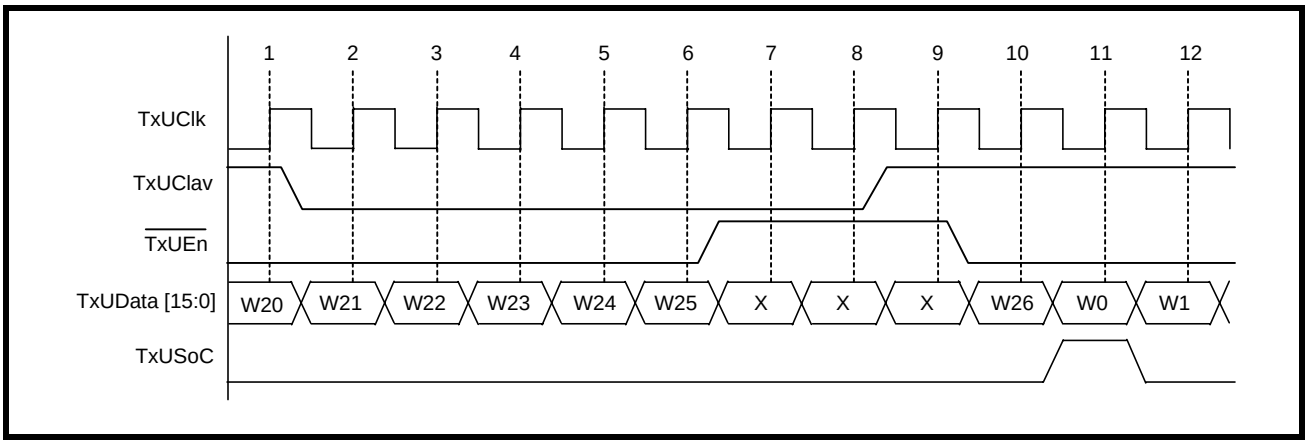
3.1.2.3.0.1 Octet-Level Handshaking

The UNI will be operating in the “Cell-Level” Handshaking Mode following power up or reset. Therefore, the bit 5 (Handshaking Mode) of the UTOPIA Configuration Register to must be set “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 TxUClav output pin. The ATM Layer processor is expected to monitor the TxUClav output pin in order to determine if it is OK to write data into the TxFIFO. The TxUClav output pin exhibits a role that is similar to CTS (Clear to Send) in RS-232 based data transmission systems. As long as TxUClav 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 TxUClav pin toggles “low”, this indicates that the TxFIFO can only accept 4 (or less) more write operations from the ATM Layer processor. Once the TxUClav 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 TxUClav 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 TxUClav signal will toggle “low” when the TxFIFO can only accept 8 (or less) bytes of ATM cell data from the ATM Layer processor.

Figure 5 presents a timing diagram illustrating the behavior of TxUClav during writes to the Transmit UTOPIA Interface block, while operating in the Octet-Level Handshaking Mode.

FIGURE 5. TIMING DIAGRAM OF TXUCLAV/TXFULLB AND VARIOUS OTHER SIGNALS DURING WRITES TO THE TRANSMIT UTOPIA, WHILE OPERATING IN THE OCTET-LEVEL HANDSHAKING MODE.



Note: regarding Figure 5

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 5 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.

In Figure 5 , TxUClav is initially “high” during clock edge # 1. However, shortly after the ATM Layer processor writes in word W20, TxUClav toggles “low”, indicating that the TxFIFO is starting to fill up. The ATM Layer processor will detect this “negation of TxUClav” 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 TxUEn. The ATM Layer processor must keep TxUEn negated until it detects that TxUClav has once again returned “high”. In Figure 5 , TxUClav is asserted after clock edge #8. The ATM Layer processor detects this transition in TxUClav at clock edge #9; and subsequently, asserts TxUEn. The ATM Layer resumes writing in more ATM cell data into the Transmit UTOPIA Interface block.

3.1.2.3.0.2 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 TxUClav is at a logic “1”, it means that the TxFIFO has enough remaining empty space for it to receive at least one more full cell of data from the ATM Layer processor. However, when TxUClav toggles from “high” to “low”, it indicates that the very next cell (following the one that is currently being written) cannot be accepted by the TxFIFO. Conversely, once TxUClav has returned to the logic “1” level, it indicates that at least one more full cell may be written into the TxFIFO by the ATM Layer processor. As in the “Octet-Level” Handshake mode, the ATM Layer processor is expected to poll the TxUClav output towards the end of transmission of the cell currently being written and to proceed with transmission of the next cell only if TxUClav 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 UNI can be configured 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 = 6Ah

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[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 5: THE RELATIONSHIP BETWEEN THE CONTENTS IN BIT FIELD 5 (HANDSHAKE MODE) WITHIN THE UTOPIA CONFIGURATION REGISTER AND THE RESULTING UTOPIA INTERFACE HANDSHAKE MODE.

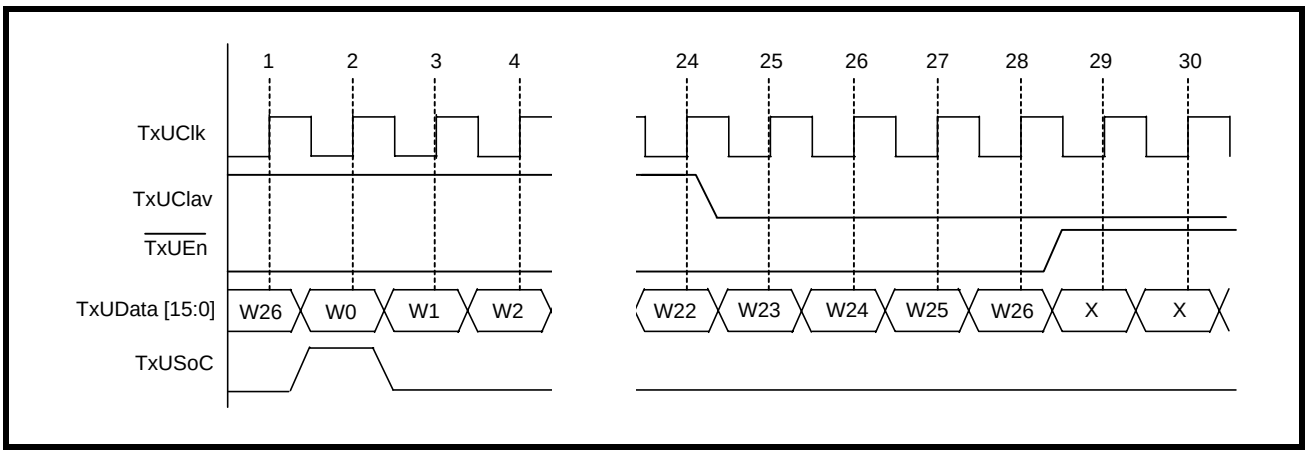
VALUE	UTOPIA INTERFACE HANDSHAKE MODE
0	The UTOPIA Interfaces operate in the octet level handshake mode.
1	The UTOPIA Interfaces operate in the cell level handshake mode.

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, a "0" must be written to this bit-field in order to configure the UNI into the "Octet Level Handshaking" mode.*

Figure 6 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 6. TIMING DIAGRAM OF VARIOUS TRANSMIT UTOPIA INTERFACE BLOCK SIGNALS, WHEN THE TRANSMIT UTOPIA INTERFACE BLOCK IS OPERATING IN THE "CELL LEVEL HANDSHAKING" MODE.



Note: regarding Figure 6

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 6 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.*

In Figure 6 , 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, TxUClav toggles “low”. In the “Cell-Level” Handshaking mode, this means that the 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 $\overline{\text{TxUEn}}$ signal after clock edge #28. At this point, the ATM Layer processor must wait until TxUClav toggle “high” once again; before writing in the next ATM cell.

3.1.2.3.1 Selecting the Operating Depth of the TxFIFO

The physical depth of the TxFIFO is 16 cells but can be operated with a smaller FIFO depth. Therefore, the UNI allows the selection of operating depths of 4, 8, 12 or the full 16 cells. This selection can be made by writing the appropriate data to Bits 1 and 2 (TFIFODepth[1, 0]) within the UTOPIA Configuration Register, as depicted below .

UTOPIA Configuration Register: Address = 6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[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 6: 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 appropriate data must be written to these two bit-fields in order to change this parameter.

3.1.2.3.2 Resetting the TxFIFO via Software Command

The UNI allows the TxFIFO to be reset 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) of the Transmit UTOPIA Interrupt Enable/Status Register as depicted below.

Transmit UTOPIA—Interrupt/Status Register (Address—6Eh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TFIFO Reset	Discard Upon PErr	TPerr IntEn	TFIFO ErrIntEn	TCOCA IntEn	TPErr IntStat	TFIFO OverInt Stat	TCOCA IntStat
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

3.1.2.3.3 Monitoring the TxFIFO Status

The local µP has the ability to poll and monitor the status of the TxFIFO via the Transmit UTOPIA FIFO

Status Register (Address = 71h). The bit format of this register is presented below.

Transmit UTOPIA 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 the corresponding meaning.

TxFIFO Full

TXFIFO FULL (BIT 1)	MEANING
0	TxFIFO is full, the ATM Layer processor risks causing an overrun if it writes to the TxFIFO now.
1	TxFIFO is not full.

TxFIFO Empty

TXFIFO EMPTY (BIT 0)	MEANING
0	TxFIFO is not empty
1	TxFIFO is empty. The TxCell Processor is currently generating IDLE cells

3.1.2.4 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.

3.1.2.4.1 Single PHY Operation

The UNI chip will be operating in the Multi-PHY mode upon power up or reset. Therefore, a “1” must be written to Bit 4 within the UTOPIA Configuration register (Address = 6Ah) in order to configure the UNI into the Single-PHY Mode.

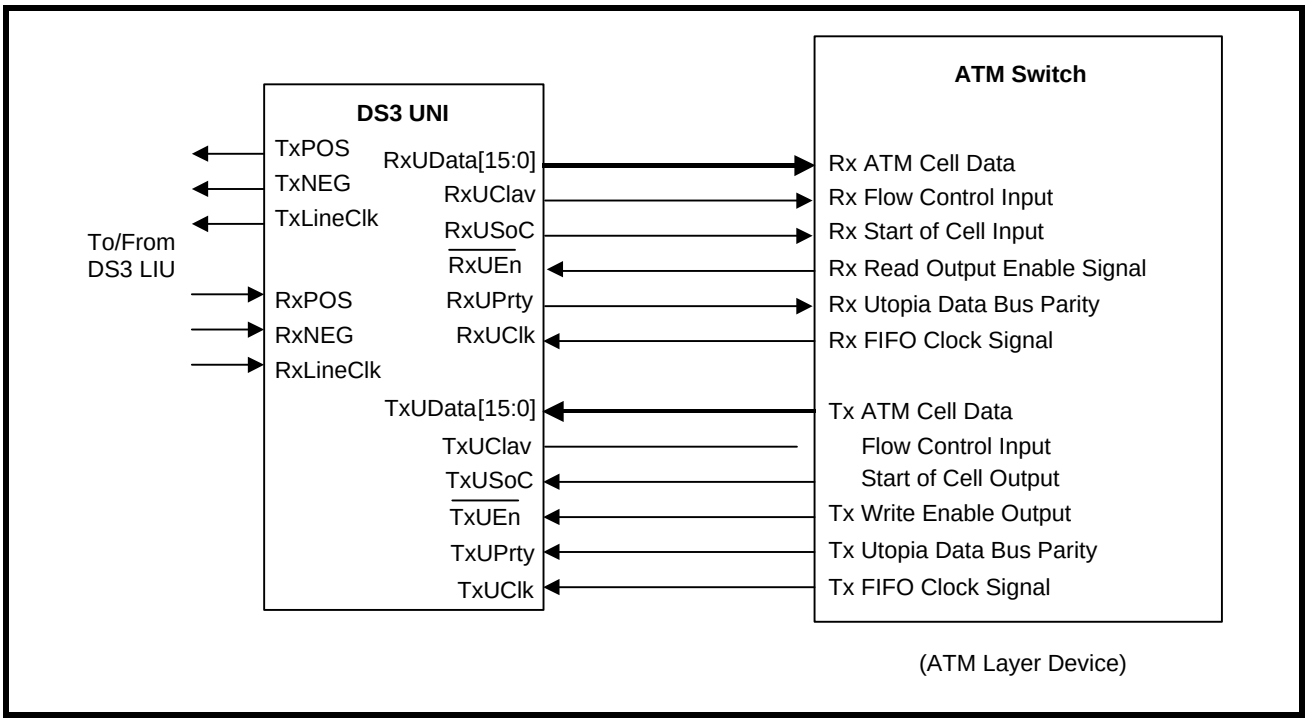
UTOPIA Configuration Register: Address = 6Ah

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Handshake Mode	S-PHY/M-PHY*	CellOf52 Bytes	TFIFODepth[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 in Figure 7 .

FIGURE 7. 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 7.4.2.2.2.1. 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 TxUClav output pin.*

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

Note: The actual meaning of TxUClav 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 TxUSoC pin "high" for one clock period of TxUClk.

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

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

4. *Assert the "Transmit UTOPIA Data Bus"—Write Enable Signal, $\overline{\text{TxUEn}}$.*

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

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

If the UNI is operating in the Octet-Level handshake mode, then the ATM Layer processor should check the level of the TxUClav 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 TxUClav signal, as it nears completion of writing in a given cell.

The above-mentioned procedure is also depicted in Flow-Chart Form in Figure 8 ; and in Timing Diagram form in Figure 9 and 10.

FIGURE 8. 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.

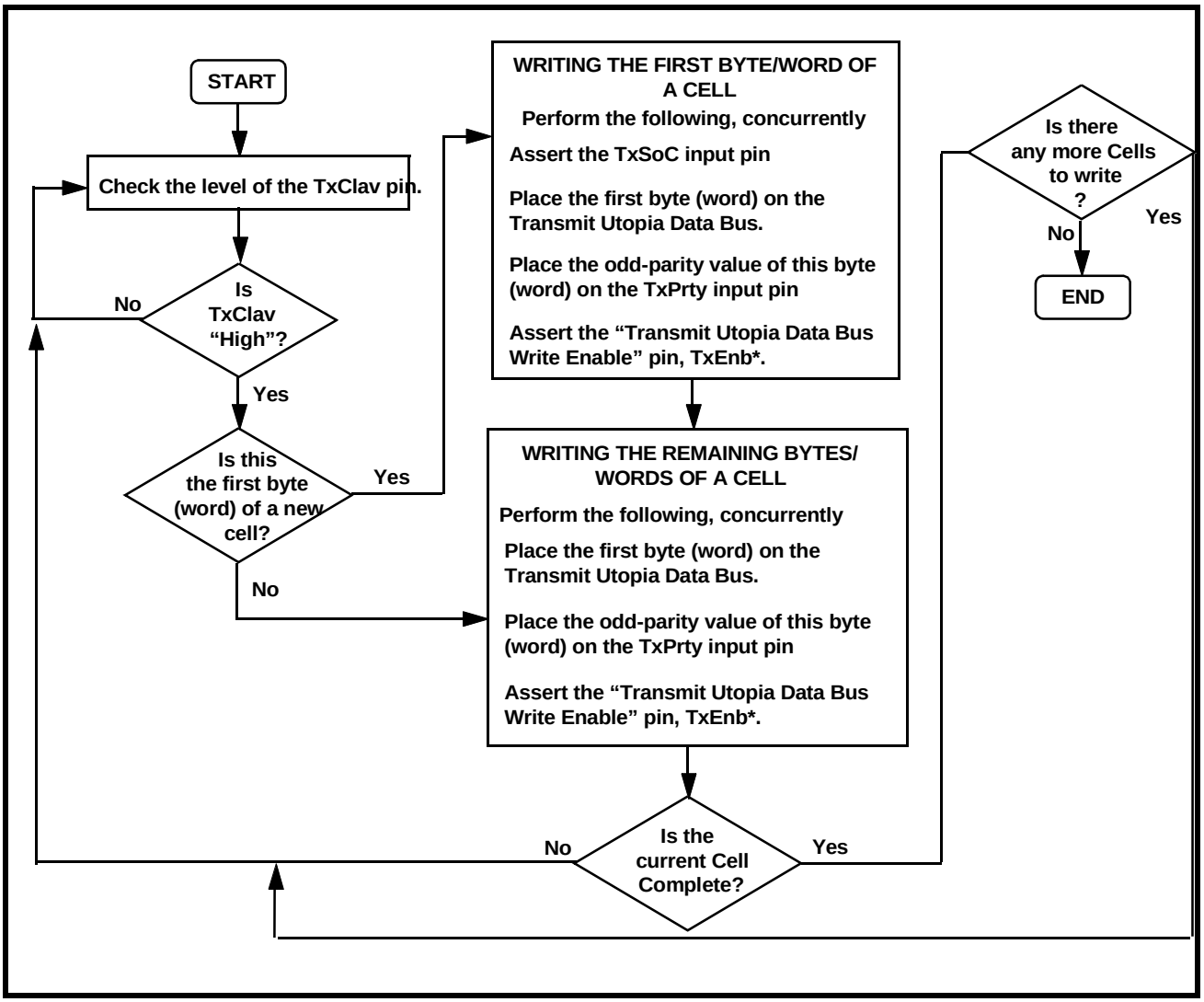
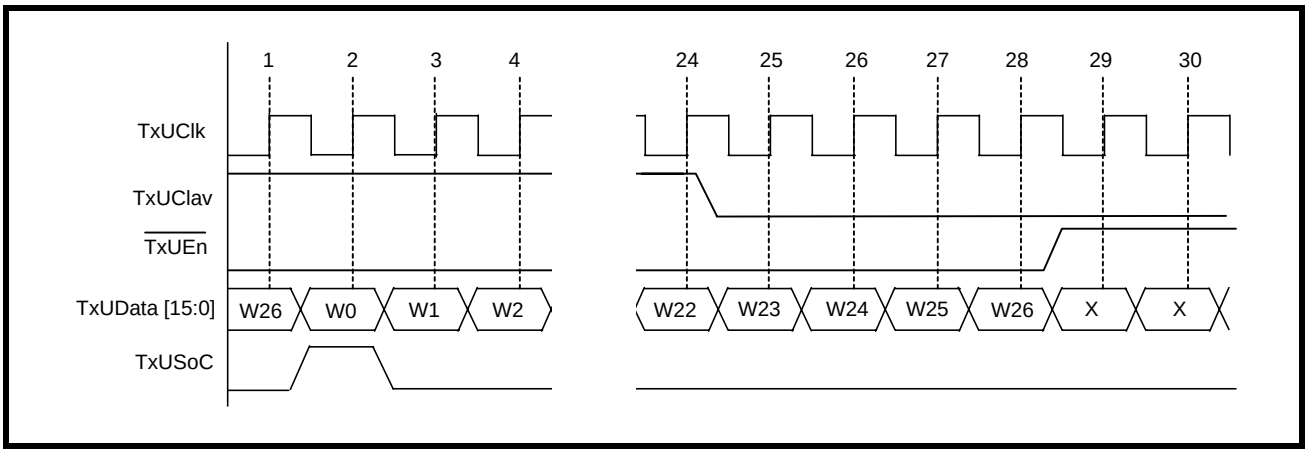


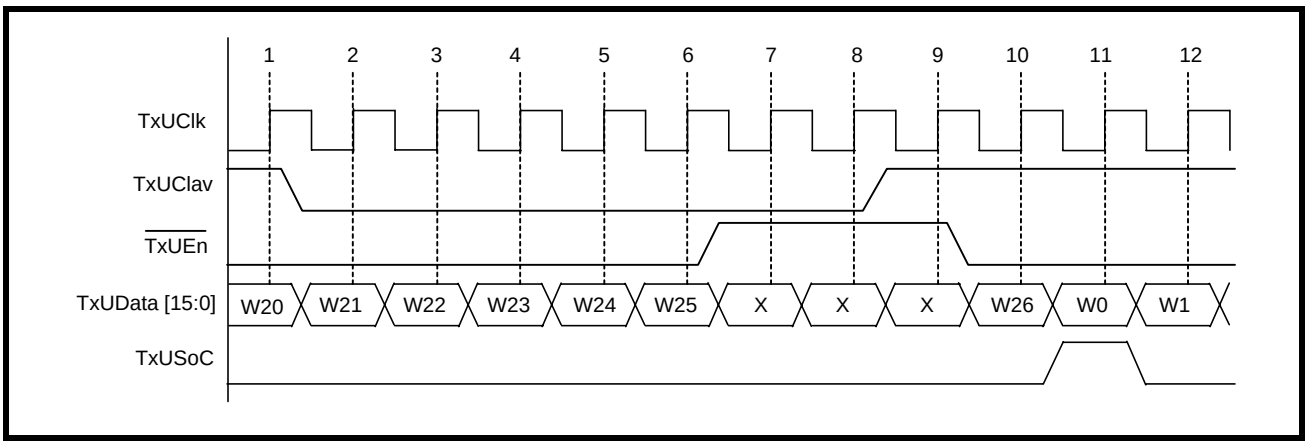
FIGURE 9. TIMING DIAGRAM OF ATM LAYER PROCESSOR TRANSMITTING DATA TO THE UNI OVER THE UTOPIA DATA BUS, (SINGLE -PHY MODE/CELL-LEVEL HANDSHAKING).



Note: regarding Figure 9

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 9 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 10. TIMING DIAGRAM OF ATM LAYER PROCESSOR TRANSMITTING DATA TO THE UNI OVER THE UTOPIA DATA BUS (SINGLE-PHY MODE/OCTET-LEVEL HANDSHAKING).



Note: regarding Figure 10

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 10 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 TxUClav pin is used as a data flow control pin, and has a role somewhat similar to RTS (Request To Send) in RS-232 based data transmission. The TxUClav pin will have a slightly different role when the 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 ex-

pected to poll the TxUClav output pin before writing the next byte, word or cell to the TxFIFO.

3.1.2.4.2 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 which 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. This address value is assigned to a particular "Transmit UTOPIA Interface block" by writing this address value into the "TxUTOPIA Address Register" (Address = 70h) within its "host" UNI device. The bit-format of the "TxUTOPIA Address Register" is presented below.

TxUTOPIA Address Register (Address = 70h)

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, a "UTOPIA address" value is assigned 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 = 6Ch) within the "host" UNI device. The bit-format of the "Rx UTOPIA Address Register" is presented below.

Rx UTOPIA Address Register (Address = 6Ch)

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 7.4.2.2.2.2.

3.1.2.4.2.1 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 11. AN ILLUSTRATION OF MULTI-PHY OPERATION WITH UNI DEVICES #1 AND #2

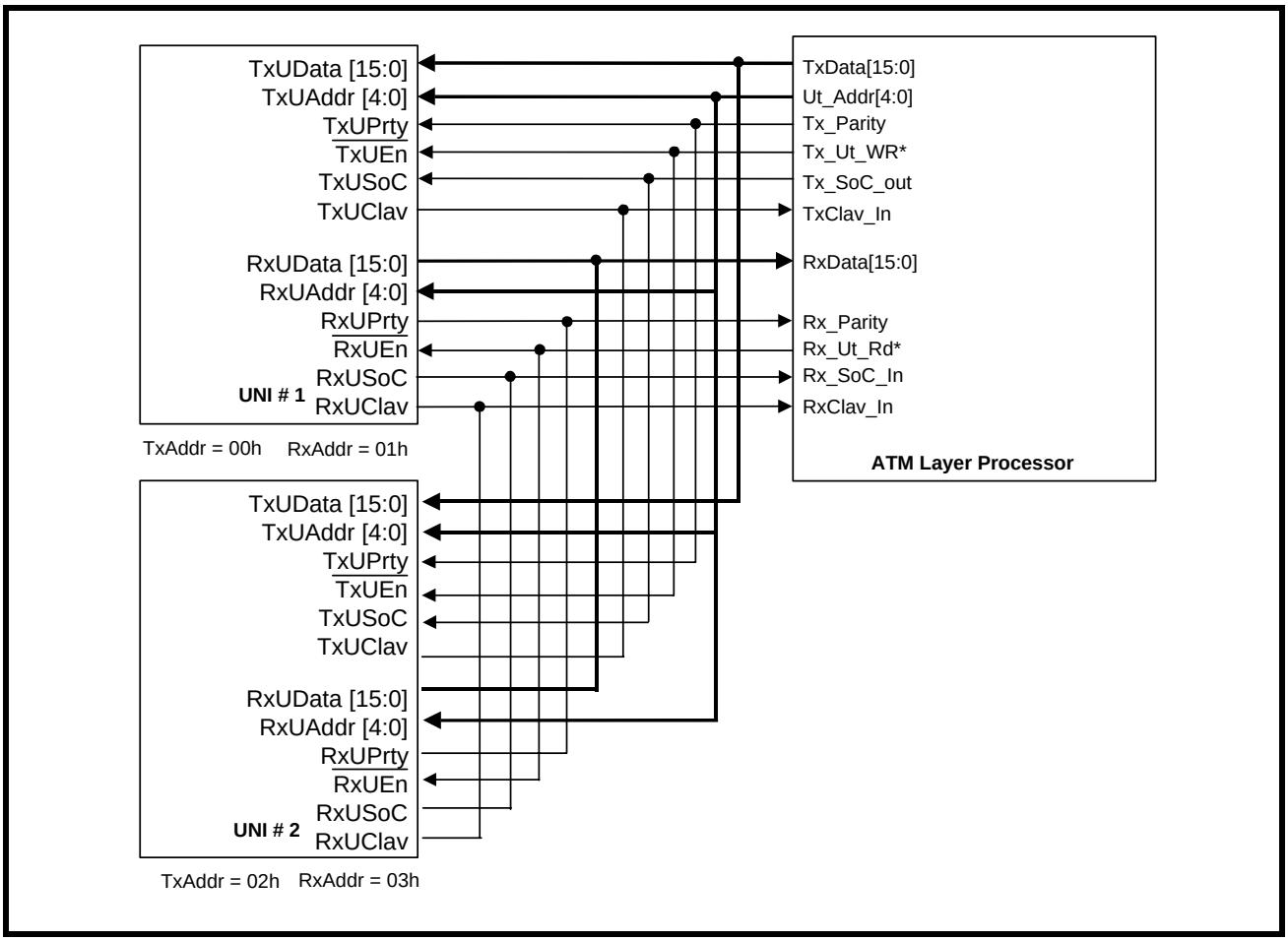


Figure 11 depicts a "Multi-PHY" system consisting of an ATM Layer processor and two (2) UNI devices, which are designated as "UNI #1" and "UNI #2". In this figure, both of the UNIs are connected to the ATM Layer processor via a common "Transmit UTOPIA" Data Bus, a common "Receive UTOPIA" Data Bus, a common "TxUClav" line, a common "RxClav" line, as well as common TxUEn, RxUEn, TxUSoC and RxUSoC lines. The ATM Layer processor will also be addressing both the Transmit and Receive UTOPIA Interface blocks via a common "UTOPIA" address bus (Ut_Addr[4:0]) Therefore, the Transmit and Receive UTOPIA Interface Blocks, within a given UNI might have different addresses; as depicted in Figure 11 .

The UTOPIA Address values that have been assigned to each of the Transmit and Receive UTOPIA Interface blocks, within Figure 11 , are listed below in Table 7 .

TABLE 7: UTOPIA ADDRESS VALUES OF THE UTOPIA INTERFACE BLOCKS ILLUSTRATED IN FIGURE 11 .

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 "TxUTOPIA Address Register" (Address = 70h) within their "host" UNI device. The discussion of the

Receive UTOPIA Interface blocks, within UNIs #1 and #2 is presented in Section 7.4.2.2.2.2.1.

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 $\overline{\text{TxUEn}}$ input pin (if it is not asserted already).*

The UNI device (being "polled") will know that this is only a "polling" operation, if the $\overline{\text{TxUEn}}$ 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 "TxUClav_in" input pin (see Figure 9).*

The UNI Devices Role in the "Polling" Operation

UNI #2 will sample the signal levels placed on its TxUTOPIA Address input pins (TxUAddr[4:0]) on the rising edge of its "Transmit UTOPIA Interface block" clock input signal, TxUClk. Afterwards, UNI #2 will compare the value of these "Transmit UTOPIA Address Bus input pin" signals with that of the contents of its "TxUTOPIA Address Register (Address = 70h).

If these values do not match, (e.g., TxUAddr[4:0] ¦02h) then UNI #2 will keep its "TxUClav" output signal "tri-stated"; and will continue to sample its "Transmit UTOPIA Address bus input" pins; with each rising edge of TxUClk.

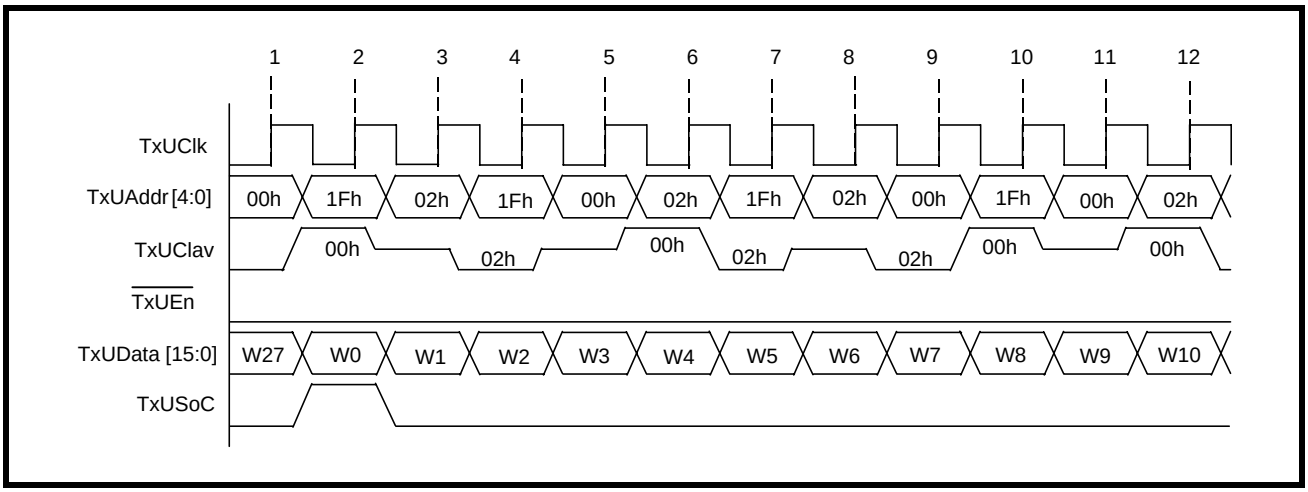
If these two values do match, (e.g., TxUAddr[4:0] = 02h) then UNI #2 will drive its "TxUClav" output pin to the appropriate level, reflecting its TxFIFO "fill-status". Since the UNI is automatically operating in the "Cell Level Handshaking" mode while it is operating in the "Multi-PHY" mode; the UNI will drive the TxUClav output signal "high" if it is capable of receiving at least one more complete cell of data from the ATM Layer processor. Conversely, the UNI will drive the "TxUClav" output signal "low" if its TxFIFO is too full and is incapable of receiving one more complete cell of data from the ATM Layer processor.

When UNI #2 has been selected for "polling", UNI #1 will continue to keeps its "TxUClav" output signal "tri-stated". Therefore, when UNI #2 is driving its "TxUClav" output pin to the appropriate level, it will be driving the entire "TxUClav" line, within the "Multi-PHY" system. Consequently, UNI#2 will also be driving the "TxUClav_in" input pin of the ATM Layer processor (see Figure 11).

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

Figure 12 presents a timing diagram that depicts the behavior of the ATM Layer processor's and the UNI's signals during polling.

FIGURE 12. TIMING DIAGRAM ILLUSTRATING THE BEHAVIOR OF VARIOUS SIGNALS FROM THE ATM LAYER PROCESSOR AND UNI, DURING POLLING.



Note: regarding Figure 12

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 12 illustrates the ATM Layer processor writing 27 words (W0 through W26) for each ATM cell.*
3. *The ATM Layer processor is currently writing ATM cell data to the Transmit UTOPIA Interface Block, within UNI #1 (TxUAddr[4:0] = 00h) during this "polling process".*
4. *The TxFIFO, within UNI#2's Transmit UTOPIA Interface block (TxUAddr[4:0] = 02h) is incapable of receiving any additional ATM cell data from the ATM Layer processor. Hence, the TxUClav line will be driven "low" whenever this particular Transmit UTOPIA Interface block is "polled".*
5. *The Transmit UTOPIA Address of 1Fh is not associated with any UNI device, within this "Multi-PHY" system. Hence, the TxUClav line is tri-stated whenever this address is "polled".*

Note: Although Figure 11 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 7.4.2.2.2.2 for a discussion on the Receive UTOPIA Interface block during Multi-PHY operation.

3.1.2.4.2.2 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 $\overline{\text{TxUEn}}$ 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 $\overline{\text{TxUEn}}$ signal.
4. Assert the TxUSoC input pin.
5. Begin applying the ATM Cell data in a byte-wide (or word-wide) manner to the Transmit UTOPIA Data Bus.

Figure 13 presents a flow-chart that depicts the "UNI Device Selection and Write" process in Multi-PHY operation.

FIGURE 13. FLOW-CHART OF THE "UNI DEVICE SELECTION AND WRITE PROCEDURE" FOR THE MULTI-PHY OPERATION.

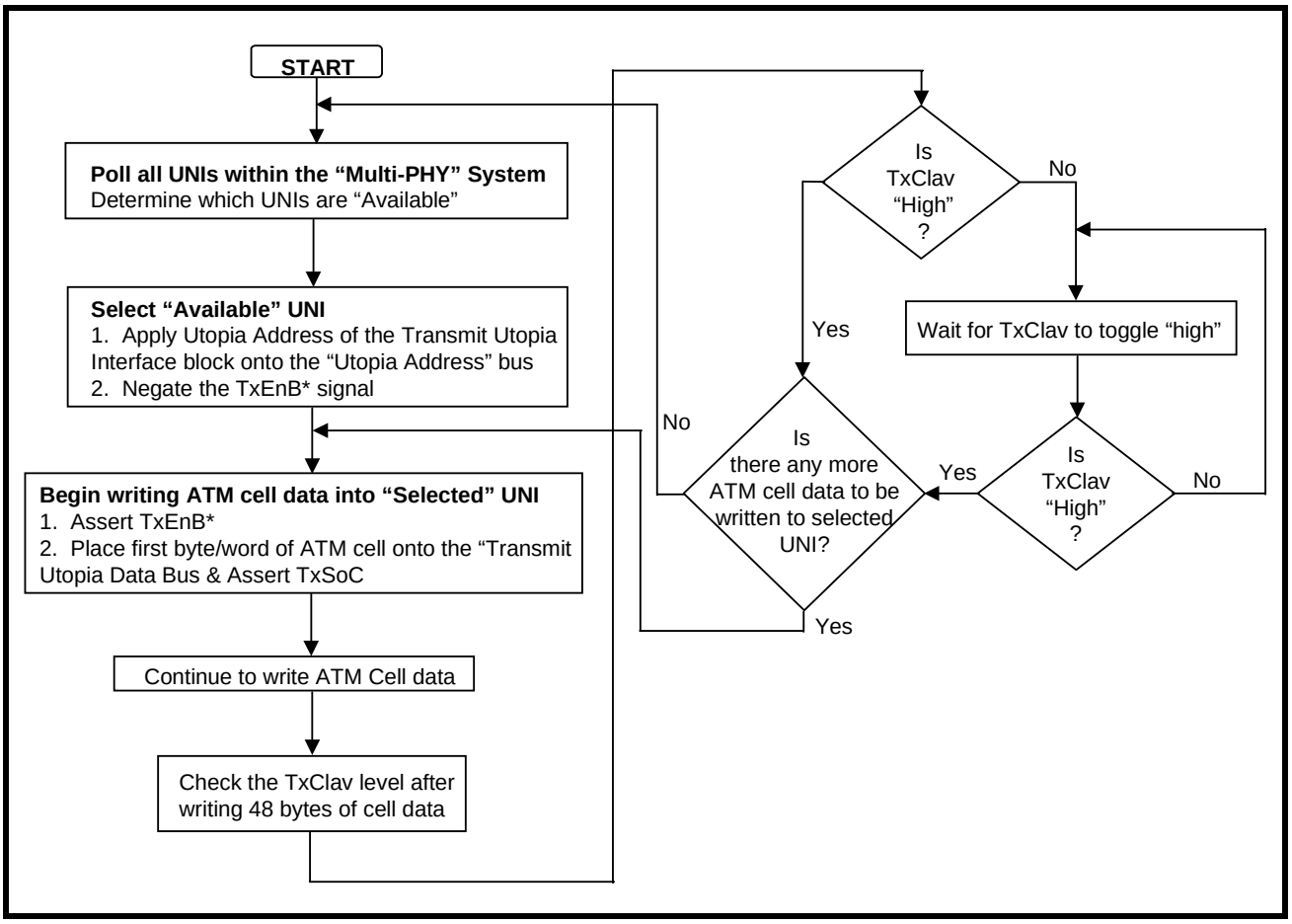
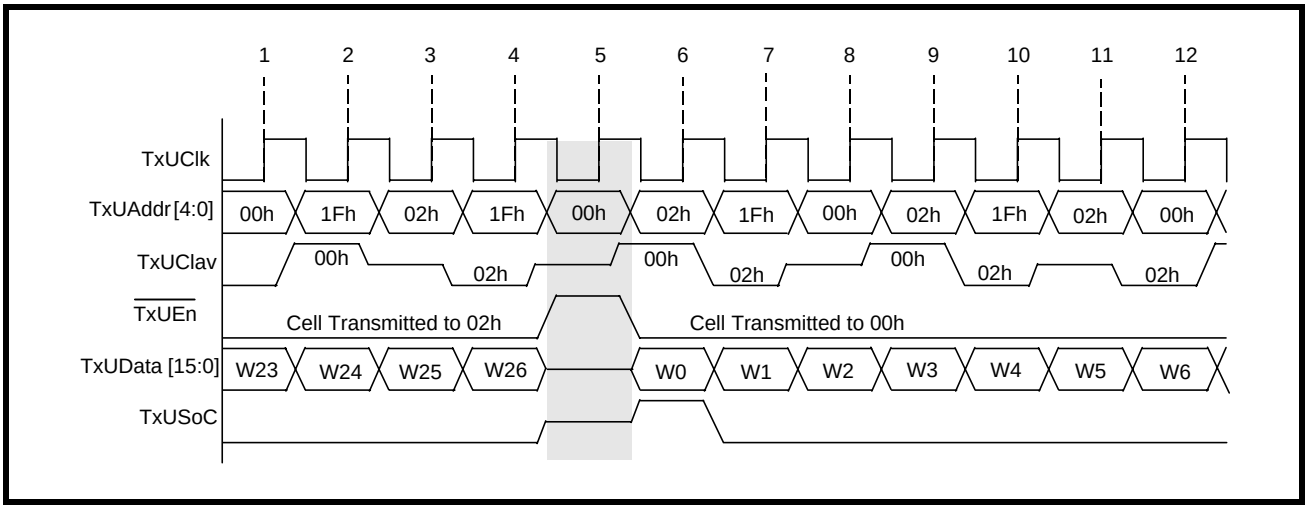


Figure 14 presents a timing diagram that illustrates the behavior of various "Transmit UTOPIA Interface block" signals during the "Multi-PHY" UNI Device Selection and Write operation.

FIGURE 14. TIMING DIAGRAM OF THE TRANSMIT UTOPIA DATA AND ADDRESS BUS SIGNALS, DURING THE "MULTI-PHY" UNI DEVICE SELECTION AND WRITE OPERATIONS.



Note: regarding Figure 14

1. *The Transmit UTOPIA Data bus is configured to be 16 bits wide. Hence, the data which the ATM Layer processor places on the Transmit UTOPIA Data bus, is expressed in terms of 16-bit words (e.g., W0–W26).*
2. *The Transmit UTOPIA Interface Block is configured to handle 54 bytes/cell. Hence, Figure 14 illustrates the ATM Layer processor writing 27 words (e.g., W0 through W26) for each ATM cell.*

In Figure 14 , the ATM Layer processor is initially writing ATM cell data to the Transmit UTOPIA Interface block within UNI #2 (TxUAddr[4:0] = 02h). However, the ATM Layer processor is also polling the Transmit UTOPIA Interface block within UNI #1 (TxUAddr[4:0] = 00h) and some "non-existent" device at TxUAddr[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 (TxUAddr[4:0] = 02h). The ATM Layer processor will indicate its intentions to select a new UNI device for writing by negating the $\overline{\text{TxUEn}}$ signal, at clock edge #5 (see the shaded portion of Figure 14). At this time, UNI #1 will notice two things:

1. The UTOPIA Address for the Transmit UTOPIA Interface block, within UNI #1 is on the Transmit UTOPIA Address bus (TxUAddr[4:0] = 00h).
2. The $\overline{\text{TxUEn}}$ 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.

3.1.2.5 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 "xxxx1xxxb" (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
Rx DS3 Interrupt Status	Rx PLCP Interrupt Status	Rx CP Interrupt Status	Rx UTOPIA Interrupt Status	TxUTOPIA Interrupt Status	TxCP Interrupt Status	TxDS3 Interrupt Status	One Sec Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RUR
x	x	x	x	1	x	x	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 TxUT Interrupt Enable/Status Register, which is located at address 6Eh within the UNI device. The bit format of this register is presented below.

TxUT Interrupt Enable /Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TFIFO Reset	Discard Upon PErr	TPerr Interrupt Enable	TxFIFO ErrInt Enable	TCOCA Interrupt Enable	TPErr Interrupt Status	TxFIFO OverInt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

The "TxUT 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 en-

able bits for the Transmit UTOPIA Interface block. Each of these "interrupt processing relevant" bit fields are defined below.

Bit 0—TCOCA Interrupt Status—Transmit UTOPIA Change of Cell Alignment Condition

If the ATM Layer Processor asserts the TxUSoC input pin prior to writing the contents of a complete cell (as configured via the CellOf52Bytes 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.

TxUT Interrupt Enable /Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO 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—TxFIFO Overrun Interrupt Status

If the TxFIFO is filled to capacity, and if the ATM Layer processor attempts to write any additional data to the TxFIFO, some of the data within the TxFIFO 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 TxUTOPIA Interrupt Enable/Status Register to "1", as depicted below.

Transmit UTOPIA Interrupt Enable /Status Register (Address—6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO 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 TxUT 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 TxUPrty 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 TxUPrty 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 (TxUT Parity Error Interrupt Status), within the Transmit UTOPIA Interrupt Enable/Status Register to “1”, as depicted below.

Transmit UTOPIA Interrupt Enable /Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO Overrun Interrupt Status	TCOCA 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 TxUT Interrupt Enable/Status register, and the INTB* output pin will all be negated, unless outstanding interrupt conditions are awaiting servicing.

Bit 3—TCOCA Interrupt Enable—Transmit UTOPIA Change of Cell Alignment Interrupt Enable

This “read/write” bit-field is used for enabling or disabling 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.

TxUT Interrupt Enable/Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO Overrun Interrupt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

Bit 4—TxFIFO ErrInt Enable—TxFIFO Overrun Condition Interrupt Enable

This “Read/Write” bit-field is used for enabling or disabling the “TxFIFO 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.

TxUT Interrupt Enable/Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO Overrun Interrupt Status	TCOCA 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 is used for enabling or disabling the “Detected Parity error” interrupt. This interrupt can be enabled 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. A “1” must be written to this bit in order to enable this interrupt.

TxUT Interrupt Enable /Status Register (Address-6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFIFO Reset	Discard Upon Parity Error	TxUT Parity Error Interrupt Enable	TxFIFO Overrun Interrupt Enable	TCOCA Interrupt Enable	TxUT Parity Error Interrupt Status	TxFIFO Overrun Interrupt Status	TCOCA Interrupt Status
R/W	R/W	R/W	R/W	R/W	RUR	RUR	RUR

3.2 Transmit Cell Processor

3.2.1 Brief Description of the Transmit Cell Processor

The Transmit Cell Processor reads in cells from the Transmit UTOPIA FIFO (TxFIFO) 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 capability to write an “out-bound” 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 provides a means to externally insert the value of the GFC (Generic Flow Control) field for each outbound cell. Figure 15 presents a simple illustration of the Transmit Cell Processor block and the associated external pins.

FIGURE 15. SIMPLE ILLUSTRATION OF THE TRANSMIT CELL PROCESSOR BLOCK AND THE ASSOCIATED EXTERNAL PINS

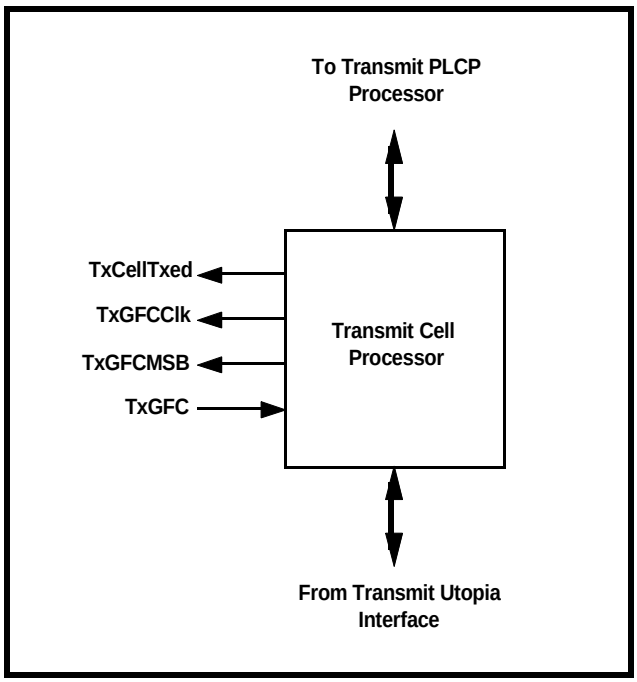


Figure 15 presents a functional block diagram of the Transmit Cell Processor.

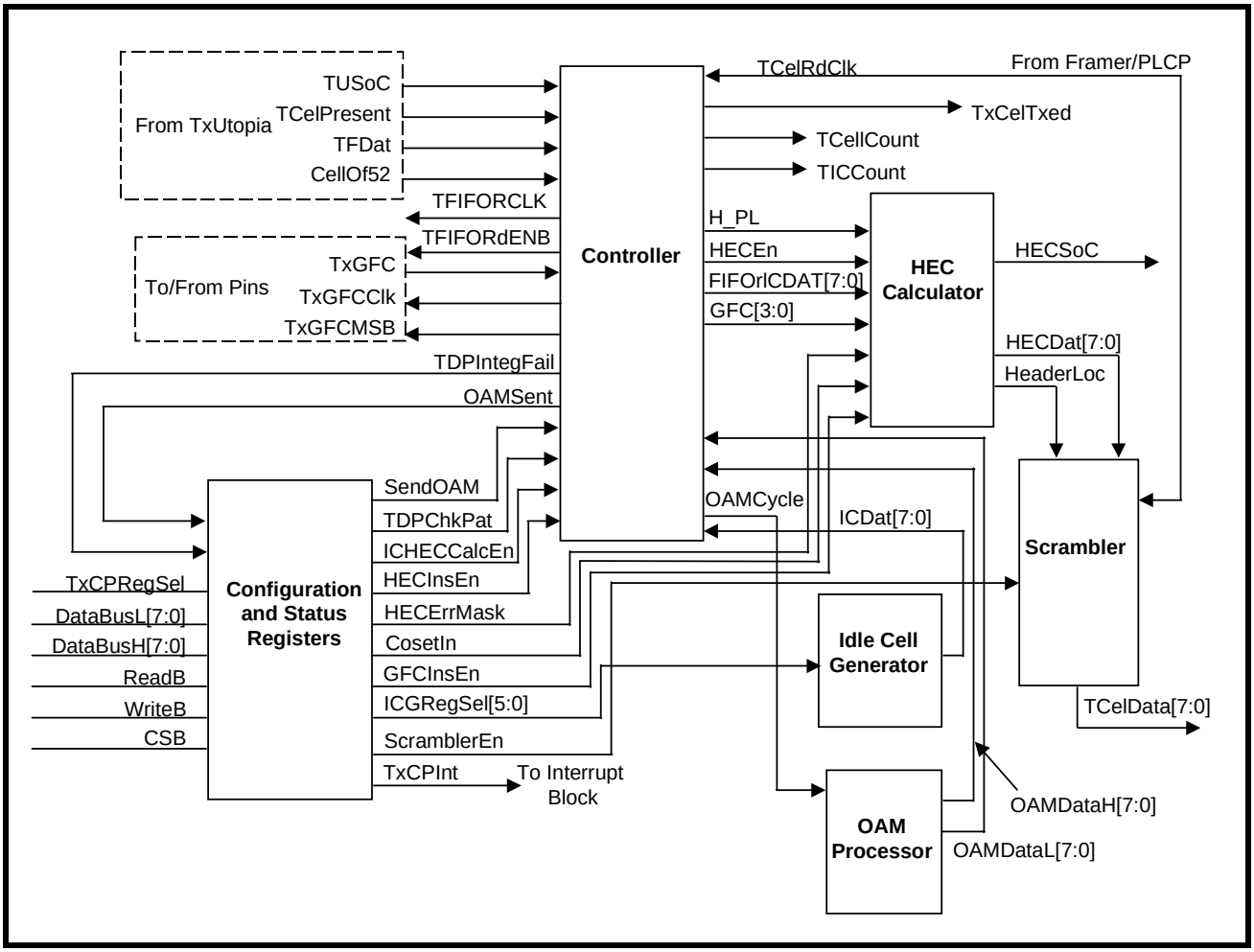
3.2.2 Functional Description of Transmit Cell Processor

The Transmit Cell Processor consists of the following functional blocks.

- Configuration and Status Register
- Controller
- HEC Byte Calculator
- OAM Cell Processor
- Cell Scrambler
- IDLE Cell Generator

• “Transmit GFC Nibble-field” serial input port

FIGURE 16. FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT CELL PROCESSOR BLOCK



Most of these functional blocks will be discussed in some detail below. The Transmit Cell Processor will read in ATM Cell Data from the TxFIFO. The first four bytes of each cell is loaded into the “HEC Byte calculator”. The fifth byte of each 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. Consequently, 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 scrambled payload), and is routed to the Transmit PLCP Processor or Transmit DS3 Framer.

When a complete cell is not available in the TxFIFO, a cell is created by the “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 provides a means 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 PLCP Processor (or Transmit DS3 Framer).

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 generator a "Data Path Integrity Check" error interrupt. After the Transmit Cell Processor has completed checking for the "Data Path Integrity Check" pattern; within a given cell, it will (optionally) overwrite this pattern by inserting the HEC byte.

The Transmit Cell Processor will inform external circuitry when a cell has been transmitted from the Transmit Cell Processor to either the Transmit PLCP Processor or the Transmit DS3 Framer, by pulsing the "TxCellTxed" output pin.

3.2.2.1 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 following are 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.

3.2.2.1.1 Configuring the HEC Byte Calculator for User and OAM Cells

The "HEC Byte Calculation and Insertion" feature can be enabled or disabled for user and OAM cells. This option is excercised by writing the appropriate value to Bit 5 of the TxCP Control Register, as depicted below.

TxCP Control Register (Address = 60h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Scrambler En	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 this feature is disable, 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 PLCP (or Transmit DS3 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 8: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT-FIELD 5 (HEC INSERT ENABLE) WITHIN THE TXCP CONTROL 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 PLCP Block (or Transmit DS3 Framer) as is

TABLE 8: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT-FIELD 5 (HEC INSERT ENABLE) WITHIN THE TXCP CONTROL REGISTER, AND THE HEC BYTE CALCULATOR'S HANDLING OF VALID CELLS

HEC INSERT ENABLE	RESULT
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. A "0" must be written to this bit in order to disable this operation.

3.2.2.1.2 Configuring the "HEC Byte Calculator and Insertion" Feature for Idle Cells

The "HEC Byte Calculation and Insertion" feature can be separately enabled or disabled for the outbound Idle Cells. This option is exercised by writing the appropriate value to bit 1 (Idle Cell HEC CalEn) within the TxCP Control Register, as depicted below.

TxCP Control Register (Address = 60h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Scrambler En	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 is used for enabling or disabling the "Calculation and Insertion" of the HEC byte into the Idle Cell as illustrated below. If disabling this feature is chosen, then the 5th octet of the Idle Cells will be transmitted to the Transmit PLCP (or Transmit DS3 Framer) block as programmed in the "TxCP Idle Cell Pattern Header—Byte 5" register (Address = 68h).

TABLE 9: THE RELATIONSHIP BETWEEN THE CONTENTS WITHIN BIT 1 (IC HEC CALC EN) OF THE "TXCP CONTROL REGISTER" AND THE RESULTING HANDLING OF IDLE CELLS, BY THE "HEC BYTE CALCULATOR"

IC HEC CALC EN	RESULT
0	The entire programmed Idle Cell header is transmitted without Modification
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 into the fifth octet position of each Idle Cell. A "0" must be written to this bit-field in order to disable this feature.

3.2.2.1.3 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 $x^8 + x^2 + x + 1$. 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 $x^6 + x^4 + x^2 + 1$; 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.

This modulo-2 addition can be enabled or disabled by writing the appropriate value to bit 6 (Coset Enable)

within the “TxCP Control” Register, as depicted below.

TxCP Control Register (Address = 60h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Scrambler En	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

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 such that the coset polynomial is modulo-2 added to the HEC byte prior to insertion into the cell. A “0” must be written to this bit to disable this operation.

3.2.2.1.4 Inserting Errors into the HEC Byte via Software Control

The XRT74L74 DS3/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 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). This option can be exercised by writing the appropriate data into the TxCP Error Mask register, which is located at address 62h within the UNI.

TxCP Error Mask Register; (Address = 62h)

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 result of this operation is written back into the fifth octet position of each of these cells. To prevent injecting errors into the HEC byte, the contents of this register must be set to 00h, the default value.

3.2.2.2 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 Cell Scrambler can be enabled or disabled by setting or clearing bit 7 (Scrambler Enable) within the “TxCP Control” Register, as depicted below.

TxCP Control Register (Address = 60h)

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, a "0" must be written to this bit in order to disable cell scrambling.

3.2.2.3 GFC Nibble-Field Serial Input Port

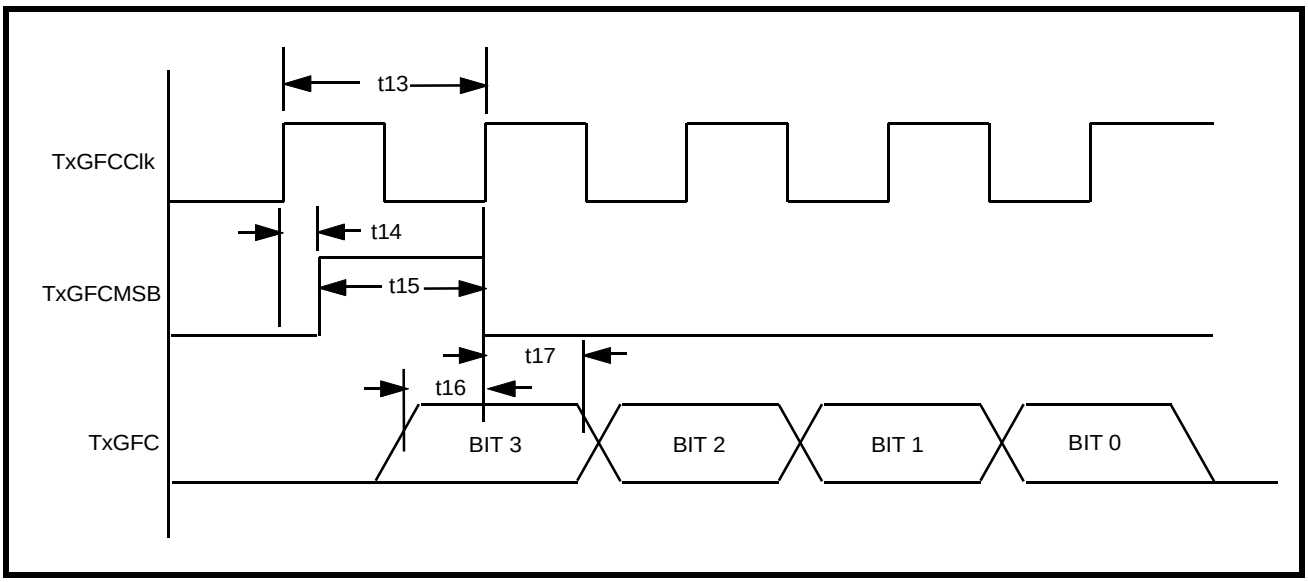
The first four bits in the first header byte of each cell are allocated for carrying "Generic Flow Control" (GFC) information. The user can externally insert their own values for the GFC nibble-field into each outbound cell, via a serial input port. This serial input port (the "Transmit GFC-Nibble-field" Serial Input port) will be activated by writing a "1" to bit 3 (GFC Insert Enable) of the "TxCP Control" Register, as depicted below.

TxCP Control Register (Address = 60h)

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 "Transmit GFC Nibble-field" Serial input port is activated, it will accept the 4 bit GFC value via the TxGFC pin during each 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 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 field. Figure 17 presents a timing diagram illustrating the role of each of these signals during GFC insertion.

FIGURE 17. BEHAVIOR OF TXGFC, TXGFCCLK, AND TXGFCMSB DURING GFC INSERTION INTO THE "OUTBOUND" CELL



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 in 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 OAM cell is to be transmitted,

the local µP must to write a “1” to bit 7 (SendOAM) within the TxCP OAM Register as depicted below.

TxCP OAM Register (Address = 61h)

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 TxCP 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 PLCP Processor (or Transmit DS3 Framer). Bit 7 of the TxCP OAM Register will be reset (to “0”) upon completion of the transmission of the OAM cell. This bit may also be polled in order to determine whether or not the OAM cell has been sent.

The number of valid cells (e.g., user and OAM cells) that have been generated and transmitted to the Transmit PLCP Processor or the Transmit DS3 Framercan be monitored . The Transmit Cell Processor increments the contents of the “PMON Transmitted Valid Cell Count (MSB and LSB)” Registers (Address = 3Ah, and 3Bh) 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 = 3Ah)

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 = 3Bh)

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

3.2.2.4 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 contents of these Idle Cells can be customized or the default values that are provided by the UNI chip can be used. The contents of these Idle Cells can be customized by programming six different registers:

- TxCP Idle Cell Pattern—Header Byte 1
- TxCP Idle Cell Pattern—Header Byte 2
- TxCP Idle Cell Pattern—Header Byte 3
- TxCP Idle Cell Pattern—Header Byte 4
- TxCP Idle Cell Pattern—Header Byte 5
- TxCP Transmit Cell Payload

Table 10 presents the Bit Format of each of these Registers and Table 11 presents the Address and Default values of these cells.

TABLE 10: BIT FORMAT OF THE TXCP IDLE CELL PATTERN -HEADER BYTES AND TXCP CELL PAYLOAD REGISTERS

REGISTER	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxCP Idle Cell Pattern—Header Byte 1	Transmit Idle Cell Pattern—Header Byte 1							
TxCP Idle Cell Pattern—Header Byte 2	Transmit Idle Cell Pattern—Header Byte 2							
TxCP Idle Cell Pattern—Header Byte 3	Transmit Idle Cell Pattern—Header Byte 3							
TxCP Idle Cell Pattern—Header Byte 4	Transmit Idle Cell Pattern—Header Byte 4							
TxCP Idle Cell Pattern—Header Byte 5	Transmit Idle Cell Pattern—Header Byte 5							
TxCP Idle Cell Payload	Transmit Idle Cell Payload							

TABLE 11: ADDRESS AND DEFAULT VALUES OF THE TXCP IDLE CELL PATTERN REGISTERS

ADDRESS	REGISTER	DEFAULT VALUE
64h	TxCP Idle Cell Pattern—Header Byte 1	00h
65h	TxCP Idle Cell Pattern—Header Byte 2	00h
66h	TxCP Idle Cell Pattern—Header Byte 3	00h
67h	TxCP Idle Cell Pattern—Header Byte 4	01h
68h	TxCP Idle Cell Pattern—Header Byte 5	52h
69h	TxCP Idle Cell Payload	5Ah

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 = 69h), 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 PLCP Processor (or the Transmit DS3 Framer). The Transmit Cell Processor increments the contents of the “PMON Transmitted Idle Cell Count (MSB and LSB)” Registers (Address = 38h and 39h) 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 = 38h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Idle Cell Count—High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

PMON Transmitted Idle Cell Count—LSB (Address = 39h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Idle Cell Count—Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

3.2.2.5 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 XRT74L74 DS3/E3 UNI. The Transmit Cell Processor will read in this cell from the TxFIFO, and will, prior to inserting a new HEC byte into the cell, read in the fifth octet from the TxFIFO and check it for a specific pattern or value. The Transmit Cell Processor can be configured to check for either a constant "55h" pattern or an alternating pattern of "55h" and "AAh" for each cell. The Transmit Cell Processor can also be configured to generate an interrupt if a Data Path Integrity Test fails. This can all be can accomplished by writing the appropriate data to the "TxCP Control" Register (Address = 60h). 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 to 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.*

TxCP Control Register (Address = 60h)

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 plays 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 TxFIFO. This "Read/Write" bit allows for specifying 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 12: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 4 (TDPCHK PAT) WITHIN THE TXCP CONTROL 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 PAT	"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 TxFIFO.
1	Transmit Cell Processor expects a constant "55h" pattern for the value of the fifth octet of the cells received from the TxFIFO.

The remaining shaded bits are “Interrupt service” related and will be discussed in the following section.

3.2.2.6 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 “xxxxx1xxb” (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
Rx DS3 Interrupt Status	Rx PLCP Interrupt Status	Rx CP Interrupt Status	Rx UTOPIA Interrupt Status	TxUTOPIA Interrupt Status	TxCP Interrupt Status	TxDS3 Interrupt Status	One Sec Interrupt Status
RO	RO	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 “TxCP Control” Register (Address = 60h) in order to verify and service this condition. The bit format of this register is presented below.

Transmit Cell Processor Control Register (Address = 60h)

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 is used 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 “TxCP Control” 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 “TxCP Control” 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 (TxCP Interrupt Status) within the “UNI Interrupt Status” register will also be reset to “0”.

3.3 Transmit PLCP Processor

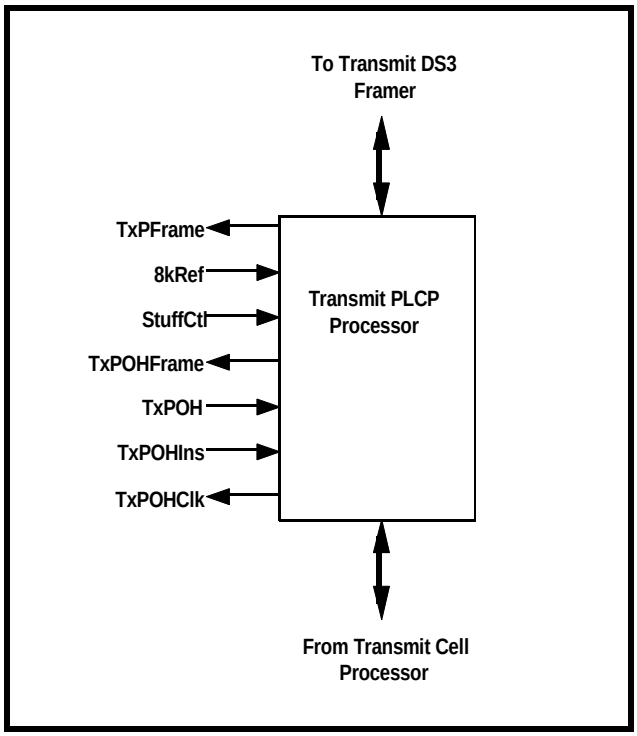
3.3.1 Brief Description of the Transmit PLCP Processor

The Transmit PLCP Processor takes the incoming cells (assigned, Idle, or OAM) from the Transmit Cell Processor and packs them into PLCP frames. Each of these PLCP frames also includes various overhead

bytes that contain information on: Path Overhead Identification, Bit Interleaved Parity Calculation results, Far-End Block Error status, and stuffing status. The generation of PLCP frames can either be synchronized to an external 8 kHz reference clock or to timing from the Receive PLCP Processor. PLCP frame generation can also be asynchronous with respect to any timing signals. The Transmit PLCP Processor can compute its "nibble-stuffing" requirements based upon its configured synchronous timing source (e.g., the external 8 kHz reference clock or Receive PLCP Timing), arbitrarily controlled via an external pin or by following a fixed stuffing pattern. Once a PLCP frame is formed, it is routed to the Transmit DS3 Framer Block of the UNI for transmission to the "Far End" Terminal. Figure 18 presents a simple illustration of the Transmit PLCP Processor and the associated external pins.

Note: The user has the option of taking advantage of the full DS3 payload bandwidth by by-passing the PLCP Processor altogether. This option will be referred to as "Direct Mapping" and is discussed in Section 6.3.3.9

FIGURE 18. SIMPLE ILLUSTRATION OF THE TRANSMIT PLCP PROCESSOR BLOCK



3.3.2 Description of the PLCP Frame and the Path Overhead (POH) Bytes

The Transmit PLCP Processor receives ATM cells from the Transmit Cell Processor. It then multiplexes these cells with some overhead (OH) bytes and frames this composite information into PLCP Frames. Table 13 presents the byte format of a PLCP Frame.

TABLE 13: FRAME FORMAT OF THE PLCP FRAME

PLCP FRAME 2 BYTES		POI 1 BYTE	POH 1 BYTE	PLCP PAYLOAD 53 BYTES	13–14 NIBBLES
A1	A2	P11	Z6	First ATM Cell	
A1	A2	P10	Z5	ATM Cell	
A1	A2	P9	Z4	ATM Cell	
A1	A2	P8	Z3	ATM Cell	
A1	A2	P7	Z2	ATM Cell	
A1	A2	P6	Z1	ATM Cell	
A1	A2	P5	X	ATM Cell	
A1	A2	P4	B1	ATM Cell	
A1	A2	P3	G1	ATM Cell	
A1	A2	P2	X	ATM Cell	
A1	A2	P1	X	ATM Cell	

TABLE 13: FRAME FORMAT OF THE PLCP FRAME

PLCP FRAME 2 BYTES		POI 1 BYTE	POH 1 BYTE	PLCP PAYLOAD 53 BYTES	13–14 NIBBLES
A1	A2	P0	C1	Twelfth ATM Cell	Trailer

Each PLCP frame consists of 12 ATM Cells, 24 bytes of Frame Alignment patterns (the A1 and A2 bytes), 12 bytes of POI (Path Overhead Identifiers), 12 bytes of POH (Path Overhead) and a 13 or 14 nibble trailer which is appended to the PLCP Frame for frequency justification. Once a PLCP Frame is formed it is routed to the Transmit DS3 Framer block of the UNI. The order of transmission of the PLCP frame begins from the upper left hand corner of the frame (A1 byte), and proceeds through the frame in a manner similar to reading this page of text, to the lower right hand corner (the 13 or 14 nibble trailer).

The definition of each of the overhead bytes within the PLCP Frame are presented below.

A1, A2 Frame Alignment Pattern Bytes

Each row within a PLCP frame will begin with two bytes of Frame Alignment patterns which are denoted as A1 and A2 in Table 13 . In accordance with the ATM Forum UNI spec, the Transmit PLCP Processor will assign the values: A1 = F6h and A2 = 28h.

POI (Path Overhead Identifier) Bytes: P0-P11

The Path Overhead Identifier (POI) bytes are used to index the adjacent Path Overhead (POH) bytes, as tabulated below in Table 14 .

TABLE 14: POI CODE AND ASSOCIATED POH BYTES

POI	POI CODE	ASSOCIATED POH BYTE
P11	2Ch	Z6
P10	29h	Z5
P9	25h	Z4
P8	20h	Z3
P7	1Ch	Z2
P6	19h	Z1
P5	15h	F1 (Frame)
P4	10h	B1 (BIP-8)
P3	0Dh	G1 (FEBE)
P2	08h	M1
P1	04h	M2
P0	01h	C1 (Stuff Indicator)

The Path Overhead bytes (POH) are defined below.

- **Z1–Z6 Bytes: Growth Octets**

The Z1–Z6 octets presently have no particular application, and are reserved for future use. The Transmit PLCP Processor will set these octets to 00h. The far-end Receive PLCP Processor will ignore the values contained in these fields.

- **F1: User Octet**

This byte is unused in the UNI and is consequently programmed to 00h. Therefore, the Far-End Receive PLCP Processor will ignore the values contained in the byte-field.

Note: This octet is used in the IEEE 802.6 MAN and in SMDS applications as a 64 kbps data link channel for proprietary use by the network provider.

- **B1–Bit Interleaved Parity–8**

The B1 byte contains the result of BIP-8 (Bit Interleaved Parity) calculations. The Bit Interleaved Parity (BIP-8) byte field supports path error monitoring. The Transmit PLCP Processor will compute the BIP-8 over a 12 x 54 octet structure, within each PLCP frame. Specifically, these calculations involve the path overhead (POH) byte fields and the associated ATM cells for a total of 648 octets. The resulting BIP-8 value is inserted into the B1 byte field within the very next

PLCP 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 octet involved in the calculation. Thus, the BIP-8 value presents the results for 8 separate even-bit parity calculations.

- **G1—PLCP Path Status**

This byte-field contains some diagnostic information which was compiled by the “Near-End” Receive PLCP Processor of this UNI device (See Section 7.2.2.2.2). The purpose of this diagnostic byte field is to inform the Far-End Terminal of whether or not the (Near End) Receive PLCP Processor of this UNI has detected errors or has had problems framing to its (the Far-End Transmit PLCP Processor’s) transmission. Table 15 presents the bit-format of the G1 octet which consists of a 4 bit Far-End Block Error (FEBE) subfield, a 1 bit RAI (Yellow) alarm and 3 X-bits (the X bits are ignored).

TABLE 15: BIT FORMAT OF G1 OCTET

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Far End Block Error (FEBE)				RAI (Yellow)	X bits (Ignored by the Receiver)		
4 Bits				1 Bit	3 Bits		

- **C1—Stuffing Status/Nibble-Trailer Length Indicator Byte**

Table 13 indicates that the PLCP frame will contain a nibble trailer of either 13 or 14 nibbles, appended to the end of each PLCP frame. This option of using either 13 or 14 nibbles presents the Transmit PLCP processor with a stuff opportunity. This octet (C1) conveys the nibble stuffing status and is also the nibble length indicator for the current PLCP frame. For more information on the C1 octet, please see Section 6.3.3.1.

3.3.3 Functional Description of the Transmit PLCP Processor Block

Figure 19 presents a functional block diagram of the Transmit PLCP Processor.

FIGURE 19. FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT PLCP PROCESSOR

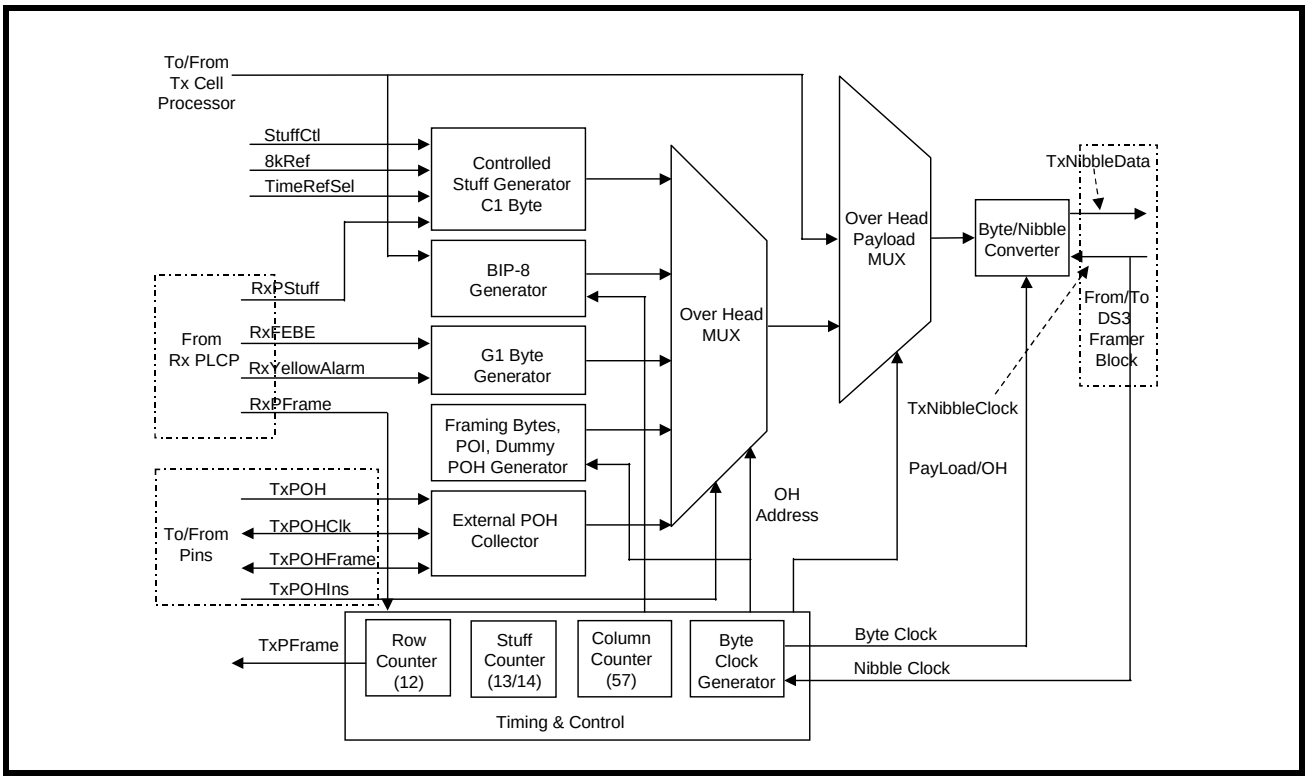


Figure 19 indicates that the Transmit PLCP Processor consists of the following functional blocks.

- Controlled Stuff Generator, C1 Byte
- BIP-8 Generator
- G1 Byte Generator
- Framing Byte, POI, Dummy POI Generator
- External POH Collector
- Transmit PLCP Framer
- Overhead MUX
- Overhead Payload MUX
- Byte/Nibble Converter

The role of some of these functional blocks will be discussed below.

3.3.3.1 Transmit PLCP Frame Timing, Stuff Control—C1 Byte

The Controlled Stuff Generator portion of the Transmit PLCP Processor is responsible for three things.

1. Determining the nibble-stuffing requirements for the current PLCP frame.
2. Fulfilling these nibble-stuffing requirements.
3. Reflecting the nibble-stuffing status in the C1 byte.

Table 16 indicates the Transmit PLCP Processor will append either a 13 or 14 nibble trailer at the end of each PLCP frame, in order to frequency justify the framing to 8 kHz. This choice between 13 or 14 nibbles presents the Transmit PLCP Processor with a “stuff” opportunity.

The Transmit PLCP Processor can be configured into one of four frame-timing/stuff-control options. These options are selected by writing the appropriate data to bit 1 and bit 0 (TimRefSel[1, 0], within the UNI Operating Mode Register. The bit format of this register is presented below.

UNI Operating Mode Register: Address = 00h

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	Cell Loop-back	PLCP Loop-back	Reset	Direct Mapped ATM	C-Bit/M13	TimRefSel[1, 0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The values for TimRefSel[1, 0] and the corresponding options are presented in Table 16 .

TABLE 16: PLCP FRAME TIMING AND STUFF CONTROL OPTIONS

BIT 1	BIT 0	RESULT
0	0	**TimRefSel[1,0] = 00** **PLCP Frame Timing Source:** Receive PLCP Processor Timing. In this configuration, the Transmit PLCP Processor takes its timing from the Receiver Start of Frame signal (from the Receive PLCP Processor, within the UNI) to start a PLCP frame. The Transmit PLCP Processor will also use this signal to calculate stuff opportunities. **Stuff Control:** The Transmit PLCP Framer has a stuff-opportunity that occurs once every three PLCP frames. Therefore, the stuff-control algorithm is based on a repeating "Stuff-Control" cycle that consists of these three (3) PLCP cycles (or a 375µs interval). The three composite PLCP frames of a stuff control cycle, when TimRefSel[1, 0] = 00, is presented below. **PLCP Frame 1 (**the first of the 3 frames) will contain 13 trailer nibbles. The C1 byte, within this PLCP frame, will contain the value FFh. This value identifies the current PLCP Frame as Frame #1 in this 3 Frame Cycle, and informs the Far-End Receive PLCP Processor that the trailer length is 13. **PLCP Frame 2** (the second of the 3 frames) will contain 14 trailer nibbles. The C1 byte, within this PLCP frame, will contain the value 00h. This value identifies the current PLCP Frame as Frame #2 in this 3 Frame Cycle, and informs the Far-End Receive PLCP Processor that the trailer nibble length is 14. **PLCP Frame 3** (the last of the 3 frames) will contain either 13 or 14 trailer nibbles, depending upon the calculated stuffing requirements. Hence, the Transmit PLCP can generate two versions of Frame #3, "No Stuff" Frame #3 and "Stuff" Frame #3. **"No Stuff" Frame #3:** If the Transmit PLCP Processor has determined that no stuff is required then it will append only 13 trailer nibbles at the end of the current PLCP frame. The C1 byte, within this PLCP frame, will identify the current frame as a "No Stuff" Frame #3, in the 3 Frame cycle, by carrying the value 66h. **"Stuff" Frame #3:** If the Transmit PLCP Processor has determined that a stuff is required, then it will append 14 trailer nibbles at the end of the current PLCP frame. The C1 byte, within this PLCP frame, will identify the current frame as a "Stuff" Frame #3, in the 3 Frame cycle, by carrying the value 99h. Once the Stuff Control algorithm has processed through PLCP Frame #3, the Transmit PLCP Processor will proceed to generate a PLCP Frame #1, and repeat this 3 frame cycle.
0	1	**TimRefSel[1,0] = 01** **PLCP Frame Timing Source:** External 8 kHz Clock Signal In this configuration, the Transmit PLCP Processor takes its timing from an 8 kHz signal which is applied at the 8KRef input pin. The Transmit PLCP Processor will also use this signal to calculate stuff opportunities. **Stuff Control:** As mentioned earlier, a stuff opportunity for the Transmit PLCP Processor occurs once in a period of three (3) PLCP Frames. These composite PLCP frames and the resulting C1 values are the same as presented in the above "PLCP Frame Timing/Stuff Control" Option (TimRefSel = 00).
1	0	**TimRefSel[1,0] = 10; - StuffCtl Input Pin** **PLCP Frame Timing Source:** PLCP Frame timing is asynchronous upon power up or reset. In this configuration, the Transmit PLCP Processor will start PLCP frames based upon an asynchronous timing signal. The stuffing opportunities are not computed based on this timing, but on the logic state of an input pin.

TABLE 16: PLCP FRAME TIMING AND STUFF CONTROL OPTIONS (CONTINUED)

BIT 1	BIT 0	RESULT
		Stuff Control: The Stuff Control algorithm is controlled by the logic state of the external pin, StuffCtl. As with the previous two Stuff Control options, the Transmit PLCP Framer has a stuff-opportunity that occurs once every three PLCP frames. Each of these composite PLCP frames are discussed below. • **PLCP Frame 1** (the first of the 3 frames) will contain 13 trailer nibbles. The C1 byte, within this PLCP frame, will identify the current frame as a Frame #1, by carrying the value FFh. Note this frame will be created independent of the state of the StuffCtl pin. • **PLCP Frame 2** (the second of the 3 frames) will contain 14 trailer nibbles. The C1 byte, within this PLCP frame, will identify the current frame as a Frame #2, by carrying the value 00h. Note this frame will be created independent of the state of the StuffCtl pin. • **PLCP Frame 3** (the last of the 3 frames) will contain either 13 or 14 trailer nibbles, depending upon the logic state of the "StuffCtl" input pin. Therefore, the Transmit PLCP can generate one of two versions of Frame #3: "No Stuff" Frame #3 and "Stuff" Frame #3. **StuffCtl = "0"—No Stuff" Frame #3:** If the StuffCtl pin is "low" then the Transmit PLCP processor will generate a "No Stuff" Frame #3. This PLCP frame will contain 13 trailer nibbles. The C1 byte will identify the current PLCP frame as a "No Stuff" Frame #3 by carrying the value 66h. **StuffCtl = "1"—"Stuff" Frame #3:** If the StuffCtl pin is "high" then the Transmit PLCP Processor will generate a "Stuff" Frame #3. This PLCP frame will contain 14 trailer nibbles. The C1 byte will identify the current PLCP frame as a "Stuff" Frame #3 by carrying the value 99h.
1	1	**TimRefSel[1,0] = 11; - Fixed Stuffing Pattern** PLCP Frame Timing: Asynchronous upon power on. **Stuff Control:** The Transmit PLCP Processor will use a fixed Stuffing Pattern which is controlled by an internal counter. This stuffing pattern results in the transmission of 13, 14, 13, 13, 14, 14, 13, 14, 14 trailer nibbles in every 9 PLCP frames repeatedly. This corresponds to 8000 - 1.5 x 10^{-5} Hz when a perfect 44.736 MHz is used as the transmit clock. Table 17 lists the contents of the C1 bytes for each of these 9 PLCP Frames.

Note: The selection of these bits also affects the operation of the Transmit DS3 Framer. This subject is presented in Section 6.4.3.4. In all cases, the C1 byte of each PLCP frame will reflect the stuffing phase and number of trailer nibbles that are appended to the current PLCP frame.

TABLE 17: VALUE OF C1 FOR THE 9 PLCP FRAMES, WHEN THE FIXED STUFFING OPTION IS SELECTED

PLCP FRAME NUMBER	NUMBER OF TRAILER NIBBLES IN FRAME	C1 BYTE VALUE
1	13	FFh
2	14	00h
3	13	66h
4	13	FFh
5	14	00h
6	14	99h
7	13	FFh
8	14	00h
9	14	99h

3.3.3.2 BIP-8 Generator—B1 Byte

The BIP-8 (Bit Interleaved Parity) generator takes a total of 12 x 54 octets per PLCP frame, (which consists of the POH byte fields and the associated ATM cells—a total of 648 octets) and performs a very specific sequence of calculations. The BIP-8 generator takes bit 7 (the MSB) of each of the 648 octets and calculates an even parity bit (based upon these 648 MSB bits). The resulting parity bit is inserted into bit 7 of the B1 byte. This same calculation is also performed for each of the remaining 7 bits in each octet. The resulting parity bits are grouped together and inserted into the B1 byte field. Therefore, the content of the B1 byte is the result of 8 separate parity bit calculations. The BIP-8 Calculation results that are obtained based upon the data within a given PLCP frame, will be inserted into the B1 octet position of the very next PLCP frame.

The B1 byte will ultimately be used by the “Far-End” Receive PLCP Processor, in order to monitor the transmission performance between the “Near-End” Transmitter and the “Far-End” Receiver. For more information on how the Receive PLCP Processor handles the B1 byte, please see Section 7.2.2.3.1.

3.3.3.3 G1 Byte Generator

The purpose of the G1 byte is to provide the “Far-End” Transmitter with diagnostic information on how well the “Near-End” Receive PLCP (e.g., the on-chip Receive PLCP) Processor is receiving and processing its PLCP frames. The bit field of the G1 byte is presented below.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Far End Block Error (FEBE)				RAI (Yellow)	X Bits (Ignored by the Receiver)		
4 Bits				1 Bit	3 Bits		

Each of these bit-fields are discussed below.

Far-End Block Error (FEBE)

The Receive PLCP Processor will receive and extract the PLCP Overhead bytes from incoming PLCP frames, originating from a “Far-End” Transmit PLCP Processor. While the Receive PLCP Processor is receiving a PLCP frame, it will calculate its own BIP-8 value for that frame. Afterwards, the Receive PLCP Processor will then compare its BIP-8 value with the contents of the B1 byte that it extracts from the very next PLCP frame. If these two BIP-8 values match, then the Receive PLCP Processor will reflect this fact by writing a FEBE value of 0h into a G1 byte. At some phase during PLCP frame processing, the Receive PLCP Processor will route the contents of the G1 byte to the Transmit PLCP Processor (on the same chip). This G1 byte will be packed in the next outbound PLCP frame, which is in turn routed to the Transmit DS3 Framer. The G1 byte is ultimately transmitted to the “Far-End” Receive PLCP Processor over the DS3 transport medium, where it will be processed and evaluated.

If the Receive PLCP Processor determines that the two BIP-8 values do not match, then the Receive PLCP Processor will count the number of bit-errors (e.g., the number of bit-by-bit discrepancies between these two BIP-8 values) and write this value into the FEBE nibble of the G1 byte. This G1 Byte will be routed to the Transmit PLCP Processor, inserted into the next outbound PLCP frame, and received and processed by the Far-End Receive PLCP Processor, as described above.

Note:

1. *Since the BIP-8 value only contains 8-bits, the largest number of errors that the Receive PLCP processor can detect is “8”. Therefore, the “FEBE” nibble-field, within the G1 byte must not contain a value exceeding the number “8”.*
2. *For more information on how the Receive PLCP Processor handles the G1 byte, from the Far-End Transmit PLCP Processor, please see Section 7.2.2.2.2.*

RAI (Yellow Alarm)

If the Receive PLCP Processor has had sufficient trouble framing to the incoming PLCP frames, (e.g., if the Receive PLCP remains “Un-framed” for 2 to 10 seconds), then the Receive PLCP Processor will assert the RAI bit in the G1 byte. The contents of the G1 byte will be routed to the Transmit PLCP Processor and subjected to the processing that was described above.

3.3.3.4 Inserting Errors into the PLCP Path Overhead Bytes

The XRT74L74 DS3/E3 UNI has provisions to allow the insertion of errors into the POH bytes of each outbound PLCP frames. This may desireable to do for chip/equipment test purposes.

The following sections briefly discuss these options.

3.3.3.4.1 Inserting Errors into the B1 Byte

There are occasions when it is desireable to inject errors into the B1 byte of the PLCP frame in order to verify that the Far-End Receiving hardware is functioning properly and will detect these errors and respond accordingly. The UNI allows the injection these errors into the B1 byte via the TxPLCP BIP-8 Error Mask Register, as depicted below.

TxPLCP BIP-8 Error Mask Register, Address = 4Ah

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
B1 Error Mask							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The B1 (BIP-8) byte of a PLCP frame is always XORed with this mask byte. The results of this operation are written back into the B1-byte position, prior to transmission. An error can be inserted into a particular bit of a B1 byte, by writing a "1" into the corresponding bit in this register.

Note: This register must be 00h for normal operation. This register is of value 00h following power up or reset.

3.3.3.4.2 Inserting Errors into the A1, A2 Bytes

The UNI allows the for the insertion of errors into each of the "Frame Alignment" bytes A1 and A2. These errors can be inserted by writing the appropriate data to the "TxPLCP A1 Byte Error Mask Register (Address = 48h); and the "TxPLCP A2 Byte Error Mask Register (Address = 49h). The bit formats of these two registers follows.

TxPLCP A1 Byte Error Mask Register (Address = 48h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
A1 Error Mask							
0	0	0	0	0	0	0	0

TxPLCP A2 Byte Error Mask Register (Address = 49h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
A2 Error Mask							
0	0	0	0	0	0	0	0

The UNI IC automatically takes each A1 byte from within an outbound PLCP frame, and performs an XOR operation with the contents of the "TxPLCP A1 Byte Error Mask" Register. The results of this operation are written back into the A1 Byte fields of the PLCP frame, prior to transmission.

The UNI IC also performs the same set of operations on the A2 bytes of the PLCP frame, with the "TxPLCP A2 Byte Error Mask" register.

To insure errors are not inserted in the A1 and A2 byte fields of each outbound PLCP frame, these two registers must contain the value 00h (the default value).

3.3.3.5 Manipulating the FEBE-Nibble Field within the G1 Bytes

The UNI can either transmit G1 bytes with a FEBE value of '0h', or to transmit a G1 byte with the correct FEBE count, as determined by the "Near-End" Receive PLCP Processor.

This option can be exercised by writing the appropriate data to bit 4 of the TxPLCP G1 Byte Register (Address = 4Bh). The bit-format of this register is presented below.

TxPLCP G1 Byte Register (Address = 4Bh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxPLCP FEBE Mask	Yellow Alarm	LSS(2)	LSS(1)	LSS(0)
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a '1' to this bit-field will cause the Transmit PLCP Processor to transmit G1 bytes with the FEBE nibble value of '0h' (independent of the number of BIP-8 errors detected by the Receive PLCP Processor). Writing a '0' to this bit-field will cause the Transmit PLCP Processor to transmit G1 bytes with the correct FEBE count, as determined by the "Near-End" Receive PLCP Processor.

3.3.3.6 Forcing a Yellow Alarm—Via Software Control

The UNI allows for the generation a "Yellow Alarm (PLCP Version thereof)" via software control. In this case, the Transmit PLCP Processor will generate a "Yellow Alarm" by automatically setting the "RAI" bit within each G1 byte to '1'. This option can be exercised by writing the appropriate bit to bit-field 3 of the TxPLCP G1 Byte Register (Address = 4Bh). The bit format of this register follows.

TxPLCP G1 Byte Register (Address = 4Bh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxPLCP FEBE Mask	Yellow Alarm	LSS(2)	LSS(1)	LSS(0)
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Writing a '1' to this bit-field forces the "PLCP—Yellow Alarm" condition. Writing a '0' to this bit-field allows the state of the RAI bit to be based upon the framing conditions of the "Near-End" Receive PLCP Processor.

3.3.3.7 Transmitting Data Link Messages via the G1 Byte

The "TxPLCP G1 Byte" Register contains three bit-fields that can be used to support a 24 kbps data link between the Near-End Transmit PLCP Processor, and the Far-End Receive PLCP Processor, as depicted below.

TxPLCP G1 Byte Register (Address = 4Bh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxPLCP FEBE Mask	Yellow Alarm	LSS(2)	LSS(1)	LSS(0)
RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Whatever data is written into the three bit-fields will appear in Bits 2–0 of the incoming G1 byte at the Far-End Receive PLCP Processor.

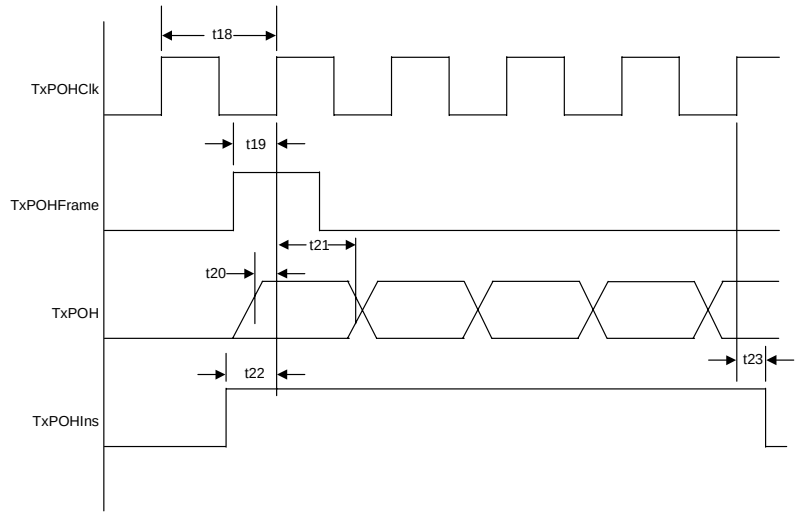
3.3.3.8 Inserting POH Bytes via the TxPOH Serial Input Port

The UNI allows the users to externally insert their own PLCP POH (Path Overhead) bytes via a serial input interface consisting of the pins: TxPOHIns, TxPOH, TxPOHFrame, and TxPOHClk. This serial input port can be activated by asserting the TxPOHIns input pin (e.g., setting it "high"). When this pin is "low", the UNI will internally generate the POH bytes. However, when this pin is "high", the users will be expected to provide their own value for the POH bytes via the TxPOH input pin. The UNI will assert (toggle "high") the TxPOHFrame output pin when it expects the MSB of the Z6 byte. The users will be expected to provide their value for the Z6 byte, with the MSB first, in descending order. Immediately after the LSB of the Z6 byte, the TxPOH Serial Input port will be expecting the MSB of the Z5 byte, and so on. The byte order that this serial input port expects is as presented in Table 16 . Once the TxPOH serial input port has read in the LSB of the C1 byte, it will repeat this sequence of bytes, beginning with the Z6 byte first. The POH data will be serially latched into the TxPOH input on the rising edge of the TxPOHClk output signal. The

clock rate of the TxPOHClk signal is nominally 768 kHz.

Figure 20 presents a timing diagram depicting the behavior of the signals associated with the TxPOH serial input interface during its use.

FIGURE 20. AN ILLUSTRATION OF THE BEHAVIOR OF THE TXPOH SERIAL INTERFACE SIGNALS DURING USER INPUT OF POH DATA.



The TxPOH Serial Input Port also allows the users to externally insert their POH bytes selectively (e.g., some POH bytes are internally generated, others are externally inserted). This can be accomplished by asserting the TxPOHIns and inserting data into the TxPOH input at a time when the TxPOH input is expecting this data, per the byte/bit order described above. If the remainder of the data is to be "internally" generated, the TxPOHIns pin must be negated during the time-slot periods for those POH bytes.

3.3.3.9 The "Direct Mapped ATM" Option

The UNI allows for the disabling (or by-passing) the Transmit PLCP processor and to directly insert the ATM cells, from the Transmit Cell Processor into the DS3 payload. This option can be exercised by writing to Bit 3 of 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
Local Loopback	Cell Loopback	PLCP Loopback	Reset	Direct Mapped ATM	C-Bit/M13	TimRefSel[1, 0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The following table presents the relationship between the value of this bit and the type of ATM Mapping incorporated.

TABLE 18: THE RELATIONSHIP BETWEEN BIT 3 OF THE UNI OPERATING MODE REGISTER AND THE RESULTING "ATM CELL" MAPPING MODE.

BIT 3	MAPPING MODE
0	**PLCP Mode:** The PLCP is enabled. PLCP Frames will be mapped into the "outbound" DS3 Frame

TABLE 18: THE RELATIONSHIP BETWEEN BIT 3 OF THE UNI OPERATING MODE REGISTER AND THE RESULTING "ATM CELL" MAPPING MODE.

BIT 3	MAPPING MODE
1	**Direct-Mapped ATM Mode:** The PLCP Processor block is bypassed. ATM cells will be directly mapped into the "outbound" DS3 Frame

Final Notes about the Transmit PLCP Processor

The Transmit PLCP Processor will be disabled, upon power up or reset. Therefore, a "1" must be written to this bit in order to enable the PLCP Processor. Selection of this bit affects both the Transmit PLCP Processor and the Receive PLCP Processor.

The advantage of selecting the "Direct-Mapped ATM" option is to result in a more efficient use of the DS3 Bandwidth. This is because in the Direct Mapped ATM mode, it is not required to include all of the POH bytes that must be included in PLCP frames.

The Transmit PLCP Processor will inform the external circuitry that a PLCP frame has been assembled and transmitted out of the PLCP Processor by pulsing the TxPFrame output pin 'high' during the transmission of the last trailer nibble.

3.4 Transmit DS3 Framer

3.4.1 Brief Description of the Transmit DS3 Framer

The Transmit DS3 Framer takes the incoming data, which can be either PLCP frames from the Transmit PLCP Processor or ATM Cells from the Transmit Cell Processor and maps it into the payload portion of the DS3 frame. The Transmit DS3 Framer supports either the M13 or C-Bit Parity frame formats. The Transmit DS3 Framer operates at 44.736 MHz and framing is derived from an input clock signal. The framing overhead bits are generated and inserted with the DS3 payload bits to make up the complete DS3 frame. The DS3 frame is then encoded into either the Unipolar, AMI or B3ZS line codes. When the Transmit DS3 Framer is operating in the C-Bit Parity Framing format, it provides an interface that supports the transmission of path maintenance data link messages on the outgoing DS3 frames via the on-chip LAPD Transmitter. The Transmit DS3 Framer also includes an on-chip Transmit FEAC Processor that supports the transmission of FEAC (Far End Alarm and Control) messages over the outgoing DS3 frame. Different transmission conditions like AIS (Alarm Indication Signal), Idle Condition and the Yellow Alarm can be generated upon software command. Further, the LOS (Loss of Signal) condition can be simulated upon software command.

3.5 TRANSMIT E3 FRAMER

3.5.1 Brief Description of the Tansmit E3 Framer

4.0 THE RECEIVE SECTION

The purpose of the Receiver Section of the XRT74L74 DS3/E3 ATM UNI 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 DS3 transport medium.

The Receive Section of the DS3 UNI chip consists of the following functional blocks:

- Receive DS3 Framer
- Receive PLCP Processor
- Receive Cell Processor
- Receive UTOPIA Interface

The Receive DS3 Framer will synchronize itself to this incoming DS3 Data Stream (containing ATM cells) via the RxPOS, RxNEG, and RxLineClk input pins, and proceed to “strip off” and process the OH bits of the DS3 frame. Once all of the OH bits have been removed, the payload portion of the received DS3 Frame should consist of either PLCP frames or ATM cells (if the Direct-Mapped ATM option was selected). The PLCP frames are routed to the Receive PLCP Processor and the “Direct-Mapped” ATM Cells are sent onto the Receive Cell Processor.

The Receive PLCP Processor will take the PLCP frame data and search for the A1/A2 Frame Alignment pattern bytes, in order to determine the PLCP frame boundaries. Once PLCP framing is established, the Receive PLCP Processor will proceed to check and process the OH bytes, within the PLCP frame. The PLCP Frames, along with framing information are sent on to the Receive Cell Processor.

The Receive Cell Processor takes delineated PLCP frames from the Receive PLCP Processor, and performs the following operations:

- Performs 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 computed HEC value with that of the fifth octet, within the cell. If the two HEC values are equal, the cell is then retained for further processing. If the two HEC values are not equal, then the cells with single-bit errors are corrected. However, the cell is optionally discarded if multile-bit errors are detected.

- Idle Cell Filtering

The Receive Cell Processor will detect and remove Idle Cells and can be configured to filter User and OAM cells.

- 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 block.

The following sections discuss the blocks comprising the Receiver portion of the DS3 UNI in detail.

4.1 Receive DS3 Framer

4.1.1 Brief Description of the Receive DS3 Framer

The Receive DS3 Framer synchronizes itself to the incoming DS3 data-stream. It decodes and frames the incoming data into DS3 frames. It supports both the M13 and C-bit Parity framing formats. It detects Line Code Violations (LCV), the Loss of Signal (LOS) condition, the Alarm Indication Signal (AIS) and Idle patterns, Out of Frame (OOF) and Loss of Frame (LOF) conditions. The Receive DS3 Framer computes parity over a given DS3 M-frame and compares it with the P-bits that it receives in the very next DS3 M‘-hframe. It extracts and processes the DS3 frame overhead bits and provides them to a serial output port. It “validates” FEAC messages received from the “Far-End” Transmit DS3 Framer. Additionally, the Receive DS3 Framer will receive “LAPD Messages” from the “Far End” Transmit DS3 Framer; and will write this message into the “Receive LAPD Message” buffer.

The Receive DS3 Framer will detect and generate interrupts upon error conditions. The status of the Receive DS3 Framer can be read by registers through the UNI-Microprocessor interface. If the UNI is operating in the “Direct-Mapped” ATM Mode, then the Receive DS3 Framer will route the contents of the DS3 payload to the Receive Cell Processor. Otherwise, if the UNI is operating in the PLCP mode, then the Receive DS3 framer will route the payload to the Receive PLCP Processor.

Figure 21 presents a simple block diagram of the Receiver DS3 Framer along with the associated pins. Additionally, Figure 22 presents a more in-depth functional block diagram of the Receive DS3 Framer.

FIGURE 21. BLOCK DIAGRAM OF THE RECEIVER DS3 FRAMER, WITH ASSOCIATED PINS.

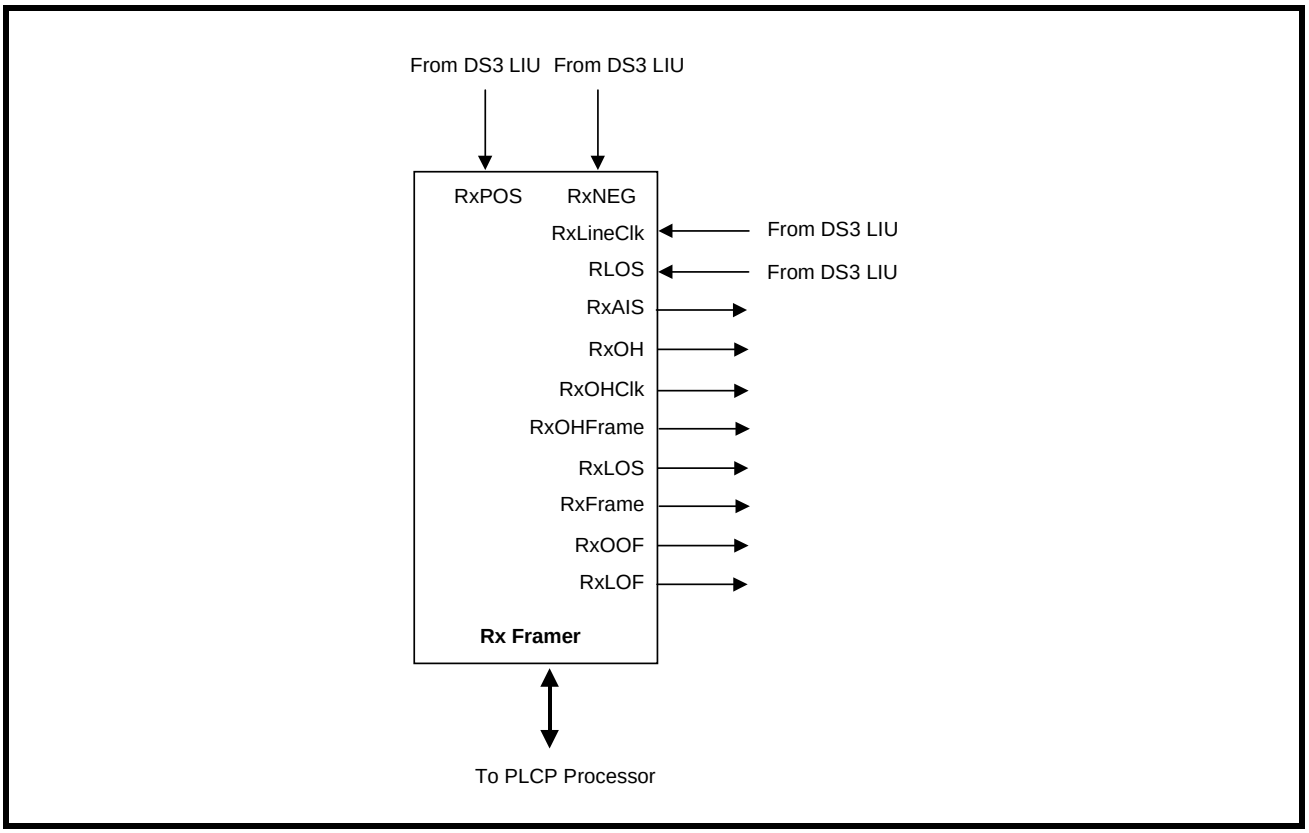
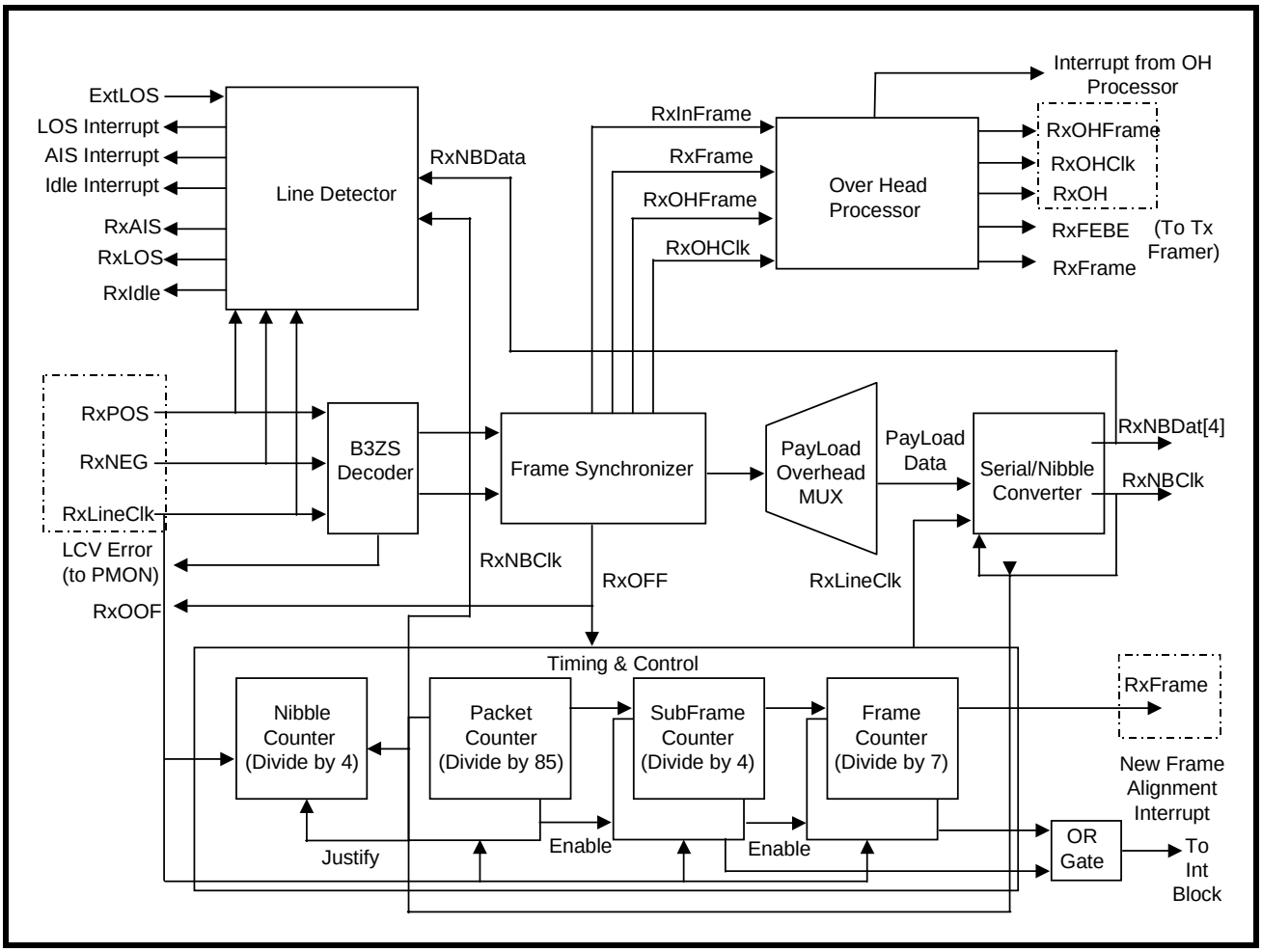


FIGURE 22. FUNCTIONAL BLOCK DIAGRAM OF RECEIVER FRAMER



4.2 Receive PLCP Processor

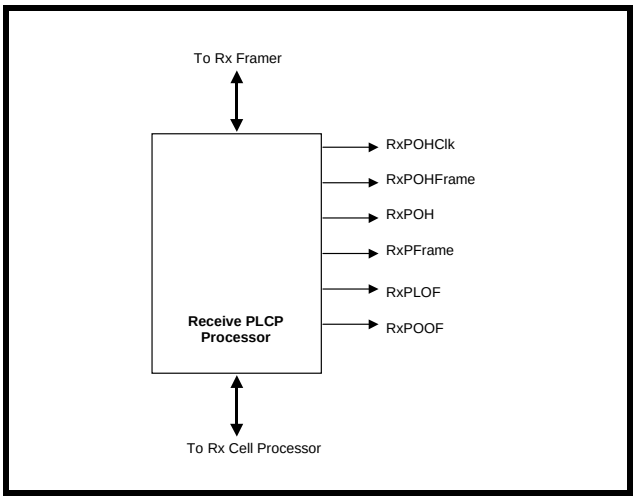
4.2.1 Operation of the Receive PLCP Processor

The Receive PLCP Processor receives PLCP frame data from the Receive DS3 Framer and locates the boundaries of these incoming PLCP frames. The Receive PLCP processor also extracts the PLCP overhead bytes, computes and verifies the incoming BIP-8 (B1) byte, transfers FEBE and Yellow Alarm information to the “Near-End” Transmit PLCP Processor, for transmittal back to the Far-End Terminal. Finally, these PLCP frames (and their designated boundaries) are routed to the Receive Cell Processor, for further processing.

Note: The Receive PLCP Processor is disabled when the UNI is operating in the “Direct Mapped ATM” mode.

Figure 23 presents a simple illustration of the Receive PLCP Processor block along with the associated external pins.

FIGURE 23. ILLUSTRATION OF THE SIMPLE BLOCK DIAGRAM OF THE RECEIVE PLCP PROCESSOR



4.2.2 Functional Description of the Receive PLCP Processor

The Receive PLCP Processor receives and operates on data extracted from the payload-portion of the incoming DS3 data stream (via the Receive DS3 Framer). Once the Receive DS3 Framer reaches the "In-Frame" state, then the Receive PLCP Processor will take this incoming data and begin searching for the PLCP frame boundaries. The Receive PLCP Processor will inform the "outside world" that it has began detecting these PLCP frame boundaries by pulsing the RxPFrame output pin. Figure 24 , presents a Functional Block Diagram of the Receive PLCP Processor and Table 19 presents the Byte Format for a PLCP Frame.

FIGURE 24. FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE PLCP PROCESSOR BLOCK

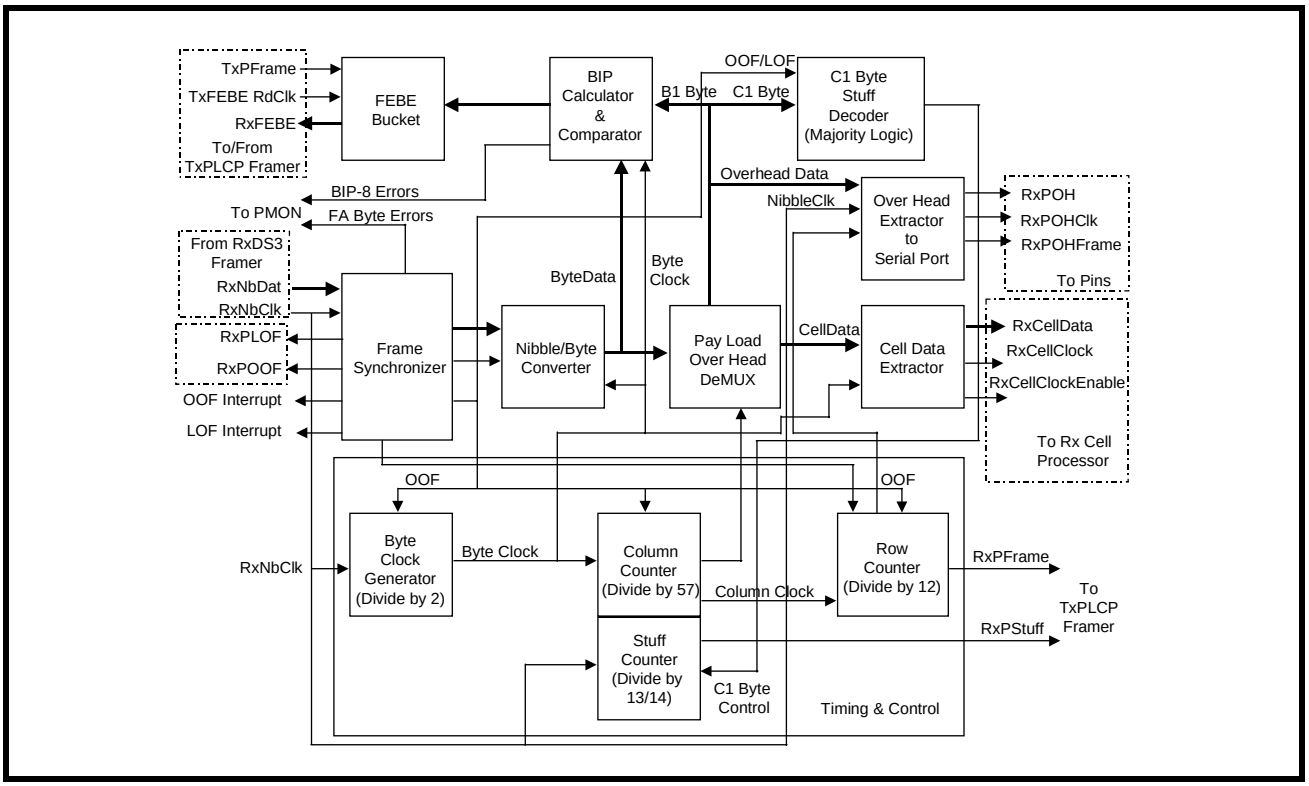


Figure 24 indicates that the PLCP Frame consists of 12 ATM Cells, 48 bytes of Overhead (OH) bytes, and 13 or 14 nibbles of “trailer” for frequency justification.

TABLE 19: BYTE FORMAT OF THE PLCP FRAME

PLCP FRAME 2 BYTES		POI 1 BYTE	POH 1 BYTE	PLCP PAYLOAD 53 BYTES	13-14 NIBBLES
A1	A2	P11	Z6	First ATM Cell	
A1	A2	P10	Z5	ATM Cell	
A1	A2	P9	Z4	ATM Cell	
A1	A2	P8	Z3	ATM Cell	
A1	A2	P7	Z2	ATM Cell	
A1	A2	P6	Z1	ATM Cell	
A1	A2	P5	X	ATM Cell	
A1	A2	P4	B1	ATM Cell	
A1	A2	P3	G1	ATM Cell	
A1	A2	P2	X	ATM Cell	
A1	A2	P1	X	ATM Cell	
A1	A2	P0	C1	Twelfth ATM Cell	Trailer

The contents of the Path Overhead (POH) bytes (e.g., Z6 through C1) of the incoming PLCP frame is output via a serial port consisting of the RxPOH, RxPOHClk, and RxPOHFrame output pins. This serial output port is discussed in greater detail in section 7.2.2.3.

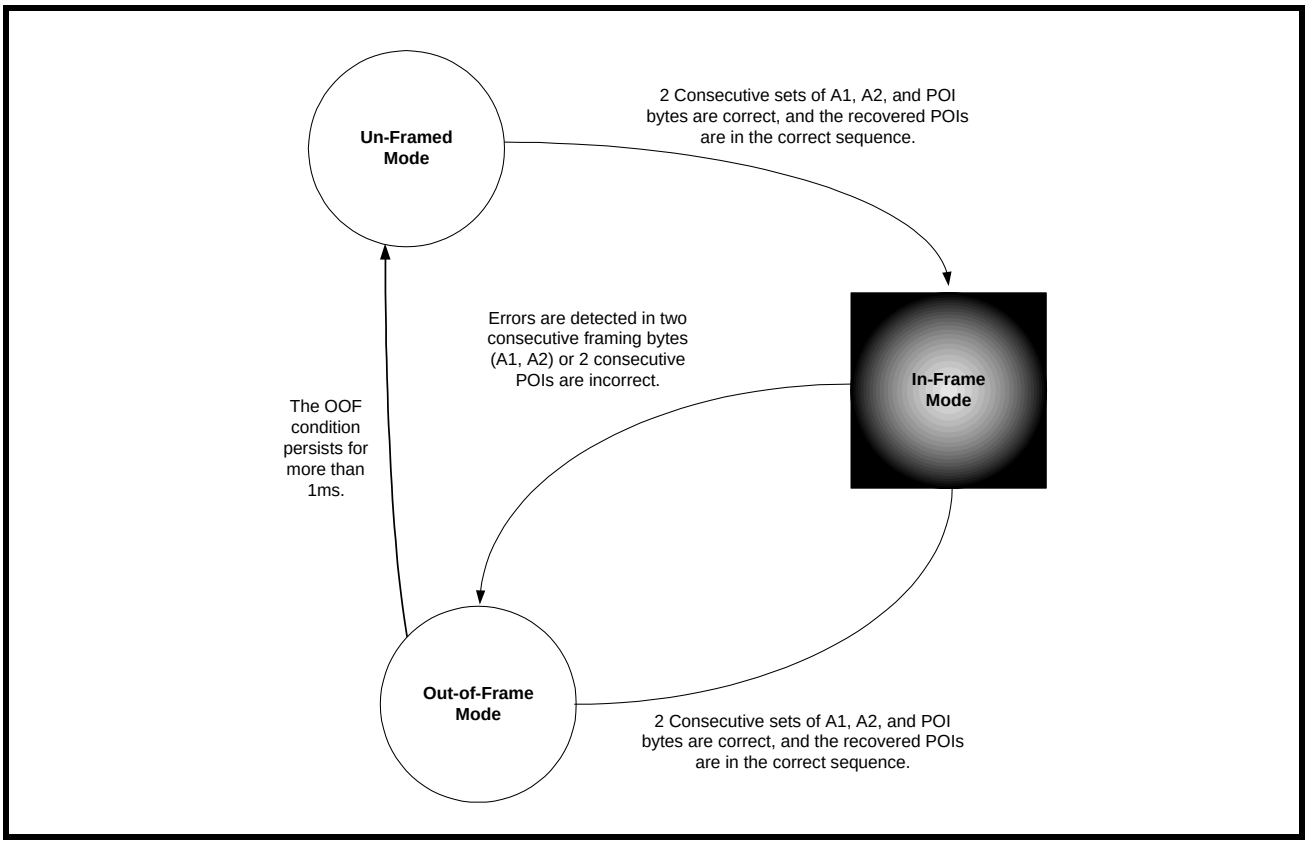
4.2.2.1 PLCP Framing

At any given time, the Receive PLCP Processor will be operating in any one of three (3) “framing” modes.

- Un-Framed
- Out-of-Frame (OOF)
- In-Frame

The State Machine diagram of the Receive PLCP Processor framing algorithm is presented in Figure 25 , and each of these framing modes are discussed.

FIGURE 25. STATE MACHINE DIAGRAM OF THE RECEIVE PLCP PROCESSOR FRAMING ALGORITHM



4.2.2.1.1 The Un-Framed Mode

When the Receive PLCP processor is operating in the “Un-Framed” mode, it does not have any form of frame synchronization with the incoming PLCP data.

The Receive PLCP Processor will indicate that it is in the “Un-Framed” Mode to external circuitry by asserting both the RxPOOF and RxPLOF output pins and the “POOF Status” and “PLOF Status” bits within the RxPLCP Configuration/Status Register, as depicted below.

RxPLCP Configuration/Status Register (Address = 44h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Reframe	POOF Status	PLOF Status	Yellow Status
x	x	x	x	x	1	1	x

The Receive PLCP Processor will attempt to acquire PLCP framing once the Receive DS3 Framer has reached the “In-Frame” state. Specifically, the Receive PLCP Processor will attempt to find the boundaries of the PLCP frames by first searching for the Frame Alignment bytes: A1 and A2. The value of the A1 and A2 bytes are F6h and 28h, respectively. After the Receive PLCP Processor locates the Frame Alignment bytes, it will then begin to read and align itself in accordance with the POI (Path Overhead Indicator) bytes.

The Receive PLCP processor will declare itself “in-frame” if two consecutive sets of A1, A2 and POI bytes are correct and if the received POIs are in the correct sequence.

4.2.2.1.2 In-Frame (Frame Maintenance Mode)

When the Receive PLCP Processor is operating in the “In-Frame” mode, it means that it is continually correctly locating the boundaries of the incoming PLCP frames. This also enables the Receive PLCP Processor to perform its tasks of POH byte extraction and processing. The Receive PLCP processor will indicate its detection of a PLCP frame boundary by pulsing the RxPFrame output pin “high” at the end of each frame. Therefore, the pulse rate of this output pin is nominally 8 kHz. The Receive PLCP Processor will notify the localµC/µP of its transition from the “Unframed” to the “In-frame” state by:

1. Negating both the RxPOOF and RxPLOF output pins
2. Negating both the POOF Status and PLOF Status bits in the RxPLCP Configuration/Status Register.
3. Generating a “Change of OOF/LOF” status interrupt request to the local µC/µP.

Additionally, while the Receive PLCP Processor is operating in the “In-frame” mode, it also will be performing “Frame Maintenance” functions by continually checking for and report framing errors. To monitor the number of Framing Errors that have been detected by the Receive PLCP Processor read the PMON PLCP Framing Byte Error Count Registers which are located at Addresses 2Ah and 2Bh. The bit-formats of these two registers are presented below.

Address = 2Ah, PMON PLCP Framing Byte Error Count Register—MSB

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FA Error Count—High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Address = 2Bh, PMON PLCP Framing Byte Error Count Register—LSB

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FA Error Count—Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

The contents of these registers reflect the total number of PLCP Framing Errors that have been detected since the last read of these registers. These registers are reset upon read.

4.2.2.1.3 Out-of-Frame (OOF) Mode

The Receive PLCP Processor will declare an “Out-of-Frame” (OOF) condition, if:

- Errors are detected in two consecutive framing bytes (A1, A2), or
- Two consecutive POIs values are both incorrect.

Once the Receive PLCP Processor declares “OOF”, then it will enter the “Out-of-Frame” state (per Figure 25).

Please note that this mode should not be confused with the “Un-Framed” mode.

When the Receive PLCP Processor is operating in the “OOF” mode, it will attempt to re-acquire the “In-frame” status. However, the Receive PLCP Processor will continue to use the previous frame synchronization, while operating in this mode. If the Receive PLCP Processor cannot re-acquire the “In-Frame” status after being in the “OOF” mode for 1ms (approximately 8 PLCP frames) or more, then the Receive PLCP Processor will declare a “Loss of Frame” and will transition back to the “Un-Framed Mode”.

The Receive PLCP Processor will indicate its transition to the “Out-of-Frame” mode by

1. Asserting the RxPOOF pin (Note: the RxPLOF pin will still remain negated).
2. Asserting the “POOF” status bit in the RxPLCP Configuration/Status Register.
3. Generating a “Change of OOF” status interrupt request to the local µC/µP.

If the Receive PLCP Processor is able to regain Frame Synchronization, it will negate the RxOOF output pin and “POOF Status” bit-field in the “RxPLCP Configuration/Status Register. The Receive PLCP Processor

will also alert the local µP/µC of this occurrence by generating the "Change in OOF Condition" interrupt.

To determine the framing state that the Receive PLCP Processor is operating in, read bits 1 and 2 of the Receive PLCP Configuration Status Register. The bit-format of this register is presented below.

RxPLCP Configuration/Status Register (Address = 44h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused				Reframe	POOF Status	PLOF Status	Yellow Status
RO	RO	RO	RO	R/W	RO	RO	RO

Bit 1—PLOF Status

A "1" in this bit-field indicates a "Loss of Frame" status. Consequently, the Receive PLCP Processor will be operating in the "Un-framed" state. Conversely, a "0" in this bit-field indicates that the Receive PLCP Processor is either in the "In-Frame" or "Out-of-Frame" state.

Note: the state of this bit-field (and the RxLOF output pin) is controlled by the contents of an Up/Down Counter. This counter is incremented whenever the "POOF Status" bit is "1" and is decremented when the "POOF Status bit is '0'. However, the counter is decremented at 1/12th of the rate that it is incremented. Therefore, when the Receive PLCP Processor goes into the "OOF" condition, this Up/Down Counter will increment. If the Receive PLCP Processor requires 1ms to regain Frame-Synchronization, the PLOF bit-field might very well be asserted, denoting an "LOF condition". However, even after the Receive PLCP Processor has declared itself "In-Frame", the PLOF bit-field will not be negated until the POOF bit-field has been negated for 12 ms.

Bit 2—POOF Status

A "1" in this bit-field indicates an "Out-of-Frame" condition. This condition necessarily indicates that the Receive PLCP Processor is not in the "In-frame" condition. Therefore, the user will have to read-in the value of bit 1 in order to determine if the Receive PLCP Processor is operating in the "Out-of-Frame" or "Un-Framed" state.

The following table relates the "read-in" values for bits 1 and 2 to the framing state of the Receive PLCP Processor.

Table 20: The Relationship between the logic states of the POOF and PLOF bit-fields, and the corresponding Receive PLCP Framing State

POOF Bit 2	PLOF Bit 1	Receive PLCP Framing State
0	0	In-Frame
0	1	In-Frame—PLOF is still "1" during the "12 ms period" that POOF is "0"
1	0	Out of Frame
1	1	Un-frame

4.2.2.1.4 Reframe via Software Command

The Receive PLCP Processor can be forced into the "OOF" mode, via software command. This is accomplished by writing a "1" to Bit 3 in the RxPLCP Configuration/Status Register, as depicted below.

RxPLCP Configuration/Status Register (Address = 44h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused				Reframe	POOF Status	PLOF Status	Yellow Status
x	x	x	x	1	x	x	x

4.2.2.2 Overhead Byte Processing

Once the Receive PLCP Processor enters into the "In-frame" mode, the 12 POH bytes are then extracted and output via a serial output port. Presently, the Receive PLCP Processor is only concerned with three (3) of these POH bytes: B1, G1, and C1. The manner in which the Receive PLCP Processor handles these POH bytes follows.

4.2.2.2.1 B1 (BIP-8) Byte

The Receive PLCP Processor will perform a BIP-8 calculation over an entire PLCP frame (excluding the A1, A2 and POI bytes) that it receives from the Receive DS3 Framer. Afterwards, the Receive PLCP Processor will read in the B1 byte, of the very next incoming PLCP frame, and perform a bit-by-bit comparison between this B1 byte and this locally-computed BIP-8 value. By the nature of the BIP-8 values, it is possible to have as many as 8 bit errors in this comparison. If the Receive PLCP Processor detects any BIP-8 errors, then it will do two things:

- increment the PMON BIP-8 Error Count Registers (Address = 28h and 29h) by the number of detected bit-errors, and,
- Inform the “Far-End” Terminal (e.g., the source of the errored data) of this occurrence by routing the number of bit-errors that were detected in this frame to the “Near-End” Transmit PLCP Processor. The Transmit PLCP Processor will then insert this number into the FEBE-nibble within the G1 byte of an outbound PLCP frame. Then the outbound PLCP frame (containing the information on the B1 byte error) will be transmitted to the “Far-End” terminal where it will be processed appropriately.

Table 21 presents the bit format of the G1 byte. The Receive PLCP processor performs this function in order to inform the “Far-End Terminal that bit errors have been detected in its transmission.

TABLE 21: BIT FORMAT OF THE G1 BYTE

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Far End Block Error (FEBE)				RAI (Yellow)	X bits (Ignored by the Receiver)		
4 Bits				1 Bit	3 Bits		

The bit-format of the PMON BIP-8 Error Count Register (Address = 28h and 29h) are presented below.

Address = 28h, PMON BIP-8 Error Count Register—MSB

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
0	0	0	0	0	0	0	0

Address = 29h, PMON BIP-8 Error Count Register—LSB

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
0	0	0	0	0	0	0	0

The contents of these registers reflect the total number of BIP-8 Errors that have been detected since the last read of these registers. These registers are reset upon read.

4.2.2.2.2 G1 Byte

The incoming G1 Byte serves to provide the “Near-End” Terminal with diagnostic information on the quality of the transmission link between the “Near-End” Transmit PLCP Processor and the “Far-End” Receive PLCP Processor. The bit-format of the G1 byte, presented in Table 21 , indicates that 5 of the 8 bits in this byte are relevant to transmission diagnosis.

Bit 3—RAI—Yellow Alarm Indicator

This bit-field serves as a “Yellow Alarm” indicator. The “Far-End” Transmit PLCP Processor will assert this bit-field if the “Far End” Receive PLCP Processor has had sufficient trouble receiving valid data from the “Near-End” Transmit PLCP Processor; and that this condition has persisted for 2 to 10 seconds. If this bit-field is asserted for 10 consecutive incoming PLCP

frames then the Receive PLCP Processor will assert the “Yellow Alarm” status bit (Bit 0) within the Receive PLCP Configuration/Status Register, as depicted below.

RxPLCP Configuration/Status Register (Address = 44h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Reframe	POOF Status	PLOF Status	Yellow Status
x	x	x	x	x	x	x	1

Bit 0, within the Receive PLCP Configuration Status register will be negated when the Receive PLCP Processor has received 10 consecutive G1 bytes with the RAI bit-field being “0”.

Bits 4 through 7—FEBE

This nibble-field represents the number of “BIP-8” bit-errors that were detected by the “Far-End” Receive PLCP Processor in a given PLCP frame. Because of the nature of the BIP-8 value, the FEBE nibble-field can indicate as many as 8 bit-errors. If the “Near-End” Receive PLCP Processor receives a G1 byte that contains a non-zero FEBE value, then the “Near-End” Receive PLCP Processor will increment the PMON PLCP FEBE Count Register (Address = 2C, 2D) by the value of the FEAC nibble-field within the received G1 byte. The bit-format of these registers is presented below.

Address = 2Ch, PMON PLCP FEBE Count Register—MSB

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PFEBE Count—High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Address = 2Dh, PMON PLCP FEBE Count Register—LSB

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PFEBE Count—Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

4.2.2.2.3 C1 Byte

The Receive PLCP processor will determine the number of trailer nibbles that exist in a given frame by reading the contents of the incoming C1 byte which is the POH byte of the 12th row of a PLCP frame. For a detailed discussion on the meaning of the C1 Byte, please see Section 6.3.3.1.

4.2.2.3 Extracting PLCP Overhead Bytes via the Serial Output Port

Once the Receive PLCP Processor declares itself “In-Frame”, then it will begin to output data via the “Receive PLCP Processor POH Byte” serial output port.

The “Receive PLCP Processor POH Byte” serial output port consists of the following output pins.

- RxPOH
- RxPOHFrame
- RxPOHClk

Table 22 presents the byte format of the PLCP frame. The “shaded” bytes represent the data that is output via the RxPOH pin. Each POH byte is output with the MSB (most significant bit) first. Each bit, within each of these POH bytes is output on the rising edge of the RxPOHClk signal. The RxPOHClk signal has a nominal frequency of 768 kHz. The Receive PLCP Processor will assert the RxPOHFrame signal

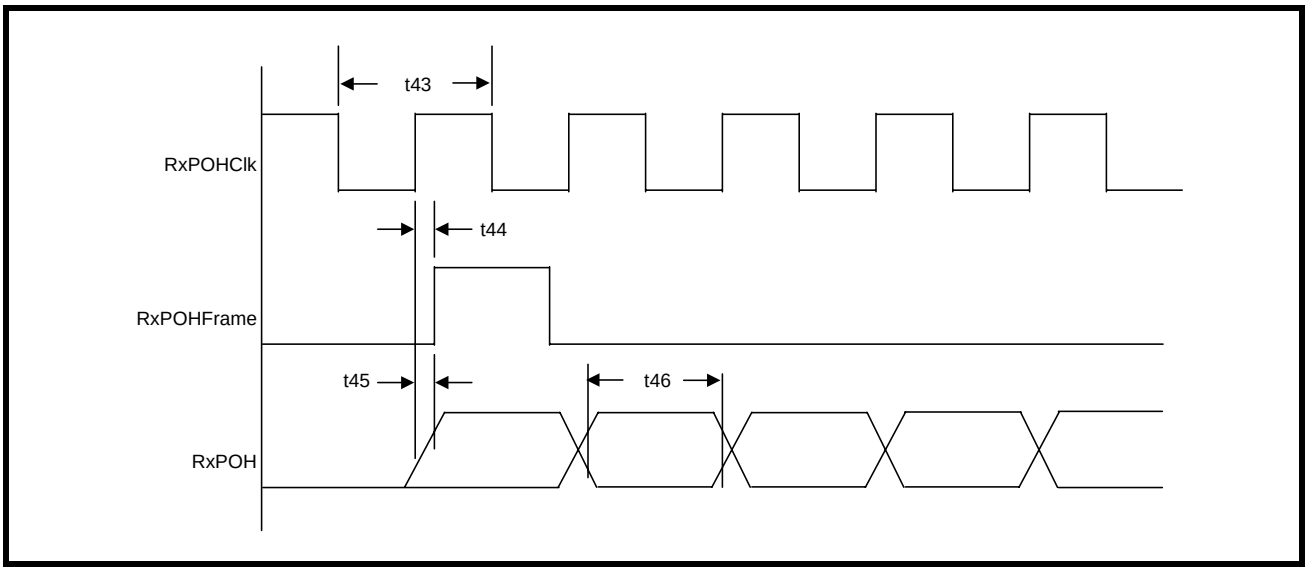
when the MSB of the Z6 byte is output via the RxPOH output pin.

TABLE 22: BYTE FORMAT OF PLCP FRAME–POH BYTES HIGHLIGHTED.

PLCP FRAME 2 BYTES		POI 1 BYTE	POH 1 BYTE	PLCP PAYLOAD 53 BYTES	
A1	A2	P11	Z6	First ATM Cell	13–14 Nibbles
A1	A2	P10	Z5	ATM Cell	
A1	A2	P9	Z4	ATM Cell	
A1	A2	P8	Z3	ATM Cell	
A1	A2	P7	Z2	ATM Cell	
A1	A2	P6	Z1	ATM Cell	
A1	A2	P5	X	ATM Cell	
A1	A2	P4	B1	ATM Cell	
A1	A2	P3	G1	ATM Cell	
A1	A2	P2	X	ATM Cell	
A1	A2	P1	X	ATM Cell	
A1	A2	P0	C1	Twelfth ATM Cell	Trailer

Figure 26 presents a drawing of waveforms illustrating the timing relationship between RxPOH, RxPOHFrame, and RxPOHClk.

FIGURE 26. TIMING RELATIONSHIP BETWEEN THE RECEIVE PLCP POH BYTE SERIAL OUTPUT PORT PINS—RXPOH, RXPOHFRAME AND RXPOHCLK.



4.2.2.4 Direct-Mapped ATM Mode

The Receive PLCP Processor will be disabled if the XRT74L74 DS3/E3 UNI is configured to operate in the “Direct Mapped ATM” Mode.

4.2.2.5 Receive PLCP Processor-related Interrupts

The Receive PLCP Processor will generate interrupts upon the following conditions:

- Change in OOF status

• Change in LOF status

If one of these conditions occur, and if that particular condition is enabled for interrupt generation, then when the local µC/µP reads the UNI Interrupt Status Register, as shown below; it should read "x1xxxxxb" (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
RxDS3 Interrupt Status	RxPLCP Interrupt Status	RxCP Interrupt Status	RxUTOPIA Interrupt Status	TxUTOPIA Interrupt Status	TxCP Interrupt Status	TxDS3 Interrupt Status	One Sec Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RUR

At this point, the local µC/µP will have determined that the Receive PLCP Processor block is the source of the interrupt, and that the Interrupt Service Routine should branch accordingly. In order to accomplish this the local µP/µC should now read the RxPLCP Interrupt Status Register. The bit-format of the RxPLCP Interrupt Status register is presented below.

RxPLCP Interrupt Status Register (Address = 46h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						**POOF Interrupt Status**	**RLOF Interrupt Status**
RO	RO	RO	RO	RO	RO	**RUR**	**RUR**

The bit format of the RxPLCP Interrupt Status Register indicates that only two (2) bit-fields, within this register, are active. The role of each of these bit fields follows.

Bit 0—"PLOF Interrupt Status

A "1" in this bit-field indicates that the Receive PLCP Processor has requested a "Change of PLOF" interrupt. Note, this type of interrupt could occur due to a transition in the framing state from the "Out-of-Frame" state to the "Un-framed" state; during which the RxLOF pin will toggle "high". This type of interrupt could also occur due to a transition from the "Un-framed" state to the "In-frame" state. It is possible to distinguish between these two possibilities based upon the read-in content of the RxPLCP Configuration/Status register. If the local µC/µP reads in a 'xxxxx00xb" value from this register, then the "Change in PLOF" interrupt request was due to a transition from the "Un-framed" to the "In-frame" condition. Conversely, if the local µC/µP reads in the value "xxxxx11xb" then the "Change in PLOF" interrupt request was due to a transition from the "Out-of-Frame" state to the "Un-framed" state.

Bit 1—POOF Interrupt Status

A "1" in this bit-field indicates that the Receive PLCP Processor has requested a "Change of OOF status" interrupt. Note, this type of interrupt request could occur due to a transition from the "Un-framed" state to the "In-frame" state;' during which the RxOOF pin will toggle "low". This type of interrupt could also occur due to a transition from the "In-frame" to the "Out-of-Frame" state. It is possible to distinguish between these two possibilities based upon the read-in content of the RxPLCP Configuration/Status register. If the local µC/µP reads in a "xxxxx0xxb" value from this register, then the Receive PLCP Processor has transitioned from the "Un-framed" state to the "In-frame" state. Conversely, if the local µC/µP reads in "xxxxx1xxb", then this indicates the transition from the "In-frame" state to the "Out-of Frame" state.

Each of these interrupts can be enabled/disabled by writing the appropriate data to the Receive PLCP Interrupt Enable Register. This register has the exact same bit-format as does the Receive PLCP Interrupt Status Register. The bit-format of this register is presented below.

Receive PLCP Interrupt Enable Register

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						POOF Interrupt Enable	PLOF Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	x	x	x	x	x	0	0

To enable these interrupts write a “1” to their corresponding bit-fields, in this register. Conversely, to disable these interrupts write a “0” to these bit fields. These bit-fields are “0” upon power-up or reset of the UNI chip.

4.3 Receive Cell Processor

4.3.1 Brief Description of the Receive Cell Processor

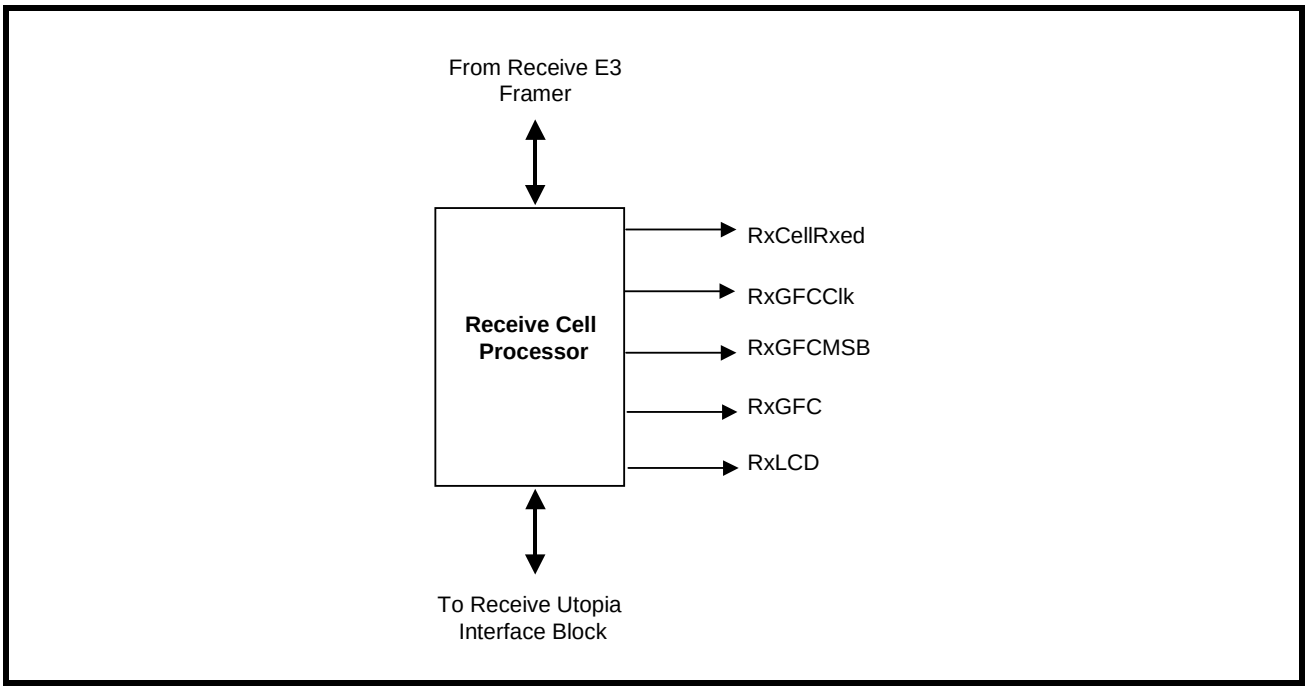
The Receive Cell Processor receives either delineated PLCP frames from the Receive PLCP Processor, or “Direct Mapped ATM” cells from the Receive DS3 Framer. The Receive Cell Processor will then perform the following operations on this data.

- 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 “Receive GFC Nibble Field” Serial Output port.

Figure 27 presents a simple block diagram of the Receive Cell Processor block along with its external pins.

FIGURE 27. SIMPLE ILLUSTRATION OF THE RECEIVE CELL PROCESSOR, WITH ASSOCIATED PINS



4.3.2 Functional Description of Receive Cell Processor

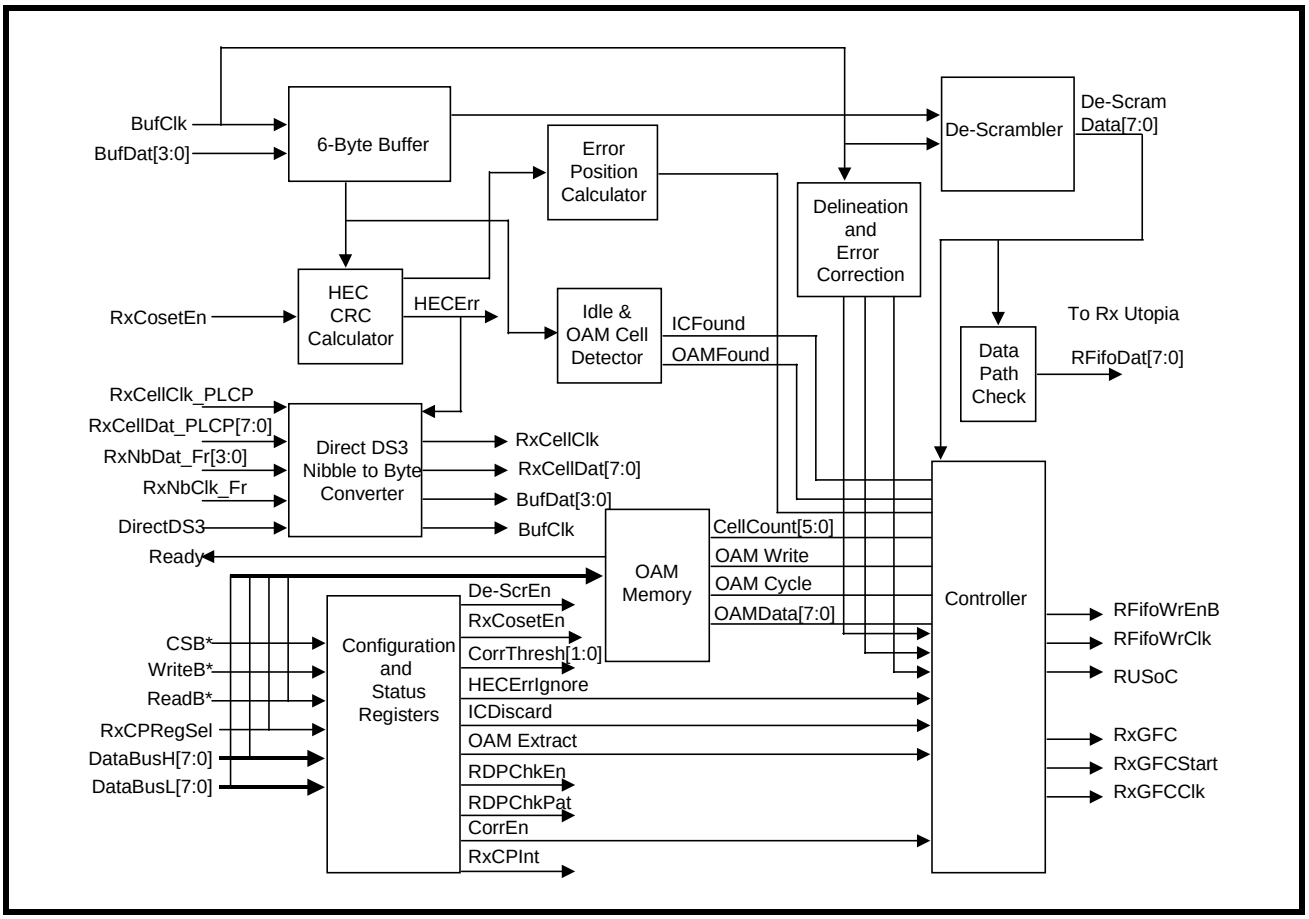
The Receive Cell Processor receives delineated frames from the Receive PLCP Processor (or ATM Cells from the Receive DS3 Framer). Once the Receive Cell Processor receives this information then it will proceed to perform the following functions.

- 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 28 presents a functional block diagram of the Receive Cell Processor.

FIGURE 28. FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE CELL PROCESSOR



4.3.2.1 Cell Delineation

The approach that the Receive Cell Processor will use to perform cell-delineation depends upon whether the UNI is operating in the "PLCP" mode (e.g., with the PLCP Processors active) or in the "Direct-Mapped ATM" mode (e.g., with the PLCP Processors disabled). The cell-delineation process for each of these modes are discussed below.

4.3.2.1.1 Cell Delineation while the UNI is Operating in the PLCP Mode

The Receive PLCP Processor determines the frame boundaries of the PLCP frame data that it receives from the Receive DS3 Framer. Afterwards, the Receive PLCP Processor will transfer these PLCP frames, along with the frame boundary information to the Receive Cell Processor. Table 23 presents the byte-format of the PLCP frame. It is easy to see, from this figure, that if the Receive Cell Processor is aware of the locations of the boundaries of these PLCP frames, then the comprising ATM cells are easily located and thus delineated.

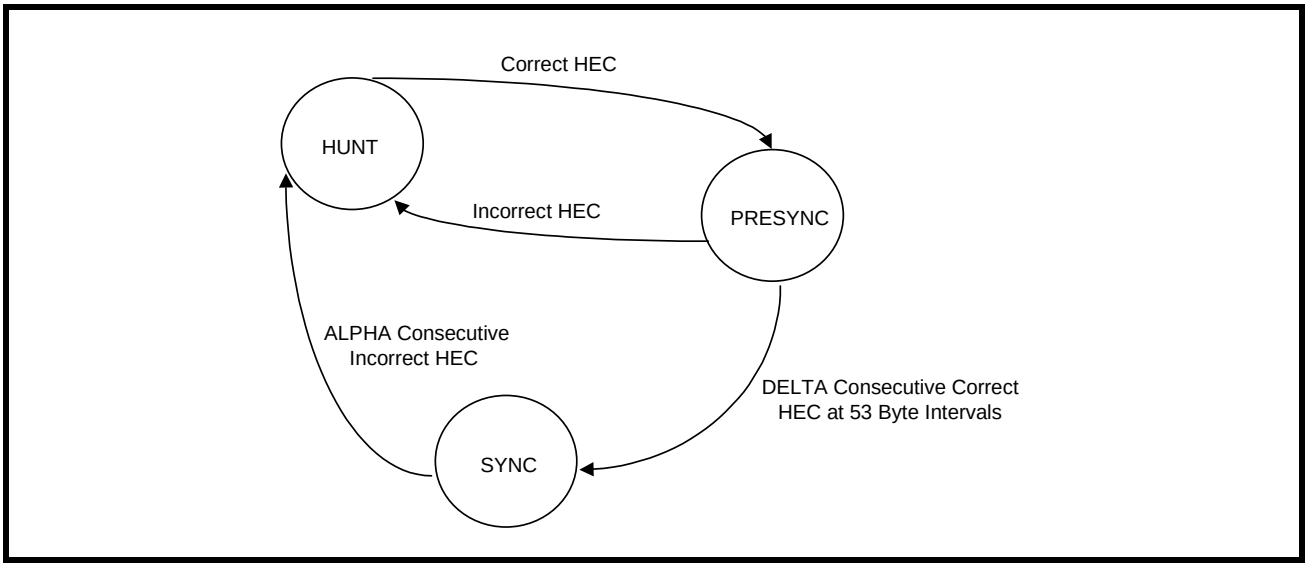
TABLE 23: BYTE-FORMAT OF THE PLCP FRAME

PLCP FRAME 2 BYTES		POI 1 BYTE	POH 1 BYTE	PLCP PAYLOAD 53 BYTES	
A1	A2	P11	Z6	First ATM Cell	13–14 nibbles
A1	A2	P10	Z5	ATM Cell	
A1	A2	P9	Z4	ATM Cell	
A1	A2	P8	Z3	ATM Cell	
A1	A2	P7	Z2	ATM Cell	
A1	A2	P6	Z1	ATM Cell	
A1	A2	P5	X	ATM Cell	
A1	A2	P4	B1	ATM Cell	
A1	A2	P3	G1	ATM Cell	
A1	A2	P2	X	ATM Cell	
A1	A2	P1	X	ATM Cell	
A1	A2	P0	C1	Twelfth ATM Cell	Trailer

4.3.2.1.2 Cell Delineation while the UNI is Operating in the "Direct-Mapped ATM" mode.

When the UNI is operating in the "Direct-Mapped ATM" mode, then the Receive Cell Processor is receiving unframed cell data from the Receive DS3 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 29 . Each of these states are discussed below. .

FIGURE 29. CELL DELINEATION ALGORITHM EMPLOYED BY THE RECEIVE CELL PROCESSOR, WHEN THE UNI IS OPERATING IN THE "DIRECT-MAPPED" ATM MODE.



The HUNT State

When the UNI chip is first powered up and configured to operate in the "Direct-Mapped ATM" mode, 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 DS3 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 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 µP (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 “RxCP Interrupt Status” Register, as depicted below.

RxCP Interrupt Status Register (Address = 4Eh)

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 (RxLCD) within the RxCP Configuration Register to “0”.

The SYNC State

When the Receive Cell Processor is operating in 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 is is not properly delineating cells by doing the following.

- Generating a “Change in LCD State” interrupt.
- Assert the RxLCD output pin (e.g., toggling it “high”).
- Setting bit 7 (RxLCD) within the “RxCP Configuration Register” to “0”, as depicted below.

RxCP Configuration Register (Address = 4Ch)

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 be written to the RxFIFO within the Receive UTOPIA Interface block.

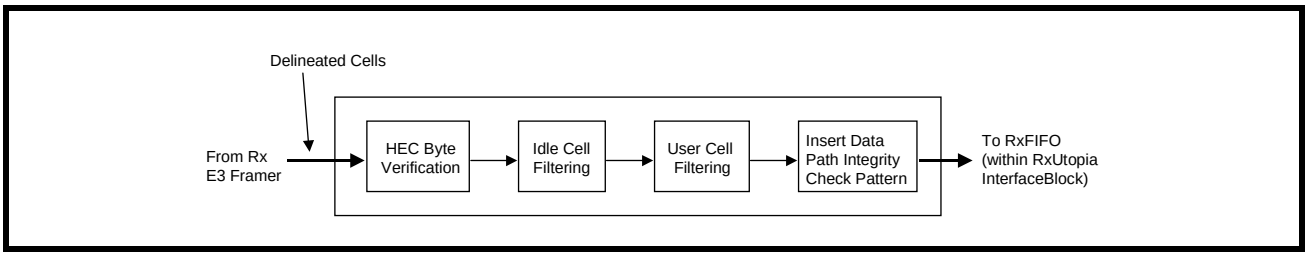
The sequence of filtering/processing that each cell must go through is listed below in sequential order.

- HEC Byte Verification
- Idle Cell Filtering
- User Cell Filteing
- 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 30 .

Each of these "Filtering/Processing" steps (within the Receive Cell Processor) are discussed in detail below.

FIGURE 30. ILLUSTRATION OF OVERALL CELL FILTERING/PROCESSING PROCEDURING THE OCCURS WITHIN THE RECEIVE CELL PROCESSOR



4.3.2.2 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 PLCP Processor (or Receive DS3 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 PLCP Processor (or the Receive DS3 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 employs 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 (RxCoset Enable) of the RxCP Configuration Register; as depicted below.

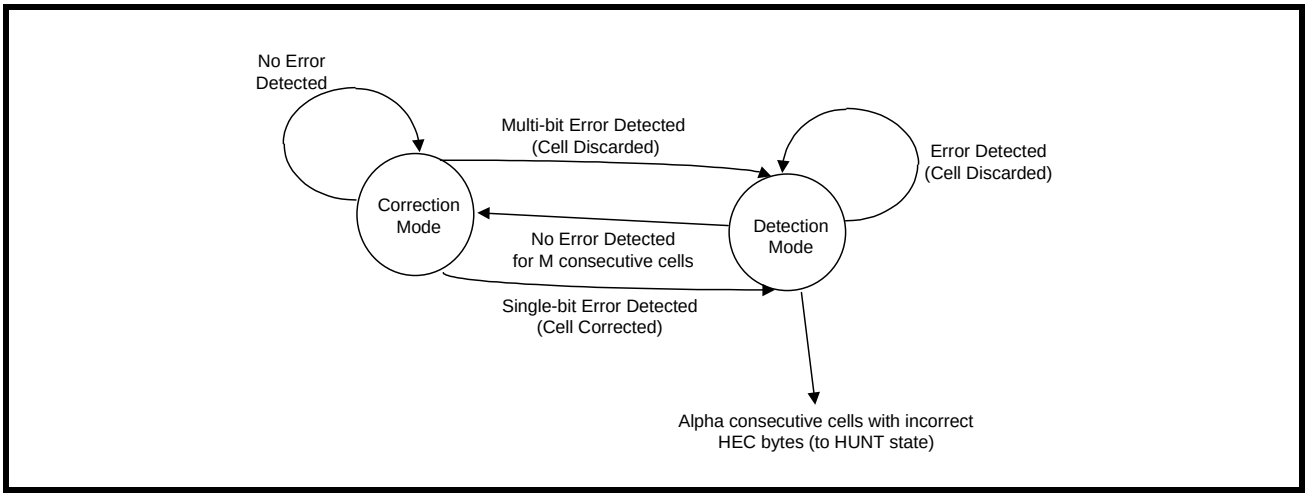
RxCP Configuration Register (Address = 4Ch)

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 "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 31 presents a State Machine Diagram of the "HEC Byte Error Correction/Detection" Algorithm. Each of these states are discussed below.

FIGURE 31. 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. To configure the Receive Cell Processor to retain these cells with multi-bit errors, write to bit 0 (HEC Error Ignore) of the RxCP Configuration Register, as depicted below.

RxCP Configuration Register (Address = 4Ch)

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

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 2Eh and 2Fh and their bit-formats are presented below.

PMON Received Single HEC Error Count—MSB (Address = 2Eh)

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

PMON Received Single HEC Error Count—LSB (Address = 2Fh)

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 = 30h and 31h). 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 = 30h)

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 = 31h)

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. To configure the Receive Cell Processor to retain errored cells, write to bit 0 (HEC Error Ignore) of the RxCP Configuration register (Address = 4Ch), 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. To configure the UNI to use any of these values for M, write the appropriate values to the "RxCP Additional Configuration" Register (Address = 4Dh), as depicted below.

RxCP Additional Configuration Register (Address = 4Dh)

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 is used to enable/disable the "Correction Mode" portion of the "HEC Byte Error Correction/Detection" algorithm. If a "0" is written to this bit-field, the "HEC Byte Error Correction/Detection" algorithm will be disabled from entry/operation in the "Correction" mode. Therefore, the Receive Cell Processor will only operate in the "Detection" mode. If a "1" is written to this bit field then the "HEC Byte Error Correction/Detection" algorithm will transition into and out of the "Correction" as dictated by the "Correction Threshold".

Bits 2 and 3—Correction Threshold [1, 0]

These "Read/Write" bit-fields are used 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 24: 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

4.3.2.3 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 7.3.2.2.)
- 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 7.3.2.2.

4.3.2.3.1 Idle Cell Filtering

The Receive Cell Processor can be configured to either discard or retain Idle cells by writing to bit 4 (Idle Cell Discard) of the RxCP Configuration Register, as depicted below.

RxCP Configuration Register (Address = 4Ch)

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

If a “0” is written 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 a “1” is written 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, the header byte patterns of these Idle cells must be specified. The Idle Cell header byte pattern is defined based upon the content of 8 read/write registers. These eight registers are the four “RxCP Idle Cell Pattern Header byte registers, and the four “RxCP 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 “RxCP Idle Cell Pattern Header Byte” registers, based upon constraints specified by the contents within the “RxCP Idle Cell Mask Header Byte” registers. The use 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 “RxCP Idle Cell Pattern Header Byte-1” register (Address = 50h). 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 “RxCP Idle Cell Mask Header Byte-1” register (Address = 54h) also plays a role in this comparison process. For instance, if bit-field “0” within the “RxCP 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 “RxCP 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 “RxCP 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 Table 25 , below.

TABLE 25: ILLUSTRATION OF THE ROLE OF THE “RXCP IDLE CELL PATTERN HEADER BYTE” REGISTER, AND THE “RXCP IDLE CELL MASK HEADER BYTE” REGISTER

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 “RxCP 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 “RxCP 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

TABLE 25: ILLUSTRATION OF THE ROLE OF THE "RXCP IDLE CELL PATTERN HEADER BYTE" REGISTER, AND THE "RXCP IDLE CELL MASK HEADER BYTE" REGISTER (CONTINUED)

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 RxCP Idle Cell Mask Header Byte-1 Register)				Don't Care	Don't Care	Don't Care	Don't Care

Results of Comparison

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1	0	1	0	x	x	x	x

Based upon these register settings, any cell containing values in the range of A0h–AFh are considered to 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.

RxCP Idle Cell Pattern Header Byte-1 Register (Address = 50h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Pattern Header Byte-2 Register (Address = 51h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Pattern Header Byte-3 Register (Address = 52h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Pattern Header Byte-4 Register (Address = 53h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle Cell Pattern—Header Byte							

RxCP Idle Cell Pattern Header Byte-4 Register (Address = 53h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

RxCP Idle Cell Mask Header—Byte 1 (Address = 54h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Mask Header—Byte 2 (Address = 55h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Mask Header—Byte 3 (Address = 56h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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

RxCP Idle Cell Mask Header—Byte 4 (Address = 57h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle 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 = 32h, 33h). The bit-format of these registers are presented below.

PMON Received Idle Cell Count—MSB (Address = 32h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle Cell Count—High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

PMON Received Idle Cell Count—LSB (Address = 33h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxIdle Cell Count—Low Byte							
RO	RO	RO	RO	RO	RO	RO	RO
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.

4.3.2.3.2 User Cell Filtering

The Receive Cell Processor can be configured to filter incoming user or OAM cells based upon the value of their header bytes. 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 RxFIFO, within the Receive UTOPIA Interface block.

RxCP Additional Configuration Register (Address = 4Dh)

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]		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 “RxCP 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 “RxCP User Cell Filter Pattern Header byte” registers and the four “RxCP 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 “RxCP User Cell Filter Pattern Header Byte” registers based upon constraints specified by the contents of the “RxCP 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

For example, header byte 1 of a given incoming User cell will be subjected to a bit-by-bit comparison to the contents of the “RxCP User Cell Filter Pattern Header Byte-1” register (Address = 58h). However, the contents of the “RxCP User Cell Filter Mask Header Byte-1” register (Address = 5Ch) also plays a role in this comparison process. For example, if bit-field “0” within the “RxCP 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 “RxCP 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 “RxCP 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 defined in Table 26 , on following page.

TABLE 26: ILLUSTRATION OF THE ROLE OF THE "RXCP USER CELL FILTER PATTERN HEADER BYTE" REGISTER AND THE "RXCP USER CELL FILTER MASK HEADER BYTE" REGISTER.

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 "RxCP 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 "RxCP 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	1	1	1	0	0	0	0

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 RxCP User Cell Filter Mask Header Byte-1 Register)				Don't Care	Don't Care	Don't Care	Don't Care

Resulting "User Cell Fiilter" 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

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 "RxCP Additional Configuration Register (Address = 4Dh). If this bit-field is '0', then only-matching cells will be retained, and 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

RxCP User Filter Cell Pattern Header—Byte 1 (Address = 58h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Pattern Header—Byte 2 (Address = 59h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Pattern Header—Byte 3 (Address = 5Ah)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Pattern Header—Byte 4 (Address = 5Bh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Mask Header—Byte 1 (Address = 6Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Mask Header—Byte 2 (Address = 5Dh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser Cell Mask Header—Byte 2							

RxCP User Filter Cell Mask Header—Byte 2 (Address = 5Dh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

RxCP User Filter Cell Mask Header—Byte 3 (Address = 5Eh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser 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

RxCP User Filter Cell Mask Header—Byte 4 (Address = 5Fh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxUser Cell Mask Heade—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

4.3.2.4 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 27: 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 XRT74L74 DS3/E3 UNI 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.

4.3.2.4.1 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, transmission across a single link, to a destination node; and then terminated at the 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 RxFIFO, 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 Cell Filter is configured 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 allow the "Segment"-type OAM Cell to be written to the RxFIFO, within the Receive UTOPIA Interface Block.

Although the Receive Cell Processor will discard this "Segment" OAM cell, the Receive Cell Processor can be configured 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 a "1" is written to bit 3 (OAM Check Bit) within the "RxCP Configuration" register (Address = 4Ch), 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).

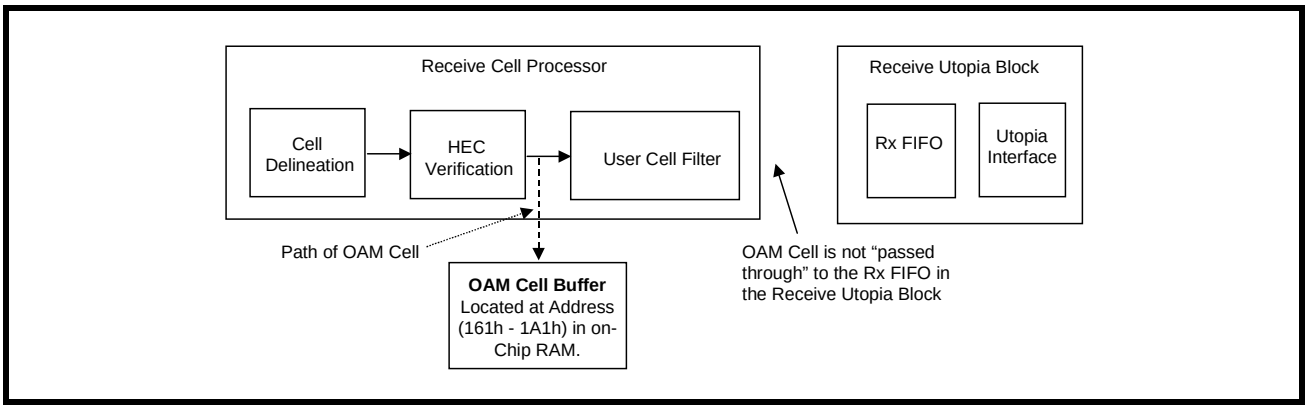
RxCP Configuration Register (Address = 4Ch)

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 a "0" is written to bit 3 of the "RxCP 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 32 presents an illustration depicting how the Receive Cell Processor handles incoming Segment-type OAM cells, if a "1" has been written to bit 3 (OAM Check Bit) of the "RxCP Configuration" register.

FIGURE 32. AN APPROACH TO PROCESSING SEGMENT OAM CELLS, VIA THE RECEIVE CELL PROCESSOR.



4.3.2.4.2 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

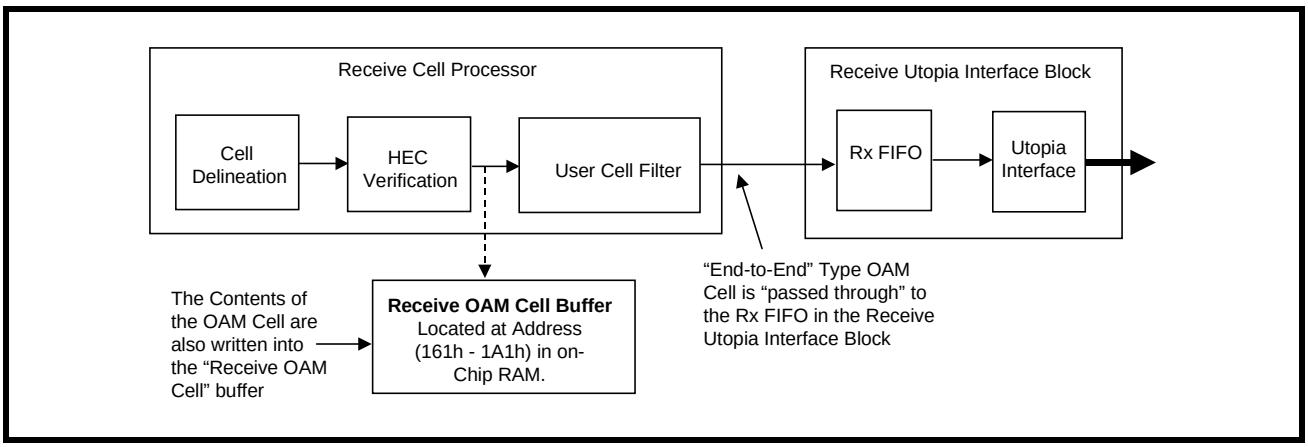
If the Receive Cell Processor determines that the incoming OAM Cell is an "End-to-End" type then it will be written into the RxFIFO, within the Receive UTOPIA Interface block. This act will allow the ATM Layer processor to read in this OAM cell, from the UNI and propagate this cell to other nodes in the network.

Note: The Receive Cell Processor will write the "End-to-End" OAM cell to the RxFIFO, independent of the User Cell Filter settings.

The Receive Cell Processor can also be configured to write the contents of the "End-to-End" OAM celli into the Receive OAM Cell Buffer. For details on how this can be done, please see Section 7.3.2.4.1.

Figure 33 presents an illustration which depicts how the Receive Cell Processor handles incoming End-to-End type OAM Cells, if a "1" has been written to bit 3 (OAM Check Bit) of the "RxCP Configuration" register.

FIGURE 33. APPROACH TO PROCESSING "END-TO-END" OAM CELLS



Monitoring the Number of User/OAM Cells

To monitor the number of Valid cells (User and OAM) that have been received by the Receive Cell Processor, read the PMON Received Valid Cell Count Registers (Address = 34h, and 35h). The bit-format of these registers are presented below.

PMON Received Valid Cell Count—MSB (Address = 34h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxValid 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 = 35h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxValid 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 reflect 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 errors, Idle Cell removal, or User cell filtering) by reading the PMON Discarded Cell Count Registers (Address = 36h, 37h). The bit-format of this register is presented below.

PMON Discarded Cell Count—MSB (Address = 36h)

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 = 37h)

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.

4.3.2.5 Cell Payload De-Scrambling

In numerous applications the payload portion of the incoming cells will be scrambled by the Transmit Cell Processor, within 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 with 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 $x^{43} + 1$.) This option can be configured by writing a “1” to Bit 2 (De-Scramble Enable) of the RxCP Configuration Register, as depicted below.

RxCP Configuration Register (Address = 4Ch)

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

4.3.2.6 Data Path Integrity Check

The "Data Path Integrity" check is a test that is continually run 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) are 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 process, it will be written to the RxFIFO, within the Receive UTOPIA Interface Block. However, prior to being written into the RxFIFO, 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 TxFIFO; prior to computing and overwriting this byte with the HEC byte. For more information on how the Transmit Cell Processor andles the "Data Path Integrity Check" test patterns, please see section 1.2.2.6.

The Receive Cell Processor's Handling of the Data Path Integrity Test pattern

There are a variety of options for configuring the Receive Cell Processor to support the Data Path Integrity Test. First it must be decided whether or not to 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 RxFIFO. The Receive Cell Processor can be configured by writing the appropriate value into bit 5 (RDPChk Pattern Enable) within the "RxCP Configuration Register (Address = 4Ch) as depicted below.

RxCP Configuration Register (Address = 4Ch)

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

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 for chooinge between two possible Data Path Integrity Test patterns,by writing the appropriate value to Bit 6 (RDPChk Pattern) within the "RxCP Configuration" Register (Address = 4Ch). 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 RxFIFO. 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 of the RxCP Configuration Register, is ignored if Bit 5 is set to "0".

4.3.2.7 GFC Nibble Extraction—via the RxGFC Serial Output Port

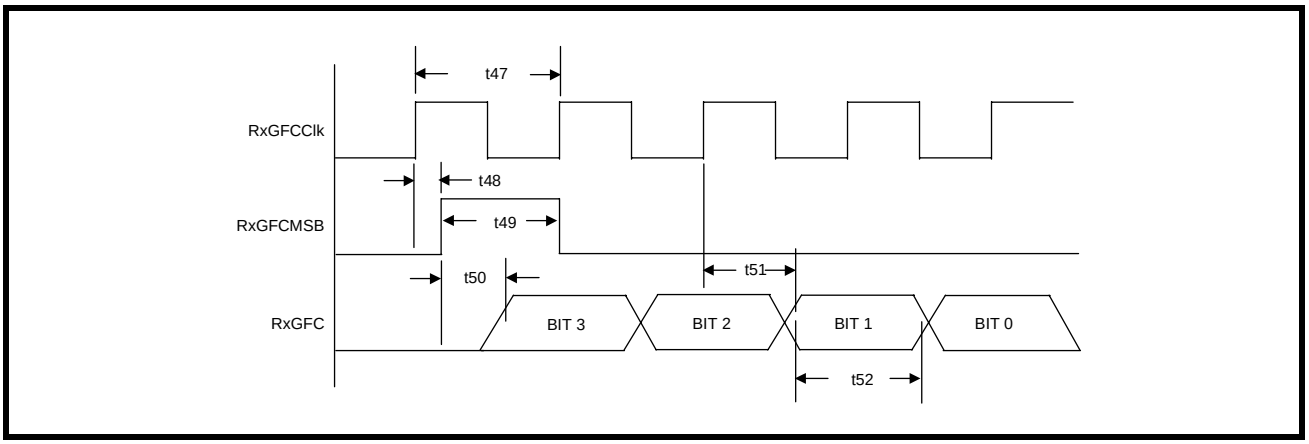
The first four bit-field of each cell header are the GFC bits. The Receive Cell processor will output the contents of the GFC Nibble-field for each cell that it receives, via the "GFC Nibble Field" serial output port.

The "Receive GFC Nibble-Field" serial output port consists of the following pins.

- RxGFC
- RxGFCClk
- RxGFCMSB

The data is output via the RxGFC output pin. The order of transmission, within a given cell, is with the MSB first and in descending order until transmitting the LSB bit. Afterwards, the "GFC Nibble-field" serial output port will output the MSB for the GFC Nibble-field of the next cell. This data is clocked out on the rising edge of the RxGFCClk output signal. The RxGFCMSB output pin will be pulsed "high" each time the MSB of the GFC Nibble field, for a given cell, is present at the RxGFC input. Figure 34 presents an illustration depicting the behavior of the RxGFC Serial Output Port signals.

FIGURE 34. ILLUSTRATION OF THE BEHAVIOR OF THE RXGFC SERIAL OUTPUT PORT SIGNALS



4.3.2.8 Receive Cell Processor Interrupts

The Receive Cell Processor will generate interrupts upon

- HEC Errors
- OAM Cell received
- Loss of Cell Delineation

If one of these conditions occur, and if that particular condition is enabled for interrupt generation, then when the local µC/µP reads the UNI Interrupt Status Register, as shown below, it should read 'xx1xxxxxb' (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
RxDS3 Interrupt Status	RxPLCP Interrupt Status	RxCP Interrupt Status	RxUTOPIA Interrupt Status	TxUTOPIA Interrupt Status	TxCP Interrupt Status	TxDS3 Interrupt Status	One Sec Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RUR
x	x	1	x	x	x	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 "RxCP Interrupt Status Register" (Address = 4Fh). The bit format of this register is presented below.

RxCP Interrupt Status Register (Address = 4Ch)

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 RxCP 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 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 RxLCD bit-field (bit 7) in the RxCP Configuration Register (Address = 4Ch).

Bit 2—Received OAM Cell Interrupt Status

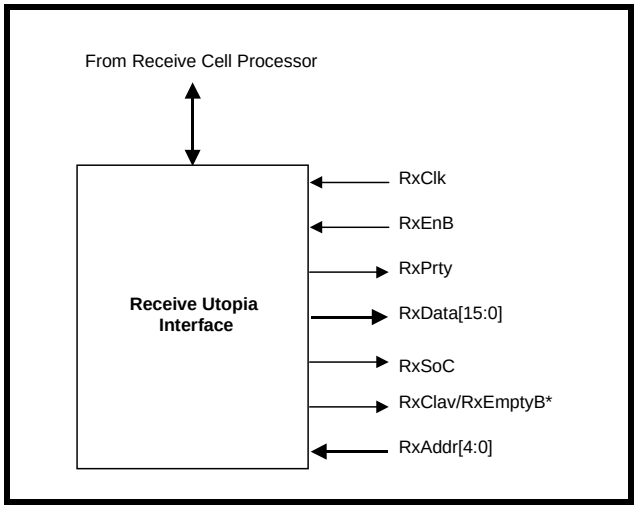
A “1” in this “Reset-upon-Read” bit-field indicates that the Receive Cell Processor has detected an OAM Cell in the path of “incoming cells”; and has stored the contents of this OAM cell in the “Receive OAM Cell Buffer”, since the last read of this register. The purpose of this interrupt is to alert the local µP/µC that the “Receive OAM Cell Buffer” (within the UNI) contains an OAM cell that needs to be read and processed.

4.4 Receive UTOPIA Interface Block

4.4.1 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 RxFIFO throughout this document. The Receive Cell Processor will delineate, check for HEC byte errors, filter and descramble ATM Cells. Whatever cells were not discarded by the Receive Cell Processor will be written into the RxFIFO, where it can be read out from the UNI device by the ATM Layer Processor. The Receive UTOPIA Interface Block will inform the ATM Layer processor that it has cell data available for reading, by asserting the RxUClav pin “high”.
Figure 35 on the following page presents a simple illustration of the Receive UTOPIA Interface block and the associated pins.

FIGURE 35. SIMPLE BLOCK DIAGRAM OF RECEIVE UTOPIA BLOCK OF UNI.



4.4.2 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 RxUSoC (Receive Start of Cell) pin each time the first byte (or word) of a new cell is present on the Receive UTOPIA Data Bus.

The Receive UTOPIA Interface Block consists of the following sub-blocks:

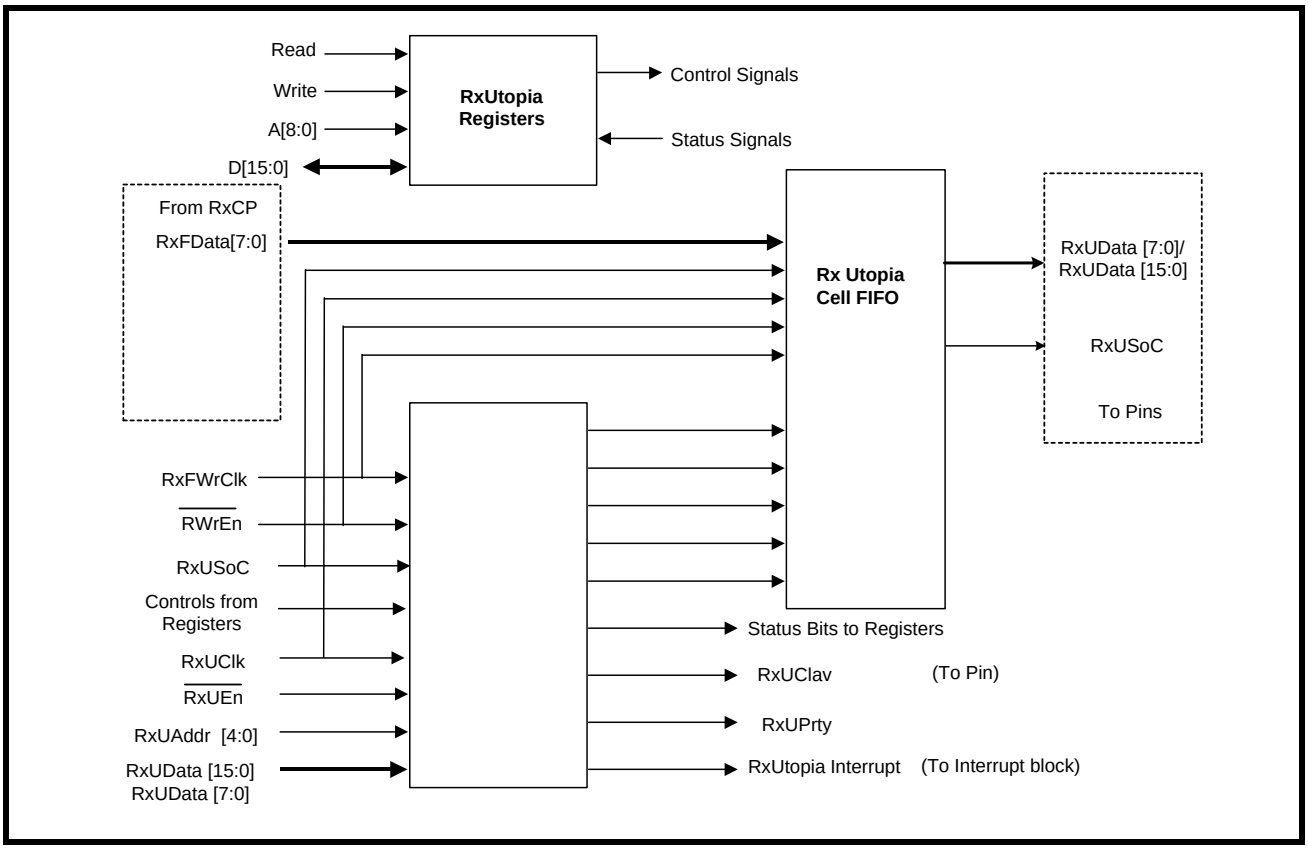
- Receive UTOPIA Output Interface
- Receive UTOPIA Cell FIFO (RxFIFO)
- Receive UTOPIA FIFO Manager

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 RxUPrty pin, while the corresponding data byte (word) is present at the RxUData[15:0] output pins.

The Receive UTOPIA Interface block can be configured to process 52, 53, and 54 bytes per cell; and will assert the RxUSoC (Receive “Start of Cell”) output pin at the cell boundaries. If the Receive UTOPIA Interface block detects a “runt” cell (e.g., a cell that is smaller than what the Receive UTOPIA Interface block has been configured to handle), it will generate an interrupt to the local µP, discard this “runt” cell, and resume normal operation.

The physical size of the RxFIFO is four cells. The incoming data (from the Receive Cell Processor) is written into the RxFIFO, where it can be read in and processed by the ATM Layer Processor. A FIFO Manager maintains the RxFIFO and indicates the FIFO Empty and FIFO Full to the local µP. Additionally the FIFO Manager will indicate that ATM Cell Data is available in the RxFIFO, by asserting the RxUClav output pin. Figure 36 presents a Functional Block Diagram of the Receive UTOPIA Interface Block.

FIGURE 36. FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE UTOPIA INTERFACE BLOCK



The following sections discuss each functional sub-block of the Receive UTOPIA Interface block in detail. Additionally, these sections discuss many of the features associated with the Receive UTOPIA Interface block as well as how these features can be optimized to suit selected 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.

4.4.2.1 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 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.

4.4.2.1.1 The Pins of the Receive UTOPIA Bus Interface

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

- RxUData[15:0]—Receive UTOPIA Data Bus output pins.
- RxUAddr[4:0]—Receive UTOPIA Address Bus input pins.
- RxUClk—Receive UTOPIA Interface Block clock input pin.
- RxUSoC—Receive “Start of Cell” Indicator output pin.
- RxUPrty—Receive UTOPIA—Odd Parity output pin.
- $\overline{\text{RxUEn}}$—Receive UTOPIA Data Bus—Output Enable input pin.
- RxUClav/RxFullB*—RxFIFO Cell Available output pin.

Each of these signals are discussed below.

RxUData[15:0]—Receive UTOPIA Data Bus Outputs

The ATM Layer Processor will read ATM cell data from the Receive UTOPIA Interface block in a byte-wide (or 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 7.4.2.1.2). If the "8-bit wide" mode is selected, then only the RxUData[7:0] output pins will be active and capable of transmitting data. If the 16-bit wide mode is selected, then all 16 output pins (e.g., RxUData[15:0]) will be active. The Receive UTOPIA Data bus is tri-stated while the active low RxUEn (Receive UTOPIA Bus—Output Enable) input signal is "high". Therefore, the ATM Layer Processor must assert this signal (e.g., toggle RxUEn low) in order to read the ATM cell data from the Receive UTOPIA Interface block. The data on the Receive UTOPIA Data Bus output pins are updated on the rising edge of the Receive UTOPIA Interface block clock signal, RxUClk.

RxUAddr[4:0]—Receive UTOPIA Address Bus Inputs

These input pins are used only when the UNI is operating in the Multi-PHY mode. Therefore, for more information on the Receive UTOPIA Address Bus, please see Section 7.4.2.2.2.2.

RxUClk—Receive UTOPIA Interface Block—Clock Signal Input Pin

The Receive UTOPIA Interface block uses this signal to update the data on the Receive UTOPIA Data Bus. The Receive UTOPIA Interface block also uses this signal to sample and latch the data on the Receive UTOPIA Address bus pins (during Multi-PHY operation), into the Receive UTOPIA Interface block circuitry. This clock signal can run at frequencies of 25 MHz, 33 MHz, or 50 MHz.

$\overline{\text{RxUEn}}$—Receive UTOPIA Data Bus—Output Enable Input

The Receive UTOPIA Data bus is tri-stated while this input signal is negated. Therefore, the user must assert this "active-low" signal (toggle it "low") in order to read the byte (or word) from the Receive UTOPIA Interface block via the Receive UTOPIA Data bus.

RxUPrty—Receive UTOPIA—Odd Parity Bit Output Pin

The Receive UTOPIA Interface Block will compute the odd-parity of each byte (or word) of ATM cell data that it will place on the Receive UTOPIA Data bus. The Receive UTOPIA Data bus will output the value of the computed parity bit at the RxUPrty output pin, while the corresponding byte (or word) is present on the Receive UTOPIA Data Bus. This features allows the ATM Layer Processor to perform parity checking on the data that it receives from the Receive UTOPIA Interface Block.

RxUSoC—Receive UTOPIA—"Start of Cell" Indicator Output Pin

The Receive UTOPIA Interface block will pulse this output signal "high", for one clock period of RxUClk, when the first byte (or word) of a new cell is present on the Receive UTOPIA Data Bus. This signal will be "low" at all other times.

RxUClav/RxEmptyB*—RxFIFO Cell Available/ RxEmpty*

This output signal is used to alert the ATM Layer Processor that the RxFIFO contains some ATM cell data that is available for reading. Please see Section 7.4.2.2.1 for more information regarding this signal.

4.4.2.1.2 Selecting the UTOPIA Data Bus Width

The UTOPIA data bus width can be selected to be either 8 or 16 bits by writing the appropriate data into the UtWidth16 bit (bit 0) within the UTOPIA Configuration Register, as shown below.

UTOPIA Configuration Register: (Address = 6Ah)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[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 Receive UTOPIA Data output pins: RxUData[7:0] will be active. (The output pins:RxUData[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: RxUData[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 28: 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 affects the width of the Transmit UTOPIA Data bus.*
2. *The UTOPIA Data Bus width will be 8 bits, upon power up or reset. Therefore, a "1" must be written to this bit in order to set the width of the Receive UTOPIA (and the Transmit UTOPIA data bus) to 16 bits.*

4.4.2.1.3 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 following cell size options are available.

- If the UTOPIA Data Bus is 8 bits wide 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 is 16 bits wide 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 selection is made by writing the appropriate data to bit 3 (CellOf52Bytes) within the UTOPIA Configuration Register, as depicted below.

UTOPIA Configuration Register: (Address = 6Ah)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[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 Receive UTOPIA Interface block will process.

TABLE 29: 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 wide. 54 bytes when the UTOPIA Data Bus width is 16 bits wide.
1	52 bytes, regardless of the 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

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 RxUSoC output pin. If the ATM Layer processor continues to try to read-in more octets, it will end up reading in in-valid data.

4.4.2.1.4 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 RxUPrty pin. The RxUPrty 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.

4.4.2.2 Receive UTOPIA FIFO Manager

The RxFIFO 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.
- Generating the “RxFIFO Underrun Condition” interrupt to the local µP, when the RxFIFO has been depleted of ATM cell data.

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 these features can be optimized to suit particlar application needs.

The Receive UTOPIA FIFO Manager provides the following options.

- Handshaking Mode (Octet Level vs Cell Level)
- Resetting the RxFIFO
- Monitoring the RxFIFO

4.4.2.2.1 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.

4.4.2.2.1.1 Octet-Level Handshaking

The UNI will be operating in the Cell-Level Handshaking Mode following power up or reset. Therefore, bit 5 (Handshake Mode) within the UTOPIA Configuration Register to must be set to “0” in order to configure the UNI into “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 RxUClav output pin.

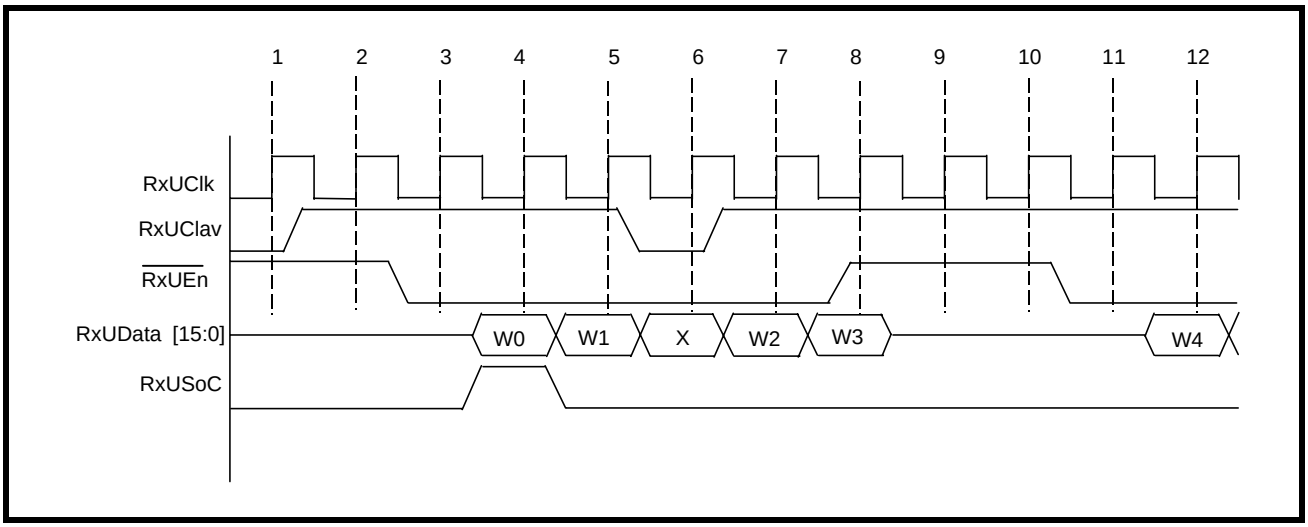
When the UNI is operating in the Octet-Level Handshake mode, the Receive UTOPIA Interface block will assert the RxUClav output pin, when the RxFIFO contains at least one “read cycle’s” worth of ATM Cell Data. In other words, if the UTOPIA Data bus width is configured to be 16 bits wide, then the RxUClav 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 RxUClav signal will be asserted when the RxFIFO contains at least one byte of ATM cell data. The Receive UTOPIA Interface block will negate RxUClav when the RxFIFO has been depleted of any data. Therefore, the RxUClav 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 RxUClav 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 RxUClav and read out the cell data in order to avoid an “Overrun Condition” in the RxFIFO. Finally, the ATM Layer processor is expected to do one of two things, whenever RxUClav 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 RxUClav 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 RxUClav has toggled “low”.

Figure 37 presents a timing diagram illustrating the behavior of the RxUClav pin during reads from the Receive UTOPIA Interface block, while operating in the Octet-Level Handshaking Mode.

FIGURE 37. TIMING DIAGRAM OF RXUCLAV/RXEMPTYB AND VARIOUS OTHER SIGNALS DURING READS FROM THE RECEIVE UTOPIA, WHILE OPERATING IN THE OCTET-LEVEL HANDSHAKING MODE.



Note: regarding Figure 37

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 37 illustrates the ATM Layer processor reading 27 words (W0 through W26) for each ATM cell.*

In Figure 37 , RxUClav 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 RxUClav 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 RxUClav” during clock edge #2. Consequently, in order to begin reading this cell data, the ATM Layer processor will then assert the RxUEn input pin. At clock edge #3, the Receive UTOPIA Interface block detects RxUEn 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 RxUClav to toggle “low”. In this figure, the ATM Layer processor will keep the RxUEn signal asserted, and will read in an “invalid” word which is denoted by the “X” in Figure 37 . Shortly thereafter, the RxFIFO receives some additional cell data from the Receive Cell Processor, which in turn causes RxUClav 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 RxUEn signal; at clock edge #8. The Receive UTOPIA Interface block detects that RxUEn 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 RxUEn (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.

4.4.2.2.1.2 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 RxUClav output is at a logic “1”, it means that the RxFIFO contains at least one complete ATM cell of data that is available for reading by the ATM Layer Processor. When RxUClav toggles from “high” to “low”, it indicates that RxFIFO contains less than one complete ATM cell. As in the “Octet-Level” Handshake mode, the ATM Layer processor is expected to monitor the RxUClav output, and quickly respond and read the RxFIFO, whenever the RxUClav 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 7.4.2.2.2.

The UNI can be configured 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 = 6Ah

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Handshake Mode	M-PHY	CellOf52 Bytes	TFIFODepth[1, 0]		UtWidth16
R/W		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 30: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 5 (HANDSHAKE MODE) WITHIN THE UTOPIA CONFIGURATION REGISTER, AND THE RESULTING UTOPIA INTERFACE HANDSHAKE MODE

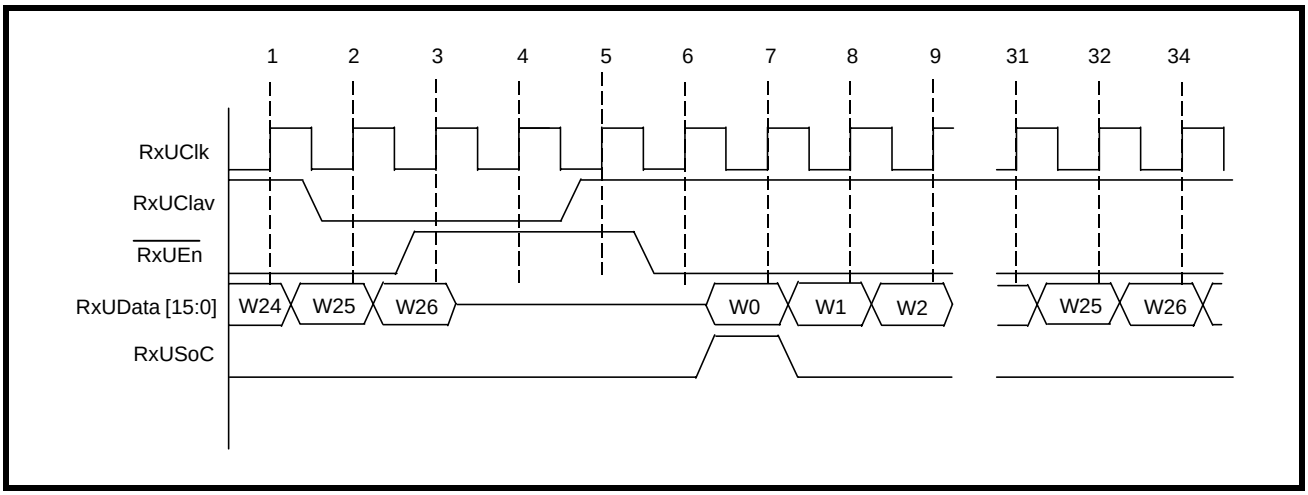
VALUE	RESULTING HANDSHAKE MODE
0	The UTOPIA Interfaces operate in the cell level handshake mode.
1	The UTOPIA Interfaces operate in the octet level handshake mode.

Note:

1. *The Handshake Mode selection applies to both the Transmit UTOPIA and Receive UTOPIA Interface blocks.*
2. *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, a "0" must be written to this bit in order to configure "Octet- Level Handshaking, mode.*

Figure 38 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 38. TIMING DIAGRAM OF VARIOUS RECEIVE UTOPIA INTERFACE BLOCK SIGNALS, WHEN THE RECEIVE UTOPIA INTERFACE BLOCK IS OPERATING IN THE "CELL LEVEL" HANDSHAKE MODE



Note: regarding Figure 39

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 39 illus-*

trates the ATM Layer processor reading in 27 words (W0 through W26) for each ATM cell.

In Figure 39 , the ATM Layer processor is just finishing up its reading of an ATM cell. Prior to clock edge #2, the RxFIFO does not contain enough ATM cell data to make up at least one cell. Hence, the Receive UTOPIA Interface block negates the RxUClav signal. The ATM Layer processor detects that the RxUClav 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 RxUEn signal and will cease to read in anymore ATM cell data from the Receive UTOPIA Interface block; until RxUClav toggles "high" again.

The RxFIFO accumulates enough cell data to make up a complete ATM cell shortly before clock edge #5. At this point the Receive UTOPIA Interface block reflects this fact by asserting the RxUClav signal. The ATM Layer processor detects that the RxUClav signal has toggled "high" at clock edge #5. Consequently, the ATM Layer processor then asserts the RxUEn signal (e.g., toggles it "low") after clock edge #5. The Receive UTOPIA Interface block detects the fact that the RxUEn 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.

4.4.2.2.1.3 Resetting the RxFIFO via Software Command

The UNI allows for reseting the RxFIFO, 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 (RxFIFO Reset) of the Receive UTOPIA Interrupt Enable/Status Register as depicted below.

Receive UTOPIA—Interrupt/Status Register (Address—6Bh)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxFIFO Reset	RxFIFO Overrun Interrupt Enable	RxFIFO Underrun Interrupt Enable	RCOCA Interrupt Enable	RxFIFO Overrun Interrupt Enable	RxFIFO Underrun Interrupt Enable	RxFIFO COCA Int. Status
R/O	R/W	R/W	R/W	R/W	RUR	RUR	RUR

Once the RxFIFO has been reset, then the contents of the RxFIFO will be "flushed" and the Receive FIFO Status register will reflect the "RxFIFO Empty" status.

4.4.2.2.1.4 Monitoring the RxFIFO Status

The local µP has the ability to poll and monitor the status of the RxFIFO via the Receive UTOPIA FIFO Status Register. The bit format of this register is presented below.

Receive UTOPIA FIFO Status Register (Address = 6Dh)

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	RxFIFO is not full.
1	RxFIFO is full, and if the next operation by the ATM Layer processor is not a read operation, then the RxFIFO could be overrun.

RxFIFO Empty

RxFIFO Empty (Bit 0)	Meaning
0	RxFIFO is not empty
1	RxFIFO is empty.

4.4.2.2.2 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.

4.4.2.2.3 Single PHY Operation

The UNI chip will be operating in the Multi-PHY mode upon power up or reset. Therefore, a “1” must be written into Bit 4 of the UTOPIA Configuration Register as depicted below in order to configure the UNI into the single-PHY Mode.

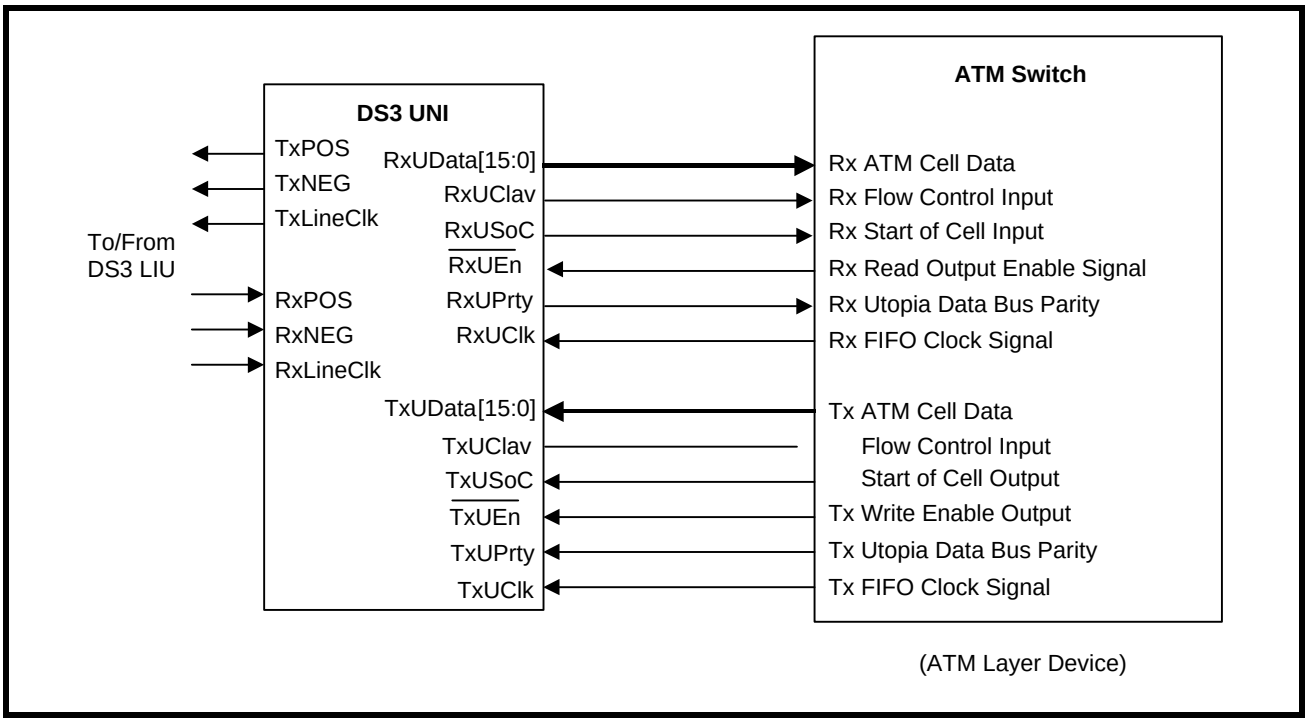
UTOPIA Configuration Register: Address = 6Ah

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	TFIFODepth [1, 0]		UtWidth16
RO		R/W	R/W	R/W	R/W		R/W
xx		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 39 . ATM Cell data is read from the RxFIFO, via the Receive UTOPIA Data Bus, provided that the Receive UTOPIA Output enable signal ($\overline{\text{RxUEn}}$) is low. The data on the Receive UTOPIA Data bus is updated on the rising edge of the Receive UTOPIA clock (RxUClk). The Receive UTOPIA Interface block will pulse the Receive start of cell signal (RxUSoC) 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 RxUPrty pin.

FIGURE 39. 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 RxUClav pin

If the RxUClav pin is "high" then the RxFIFO contains some ATM cell data that needs to be read by the ATM Layer processor. In this case, the ATM Layer processor should begin to read the cell data from the Receive UTOPIA Interface block. However, if the RxUClav pin is "low", then the RxFIFO does not contain any cell data that can be read. In this case, the ATM Layer processor should wait until RxUClav toggles "high" before attempting to read any more cell data from the "Receive UTOPIA Interface block".

Note: The actual meaning associated with RxUClav toggling "high" or "low" depends upon whether the UNI is operating in the "Cell Level" or "Octet Level" handshake modes.

2. Assert the $\overline{\text{RxUEn}}$ pin and read the first byte (or word) of the new cell from the Receive UTOPIA Data Bus.

Once the ATM Layer processor has detected that RxUClav has toggled "high", then it should assert the $\overline{\text{RxUEn}}$ input pin (e.g., toggling it "low"). Once the Receive UTOPIA Interface block has determined that the $\overline{\text{RxUEn}}$ input pin is "low", then it will begin to place some cell data onto the Receive UTOPIA Data Bus. If this first byte (or word) is the beginning of a new ATM cell, then the ATM Layer processor should verify that this byte (or word) is indeed the beginning of a new cell, by observing the RxUSoC output pin (of the UNI IC) pulsing "high" for one clock period of RxUClk.

3. Compute the odd-parity of the byte (or word) that is being read from the Receive UTOPIA Data bus, and compare the value of this parity bit with that of the RxUPrty output pin.

This operation is optional, but should be done concurrently while checking for the assertion of the RxUSoc output pin.

When reading in the subsequent bytes (or words) of the cell, the ATM Layer must do the following.

- Repeat Steps 1 and 2.
- If the UNI is operating in the Octet-Level Handshake mode, then the ATM Layer processor should check the RxUClav level prior to asserting the $\overline{\text{RxUEn}}$ (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 RxUClav signal is "high".
- If the UNI is operating in the Cell-Level Handshake mode, then the ATM Layer processor should check the RxUClav signal level just as it (the ATM Layer processor) is reading in the very last byte (or word) of a given cell. If the RxUClav level is "high", then

the ATM Layer processor should proceed to read in the next cell from the Receive UTOPIA Interface block. However, if the RxUClav level is “low”, then the ATM Layer processor should halt reading in data, when it reaches the end of the cell (that it is currently reading in).

- The ATM Layer processor should keep a count on the total number of bytes that have been read in since the last assertion of the RxUSoC output pin. This will help the ATM Layer processor to determine when it has reached the boundary of a given cell.

The previously-mentioned procedure is also depicted in “Flow Chart Form” in Figure 40 , and in Timing Diagram form in Figure 41 and 42.

FIGURE 40. 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.

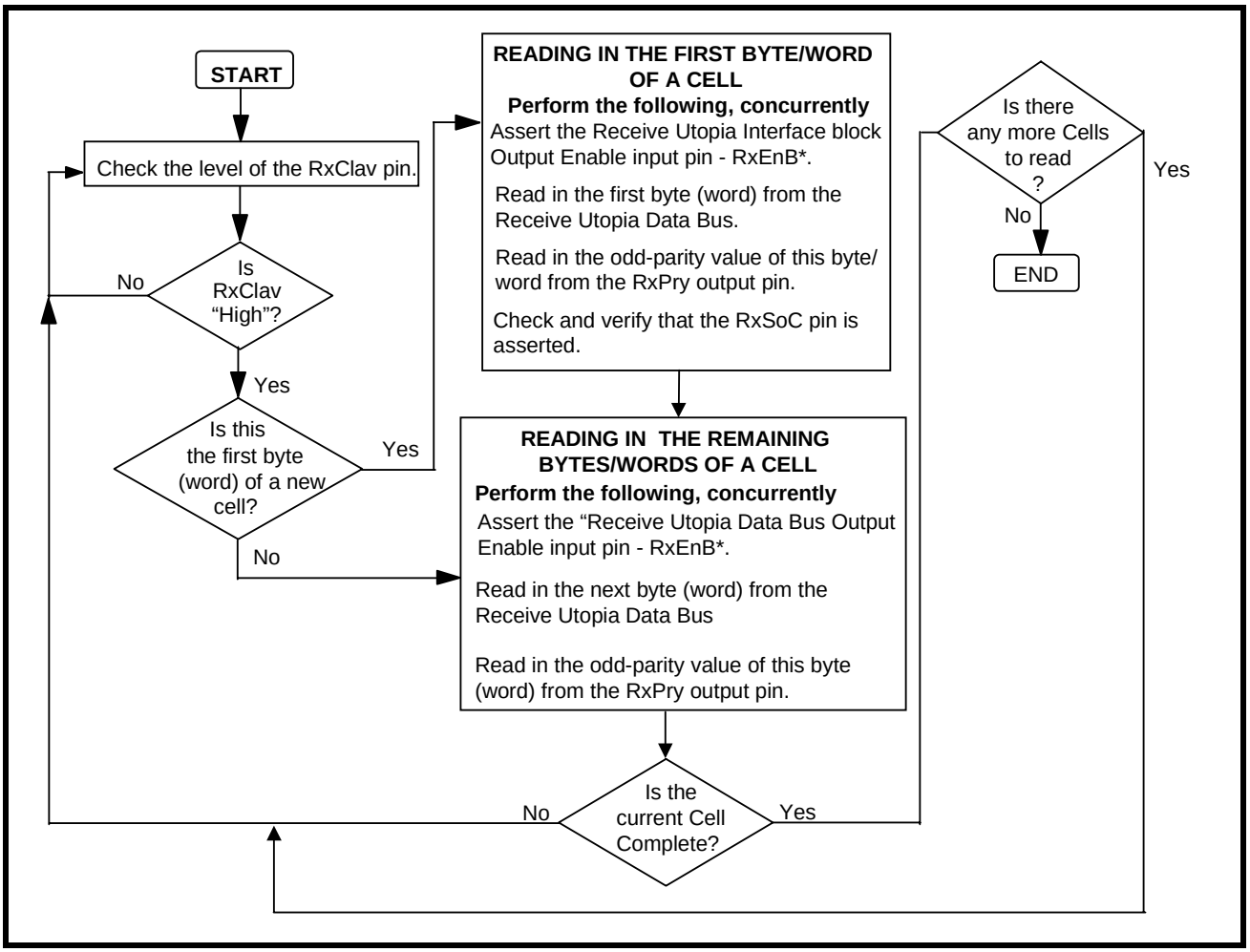
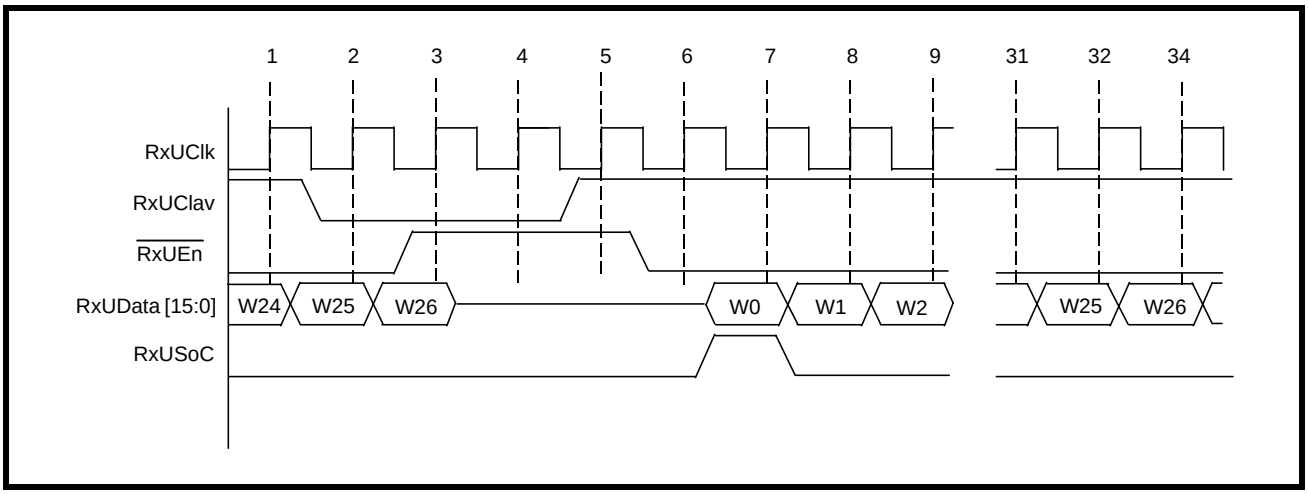


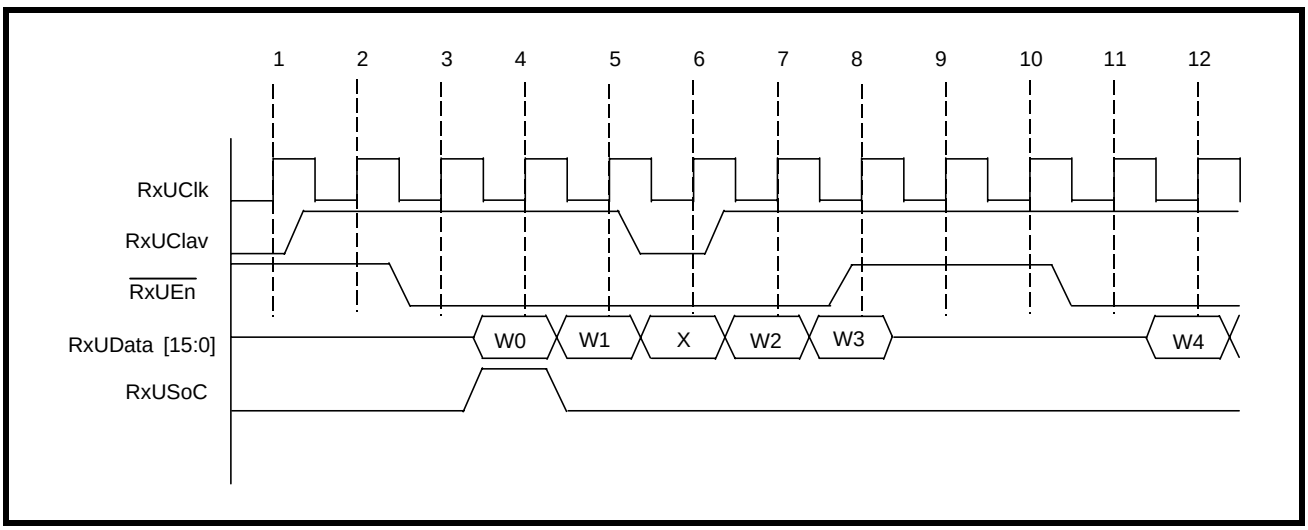
FIGURE 41. TIMING DIAGRAM OF ATM LAYER PROCESSOR RECEIVING DATA FROM THE UNI OVER THE UTOPIA DATA BUS, (SINGLE-PHY MODE/CELL LEVEL HANDSHAKING).



Note: regarding Figure 41

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 41 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 42. TIMING DIAGRAM OF ATM LAYER PROCESSOR RECEIVING DATA FROM THE UNI OVER THE UTOPIA DATA BUS, (SINGLE-PHY MODE/OCTET LEVEL HANDSHAKING).



Note: regarding Figure 42

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 42 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 RxUClav pin exhibits a role that is similar to the "Ready Ready" function in RS-232 based data com-

munication. This pin is asserted when the RxFIFO contains ATM cell data that can be read by the ATM Layer processor. The RxUClav 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 RxUClav pin before attempting to read in the next byte, word or cell from the RxFIFO.

4.4.2.2.3.1 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 "RxUTOPIA Address Register" (Address = 6Ch) within its "host" UNI device. The bit-format of the "RxUTOPIA Address Register" is presented below.

Receive UTOPIA Address Register: (Address = 6Ch)

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 "TxUTOPIA Address Register" (Address = 70h) within the "host" UNI device. The bit-format of the "TxUTOPIA Address Register" is presented below.

Tx UTOPIA Address Register (Address = 70h)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxUTOPIA_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 1.1.2.3.2.

4.4.2.2.3.1.1 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 43. AN ILLUSTRATION OF MULTI-PHY OPERATION WITH UNI DEVICES #1 AND #2

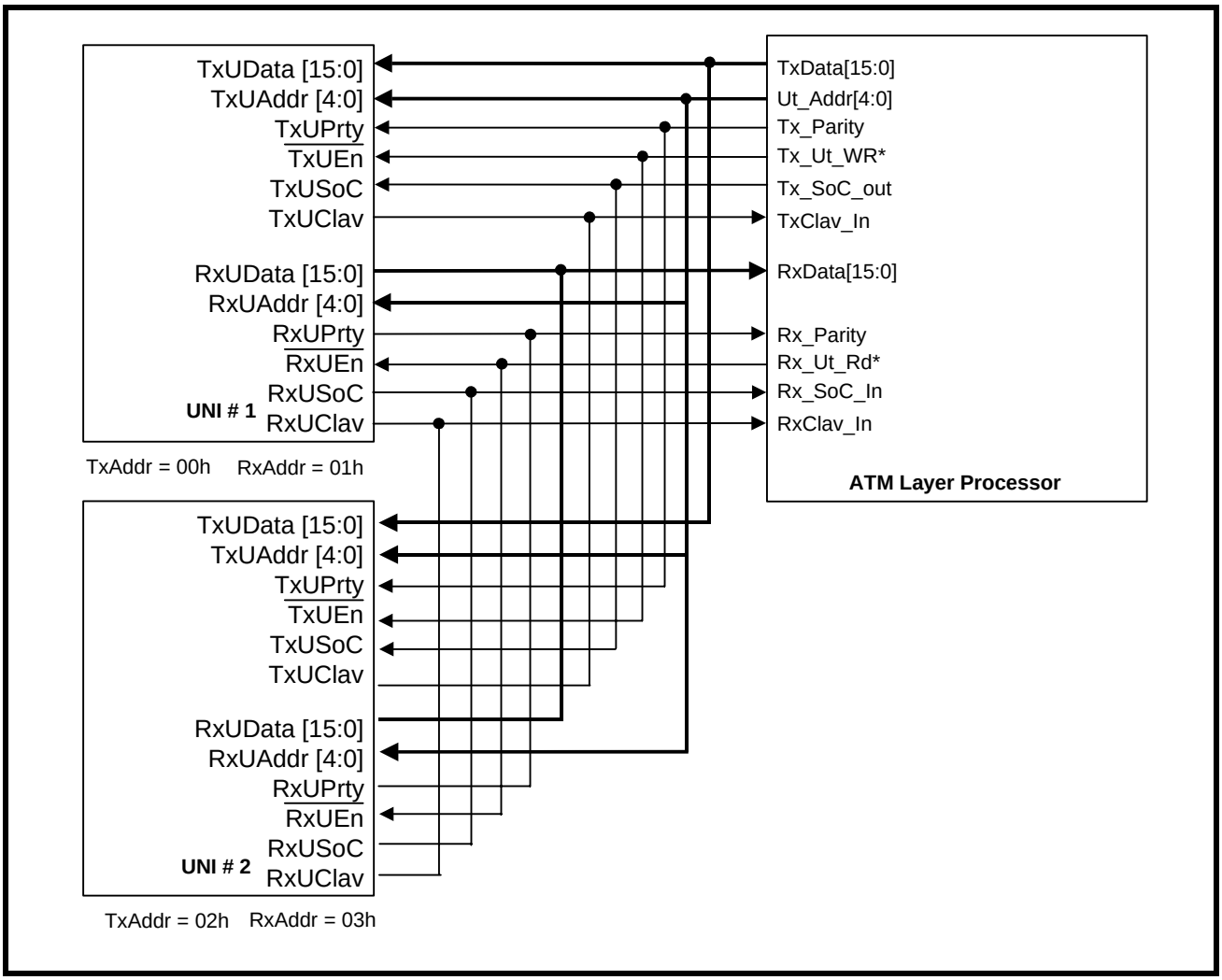


Figure 43 depicts a “Multi-PHY” system consisting of an ATM Layer processor and two (2) UNI devices, designated as “UNI #1” and “UNI #2”. In this figure, both of the UNIs are connected to the ATM Layer processor via a common “Transmit UTOPIA” Data Bus, “Receive UTOPIA” Data Bus, a common TxUClav line, a common RxUClav line, as well as common TxUEn, RxUEn, TxUSoC and RxUSoC 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 42 .

The UTOPIA Address values that have been assigned to each of the Transmit and Receive UTOPIA Interface blocks within Figure 42 , are listed below in Table 31 .

TABLE 31: UTOPIA ADDRESS VALUES OF THE UTOPIA INTERFACE BLOCKS ILLUSTRATED IN FIGURE 43

BLOCK	UTOPIA ADDRESS VALUE
Transmit UTOPIA Interface block—UNI #1	00h
Receive UTOPIA Interface block—UNI #1	01h

TABLE 31: UTOPIA ADDRESS VALUES OF THE UTOPIA INTERFACE BLOCKS ILLUSTRATED IN FIGURE 43

BLOCK	UTOPIA ADDRESS VALUE
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 "RxUTOPIA Address Register" (Address = 6Ch) 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 $\overline{\text{RxUEn}}$ input pin (if it not asserted already).

The UNI device (being "polled") will know that this is only a "polling" operation, if the $\overline{\text{RxUEn}}$ 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 "RxUClav_in" input pin (see Figure 42).

The UNI Device's Role in the "Polling" Operation

UNI #2 will sample the signal levels placed on its Rx UTOPIA Address input pins (RxUAddr[4:0]) on the rising edge of its "Receive UTOPIA Interface block" clock input signal, RxUClk. Afterwards, UNI #2 will compare the value of these "Receive UTOPIA Address Bus input pin" signals with that of the contents of its "RxUTOPIA Address Register" (Address = 6Ch).

If these values do not match (e.g., RxUAddr[4:0] µ 03h) then UNI #2 will keep its "RxUClav" output signal "tri-stated"; and will continue to sample its "Receive UTOPIA Address bus input" pins, with each rising edge of RxUClk.

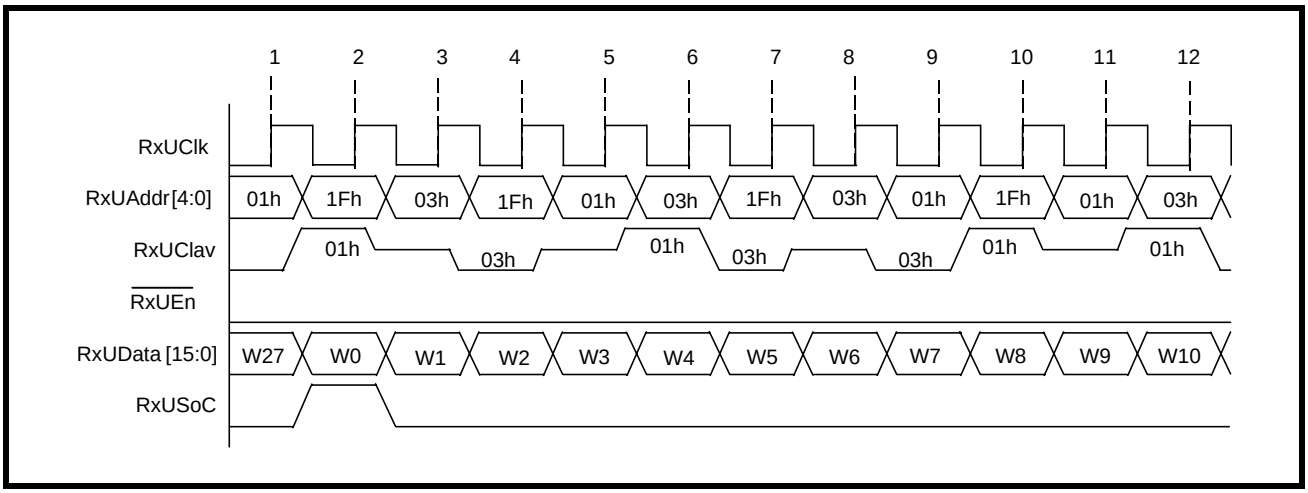
If these two values do match (e.g., RxUAddr[4:0] = 03h) then UNI #2 will drive its "RxUClav" output pin to the appropriate level, reflecting its RxFIFO "fill status". Since the UNI is automatically operating in the "Cell Level Handshaking" mode, while it is operating in the "Multi-PHY" mode, the UNI will drive the RxUClav 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 "RxUClav" 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 "RxUClav" output signal "tri-stated". Therefore, when UNI #2 is driving its "RxUClav" output pin to the appropriate level; it will be driving the entire "RxUClav" line, within the "Multi-PHY" system. Consequently, UNI#1 will also be driving the "RxUClav_in" input pin of the ATM Layer processor (see Figure 43).

If UNI #2 drives the "RxUClav" line "low", upon the application of its address on the UTOPIA Address bus, then the ATM Layer processor will "learn" that UNI #2 does not contain any ATM cell data that is ready to be read. However, if UNI #2 drives the RxUClav line "high" (during "polling"), then the ATM Layer processor will know that UNI#2 contains at least one cell of data that needs to be read.

Figure 44 presents a timing diagram, that depicts the behavior of the ATM Layer processor's and the UNI's signals during polling.

FIGURE 44. TIMING DIAGRAM ILLUSTRATING THE BEHAVIOR OF VARIOUS SIGNALS FROM THE ATM LAYER PROCESSOR AND THE UNI, DURING POLLING.



Note: regarding Figure 44

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 44 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 (RxUAddr[4:0] = 01h) during this "polling process".*
4. *The RxFIFO, within UNI#2's Receive UTOPIA Interface block (RxUAddr[4:0] = 03h) is either depleted or does not contain enough data to constitute a complete ATM cell. Hence, the RxUClav line will be driven "low" whenever this particular Receive UTOPIA Interface block is "polled".*
5. *The Receive UTOPIA Address of 1Fh is not associated with any UNI device, within this "Multi-PHY" system. Hence, the RxUClav line is tri-stated whenever this address is "polled".*

Note: Although Figure 44 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.

4.4.2.2.3.1.2 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 $\overline{\text{RxUEn}}$ 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 $\overline{\text{RxUEn}}$ signal.
4. Check and insure that the RxUSoC output pin (of the selected UNI) pulses "high" when the first byte or word of ATM cell data has been placed on the Receive UTOPIA Data Bus.
5. Begin reading the ATM Cell data in a byte-wide (or word-wide) manner from the Receive UTOPIA Data bus.

Figure 45 presents a flow-chart that depicts the "UNI Device Selection and Read" process in Multi-PHY operation.

FIGURE 45. FLOW-CHART OF THE "UNI DEVICE SELECTION AND READ PROCEDURE" FOR THE MULTI-PHY OPERATION.

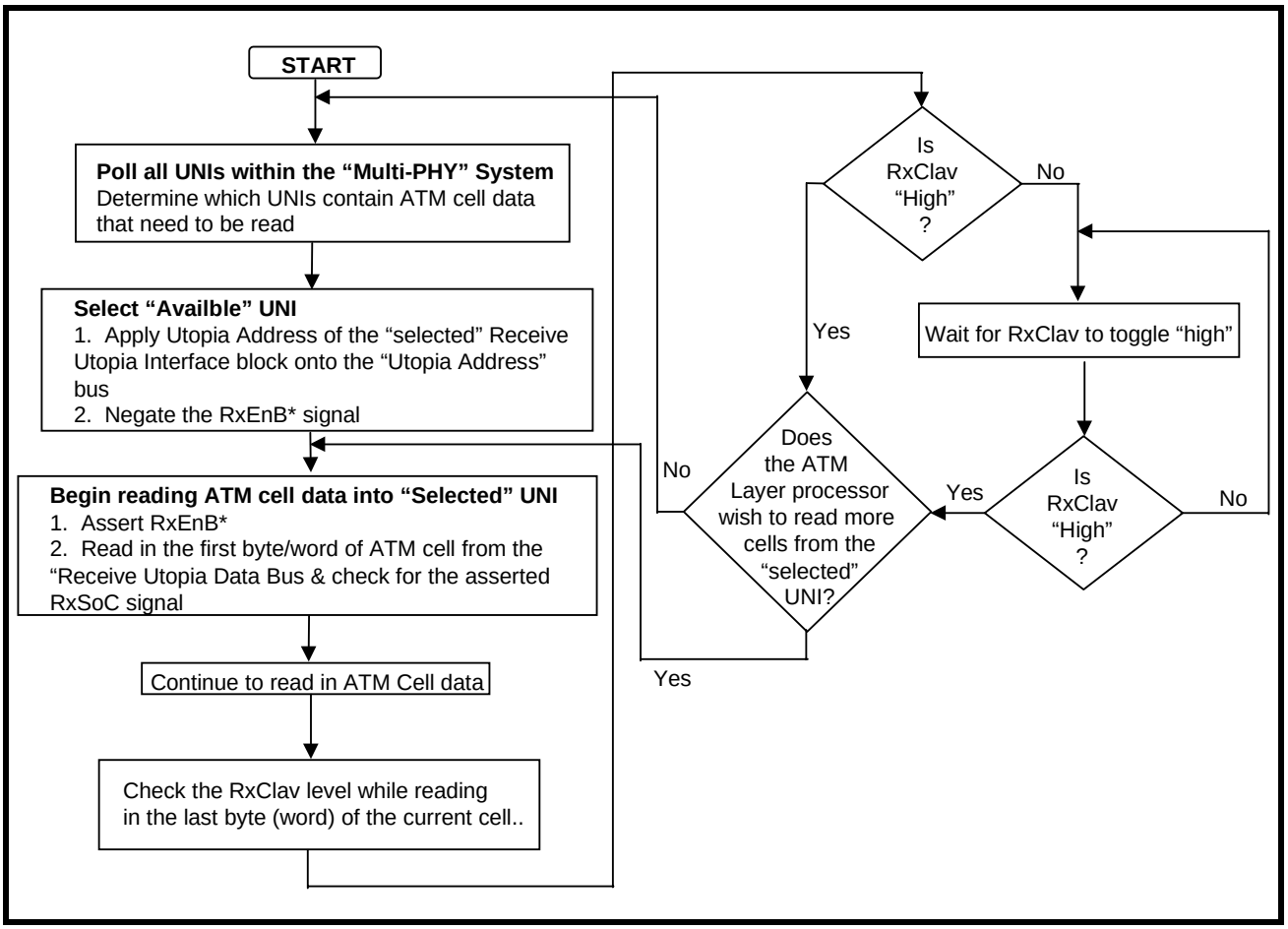
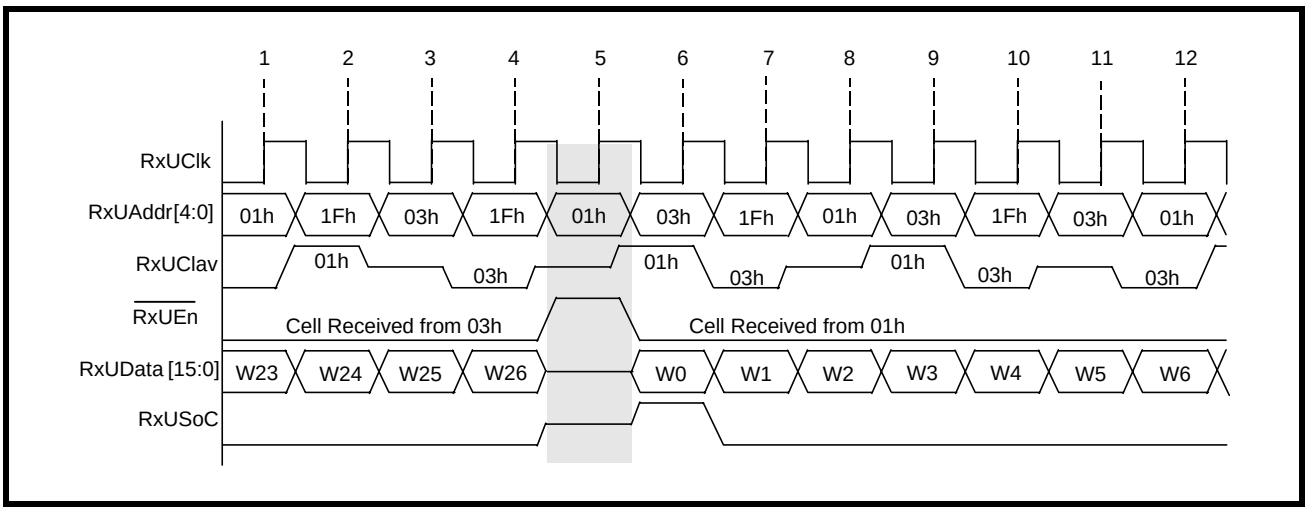


Figure 46 presents a timing diagram that illustrates the behavior of various "Receive UTOPIA Interface block" signals, during the "Multi-PHY" UNI Device Selection and Read operation.

FIGURE 46. TIMING DIAGRAM OF THE RECEIVE UTOPIA DATA AND ADDRESS BUS SIGNALS, DURING THE "MULTI-PHY" UNI DEVICE SELECTION AND WRITE OPERATIONS.



Note: regarding Figure 46

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 46 illustrates the ATM Layer processor reading 27 words (e.g., W0 through W26) for each ATM cell.*

In Figure 46 , the ATM Layer processor is initially reading ATM cell data from the Receive UTOPIA Interface within UNI #2 (RxUAddr[4:0] = 03h). However, the ATM Layer processor is also polling the Receive UTOPIA Interface block within UNI #1 (RxUAddr[4:0] = 01h) and some "non-existent" device at RxUAddr[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 (RxUAddr[4:0] = 03h). The ATM Layer processor will indicates its intention to select a new UNI device for reading by negating the $\overline{\text{RxUEn}}$ signal, at clock edge #5 (see the shaded portion of Figure 46). At this time, UNI #1 will notice two things:

1. The UTOPIA Address for the Receive UTOPIA Interface block, within UNI #1 is on the Receive UTOPIA Address bus (RxUAddr[4:0] = 01h).
2. The $\overline{\text{RxUEn}}$ 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.

4.4.2.3 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)
- RxFIFO Overrun
- RxFIFO Underrun

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 "xxx1xxxxb" (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
RxDS3 Interrupt Status	RxPLCP Interrupt Status	RxCP Interrupt Status	RxUTOPIA Interrupt Status	TxUTOPIA Interrupt Status	TxCP Interrupt Status	TxDS3 Interrupt Status	One Sec Interrupt Status
RO	RO	RO	RO	RO	RO	RO	RUR
x	x	x	x	1	x	x	x

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 three 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 "RxUT Interrupt Enable/Status Register, which is located at address 6Bh in the UNI device. The bit format of this register is presented below.

Address = 6Bh, RxUT 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	RxFIFO Underflw Interrupt Enable	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	RxFIFO Underflw Interrupt Status	RCOCA Interrupt Status
RO	R/W	R/W	R/W	R/W	RUR	RUR	RUR

The RxUT 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 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 = 6Bh, RxUT 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	RxFIFO Underflw Interrupt Enable	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	RxFIFO Underflw Interrupt Status	RCOCA Interrupt Status
RO	R/W	R/W	R/W	R/W	RUR	RUR	RUR
0	0	x	x	1	x	x	1

Bit 1—RxFIFO Underflw Interupt Status—RxFIFO Underrun Condition

Whenever the Receive UTOPIA Interface block sets its RxUClav signal to “high”, the ATM Layer processor will know that the RxFIFO has some ATM cell data that needs to be read. Hence, the ATM Layer processor will begin to read out this cell data. If the ATM Layer processor reads out all of the cell data and depletes the RxFIFO, then the UNI will generate an “RxFIFO Underrun” Interrupt. The Receive UTOPIA Interface block will indicate that it is generating this kind of interrupt by asserting Bit 1 (RxFIFO Underflw Interrupt Status) of the Receive UTOPIAn Interrupt Enable/Status Register, as depicted below.

Address = 6Bh, RxUT 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	RxFIFO Underflw Interrupt Enable	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	RxFIFO Underflw Interrupt Status	RCOCA Interrupt Status
RO	R/W	R/W	R/W	R/W	RUR	RUR	RUR
0	0	x	1	x	x	1	x

Bit 2—RxFIFO 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 = 6Bh, RxUT 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	RxFIFO Underflw Interrupt Enable	RCOCA Interrupt Enable	RxFIFO Overflw Interrupt Status	RxFIFO Underflw Interrupt Status	RCOCA Interrupt Status
RO	R/W	R/W	R/W	R/W	RUR	RUR	RUR
0	0	1	x	x	1	x	x

Bit 3—RCOCA Interrupt Enable—Receive UTOPIA Change of Cell Alignment Interrupt Enable

This “Read/Write” bit-field is used 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 4—RxFIFO Underflw Interrupt Enable—RxFIFO Underrun Condition Interrupt Enable

This “Read/Write” bit-field is used to enable or disable the generation of interrupts due to an “RxFIFO Underrun” 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.

Bit 5—RxFIFO Overflw Interrupt Enable—RxFIFO Overrun Condition Interrupt Enable

This “Read/Write” bit-field is used to enable or disable the generation of interrupts due to an “RxFIFO 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.

XRT74L74 CONFIGURATION

The XRT74L74 DS3/E3 Framer IC can be configured to support any of the following four framing formats.

- DS3/C-Bit Parity
- DS3/M13
- E3/ITU-T G.832
- E3/ITU-T G.751

As a consequence, the discussion of the XRT74L74 Framer IC will be organized as follows:

- **Section 4.0 - DS3 Mode Operation of the XRT74L74**
- **Section 5.0 - E3, ITU-T G.751 Operation of the XRT74L74**
- **Section 6.0 - E3, ITU-T G.832 Operation of the XRT74L74**
- **Section 7.0 - Framer Local Loop-back Mode Operation**
- **Section 8.0 - High Speed HDLC Controller Mode of Operation**

5.0 DS3 OPERATION OF THE XRT74L74

This section will discuss in detail, the operation of the XRT74L74 Framer IC, when it has been configured to operate in the DS3 Mode.

Configuring the XRT74L74 to Operate in the DS3 Mode

The XRT74L74 can be configured to operate in the DS3 Mode by writing a "1" into bit-field 6 within the Framer Operating Mode register, as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	1	x	0	x	x	x	x

Prior to describing the functional blocks within the Transmit and Receive Sections of the XRT74L74, it is important to describe the following two framing formats.

- M13
- C-Bit Parity

5.1 DESCRIPTION OF THE DS3 FRAMES AND ASSOCIATED OVERHEAD BITS

The role of the various overhead bits are best described by discussing the DS3 Frame Format as a whole. The DS3 Frame contains 4760 bits, of which 56 bits are overhead and the remaining 4704 bits are payload bits. The payload data is formatted into packets of 84 bits and the overhead (OH) bits are inserted between these payload packets. The XRT74L74 Framer supports the following two DS3 framing formats:

- C-bit Parity
- M13

Figures 47 and 48 present the DS3 Frame Format for C-bit Parity and M13, respectively.

FIGURE 47. DS3 FRAME FORMAT FOR C-BIT PARITY

X	I	F1	I	AIC	I	F0	I	NA	I	F0	I	FEAC	I	F1	I
								I							
X	I	F1	I	UDL	I	F0	I	NA	I	F0	I	UDL	I	F1	I

FIGURE 47. DS3 FRAME FORMAT FOR C-BIT PARITY

P	I	F1	I	CP	I	F0	I	CP	I	F0	I	CP	I	F1	I
P	I	F1	I	FEBE	I	F0	I	FEBE	I	F0	I	FEBE	I	F1	I
M0	I	F1	I	DL	I	F0	I	DL	I	F0	I	DL	I	F1	I
M1	I	F1	I	UDL	I	F0	I	UDL	I	F0	I	UDL	I	F1	I
M0	I	F1	I	UDL	I	F0	I	UDL	I	F0	I	UDL	I	F1	I

X = Signaling bit for network control
I = Payload Information (84 bit packets)
Fi = Frame synchronization bit with logic value i
P = Parity bit
Mi = Multiframe synchronization bit with logic value i
AIC = Application Identification Channel
NA = reserved for network application
FEAC = Far End Alarm and Control
DL = Data Link
CP = CP (Path)-bit parity
FEBE = Far End Block Error
UDL = User Data Link

FIGURE 48. DS3 FRAME FORMAT FOR M13

X	I	F1	I	C11	I	F0	I	C12	I	F0	I	C13	I	F1	I
								I							
X	I	F1	I	C21	I	F0	I	C22	I	F0	I	C23	I	F1	I
P	I	F1	I	C31	I	F0	I	C32	I	F0	I	C33	I	F1	I
P	I	F1	I	C41	I	F0	I	C42	I	F0	I	C43	I	F1	I
M0	I	F1	I	C51	I	F0	I	C52	I	F0	I	C53	I	F1	I
M1	I	F1	I	C61	I	F0	I	C62	I	F0	I	C63	I	F1	I
M0	I	F1	I	C71	I	F0	I	C72	I	F0	I	C73	I	F1	I

X = Signaling bit for network control
I = Payload Information (84 bit packets)
Fi = Frame synchronization bit with logic value i
Cij = jth stuff code bit of ith channel
P = Parity bit
Mi = multiframe synchronization bit with logic values i

To choose between these two frame formats, write the appropriate data to bit 2 of the Framer Operating Mode Register (Address = 0x00), as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
x	1	x	0	x	x	x	x

Table 32 lists the relationship between the value of the this bit-field and the resulting DS3 Frame Format.

TABLE 32: THE RELATIONSHIP BETWEEN THE CONTENT OF BIT 2, (C-BIT PARITY*/M13) WITHIN THE FRAMER OPERATING MODE REGISTER AND THE RESULTING DS3 FRAMING FORMAT

BIT 2	DS3 FRAME FORMAT
0	C-Bit Parity
1	M13

NOTE: *This bit setting also configures the frame format for both the Transmit and Receive Section of the XRT74L74.*

Each of the two DS3 Frame Formats, as presented in Figure 47 and Figure 48, constitute an M-frame (or a full DS3 Frame). Each M-frame consists of 7 - 680 bit F-frames (sometimes referred to as, subframes). In Figure 47 and 48, each F-frame is represented by the individual rows of payload and overhead bits. Each F-frame can be further divided into 8 blocks of 85 bits, with 84 of the 85 bits available for payload information and the remaining one bit used for frame overhead.

Differences Between the M13 and C-Bit Parity Frame Formats

The frame formats for M13 and C-bit Parity are very similar. However, the main difference between these two framing formats is in the use of the C-bits. In the M13 Format, the C-bits reflect the status of stuff-opportunities that either were or were not used while multiplexing the 7 DS2 signals into this DS3 signal. If two of the three stuff bits, within a F-frame, are "1", then the associated stuff bit, Si (not shown in Figure 48), is interpreted as being a stuff bit. In the C-bit Parity framing format, the C bits take on different roles, as presented in Table 33.

TABLE 33: C-BIT FUNCTIONS FOR THE C-BIT PARITY DS3 FRAME FORMAT

C - BIT	FUNCTION OF C-BITS WHILE IN THE C-BIT PARITY FRAMING FORMAT
C11	AIC (C-Bit Parity Mode)
C12	NA (Reserved for Network Application)
C13	FEAC (Far End Alarm & Control)
C21, C22, C23	(UDL) User Data Link (undefined for DS3 Frame)
C31,C32, C33	CP (Path) Parity Bits
C41, C42, C43	FEBE (Far End Block Error) Indicators
C51, C52, C53	(DL) Path Maintenance Data Link
C61, C62, C63, C71, C72, C73	(UDL) User Data Link (undefined for DS3 Frame)

Definition of the DS3 Frame Overhead Bits

In general, the DS3 Frame Overhead Bits serve the following three purposes:

1. Support Frame Synchronization between the Local and Remote DS3 Terminals
2. Provide parity bits in order to facilitate performance monitoring and error detection.
3. Support the transmission of Alarms, Status, and Data Link information to the Remote DS3 Terminal.

The Overhead bits supporting each of these purposes are further defined below.

5.1.1 Frame Synchronization Bits (Applies to both M13 and C-bit Parity Framing Formats)

Each DS3 Frame (M-frame) contains a total of 31 bits that support frame synchronization. Each DS3 M-frame contains three M-bits. According to Figure 47 and Figure 48, these M-bits are the first bits in F-frames 5, 6 and 7. These three bits appear in each M-frame with the repeating pattern of "010". This fact is also presented in Figure 47 and Figure 48, which contains bit-fields that are designated as: M0, M1, and M0 (where M0 = "0", and M1 = "1").

Each F-frame contains four F-bits, which also aid in synchronization between the Local and the remote

DS3 terminals. Therefore, each DS3 M-frame consists of a total of 28 F-bits. These F-bits exhibit a repeating pattern of "1001" within each F-frame. This fact is also presented in Figure 47 and Figure 48, which contains bit-fields that are designated as: F1, F0, F0, and F1 (where F0 = "0", and F1 = "1").

Each of these bit-fields will be used by the Receive DS3 Framer block, within the remote terminal equipment, to perform Frame Acquisition and Frame Maintenance functions.

NOTE: *For more information on how the Receive DS3 Framer uses these bit-fields, please see Section 5.3.2*

5.1.2 Performance Monitoring/Error Detection Bits (Parity)

The DS3 Frame uses numerous bit fields to support performance monitoring of the transmission link between the Local Transmitting Terminal and the Remote Receiving Terminal. The DS3 frame can contain two types of parity bits, depending upon the framing format chosen. P-bits are available in both the M13 and C-bit Parity Formats. However, the C-bit Parity format also includes additional CP-Parity bits.

P-Bits (Applies to M13 and C-Bit Parity Frame Formats)

Each DS3 M-frame consists of two (2) P-bits. These two P-bits carry the parity information of the previous DS3 frame for performance monitoring. These two P-bits must be identical, within a given DS3 frame. The Transmit Section will compute the even parity over all 4704 payload bits within a given DS3 frame, and insert the resulting parity information in the P-bit fields of the very next DS3 frame. The two P-bits are set to "1" if the payload of the previous DS3 frame consists of an odd number of "ones" in the frame. Conversely, the two P-bits are set to zero if an even number of "ones" is found in the payload of the previous DS3 frame.

NOTE: *For information on how the Receive DS3 Framer handles P-bits, please see Section 5.3.2.6.1.*

CP-(Path) Parity Bits (Applies to only the C-Bit Parity Framing Format)

Each DS3 M-Frame consists of tw0 (2) CP-Bits. These two bits have a very similar role to those of P-Bits. Further, the XRT74L74 Framer IC processes CP-Bits in an identical manner that it handles P-Bits.

However for some DS3 applications, there is a difference between P and CP-bits, that should be noted.

- P-Bits are used to support error detection of a DS3 data stream as it travels from one T.E. to the next. (e.g., a single DS3 link between two T.E.)
- CP-Bits are used to support error detection of DS3 data stream as it travels from the Source T.E. (where the DS3 Data Stream originated), to the Sink T.E, (where the DS3 Data Stream is terminated.)

NOTE: *This transmission path from Source T.E. to Sink T.E. may involve numerous T.E.*

- P-Bits are verified and recomputed as it passes through a Mid-Network T.E. (which is neither a Source nor Sink T.E.)
- The values of the CP-Bits (as generated by the Source T.E.) must be preserved as a DS3 frame travels to the Sink T.E. (Through any number of Mid-Network T.E.)

NOTE: *For more information on how CP-Bits are processed, please see section 5.3.2.6.2*

5.1.3 Alarm and Signaling-Related Overhead Bits

The DS3 frame consists of mumerous bit-fields which are used to support the handling of alarm and signaling information. Each of these bit-fields are defined below.

The Alarm Indication Signal (AIS) Pattern (C-Bit Parity Framing Format only)

The Alarm Indication Signal (AIS) pattern is an alarm signal that is inserted into the outbound DS3 stream when a failure is detected by the Local Terminal. The Transmit DS3 Framer will generate the AIS pattern as defined in ANSI.T1.107a-1990, which is described as follows.

- All C-bits are zeros
- All X-bits are set to "1"
- Valid M-bits, F-bits, and P-bits
- A repeating "1010..." pattern is written into the payload of the DS3 frames.

Consequently, no user (or payload) data will be transmitted while the Transmit Section of the chip is transmitting the AIS pattern.

The IDLE Condition Signal

The IDLE Condition signal is used to indicate that the DS3 channel is functionally sound, but has not yet been assigned any traffic. The Transmit Section will transmit the IDLE Condition signal as defined in ANSI T1.107a-1990, which is described as follows.

- Valid M-bits, F-bits, and P-bits
- The three CP-bits (F-frame #3) are zeros
- The X-bits are set to "1"
- A repeating "1100.." pattern is written into the payload of the DS3 frames.

FEAC - Far End Alarm & Control (Only available for the C-bit Parity Frame Format)

The third C-bit (C13 or FEAC) in the first F-frame is used as the Far End Alarm and Control (FEAC) channel between the Near-End DS3 terminal and the Remote DS3 terminal. The FEAC channel carries:

- Alarm and Status Information
- Loopback commands to initiate and deactivate DS3 and DS1 loopbacks at the distant terminals.

The FEAC message consists of a six (6) bit code word of the form [d5, d4, d3, d2, d1 d0]. This message is encapsulated with 10 framing bits to form a 16 bit FEAC Message, as illustrated below. The FEAC signals are encoded into repeating 16 bit message of the form:

0	d5	d4	d3	d2	d1	d0	0	1	1	1	1	1	1	1	1

Since each DS3 frame carries only one FEAC bit, 16 DS3 frames are required to deliver 1 complete FEAC message. The six bits labeled "dx" can represent upto 64 distinct messages, of which 43 have been defined in the standards.

NOTE: *For a more detailed discussion on the transmission of FEAC Messages, please see Section 5.2.3.1.*

FEBE - Far End Block Error (Only available for the C-bit Parity Frame Format)

F-Frame # 4 consists of 3 bit fields for the FEBE (Far-End Block Error) channel. If the (Local) Receive Section (within the Framer IC) detects P-bit parity errors, CP-bit errors or a framing error on the incoming (received) DS3 stream it will inform the Transmit Section of this fact. The Transmit Section will, in turn, set the three FEBE bits (within an outgoing DS3 Frame) to any pattern other than "111" to indicate an error. The Transmit Section will then transmit this information out to the Remote Terminal (e.g., the source of the errored-data). The FEBE bits, in the outbound DS3 frames, are set to "111" only if both of the following conditions are true:

- The Receive DS3 Framer has detected no M-bit or F-bit framing errors, and
- No P-Bit parity errors have been detected.
- No CP-Bit errors have been detected.

NOTE: *A more detailed discussion on the Transmit Section's handling of the FEBE bit-fields can be found in Section 4.2.4.2.1.9.*

The Yellow Alarm or FERF (Far-End Receive Failure) Indicator

The X-bits are used for sending Yellow Alarms or the FERF (Far-End Receive Failure) indication. When the Receive Section (of the XRT74L74), within the Remote Receiving terminal equipment, cannot identify valid framing, or detects an AIS pattern in the incoming DS3 data-stream, the Framer IC can be configured such that the Transmit Section will send a Yellow Alarm or a FERF (Far-End Receive Failure) indication to the Remote Terminal by setting both of the X-bits to zero in the outbound (returning) DS3 path. The X-bits are set to "1" during non-alarm conditions.

5.1.4 The Data Link Related Overhead Bits

UDL: User Data Link (C-bit Parity Frame Format Only)

These bit-fields are not used by the framer and are set to "1" by default. However, these bits may be used for the transmission of data via a proprietary data link. These bit-fields can be accessed via the Transmit Overhead Data Input Interface and the Receive Overhead Data Output Interface blocks.

DL: Path Maintenance Data Link (C-bit Parity Frame Format Only)

The LAPD transceiver block uses these bit-fields for the transmission and reception of path maintenance data link (PMDL) messages via ITU-T Q.921 (LAP-D) Message frames.

NOTE: *Please see Sections 5.2.3.2 and 5.3.3.2 for more information on the operation and function of the LAPD Transmitter.*

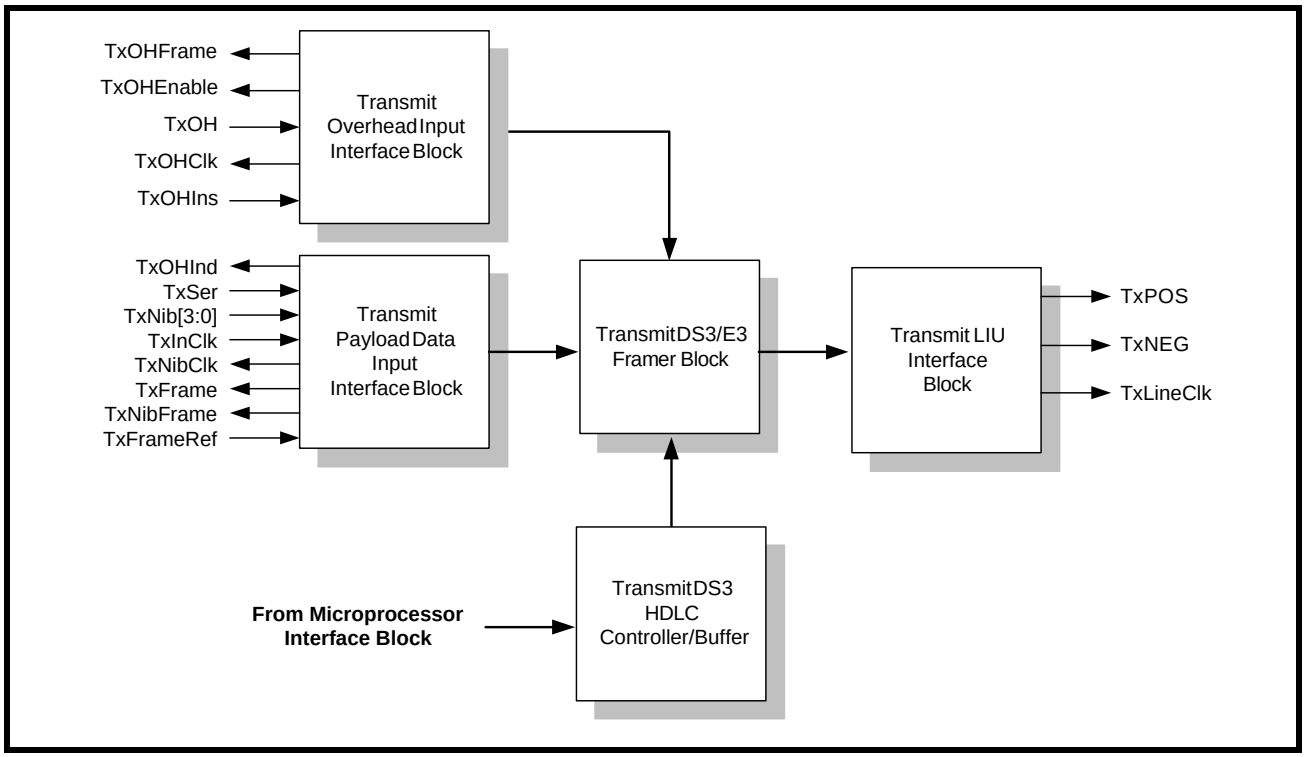
5.2 THE TRANSMIT SECTION OF THE XRT74L74 (DS3 MODE OPERATION)

When the XRT74L74 has been configured to operate in the DS3 Mode, the Transmit Section of the XRT74L74 consists of the following functional blocks.

- Transmit Payload Data Input Interface block
- Transmit Overhead Data Input Interface block
- Transmit DS3 Framer block
- Transmit DS3 HDLC Controller block
- Transmit LIU Interface block

Figure 49 presents a simple illustration of the Transmit Section of the XRT74L74 Framer IC.

FIGURE 49. A SIMPLE ILLUSTRATION OF THE TRANSMIT SECTION, WITHIN THE XRT74L74, WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE DS3 MODE

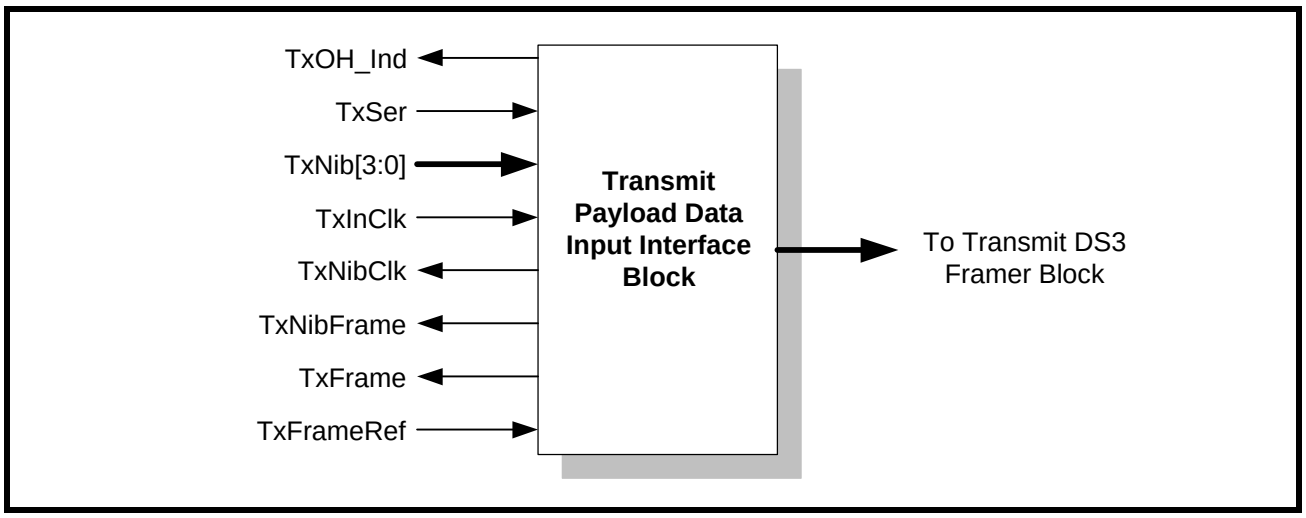


Each of these functional blocks will be discussed in detail in this document.

5.2.1 The Transmit Payload Data Input Interface Block

Figure 50 presents a simple illustration of the Transmit Payload Data Input Interface block.

FIGURE 50. A SIMPLE ILLUSTRATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK



Each of the input and output pins of the Transmit Payload Data Input Interface are listed in Table 34 and described below. The exact role that each of these inputs and output pins assume, for a variety of operating scenarios, are described throughout this section.

TABLE 34: LISTING AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE

SIGNAL NAME	TYPE	DESCRIPTION
TxSer	Input	**Transmit Serial Payload Data Input Pin:** To operate the XRT74L74 in the serial mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the outbound DS3 data stream) to this input pin. The XRT74L74 will sample the data that is at this input pin upon the rising edge either the RxOutClk or the TxInClk signal (whichever is appropriate). ***NOTE:*** *This signal is only active if the NibInt input pin is pulled "Low".*
TxNib[3:0]	Input	**Transmit Nibble-Parallel Payload Data Input pins:** To operate the XRT74L74 in the Nibble-Parallel mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the outbound DS3 data stream) to these input pins. The XRT74L74 will sample the data that is at these input pins upon the rising edge of the TxNibClk signal. ***NOTE:*** *These pins are only active if the NibInt input pin is pulled "High".*
TxNibFrame	Output	**Transmit End of Frame Output Indicator - Nibble Mode** The Transmit Section of the XRT74L74 will pulse this output pin "High" (for one nibble-period), when the Transmit Payload Data Input Interface is processing the last nibble of a given DS3 frame. The purpose of this output pin is to alert the Terminal Equipment that it needs to begin transmission of a new DS3 frame to the XRT74L74.
TxInClk	Input	**Transmit Section Timing Reference Clock Input pin:** The Transmit Section of the XRT74L74 can be configured to use this clock signal as the Timing Reference. If this configuration is selected, then the XRT74L74 will use this clock signal to sample the data on the TxSer input pin. ***NOTE:*** *If this configuration has been selected, then a 44.736 MHz clock signal must be applied to this input pin.*
TxNibClk	Output	**Transmit Nibble Mode Output** To operate the XRT74L74 in the Nibble-Parallel mode, then the XRT74L74 will derive this clock signal from the selected Timing Reference for the Transmit Section of the chip (e.g., either the TxInClk or the RxLineClk signals). It is advisable to configure the Terminal Equipment to output the outbound payload data (to the XRT74L74 Framer IC) onto the TxNib[3:0] input pins, upon the rising edge of this clock signal. ***NOTE:*** *For DS3 Applications, the XRT74L74 Framer IC will output 1176 clock edges (to the Terminal Equipment) for each outbound DS3 frame.*
TxOHInd	Output	**Transmit Overhead Bit Indicator Output:** This output pin will pulse "High" one-bit period prior to the time that the Transmit Section of the XRT74L74 will be processing an Overhead bit. The purpose of this output pin is to warn the Terminal Equipment that, during the very next bit-period, the XRT74L74 is going to be processing an Overhead bit and will be ignoring any data that is applied to the TxSer input pin. For DS3 applications, this output pin is only active if the XRT74L74 is operating in the Serial Mode. This output pin will be pulled "Low" if the device is operating in the Nibble-Parallel Mode.
TxFrame	Output	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT74L74 will pulse this output pin "High" (for one bit-period), when the Transmit Payload Data Input Interface is processing the last bit of a given DS3 frame. The purpose of this output pin is to alert the Terminal Equipment that it needs to begin transmission of a new DS3 frame to the XRT74L74 (e.g., to permit the XRT74L74 to maintain Transmit DS3 framing alignment control over the Terminal Equipment).

TABLE 34: LISTING AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE

SIGNAL NAME	TYPE	DESCRIPTION
TxFrameRef	Input	**Transmit Frame Reference Input:** The XRT74L74 permits the configuration of the Transmit Section to use this input pin as a frame reference. If this configuration is selected, then the Transmit Section will initiate its transmission of a new DS3 frame, upon the rising edge of this signal. The purpose of this input pin is to permit the Terminal Equipment to maintain Transmit DS3 Framing alignment control over the XRT74L74.
RxOutClk	Output	**Loop-Timed Timing Reference Clock Output pin:** The Transmit Section of the XRT74L74 can be configured to use the RxLineClk signal as the Timing Reference (e.g., loop-timing). If this configuration is selected, then the XRT74L74 will: • Output a 44.736 MHz clock signal via this pin, to the Terminal Equipment. • Sample the data on the TxSer input pin, upon the rising edge of this clock signal.

Operation of the Transmit Payload Data Input Interface

The Transmit Payload Data Input Interface is extremely flexible, in that it permits the following configuration options.

- The Serial or the Nibble-Parallel Interface Mode
- The Loop-Timing or the TxInClk (Local Timing) Mode

Further, if the XRT74L74 has been configured to operate in the TxInClk (Local Timing) mode, then there are two additional options.

- The XRT74L74 functions as the Frame Master (e.g., it dictates when the Terminal Equipment will initiate the transmission of data within a new DS3 frame).
- The XRT74L74 functions as the Frame Slave (e.g., the Terminal Equipment will dictate when the XRT74L74 initiates the transmission of a new DS3 frame).

Given these three set of options, the Transmit Terminal Input Interface can be configured to operate in one of the six (6) following modes.

- Mode 1 - Serial/Loop-Timed Mode
- Mode 2 - Serial/Local-Timed/Frame Slave Mode
- Mode 3 - Serial/Local-Timed/Frame Master Mode
- Mode 4 - Nibble/Loop-Timed Mode
- Mode 5 - Nibble/Local-Timed/Frame Slave Mode
- Mode 6 - Nibble/Local-Timed/Frame Master Mode

Each of these modes are described, in detail, below.

5.2.1.1 Mode 1 - The Serial/Loop-Timing Mode The Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Loop-Timing (Uses the RxLineClk signal as the Timing Reference)

Since the XRT74L74 is configured to operate in the loop-timed mode, the Transmit Section (of the XRT74L74) will use the RxLineClk input clock signal (e.g., the Recovered Clock signal, from the LIU) as its timing source. When the XRT74L74 is operating in this mode it will do the following.

1. It will ignore any signal at the TxInClk input pin.
2. The XRT74L74 will output a 44.736MHz clock signal via the RxOutClk output pin. This clock signal functions as the Transmit Payload Data Input Interface block clock signal.
3. The XRT74L74 will use the rising edge of the RxOutClk signal to latch in the data residing on the TxSer input pin.

B. Serial Mode

The XRT74L74 will accept the DS3 payload data from the Terminal Equipment, in a serial-manner, via the TxSer input pin The Transmit Payload Data Input Interface block will latch this data into its circuitry, on the rising edge of the RxOutClk output clock signal.

C. Delineation of outbound DS3 frames

The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period coincident with the XRT74L74 processing the last bit of a given DS3 frame.

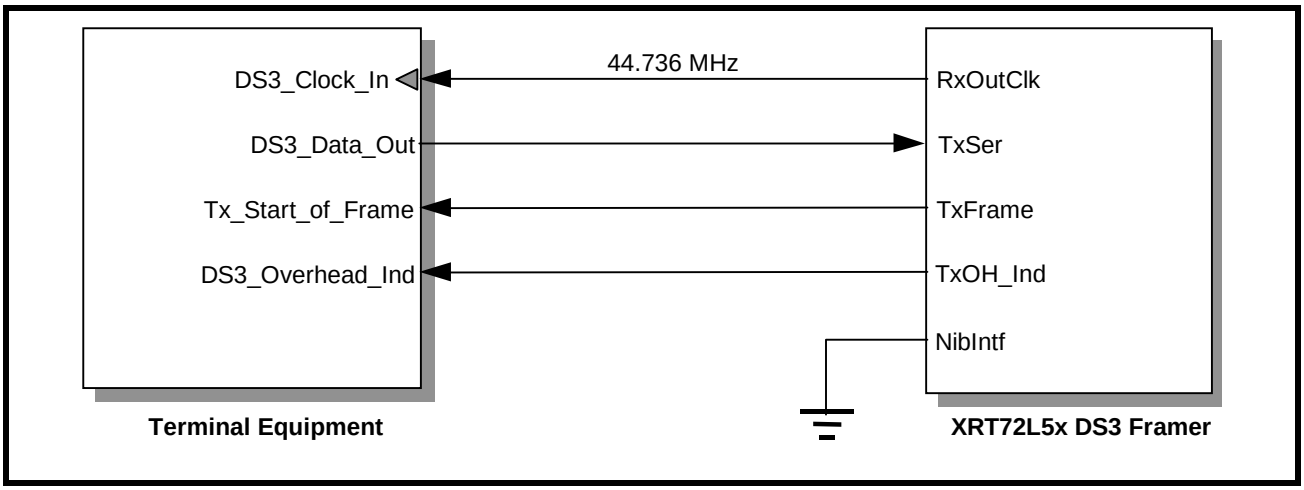
D. Sampling of Payload Data, from the Terminal Equipment

In Mode 1, the XRT74L74 will sample the data at the TxSer input, on the rising edge of RxOutClk.

Interfacing the Transmit Payload Data Input Interface block (of the XRT74L74) to the Terminal Equipment for Mode 1 Operation

Figure 51 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 1 operation.

FIGURE 51. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK (OF THE XRT74L74) FOR MODE 1(SERIAL/LOOP-TIMING) OPERATION



Mode 1, Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode, it will function as the source of the 44.736MHz clock signal (via the RxOutClk signal). This clock signal will be used as the Terminal Equipment Interface clock by both the XRT74L74 IC and the Terminal Equipment.

The Terminal Equipment will serially output the payload data of the outbound DS3 data stream via its DS3_Data_Out pin. The Terminal Equipment will update the data on the DS3_Data_Out pin upon the rising edge of the 44.736 MHz clock signal, at its DS3_Clock_In input pin (as depicted in Figure 51 and Figure 52).

The XRT74L74 will latch the outbound DS3 data stream (from the Terminal Equipment) on the rising edge of the RxOutClk signal.

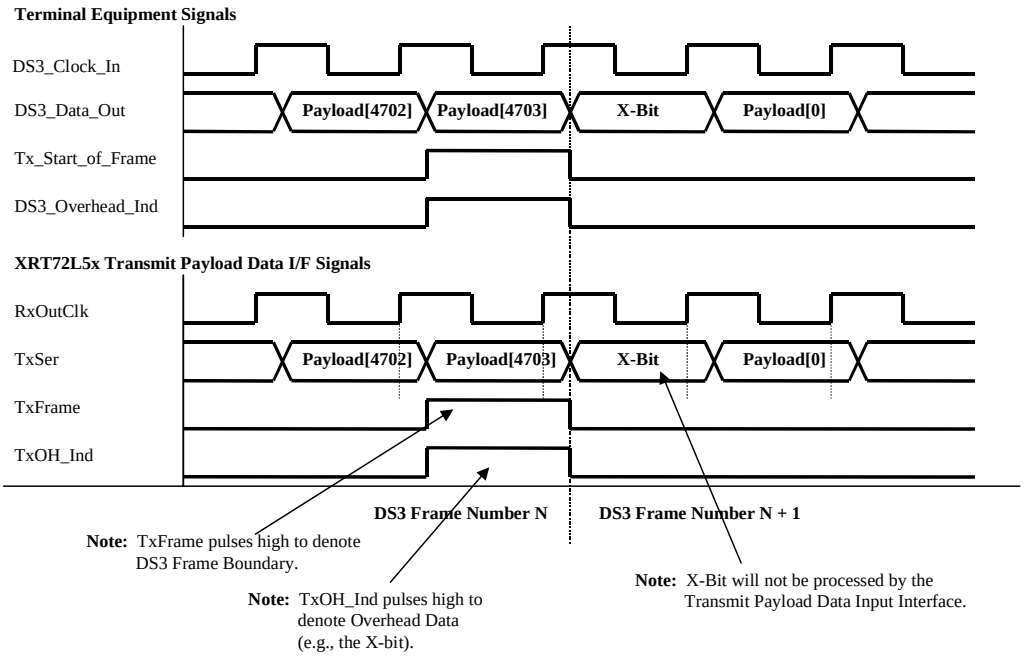
The XRT74L74 will indicate that it is processing the last bit, within a given outbound DS3 frame, by pulsing its TxFrame output pin "High" for one bit-period.

When the Terminal Equipment detects this pulse at its Tx_Start_of_Frame input, it is expected to begin transmission of the very next outbound DS3 frame to the XRT74L74 via the DS3_Data_Out (or TxSer pin).

Finally, the XRT74L74 will indicate that it is about to process an overhead bit by pulsing the TxOH_Ind output pin "High" one bit period prior to its processing of an OH (Overhead) bit. In Figure 51, the TxOH_Ind output pin is connected to the DS3_Overhead_Ind input pin of the Terminal Equipment. Whenever the DS3_Overhead_Ind pin is pulsed "High" the Terminal Equipment is expected to not transmit a DS3 payload bit upon the very next clock edge. Instead, the Terminal Equipment is expected to delay its transmission of the very next payload bit, by one clock cycle.

The behavior of the signals, between the XRT74L74 and the Terminal Equipment, for DS3 Mode 1 operation is illustrated in Figure 52.

FIGURE 52. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT (FOR MODE 1 OPERATION)



How to configure the XRT74L74 into the Serial/ Loop-Timed/Non-Overhead Interface Mode

1. Set the NibIntf input pin "Low".

2. Set the TimRefSel[1:0] bit fields (within the Framer Operating Mode Register) to "00", as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 51.

NOTE: *The XRT74L74 Framer IC cannot support the Framer Local Loop-back Mode of operation, when operating in the Loop-Timing Mode. The XRT74L74 Framer IC must be configured into any of the following modes, prior to configuring the Framer Local Loop-back Mode.*

- Mode 2 - Serial/Local-Timed/Frame-Slave Mode.
- Mode 3 - Serial/Local-Timed/Frame-Master Mode.
- Mode 5 - Nibble-Parallel/Local-Timed/Frame-Slave Mode.
- Mode 6 - Nibble-Parallel/Local-Timed/Frame-Master Mode.

NOTE: *For more detailed information on Framer Local Loop-back Mode of operation, please see the loop-back section.*

5.2.1.2 Mode 2 - The Serial/Local-Timed/ Frame-Slave Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local-Timing - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the DS3 payload data, in a serial manner, via the TxSer input pin. The Transmit Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of outbound DS3 frames (Frame Slave Mode)

The Transmit Section (of the XRT74L74) will use the TxInClk input as its timing reference, and will use the TxFrameRef input signal as its framing reference. In other words, the Transmit Section of the XRT74L74 will initiate frame generation upon the rising edge of the TxFrameRef input signal).

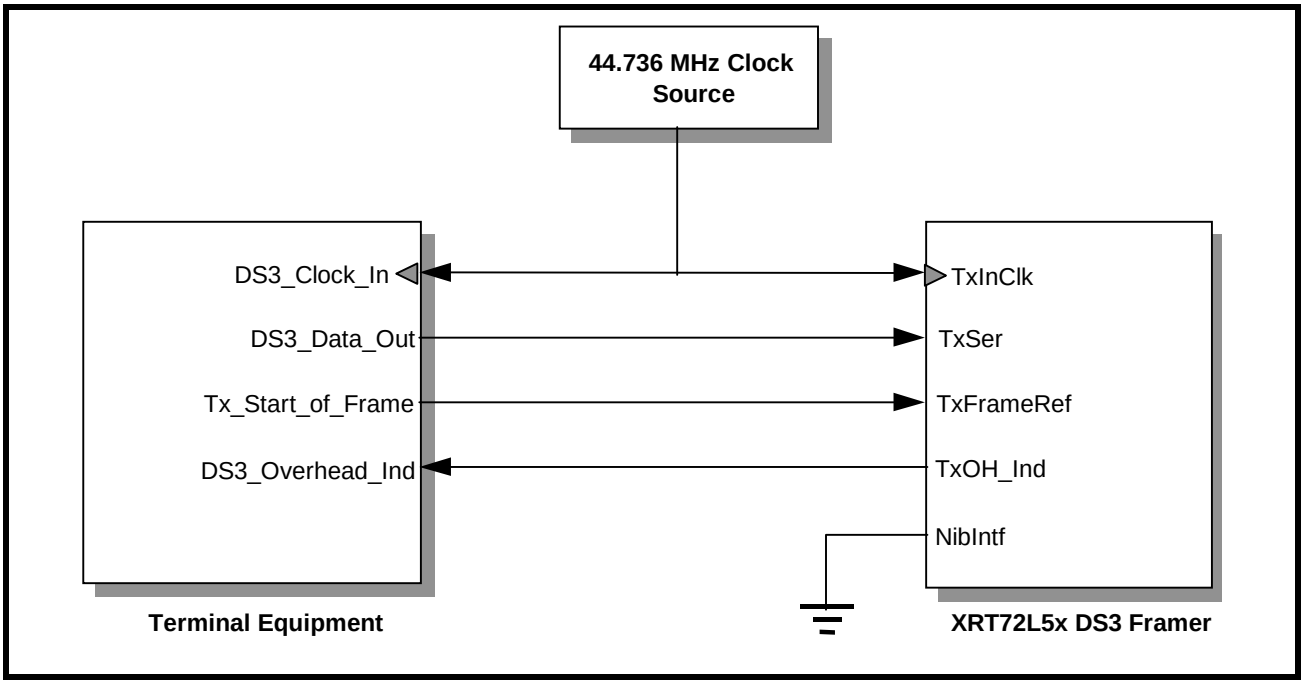
D. Sampling of payload data, from the Terminal Equipment

In Mode 2, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block (of the XRT74L74) to the Terminal Equipment for Mode 2 Operation

Figure 53 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 2 operation.

FIGURE 53. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 2 (SERIAL/LOCAL-TIMED/FRAME-SLAVE) OPERATION



Mode 2, Operation of the Terminal Equipment

As shown in Figure 53, both the Terminal Equipment and the XRT74L74 will be driven by an external 44.736MHz clock signal. The Terminal Equipment will receive the 44.736MHz clock signal via its DS3_Clock_In input pin, and the XRT74L74 Framer IC will receive the 44.736MHz clock signal via the TxInClk input pin.

The Terminal Equipment will serially output the payload data of the outbound DS3 data stream, via the DS3_Data_Out output pin, upon the rising edge of the signal at the DS3_Clock_In input pin.

NOTE: *The DS3_Data_Out output pin of the Terminal Equipment is electrically connected to the TxSer input pin.*

The XRT74L74 Framer IC will latch the data, residing on the TxSer input line, on the rising edge of the TxInClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing its Tx_Start_of_Frame output signal (and in turn, the TxFrameRef input pin of the XRT74L74), "High" for one-bit period, coincident with the first bit of a new DS3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new DS3 frame.

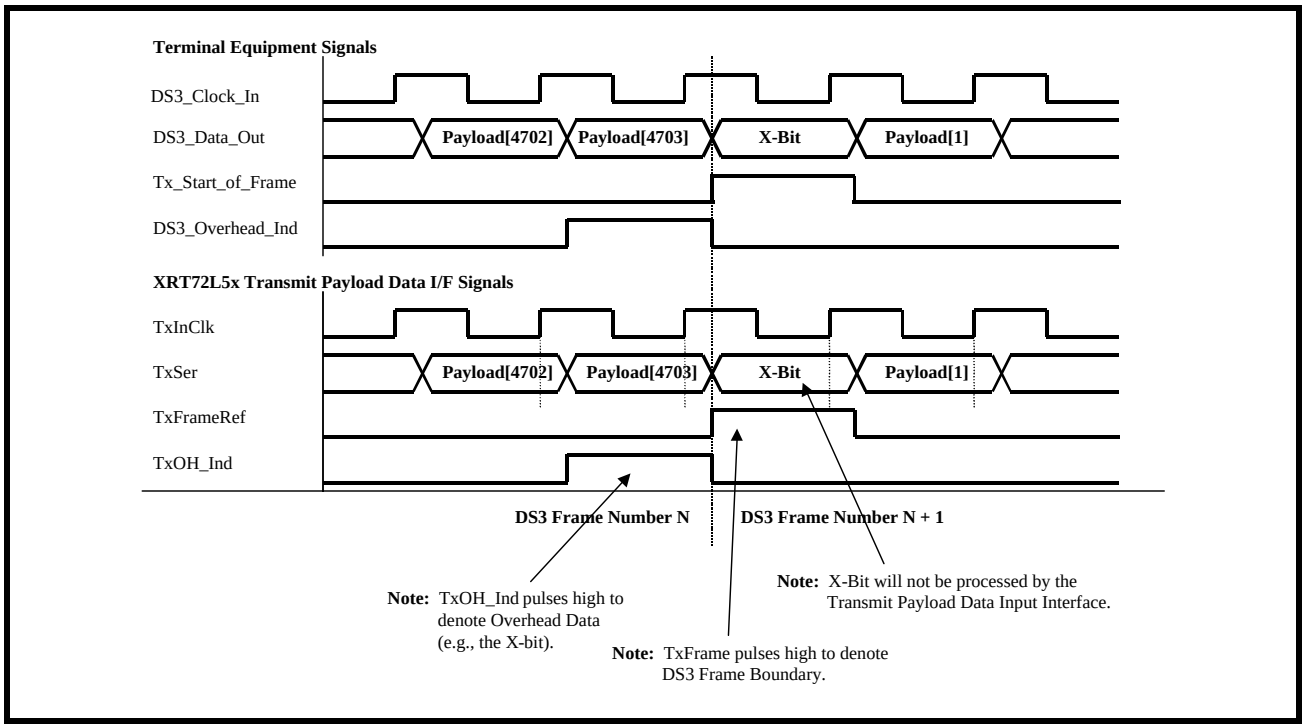
NOTES:

1. *In this case, the Terminal Equipment is controlling the start of Frame Generation, and is therefore referred to as the Frame Master. Conversely, since the XRT74L74 does not control the generation of a new DS3 frame, but is rather driven by the Terminal Equipment. Hence, the XRT74L74 is referred to as the Frame Slave.*
2. *If the XRT74L74 is configured to operate in Mode 2, it is imperative that the Tx_Start_of_Frame (or TxFrameRef) signal is synchronized to the TxInClk input clock signal.*

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the outbound DS3 frame. Since the TxOH_Ind output pin of the XRT74L74 is electrically connected to the DS3_Overhead_Ind, whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the DS3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it should delay transmission of the very next DS3 frame payload bit by one clock cycle.

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Mode 2 Operation is illustrated in Figure 54.

FIGURE 54. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 2 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 53.

5.2.1.3 Mode 3 - The Serial/Local-Timed/ Frame-Master Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local Timing - (Uses the TxInClk signal as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the DS3 payload data, in a serial manner, via the TxSer input pin. The Transmit Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of outbound DS3 frames (Frame Master Mode)

The Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference, and will initiate DS3 frame generation, asynchronously with respect to any externally applied signal. The XRT74L74 will pulse its TxFrame output pin "High" whenever it is processing the very last bit-field within a given DS3 frame.

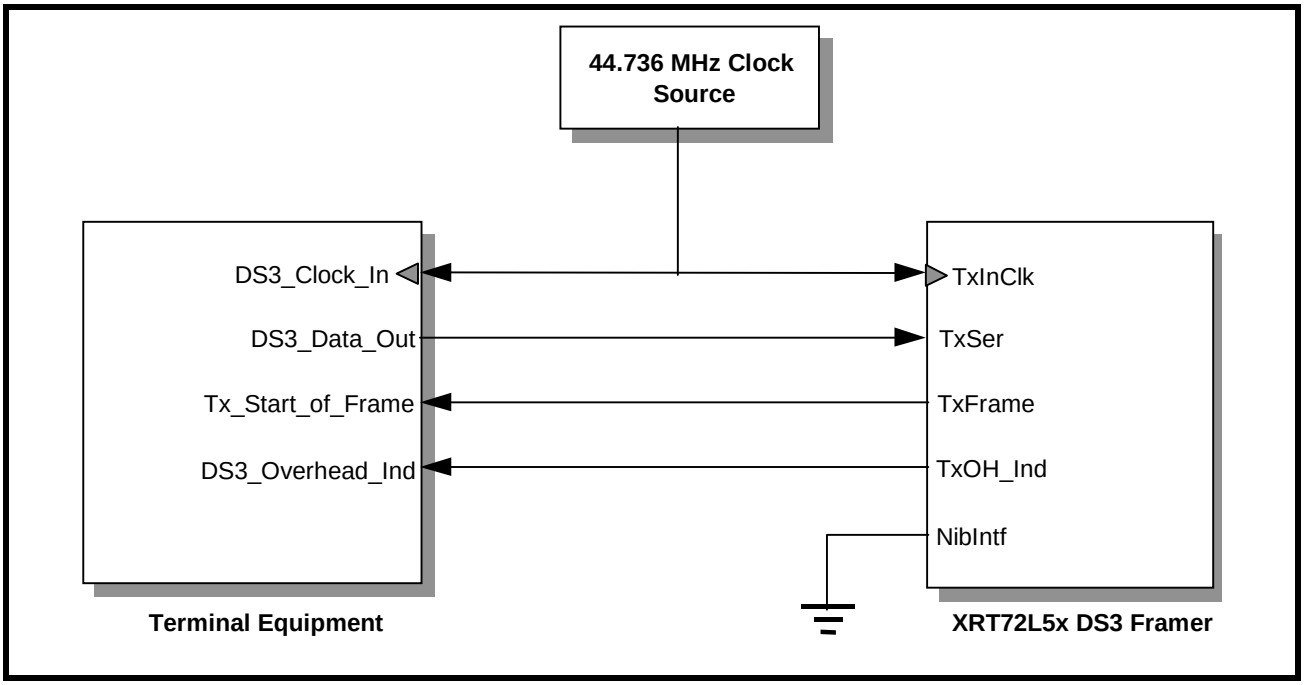
D. Sampling of payload data, from the Terminal Equipment

In Mode 3, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 3 Operation

Figure 55 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 3 operation.

FIGURE 55. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 3 (SERIAL/LOCAL-TIMED/FRAME-MASTER) OPERATION



Mode 3 Operation of the Terminal Equipment

In Figure 55, both the Terminal Equipment and the XRT74L74 are driven by an external 44.736MHz clock signal. This clock signal is connected to the DS3_Clock_In input of the Terminal Equipment and the TxInClk input pin of the XRT74L74.

The Terminal Equipment will serially output the payload data on its DS3_Data_Out output pin, upon the rising edge of the signal at the DS3_Clock_In input pin. Similarly, the XRT74L74 will latch the data, residing on the TxSer input pin, on the rising edge of TxInClk.

The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period, coincident while it is processing the last bit-field within a given outbound DS3 frame. The Terminal Equipment is expected to monitor the TxFrame signal (from the XRT74L74) and to place the first bit, within the very next outbound DS3 frame on the TxSer input pin.

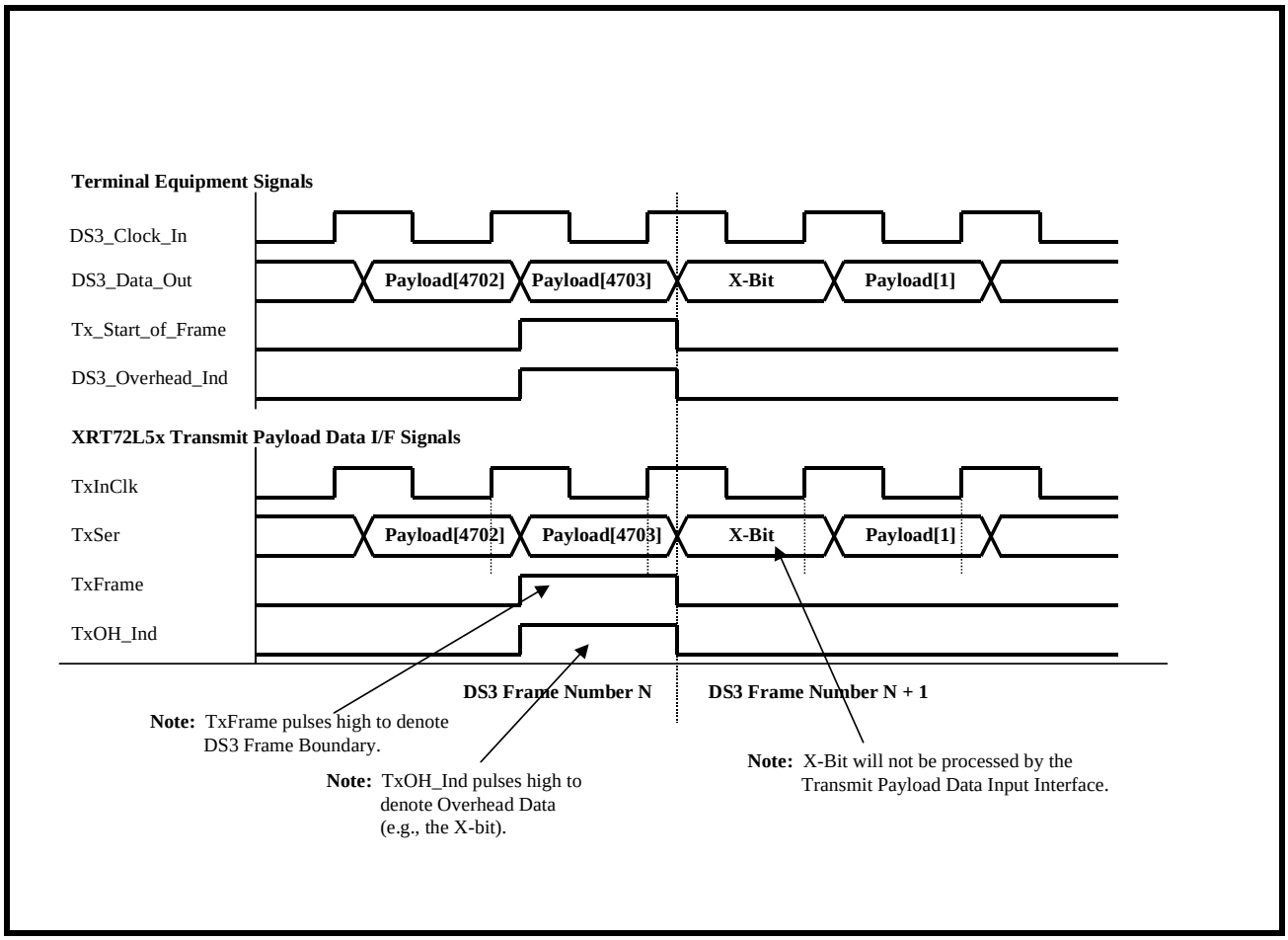
NOTE: *In this case, the XRT74L74 dictates exactly when the very next DS3 frame will be generated.*

The Terminal Equipment is expected to respond appropriately by providing the XRT74L74 with the first bit of the new DS3 frame, upon demand. Hence, in this mode, the XRT74L74 is referred to as the Frame Master and the Terminal Equipment is referred to as the Frame Slave.

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the outbound DS3 frame. Since the TxOH_Ind output pin (of the XRT74L74) is electrically connected to the DS3_Overhead_Ind whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the DS3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it should delay transmission of the very next DS3 frame payload bit by one clock cycle.

The behavior of the signal between the XRT74L74 and the Terminal Equipment for DS3 Mode 3 Operation is illustrated in Figure 56.

FIGURE 56. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (DS3 MODE 3 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "10" or "11" as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	X

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 55.

5.2.1.4 Mode 4 - The Nibble-Parallel/Loop-Timed Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Looped Timing (Uses the RxLineClk as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the RxLineClk signal as its timing reference. When the XRT74L74 is operating in the Nibble-Mode, it will internally divide the RxLineClk signal, by a factor of four (4) and will output this signal via the TxNibClk output pin.

B. Nibble-Parallel Mode

The XRT74L74 will accept the DS3 payload data, from the Terminal Equipment in a nibble-parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface block will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of the outbound DS3 frames

The XRT74L74 will pulse the TxNibFrame output pin "High" for one bit-period coincident with the XRT74L74 processing the last nibble of a given DS3 frame.

D. Sampling of payload data, from the Terminal Equipment

In Mode 4, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the RxOutClk clock signal, following a pulse in the TxNibClk signal (see Figure 58).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 11.184 MHz (e.g., 44.736 MHz divided by 4). However, for reasons described below, TxNibClk effectively operates at a Low clock frequency. The Transmit Payload Data Input Interface is only used to accept the payload data, which is intended to be carried by outbound DS3 frames. The Transmit Payload Data Input Interface is not designed to accommodate the entire DS3 data stream.*

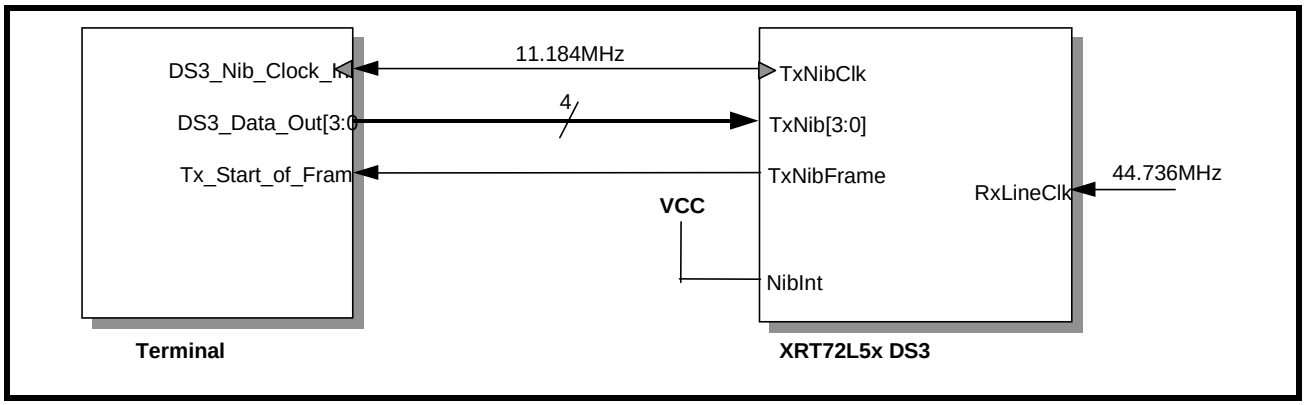
The DS3 Frame consists of 4704 payload bits or 1176 nibbles. Therefore, the XRT74L74 will supply 1176 TxNibClk pulses between the rising edges of two consecutive TxNibFrame pulses. The DS3 Frame repetition rate is 9.398kHz. Hence, 1176 TxNibClk pulses for each DS3 frame period amounts to TxNibClk running at approximately 11.052 MHz. The method by which the 1176 TxNibClk pulses are distributed throughout the DS3 frame period is presented below.

Nominally, the Transmit Section within the XRT74L74 will generate a TxNibClk pulse for every 4 RxOutClk (or TxInClk) periods. However, in 14 cases (within a DS3 frame period), the Transmit Payload Data Input Interface will allow 5 TxInClk periods to occur between two consecutive TxNibClk pulses.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 4 Operation

Figure 57 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 4 Operation.

FIGURE 57. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMED) OPERATION



Mode 4 Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode, it will function as the source of the 11.184MHz (e.g., the 44.736MHz clock signal divided by "4") clock signal, that will be used as the Terminal Equipment Interface clock by both the XRT74L74 and the Terminal Equipment.

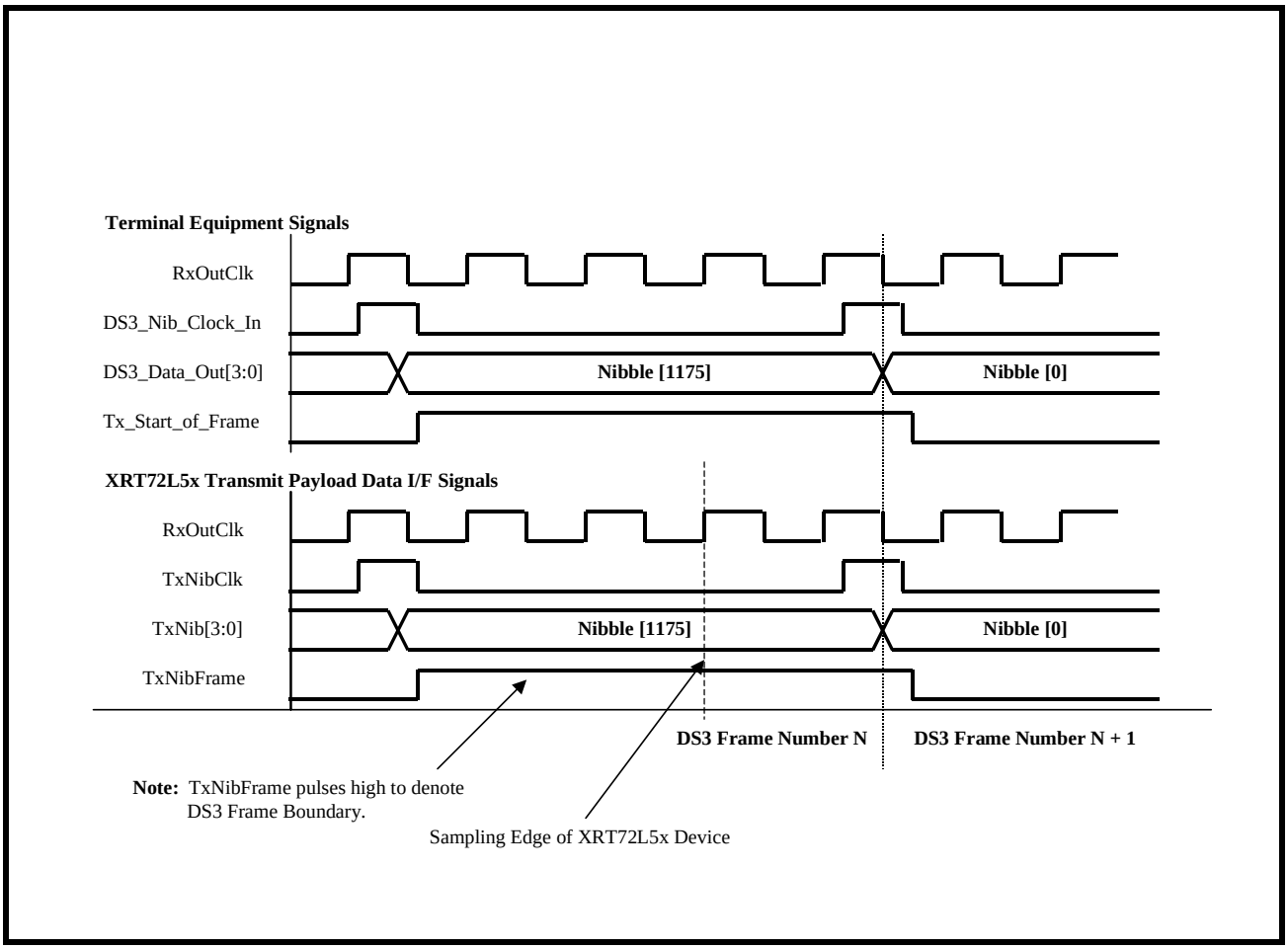
The Terminal Equipment will output the payload data of the outbound DS3 data stream via its DS3_Data_Out[3:0] pins on the rising edge of the 11.184MHz clock signal at the DS3_Nib_Clock_In input pin.

The XRT74L74 will latch the outbound DS3 data stream (from the Terminal Equipment) on the rising edge of the TxNibClk output clock signal. The XRT74L74 will indicate that it is processing the last nibble, within a given DS3 frame, by pulsing its TxNibFrame output pin "High" for one TxNibClk clock period. When the Terminal Equipment detects a pulse at its Tx_Start_of_Frame input pin, it is expected to transmit the first nibble, of the very next outbound DS3 frame to the XRT74L74 via the DS3_Data_Out[3:0] (or TxNib[3:0] pins).

Finally, for the Nibble-Parallel Mode operation, the XRT74L74 will continuously pull the TxOHInd output pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Mode 4 Operation is illustrated in Figure 58.

FIGURE 58. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 4 OPERATION)



How to configure the XRT74L74 into Mode 4

1. Set the NibIntf input pin "High".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to “00" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 57.

NOTE: *The XRT74L74 Framer IC cannot support the Framer Local Loop-back Mode of operation. The XRT74L74 Framer IC must be configured into any of the following modes, prior to configuring the Framer Local-Loop-back Mode operation.*

- Mode 2 - Serial/Local-Timed/Frame-Slave Mode.
- Mode 3 - Serial/Local-Timed/Frame-Master Mode.
- Mode 5 - Nibble-Parallel/Local-Timed/Frame-Slave Mode.
- Mode 6 - Nibble-Parallel/Local-Timed/Frame-Master Mode.

NOTE: *For more detailed information on the Framer Local Loop-back Mode Operation, please see the loop-back section.*

5.2.1.5 Mode 5 - The Nibble-Parallel/Local-Timed/Frame-Slave Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local-Timed (Uses the TxInClk signal as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the DS3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of outbound DS3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will use the TxFrameRef input signal as its Framing Reference (e.g., the Transmit Section of the XRT74L74 initiates frame generation upon the rising edge of the TxFrameRef signal).

NOTE: *In this case, the Terminal Equipment should pulse the TxFrameRef input signal (of the XRT74L74 Framer IC) coincident with it applying the first payload nibble, within a given outbound DS3 frame. Hence, the duration of this pulse should be one nibble-period of the DS3 signal (see Figure 60).*

D. Sampling of payload data, from the Terminal Equipment

In Mode 5, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 60).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 11.184 MHz (e.g., 44.736 MHz divided by 4). However, for reasons described below, TxNibClk effectively operates at a Low clock frequency. The Transmit Payload Data Input Interface is only used to accept the payload data, which is intended to be carried by outbound DS3 frames. The Transmit Payload Data Input Interface is not designed to accommodate the entire DS3 data stream.*

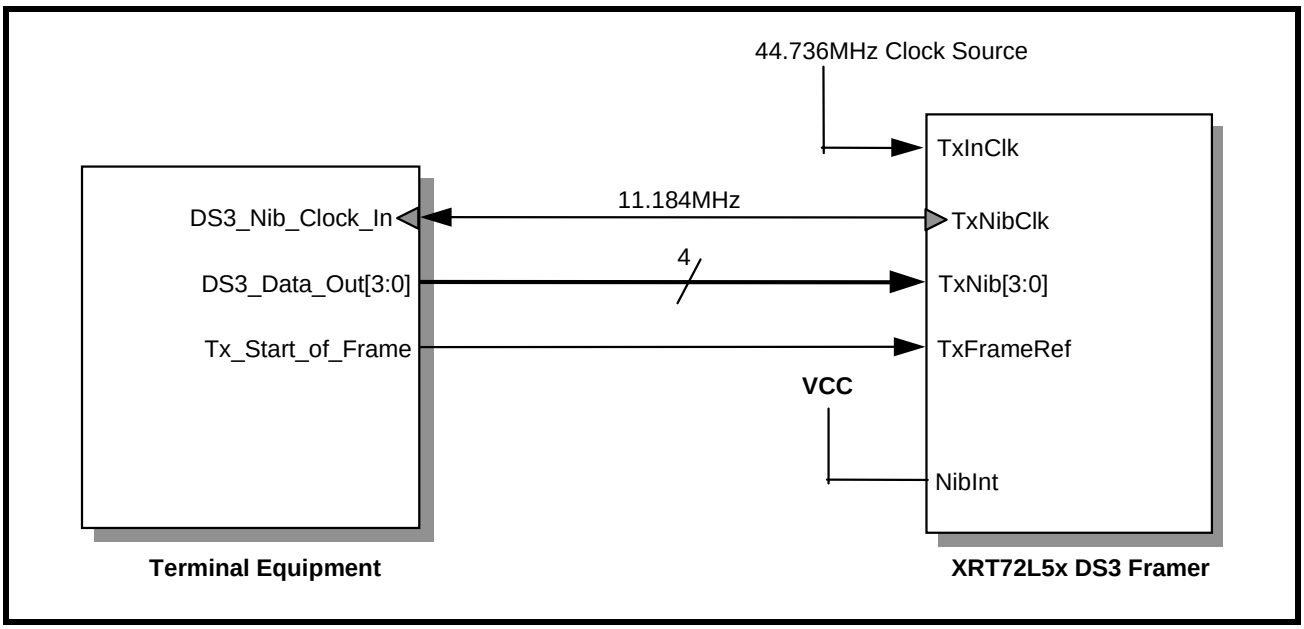
The DS3 Frame consists of 4704 payload bits or 1176 nibbles. Therefore, the XRT74L74 will supply 1176 TxNibClk pulses between the rising edges of two consecutive TxNibFrame pulses. The DS3 Frame repetition rate is 9.398kHz. Hence, 1176 TxNibClk pulses for each DS3 frame period amounts to TxNibClk running at approximately 11.052 MHz. The method by which the 1176 TxNibClk pulses are distributed throughout the DS3 frame period is presented below.

Nominally, the Transmit Section within the XRT74L74 will generate a TxNibClk pulse for every 4 RxOutClk (or TxInClk) periods. However, in 14 cases (within a DS3 frame period), the Transmit Payload Data Input Interface will allow 5 TxInClk periods to occur between two consecutive TxNibClk pulses.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 5 Operation

Figure 59 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 5 Operation.

FIGURE 59. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMED/FRAME-SLAVE) OPERATION



Mode 5 Operation of the Terminal Equipment

In Figure 59 both the Terminal Equipment and the XRT74L74 will be driven by an external 11.184MHz clock signal. The Terminal Equipment will receive the 11.184MHz clock signal via the DS3_Nib_Clock_In input pin. The XRT74L74 will output the 11.184MHz clock signal via the TxNibClk output pin.

The Terminal Equipment will serially output the data on the DS3_Data_Out[3:0] pins, upon the rising edge of the signal at the DS3_Clock_In input pin.

NOTE: *The DS3_Data_Out[3:0] output pins of the Terminal Equipment is electrically connected to the TxNib[3:0] input pins.*

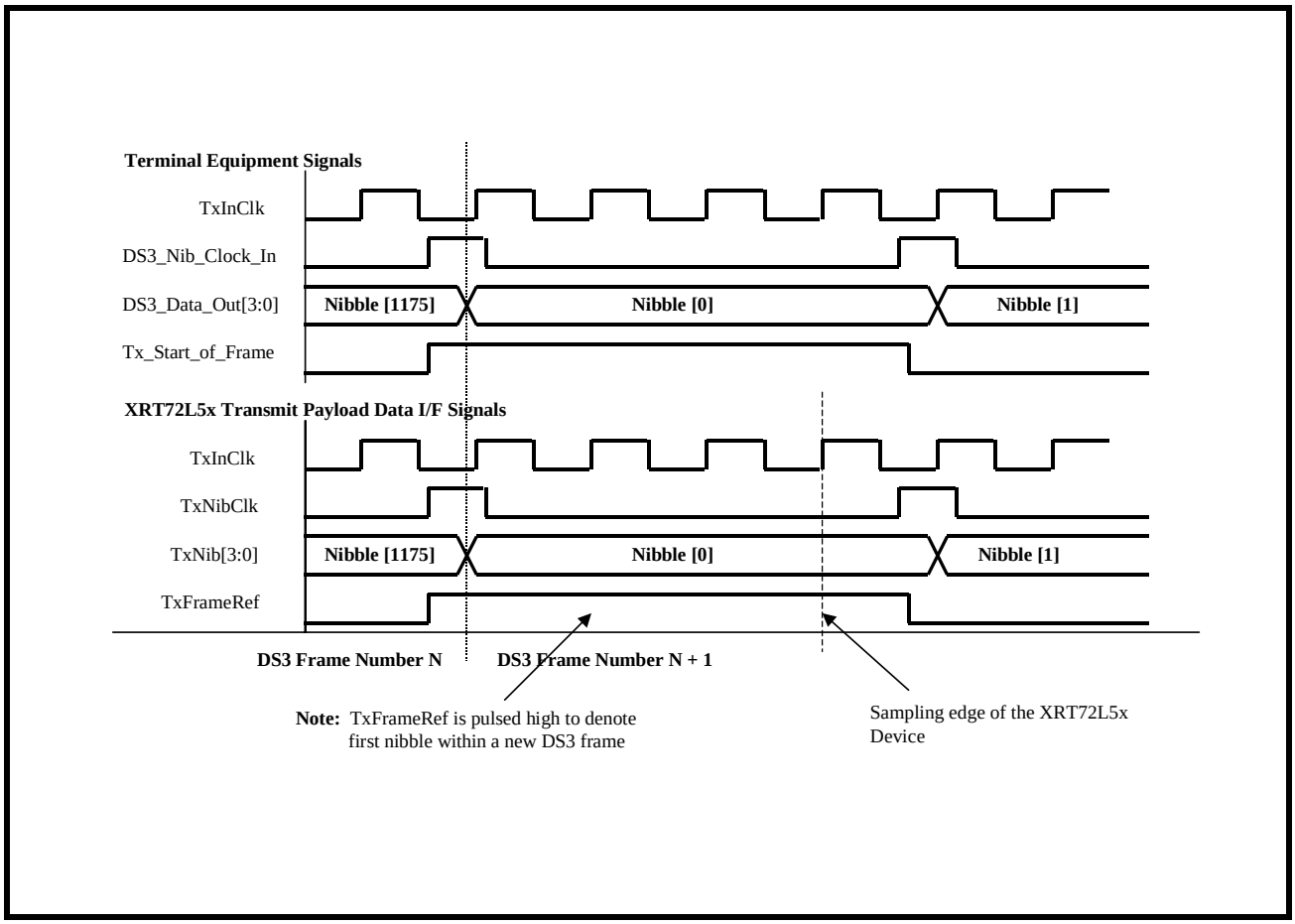
The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing the Tx_Start_of_Frame output pin (and in turn, the TxFrameRef input pin of the XRT74L74) "High" for one bit-period, coincident with the first nibble of a new DS3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new DS3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the DS3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Mode 5 Operation is illustrated in Figure 60.

FIGURE 60. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (DS3 MODE 5 OPERATION)



How to configure the XRT74L74 into Mode 5

1. Set the NibIntf input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 59.

5.2.1.6 Mode 6 - The Nibble-Parallel/TxInClk/ Frame-Master Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local-Timed (Uses the TxInClk signal as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the

TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the DS3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of outbound DS3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will initiate the generation of DS3 frames, asynchronous with respect to any external signal. The XRT74L74 will pulse the TxFrame output pin "High" whenever it is processing the last nibble, within a given outbound DS3 frame.

D. Sampling of payload data, from the Terminal Equipment

In Mode 6, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 62).

NOTE: *The TxNibClk signal from the XRT74L74, operates nominally at 11.184 MHz (e.g., 44.736 MHz divided by 4). However, for reasons described below, TxNibClk effectively operates at a Low clock frequency. The Transmit Payload Data Input Interface is only used to accept the payload data, which is intended to be carried by outbound DS3 frames. The Transmit Payload Data Input Interface is not designed to accommodate the entire DS3 data stream.*

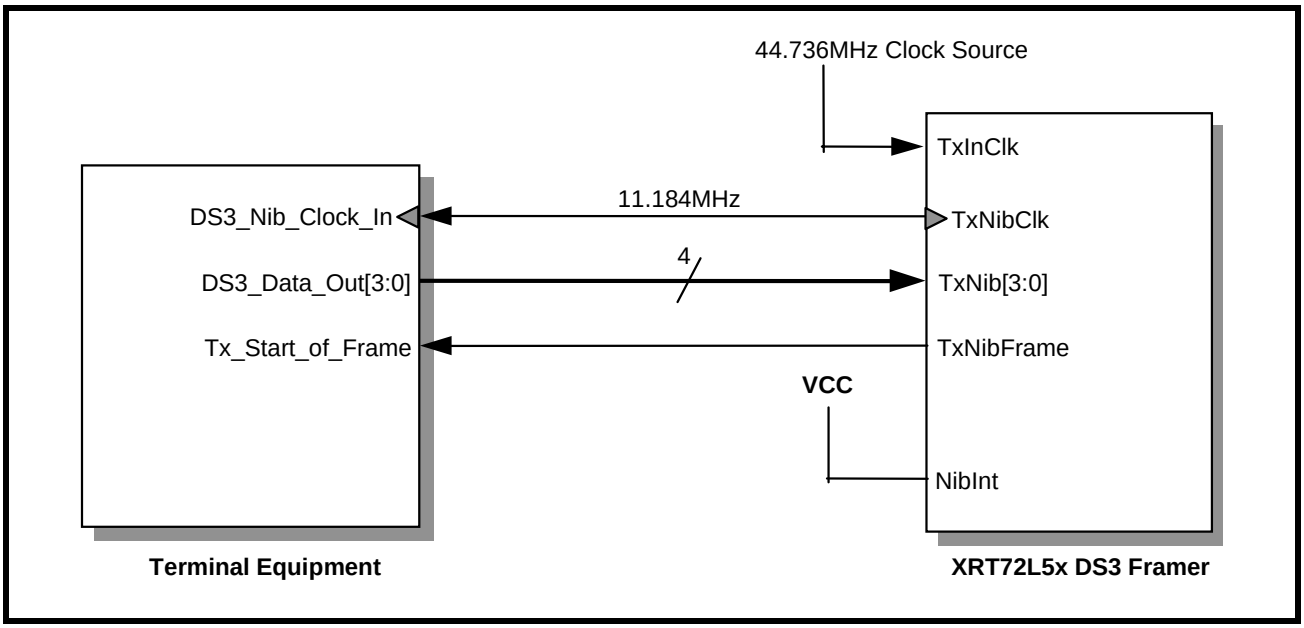
The DS3 Frame consists of 4704 payload bits or 1176 nibbles. Therefore, the XRT74L74 will supply 1176 TxNibClk pulses between the rising edges of two consecutive TxNibFrame pulses. The DS3 Frame repetition rate is 9.398kHz. Hence, 1176 TxNibClk pulses for each DS3 frame period amounts to TxNibClk running at approximately 11.052 MHz. The method by which the 1176 TxNibClk pulses are distributed throughout the DS3 frame period is presented below.

Nominally, the Transmit Section within the XRT74L74 will generate a TxNibClk pulse for every 4 RxOutClk (or TxInClk) periods. However, in 14 cases (within a DS3 frame period), the Transmit Payload Data Input Interface will allow 5 TxInClk periods to occur between two consecutive TxNibClk pulses.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 6 Operation

Figure 61 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 6 Operation.

FIGURE 61. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMED/FRAME-MASTER) OPERATION



Mode 6 Operation of the Terminal Equipment

In Figure 61 both the Terminal Equipment and the XRT74L74 will be driven by an external 11.184MHz clock signal. The Teriminal Equipment will receive

the 11.184MHz clock signal via the DS3_Nib_Clock_In input pin. The XRT74L74 will output the 11.184MHz clock signal via the TxNibClk output pin.

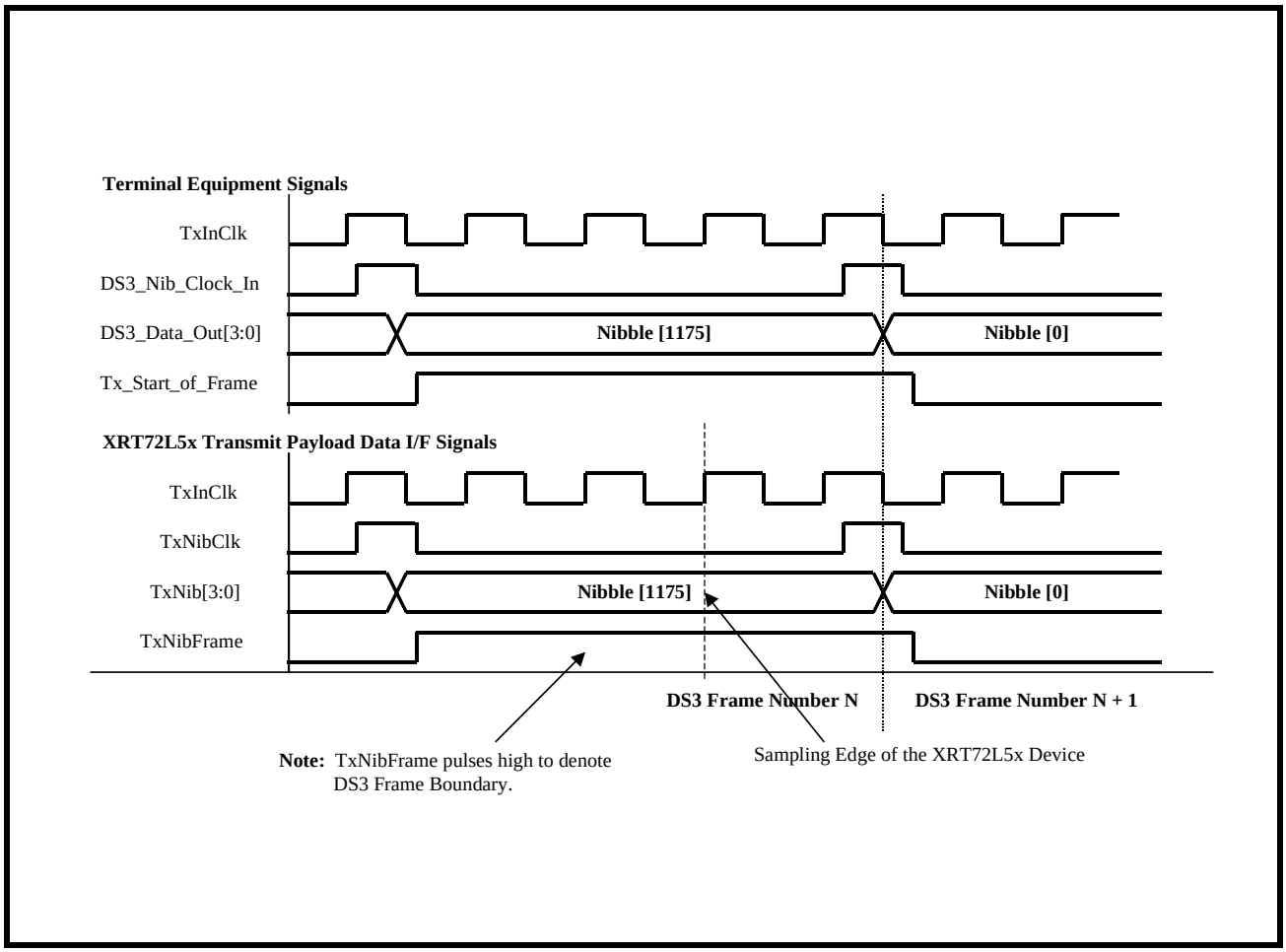
The Terminal Equipment will serially output the data on the DS3_Data_Out[3:0] pins upon the rising edge of the signal at the DS3_Clock_In input pin. The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case the XRT74L74 has the responsibility of providing the framing reference signal by pulsing the TxFrame output pin (and in turn the Tx_Start_of_Frame input pin of the Terminal Equipment) "High" for one nibble-period, coincident with the last nibble within a given DS3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the DS3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Mode 6 Operation is illustrated in Figure 62.

FIGURE 62. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (DS3 MODE 6 OPERATION)



How to configure the XRT74L74 into Mode 6

1. Set the NibInt input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to 1X as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

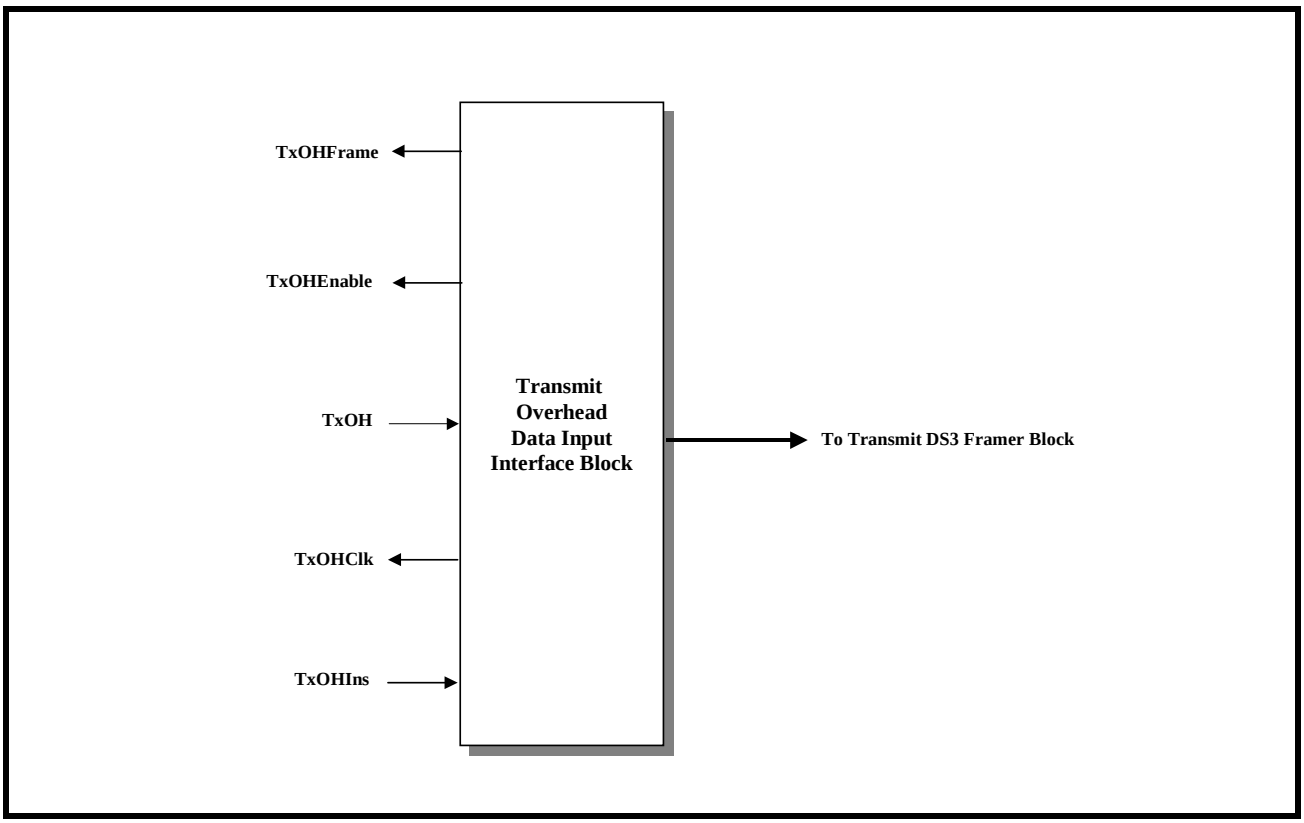
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	x

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 61.

5.2.2 The Transmit Overhead Data Input Interface

Figure 63 presents a simple illustration of the Transmit Overhead Data Input Interface block within the XRT74L74.

FIGURE 63. SIMPLE ILLUSTRATION OF THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK



The DS3 Frame consists of 4760 bits. Of these bits, 4704 bits are payload bits and the remaining 56 bits are overhead bits. The XRT74L74 has been designed to handle and process both the payload type and overhead type bits for each DS3 frame. Within the Transmit Section within the XRT74L74, the Transmit Payload Data Input Interface has been designed to handle the payload data. Likewise, the Transmit Overhead Data Input Interface has been designed to handle and process the overhead bits.

The Transmit Section of the XRT74L74 generates or processes the various overhead bits within the DS3 frame, in the following manner.

The Frame Synchronization Overhead Bits (e.g., the F and M bits)

The F and M bits are always internally generated by the Transmit Section of the XRT74L74. These overhead bits are used (by the Remote Terminal Equipment) for Frame Synchronization purposes. Hence, user values cannot be inserted for the F and M bits into the outbound DS3 data stream, via the Transmit Overhead Data Input Interface. Any attempt to externally insert values for the “F” and “M” bits, will be ignored by the Transmit Overhead Data Input Interface"High" block.

The Performance Monitoring Overhead Bits (P and CP Bits)

The P-bits are always internally generated by the Transmit Section of the XRT74L74. The “P” bits are used by the Remote Terminal Equipment to perform error-checking/detection of a DS3 data stream, as it is transmitted from one Terminal Equipment to adjacent Terminal Equipment (e.g., point-to-point checking). Hence, user values cannot be inserted for the P-bits into the outbound DS3 data stream, via the Transmit Overhead Data Input Interface.

In contrast to “P” bits, “CP” bits are used perform error-checking/detection of a DS3 data stream from the Source Terminal Equipment to the Sink Terminal Equipment. In applications where a given DS3 data stream is received via one port, and is output via another port, it is necessary that the “CP” bit-values remain constant. The only way to insure this to (1) extract out the “CP” bit values, via the Receiving Line Card and (2) insert these CP-bit values into the outbound DS3 data stream, via the Transmit Overhead Data Input Interface block. Hence, the Transmit Overhead Data Input Interface block will permit the user to externally insert the “CP” bits into the outbound DS3 data stream.

The Alarm and signaling related Overhead bits

Bits that are used to transport the alarm conditions can be either internally generated by the Transmit Section within the XRT74L74, or can be externally generated and inserted into the outbound DS3 data stream, via the Transmit Overhead Data Input Interface. The DS3 frame overhead bits that fall into this category are:

- The X bits
- The FEAC bits
- The FEBE bits.

The Data Link Related Overhead Bits

The DS3 frame structure also contains bits which can be used to transport User Data Link information and Path Maintenance Data Link information. The UDL (User Data Link) bits are only accessible via the Transmit Overhead Data Input Interface. The Path Maintenance Data Link (PMDL) bits can either be sourced from the Transmit LAPD Controller/Buffer or via the Transmit Overhead Data Input Interface.

Table 35 lists the Overhead Bits within the DS3 frame. Additionally, this table also indicates whether or not these overhead bits can be sourced by the Transmit Overhead Data Input Interface or not.

TABLE 35: A LISTING OF THE OVERHEAD BITS WITHIN THE DS3 FRAME, AND THEIR POTENTIAL SOURCES, WITHIN THE XRT74L74 IC

OVERHEAD BIT	INTERNALLY GENERATED	ACCESSIBLE VIA THE TRANSMIT OVERHEAD DATA INPUT INTERFACE	BUFFER/REGISTER ACCESSIBLE
P	Yes	No	Yes*
X	Yes	Yes	Yes
F	Yes	No	Yes*
M	Yes	No	Yes*
FEAC	No	Yes	Yes
FEBE	Yes	Yes	Yes
DL	No	Yes	Yes+
UDL	No	Yes	No
CP	No	Yes	No

NOTES:

** The XRT74L74 contains mask register bits that permit the altering tof he state of the internally generated value for these bits.*

+ The Transmit LAPD Controller/Buffer can be configured to be the source of the DL bits, within the outbound DS3 data stream.

In all, the Transmit Overhead Data Input Interface permits the insertion of overhead data into the outbound DS3 frames via the following two different methods.

- Method 1 - Using the TxOHClk clock signal
- Method 2 - Using the TxInClk and the TxOHEnable signals.

Each of these methods are described below.

5.2.2.1 4.2.2.1 Method 1 - Using the TxOHClk Clock Signal

The Transmit Overhead Data Input Interface consists of the five signals. Of these five (5) signals, the following four (4) signals are to be used when implementing Method 1.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

Each of these signals are listed and described below.

Table 36.

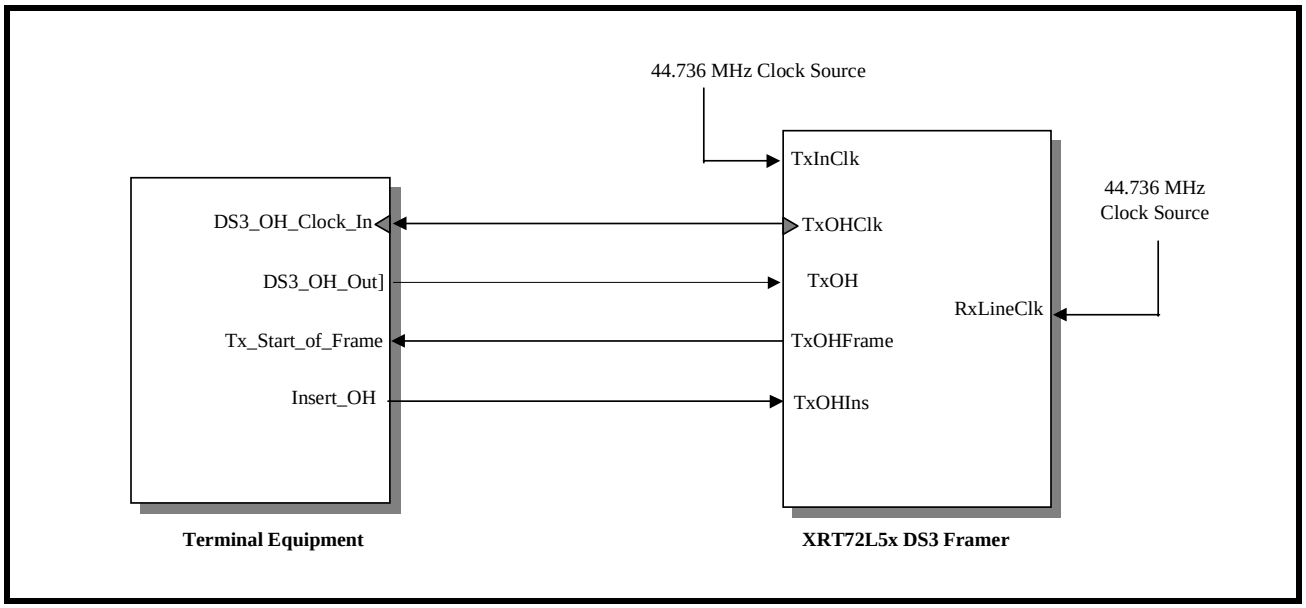
TABLE 36: DESCRIPTION OF METHOD 1 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next outbound DS3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.
TxOHClk	Output	**Transmit Overhead Input Interface Clock Output signal:** This output signal serves two purposes: 1. The Transmit Overhead Data Input Interface will provide a rising clock edge on this signal, one bit-period prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit. 2. The Transmit Overhead Data Input Interface will sample the data at the TxOH input, on the falling edge of this clock signal (provided that the TxOHIns input pin is "High"). ***NOTE:*** *The Transmit Overhead Data Input Interface will supply a clock edge for all overhead bits within the DS3 frame (via the TxOHClk output signal). This includes those overhead bits that the Transmit Overhead Data Input Interface will not accept from the Terminal Equipment.*
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given DS3 frame. The purpose of this output signal is to alert the Terminal Equipment that the Transmit Overhead Data Input Interface block is about to begin processing the overhead bits for a new DS3 frame.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment.

Figure 64 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment, when using Method 1.

FIGURE 64. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 1)



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the outbound DS3 data stream, (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of the TxOHFrame signal (e.g., the Tx_Start_of_Frame input signal) on the rising edge of the TxOHClk (e.g., the DS3_OH_Clock_In signal).
2. To keep track of the number of rising clock edges that have occurred, via the TxOHClk (e.g., the DS3_OH_Clock_In signal) since the last time the TxOHFrame signal was sampled "High". By doing this the Terminal Equipment will be able to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface block at any given time. When the Terminal Equipment knows which overhead bit is being processed, at a given TxOHClk period, it will know when to insert a desired overhead bit value into the outbound DS3 data stream. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin (of the XRT74L74).

Table 37relates the number of rising clock edges (in the TxOHClk signal, since TxOHFrame was sampled "High") to the DS3 Overhead Bit, that is being processed.

TABLE 37: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN TXOHCLK, (SINCE TXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING PROCESSED

NUMBER OF RISING CLOCK EDGES IN TXOHCLK	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (Clock edge is coincident with TxOHFrame being detected "High")	X	Yes
1	F1	No
2	AIC	Yes
3	F0	No
4	NA	Yes
5	F0	No
6	FEAC	Yes
7	F1	No
8	X	Yes
9	F1	No
10	UDL	Yes
11	F0	No
12	UDL	Yes
13	F0	No
14	UDL	Yes
15	F1	No
16	P	No
17	F1	No
18	CP	Yes
19	F0	No
20	CP	Yes
21	F0	No
22	CP	Yes
23	F1	No
24	P	No
25	F1	No
26	FEBE	Yes
27	F0	No
28	FEBE	Yes
29	F0	No
30	FEBE	Yes
31	F1	No
32	M0	No
33	F1	No
34	DL	Yes
35	F0	No
36	DL	Yes
37	F0	No

TABLE 37: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN TXOHCLK, (SINCE TXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING PROCESSED

NUMBER OF RISING CLOCK EDGES IN TXOHCLK	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
38	DL	Yes
39	F1	No
40	M1	No
41	F1	No
42	UDL	Yes
43	FO	No
44	UDL	Yes
45	FO	No
46	UDL	Yes
47	F1	No
48	M0	No
49	F1	No
50	UDL	Yes
51	F0	No
52	UDL	Yes
53	F0	No
54	UDL	Yes
55	F1	No

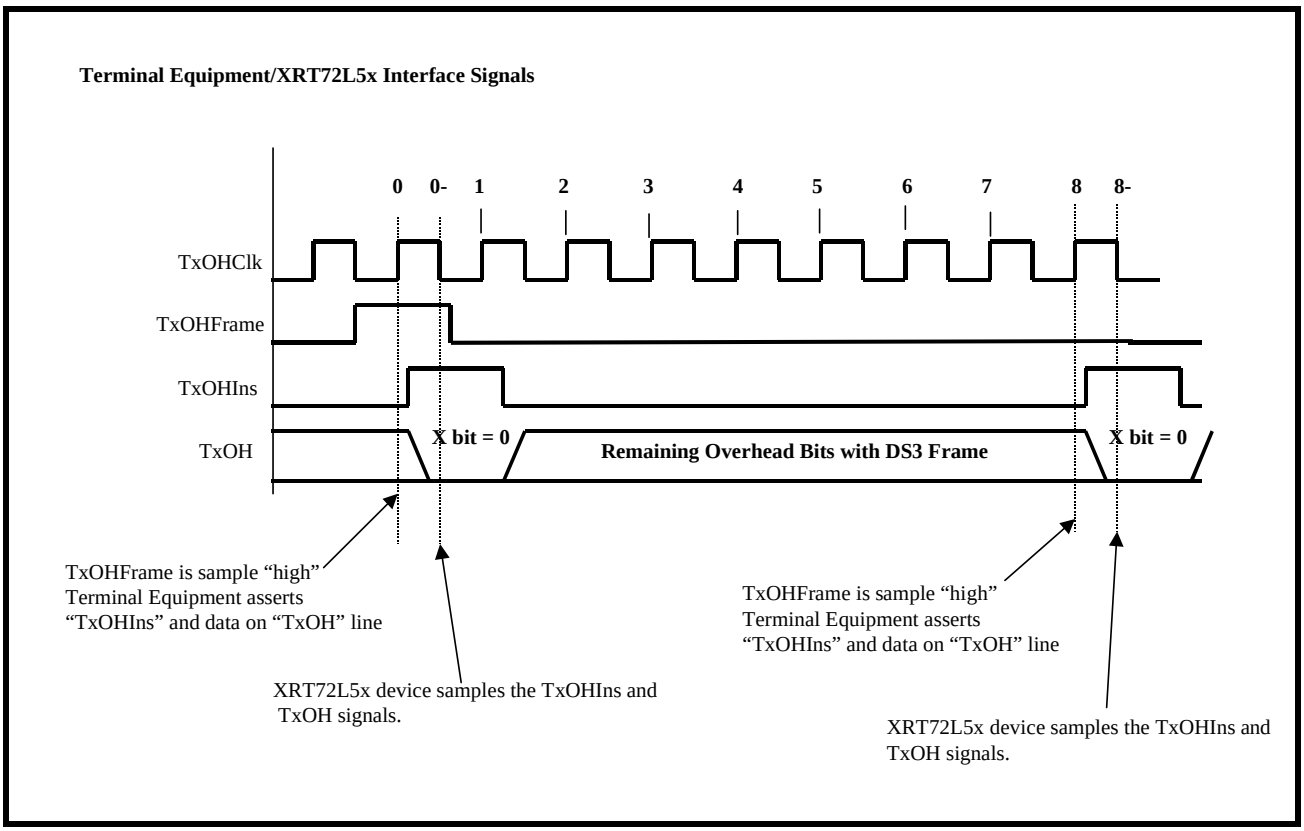
3. After the Terminal Equipment has waited the appropriate number of clock edges (from the TxOHFrame signal being sampled "High"), it should assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal, stable until the next rising edge of TxOHClk is detected.

Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using Method 1) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this example, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface, such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for DS3 Applications, a Yellow Alarm is transmitted by setting both of the X bits (within each outbound DS3 frame) to 0.

If one assumes that the connection between the Terminal Equipment and the XRT74L74 are as illustrated in Figure 64 then Figure 65 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 65. ILLUSTRATION OF THE SIGNAL THAT MUST OCCUR BETWEEN THE TERMINAL EQUIPMENT AND THE XRT74L74, IN ORDER TO CONFIGURE THE XRT74L74 TO TRANSMIT A YELLOW ALARM TO THE REMOTE TERMINAL EQUIPMENT



In Figure 65 the Terminal Equipment samples the TxOHFrame signal being "High" at the rising clock edge # 0. At this point, the Terminal Equipment knows that the XRT74L74 is just about to process the very first overhead bit within a given outbound DS3 frame. Additionally, according to Table 37, the very first overhead bit to be processed is the first X bit. In order to facilitate the transmission of the Yellow Alarm, the Terminal Equipment must set this X bit to 0. Hence, the Terminal Equipment starts this process by implementing the following steps concurrently.

a. Assert the TxOHIns input pin by setting it "High".

b. Set the TxOH input pin to 0.

After the Terminal Equipment has applied these signals, the XRT74L74 will sample the data on both the TxOHIns and TxOH signals upon the very next falling edge of TxOHClk (designated at 0- in Figure 65. Once the XRT74L74 has sampled this data, it will then insert a "0" into the first X bit position, in the outbound DS3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as clock edge 1 in Figure 65), the Terminal Equipment will negate the TxOHIns signal (e.g., toggles it "Low") and will cease inserting data into the Transmit Overhead Data Input Interface, until rising clock edge # 8 (of the TxOHClk signal). According to Table 37, rising clock edge # 8 indicates that the XRT74L74 is just about ready to process the second X bit within the outbound DS3 frame. Once again, in order to facilitate the transmission of the Yellow Alarm this X-Bit must also be set to 0. Hence, the Terminal Equipment will (once again) implement the following steps, concurrently.

a. Assert the TxOHIns input pin by setting it "High".

b. Set the TxOH input to 0.

Once again, after the Terminal Equipment has applied these signals, the XRT74L74 will sample the data on both the TxOHIns and TxOH signal upon the very next falling edge of TxOHClk (designated as 8- in Figure 65). Once the XRT74L74 has sampled this data, it will then insert a "0" into the second X bit position, in the outbound DS3 frame.

5.2.2.2 Method 2 - Using the TxInClk and TxOHEnable Signals

Method 1 requires the use of an additional clock signal, TxOHClk. However, there may be a situation in which the user does not wish to accommodate and process this extra clock signal to their design, in order to use the Transmit Overhead Data Input Interface. Hence, Method 2 is available. When using Method 2, either the TxInClk or RxOutClk signal is used to sample the overhead bits and signals which are input to the Transmit Overhead Data Input Interface. Method 2 involves the use of the following signals:

- TxOH
- TxInClk
- TxOHFrame
- TxOHEnable

Each of these signals are listed and described in Table 38.

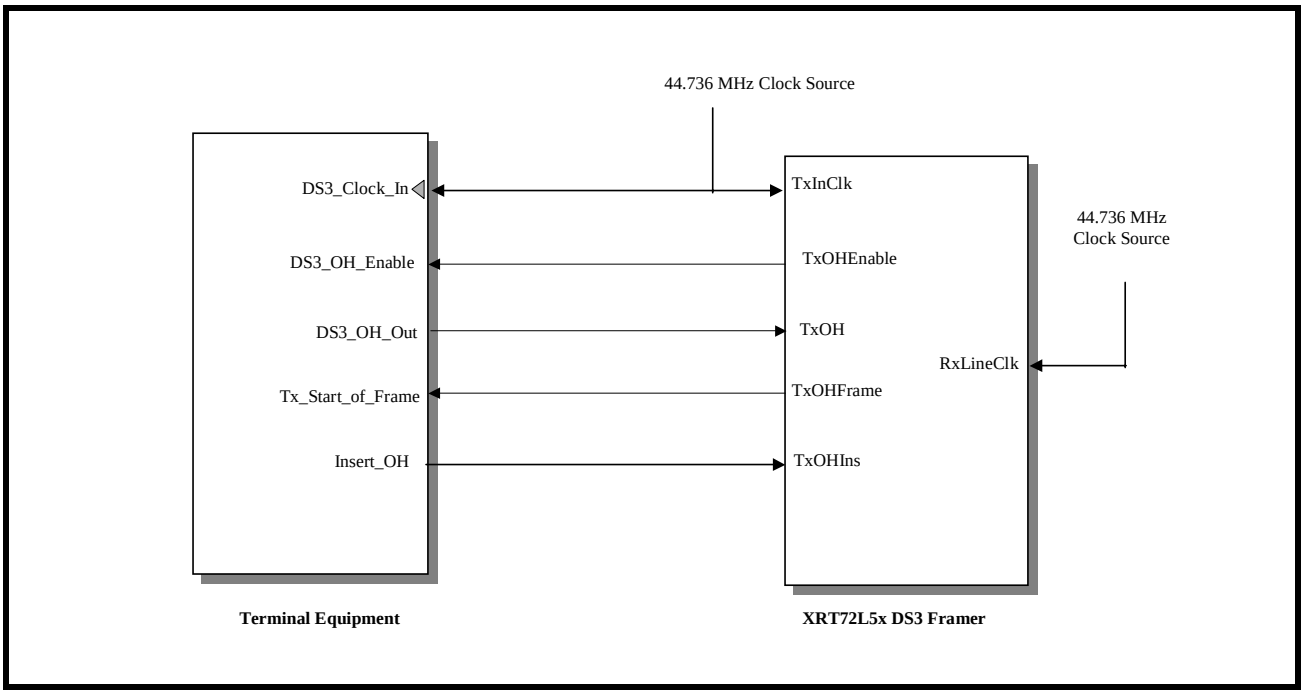
TABLE 38: DESCRIPTION OF METHOD 2 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHEnable	Output	**Transmit Overhead Data Enable Output pin** The XRT74L74 will assert this signal, for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit.
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given DS3 frame.
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxInClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next outbound DS3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment

Figure 66 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment when using Method 2.

FIGURE 66. ILLUSTRATION OF THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 2)



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the outbound DS3 data stream (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of both the TxOHFrame and the TxOHEnable input signals, via the DS3_Clock_In (e.g., either the TxInClk or the RxOutClk signal of the XRT74L74) signal. If the Terminal Equipment samples the TxOHEnable signal "High", then it knows that the XRT74L74 is about to process an overhead bit. Further, if the Terminal Equipment samples both the TxOHFrame and the TxOHEnable pins "High" (at the same time) then the Terminal Equipment knows that the XRT74L74 is about to process the first overhead bit, within a new DS3 frame.
2. To keep track of the number of times that the TxOHEnable signal has been sampled "High" since the last time both the TxOHFrame and the TxOHEnable signals were sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit the Transmit Overhead Data Input Interface is about ready to process. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pins (of the XRT74L74).

Table 39 also relates the number of TxOHEnable output pulses (that have occurred since both the TxOHFrame and TxOHEnable pins were sampled "High") to the DS3 overhead bit, that is being processed.

TABLE 39: THE RELATIONSHIP BETWEEN THE NUMBER OF TXOHENABLE PULSES (SINCE THE LAST OCCURRENCE OF THE TXOHFRAME PULSE) TO THE DS3 OVERHEAD BIT, THAT IS BEING PROCESSED BY THE XRT74L74

NUMBER OF TXOHENABLE PULSES	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (The TxOHEnable and TxOHFrame signals are both sampled "High")	X	Yes
1	F1	No
2	AIC	Yes
3	F0	No
4	NA	Yes
5	F0	No
6	FEAC	Yes
7	F1	No
8	X	Yes
9	F1	No
10	UDL	Yes
11	F0	No
12	UDL	Yes
13	F0	No
14	UDL	Yes
15	F1	No
16	P	No
17	F1	No
18	CP	Yes
19	F0	No
20	CP	Yes
21	F0	No
22	CP	Yes
23	F1	No
24	P	No
25	F1	No
26	FEBE	Yes
27	F0	No
28	FEBE	Yes
29	F0	No
30	FEBE	Yes
31	F1	No
32	M0	No
33	F1	No
34	DL	Yes
35	F0	No
36	DL	Yes
37	F0	No

TABLE 39: THE RELATIONSHIP BETWEEN THE NUMBER OF TXOHENABLE PULSES (SINCE THE LAST OCCURRENCE OF THE TXOHFRAME PULSE) TO THE DS3 OVERHEAD BIT, THAT IS BEING PROCESSED BY THE XRT74L74

NUMBER OF TXOHENABLE PULSES	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
38	DL	Yes
39	F1	No
40	M1	No
41	F1	No
42	UDL	Yes
43	FO	No
44	UDL	Yes
45	FO	No
46	UDL	Yes
47	F1	No
48	M0	No
49	F1	No
50	UDL	Yes
51	F0	No
52	UDL	Yes
53	F0	No
54	UDL	Yes
55	F1	No

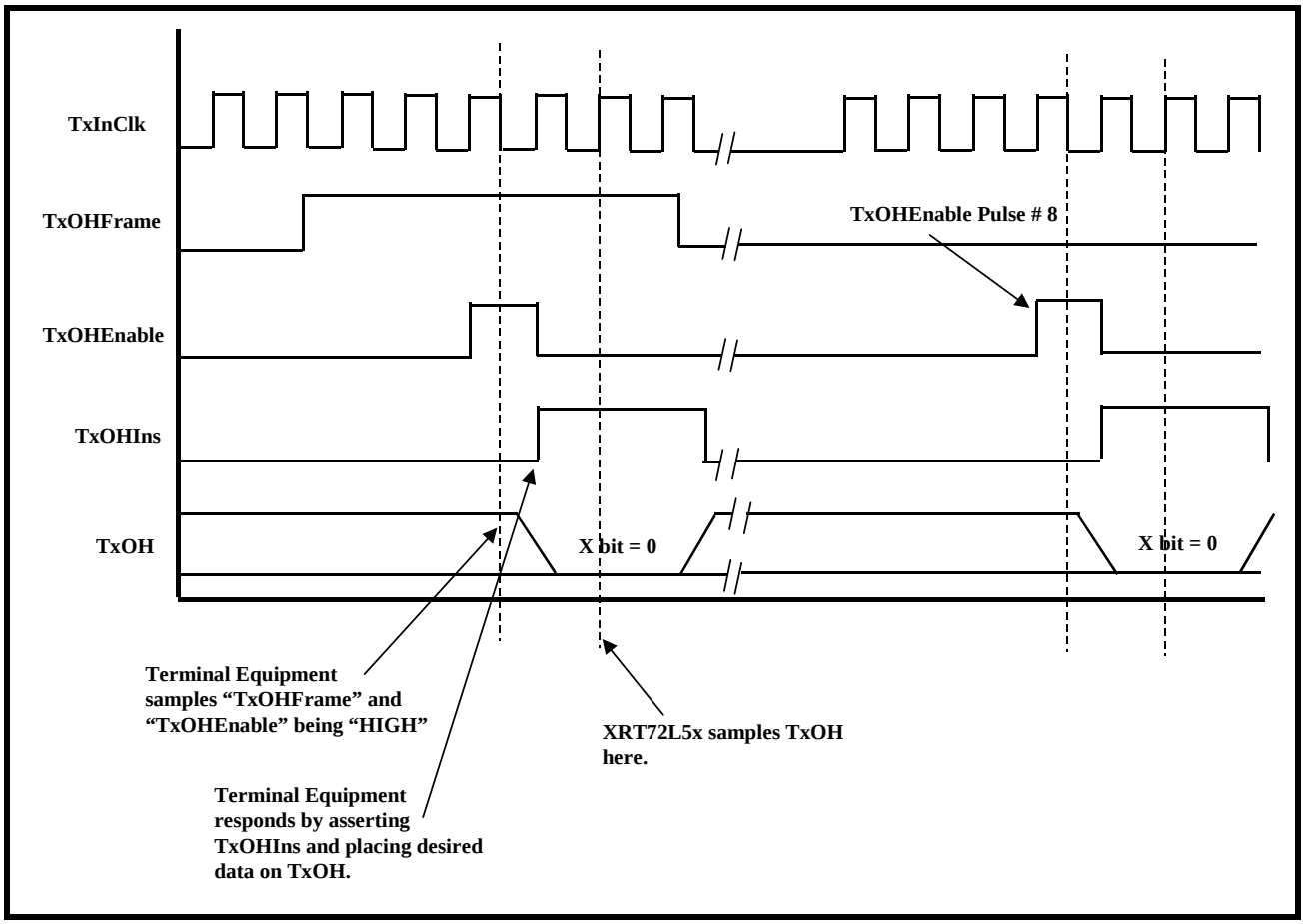
3. After the Terminal Equipment has waited through the appropriate number of pulses via the TxOHEnable pin, it should then assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal stable, until the next TxOHEnable pulse is detected.

Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using Method 2) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this case, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for DS3 applications, a Yellow Alarm is transmitted by setting all of the X bits to 0.

If one assumes that the connection between the Terminal Equipment and the XRT74L74 is as illustrated in Figure 66 then, Figure 67 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 67. BEHAVIOR OF TRANSMIT OVERHEAD DATA INPUT INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (FOR METHOD 2)



5.2.3 The Transmit DS3 HDLC Controller

The Transmit DS3 HDLC Controller block can be used to transport either Bit-Oriented Signaling (BOS) or Message-Oriented Signaling (MOS) type messages or both types of messages to the remote terminal equipment. Both BOS and MOS types of HDLC message processing are discussed in detail below.

5.2.3.1 Bit-Oriented Signaling (or FEAC Message) processing via the Transmit DS3 HDLC Controller.

The Transmit DS3 HDLC Controller block consists of two major blocks:

- The Transmit FEAC Processor.
- The LAPD Transmitter.

This section describes how to operate the Transmit FEAC Processor. If the Transmit DS3 Framer is operating in the C-bit Parity Framing Format then the FEAC (Far-End Alarm & Control) bit-field of the DS3 Frame can be used to transmit the FEAC messages (See Figure 42). The FEAC code word is a 6-bit value which is encapsulated by 10 framing bits, forming a 16-bit FEAC message of the form:

0	d5	d4	d3	d2	d1	d0	0	1	1	1	1	1	1	1	1

where '[d5, d4, d3, d2, d1, d0]' is the FEAC code word. The rightmost bit (e.g., a 1) of the FEAC Message, is transmitted first. Since each DS3 frame contains only 1 FEAC bit, 16 DS3 Frames are required to transmit the 16 bit FEAC Code Message.

The XRT74L74 contains the following two registers that support FEAC Message Transmission.

- Tx DS3 FEAC Register (Address = 0x32)
- Tx DS3 FEAC Configuration and Status Register (Address = 0x31)

Operating the Transmit FEAC Processor

In order to transmit a FEAC message to the remote terminal, the following steps must be executed.

1. Write the 6-bit FEAC code (to be sent) into the Tx DS3 FEAC Register.
2. Enable the Transmit FEAC Processor.
3. Initiate the Transmission of the FEAC Message.

Each of these steps will be described in detail below.

STEP 1 - Writing in the six bit FEAC Codeword (to be sent)

In this step, the µP/µC writes the six bit FEAC code word into the Tx DS3 FEAC Register. The bit format of this register is presented below.

TX DS3 FEAC REGISTER (ADDRESS = 0X32)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	TxFEAC[5]	TxFEAC[4]	TxFEAC[3]	TxFEAC[2]	TxFEAC[1]	TxFEAC[0]	Not Used
RO	R/W	R/W	R/W	R/W	R/W	R/W	R0
0	d5	d4	d3	d2	d1	d0	0

STEP 2 - Enabling the Transmit FEAC Processor

In order to enable the Transmit FEAC Processor (within the Transmit DS3 HDLC Controller block) a “1” must be written into bit 2 (Tx FEAC Enable) within the Tx DS3 FEAC Configuration and Status Register, as depicted below.

TRANSMIT DS3 FEAC CONFIGURATION AND STATUS REGISTER (ADDRESS = 0X31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	Not Used	Not Used	TxFEAC Interrupt Enable	TxFEAC Interrupt Status	TxFEAC Enable	TxFEAC Go	TxFEAC Busy
RO	RO	RO	R/W	RUR	R/W	R/W	R0
x	x	x	x	x	1	X	X

At this point, the Transmit FEAC Processor can be commanded to begin transmission (See STEP 3).

STEP 3 - Initiate the Transmission of the FEAC Message

The transmission of the FEAC code word (residing in the Tx DS3 FEAC register) can be initiated by writing a “1” to bit 1 (Tx FEAC Go) within the Tx DS3 FEAC Configuration and Status register, as depicted below.

TRANSMIT DS3 FEAC CONFIGURATION AND STATUS REGISTER (ADDRESS = 0X31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	Not Used	Not Used	TxFEAC Interrupt Enable	TxFEAC Interrupt Status	TxFEAC Enable	TxFEAC Go	TxFEAC Busy
RO	RO	RO	R/W	RUR	R/W	R/W	R0
x	x	x	x	x	1	1	X

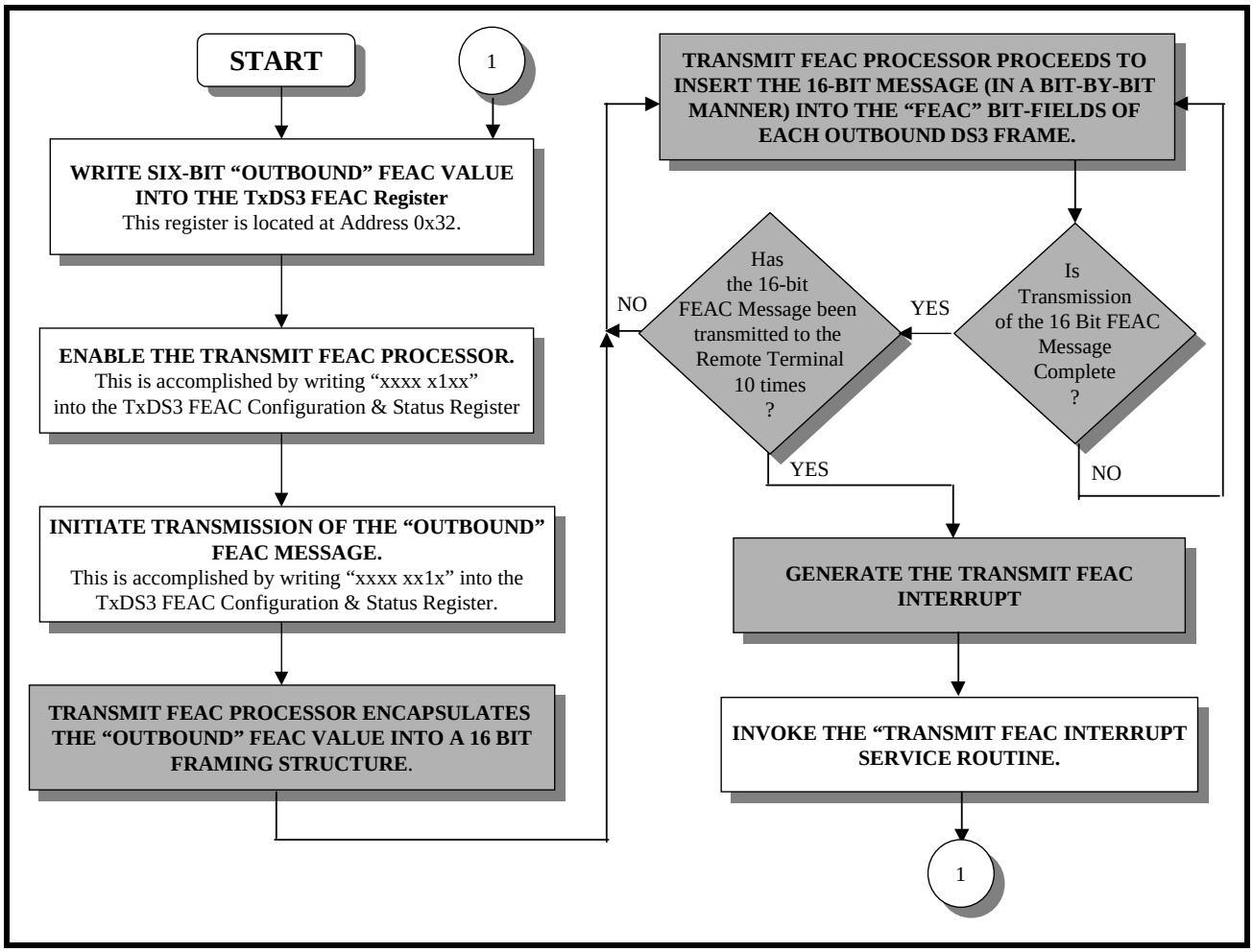
NOTE: *While executing this particular write operation, the binary value “000xx110b” should be written into the Tx DS3 FEAC Configuration and Status Register. This insures that a “1” is also being written to Bit 2 (Tx FEAC Enable) of the register, in order to keep the Transmit FEAC Processor enabled.*

Once this step has been completed, the Transmit FEAC Processor will proceed to transmit the 16 bit FEAC code via the outbound DS3 frames. This 16 bit FEAC message will be transmitted repeatedly 10 consecutive times. Hence, this process will require a total of 160 DS3 Frames. During this process the Tx FEAC Busy bit (Bit 0, within the Transmit DS3 FEAC Configuration and Status register) will be asserted, indicating that the Tx FEAC Processor is currently transmitting the FEAC Message to the remote Terminal. This bit-field will toggle to "0" upon completion of the 10th transmission of the FEAC Code Message. The Transmit FEAC Processor will generate an interrupt (if enabled) to the local µP/µC, upon completion of the 10th transmission of the FEAC Message. The purpose of having the Framer IC generating this interrupt is to let the local µP/µC know that the Transmit FEAC Processor is now available and ready to transmit a new FEAC message. Finally, once the Transmit FEAC Processor has completed its 10th transmission of a FEAC Code Message it will then begin sending all 1s in the FEAC bit-field of each DS3 Frame. The Receive FEAC Processor (at the remote terminal

equipment) will interpret this all 1s message as an Idle FEAC Message. The Transmit FEAC Processor will continue sending all 1s in the FEAC bit field, for an indefinite period of time, until the local µP/µC commands it to transmit a new FEAC message.

Figure 68 presents a flow chart depicting how to use the Transmit FEAC Processor.

FIGURE 68. A FLOW CHART DEPICTING HOW TO TRANSMIT A FEAC MESSAGE VIA THE FEAC TRANSMITTER



NOTE: *For a detailed description of the Receive FEAC Processor (within the Receive DS3 HDLC Controller block), please see Section 5.3.3.1.*

5.2.3.2 Message-Oriented Signaling (e.g., LAP-D) processing via the Transmit DS3 HDLC Controller

The LAPD Transmitter (within the Transmit DS3 HDLC Controller Block) allows the user to transmit Path Maintenance Data Link (PMDL) messages to the remote terminal via the outbound DS3 Frames. In this case the message bits are inserted into and carried by the 3 DL bit fields of F-Frame #5 within each DS3 M-frame. The on-chip LAPD transmitter supports both the 76 byte and 82 byte length message formats, and the Framer IC allocates 88 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer) to store the message to be transmitted. The message format complies with ITU-T Q.921 (LAP-D) protocol with different addresses and is presented below in Figure 69.

FIGURE 69. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following sections defines each of these bit/byte-fields within the LAPD Message Frame Format.

Flag Sequence Byte

The Flag Sequence byte is of the value 0x7E, and is used to for two purposes

1. To denote the boundaries of the LAPD Message Frame, and
2. To function as the Idle Pattern (e.g., Transmit HDLC Controller block transmits a continuous stream of flag sequence octets, whenever no LAPD Message is being transmitted).

SAPI - Service Access Point Identifier

The SAPI bit-fields are assigned the value of 001111b or 15 (decimal).

TEI - Terminal Endpoint Identifier

The TEI bit-fields are assigned the value of 0x00. The TEI field is used in N-ISDN systems to identify a terminal out of multiple possible terminal. However, since the Framer IC 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 Framer assigns the Control byte the value 0x03. Hence, the Framer 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 has been written into the on-chip Transmit LAPD Message buffer (which is located at addresses 0x86 through 0xDD).

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 = 0x86, within the Framer). The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. Table 40 presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT74L74 Framer and the corresponding octet value that the user must write into the first octet position within the Transmit LAPD Message buffer.

TABLE 40: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD

LAPD MESSAGE TYPE	VALUE OF FIRST BYTE, WITHIN INFORMATION PAYLOAD OF MESSAGE	MESSAGE SIZE
CL Path Identification	0x32	76 bytes
IDLE Signal Identification	0x34	76 bytes
Test Signal Identification	0x38	76 bytes
ITU-T Path Identification	0x3F	82 bytes

Frame Check Sequence Bytes

The 16 bit FCS (Frame Check Sequence) is calculated over the LAPD Message Header and Information Payload bytes, by using the CRC-16 polynomial, $x^{16} + x^{12} + x^5 + 1$.

Operation of the LAPD Transmitter

If a message is to be transmitted via the LAPD Transmitter, the information portion (or the body) of the message must be written into the Transmit LAPD Message Buffer, which is located at 0x86 through 0xDD in on-chip RAM via the Microprocessor Interface. Afterwards, three things must be done:

1. Specify the length of LAPD message to be transmitted.
2. Enable the LAPD Transmitter.
3. Initiate the Transmission of the PMDL Message.

Each of these steps will be discussed in detail.

STEP 1 - Specifying the Length of the LAPD Message

One of two different sizes of LAPD Messages can be transmitted. This is accomplish by writing the appropriate data to bit 1 within the Tx DS3 LAPD Configuration Register. The bit-format of this register is presented below.

TRANSMIT DS3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	X

The relationship between the contents of bit-fields 1 and the LAPD Message size is given in Table 41.

TABLE 41: RELATIONSHIP BETWEEN TXLAPD MSG LENGTH AND THE LAPD MESSAGE SIZE

TXLAPD MSG LENGTH	LAPD MESSAGE LENGTH
0	LAPD Message size is 76 bytes
1	LAPD Message size is 82 bytes

NOTE: *The Message Type selected must correspond with the contents of the first byte of the Information (Payload) portion, as presented in Table 40.*

STEP 2 - Enabling the LAPD Transmitter

Prior to the transmission of any data via the LAPD Transmitter the LAPD Transmitter must be enabled. This is accomplished this by writing a 1 to bit 0 of the Tx DS3 LAPD Configuration Register, as depicted below.

TRANSMIT DS3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	1

Bit 0 - TxLAPD Enable

This bit-field allows the user to enable or disable the LAPD Transmitter in accordance with Table 42.

TABLE 42: RELATIONSHIP BETWEEN TXLAPD MSG LENGTH AND THE LAPD MESSAGE SIZE

TXLAPD ENABLE	RESULTING ACTION OF THE LAPD TRANSMITTER
0	The LAPD Transmitter is disabled and the DL bits, in the DS3 frame, are transmitted as all 1s.
1	The LAPD Transmitter is enabled and is transmitting a continuous stream of Flag Sequence octets (0x7E).

Prior to executing step 2 (Enabling the LAPD Transmitter), the LAPD Transmitter will be disabled and the Transmit DS3 Framer block will be setting each of the DL bits (within the outbound DS3 data stream) to 1. After this step is executed, the LAPD Transmitter will begin transmitting the flag sequence octet (0x7E) via the DL bits.

NOTE: *Upon power up or reset, the LAPD Transmitter is disabled. Therefore, this bit must be set to a "1" in order to enable the LAPD Transmitter.*

STEP 3 - Initiate the Transmission

At this point, the LAPD Transmitter is ready to begin transmission. The user has written the information portion of the PMDL message into the on-chip Transmit LAPD Message buffer. Further, the user has specified the type of LAPD message that is wished to be transmitted, and has enabled the LAPD Transmitter. The only thing remaining to do is to initiate the transmission of this message. This process is initiated by writing a "1" to Bit 3 of the Tx DS3 LAPD Status/Interrupt Register (TxDL Start). The bit format of this register is presented below.

TRANSMIT DS3 LAPD STATUS/INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Tx DL Start	Tx DL Busy	TxLAPD Interrupt Enable	TxLAPD Interrupt Status
R/O	R/O	R/O	R/O	R/W	RO	R/W	RUR
0	0	0	0	1	X	X	X

A "0" to "1" transition of Bit 3 (TxDL Start) in this register, initiates the transmission of the data link message. While the LAPD transmitter is transmitting the message, the 'TxDL Busy' (bit 2) bit will be set to 1. This bit-field allows the user to poll the status of the LAPD Transmitter. Once the message transfer is completed, this bit-field will toggle back to '0'.

The LAPD Transmitter can be configured to interrupt the µC/µP upon completion of transmission of the LAPD Message, by setting bit-field 1 (TxLAPD Interrupt Enable) of the Tx DS3 LAPD Status/Interrupt register to 1. The purpose of this interrupt is to let the local µC/µP know that the LAPD Transmitter is available and ready to transmit a new message. Bit 0 will reflect the interrupt status for the LAPD Transmitter.

NOTE: *This bit-field will be reset on reading this register.*

Details Associated with the Transmission of a PMDL Message

Once the user has invoked the TxDL Start command, the LAPD Transmitter will do the following.

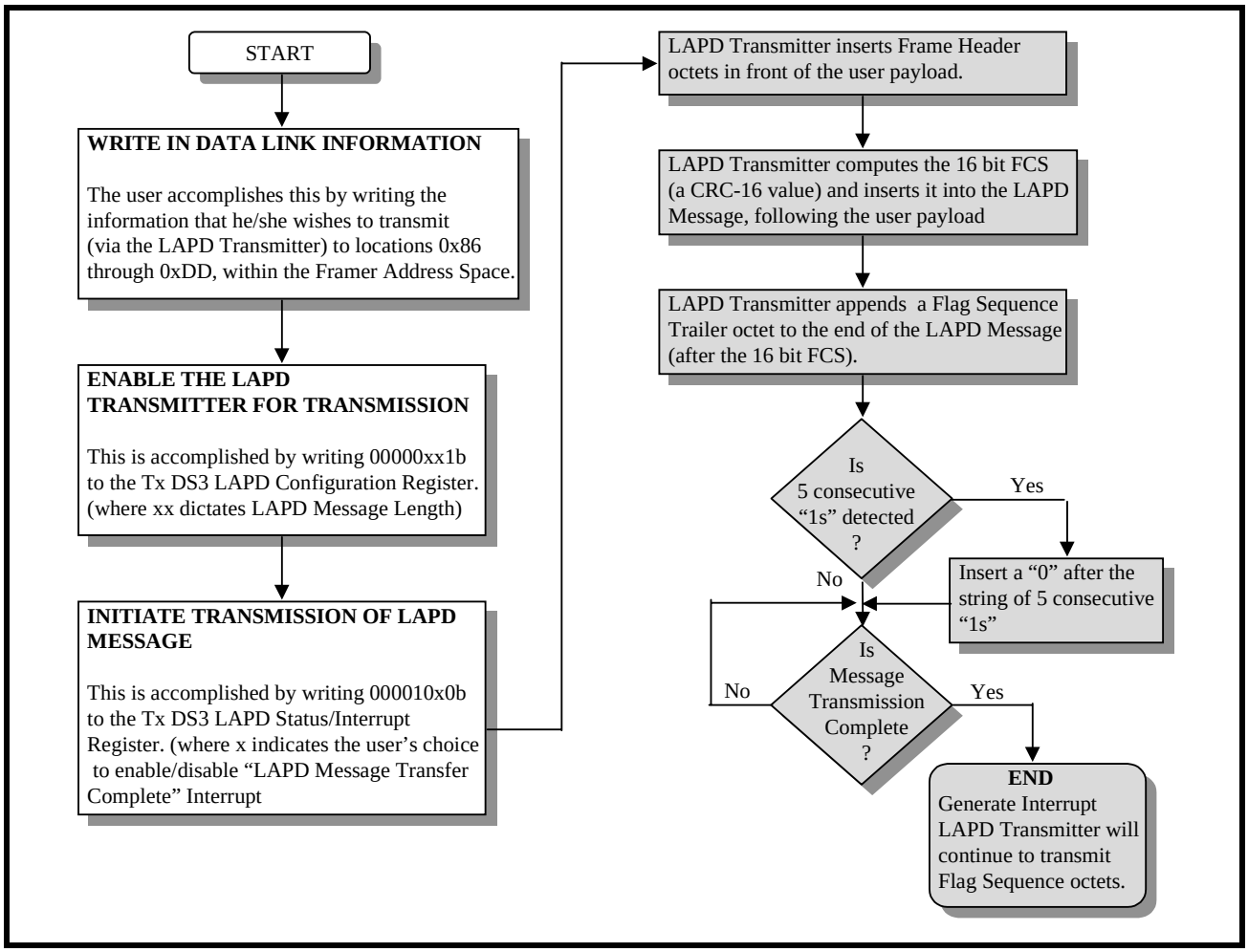
- Generate the four octets of LAPD frame header (e.g., Flag Sequence, SAPI, TEI, Control, etc.) and insert it into the LAPD Message, prior to the user's information (see the LAPD Message Frame Format in Figure 69).
- Compute the 16 bit Frame Check Sum (FCS) of the LAPD Message Frame (e.g., of the LAPD Message header and information payload) and append this value to the LAPD Message.
- Append a trailer Flag Sequence octet to the end of the message LAPD (following the 16 bit FCS value).
- Serialize the composite LAPD message and begin inserting the LAPD message into the DL bit fields of each outgoing DS3 Frame.
- Complete the transmission of the frame overhead, payload, FCS value, and trailer Flag Sequence octet via the Transmit DS3 Framer.

Once the LAPD Transmitter has completed its transmission of the LAPD Message, the Framer will generate an interrupt to the local µC/µP (if enabled). Afterwards, the LAPD Transmitter will proceed to retransmit the LAPD Message, repeatedly at one second intervals. During Idle periods (e.g., in between these transmission of the LAPD Message), the LAPD Transmitter will be sending a continuous stream of Flag Sequence Bytes. The LAPD Transmitter will continue this behavior until the user has disabled the LAPD Transmitter by writing a "0" to bit 0 (TxLAPD Enable) within the Tx DS3 LAPD Configuration Register. If the LAPD Transmitter is idle, then it will continuously send the Flag Sequence octets (via the DL bits of each outbound DS3 Frame) to the remote terminal equipment.

NOTE: *In order to prevent the user's data (e.g., the payload portion of the LAPD Message Frame) from mimicking the Flag Sequence byte, the LAPD Transmitter will insert a "0" into the LAPD data stream immediately following the detection of five (5) consecutive 1s (this stuffing occurs only while the information payload is being transmitted). The 'remote' LAPD Receiver (see Section 5.3.3.2) will have the responsibility of detecting the 5 consecutive 1s and removing the subsequent "0" from the payload portion of the incoming LAPD message.*

Figure 70 presents a flow chart depicting the procedure (in white boxes) that the user should use in order to transmit a LAPD message. This figure also indicates (via the shaded boxes) what the LAPD Transmitter circuitry will do before and during message transmission.

FIGURE 70. FLOW CHART DEPICT HOW TO USE THE LAPD TRANSMITTER



The Mechanics of Transmitting a New LAPD Message

As mentioned above, after the LAPD Transmitter has been enabled, and commanded to transmit the message, residing in the Transmit LAPD Message buffer, it will continue to transmit this message at one-second intervals. If another (e.g., different) PMDL message is to be transmitted to the Remote LAPD Receiver, the new message will have to be written into the Transmit LAPD Message buffer, via the Microprocessor Interface section of the Framer. However, care must be taken when writing in this new message. If this message is written into the Transmit LAPD Message buffer at the wrong time (with respect to these one-second transmissions), the user's action could interfere with these transmissions, thereby causing the LAPD Transmitter to transmit a corrupted message to the Remote LAPD Receiver. In order to avoid this problem, while writing the new message into the Transmit LAPD Message buffer, the following should be done:

1. Configure the Framer to automatically reset activated interrupts

This can be done by writing a "1" into Bit 3 of the Framer Operating Mode Register, as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	1	0	1	X	X	X

This action will prevent the LAPD Transmitter from generating its own one-second interrupts.

This can be done by writing a "1" into Bit 0 of the Block Interrupt Enable Register, as depicted below.

2. Enable the One-Second Interrupt

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	0	X

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 transmissions of the LAPD Message, and will transmit the correct messages to the remote LAPD Receiver.

5.2.4 The Transmit DS3 Framer Block

5.2.4.1 Brief Description of the Transmit DS3 Framer

The Transmit DS3 Framer block accepts data from any of the following three sources, and uses it to form the DS3 data stream.

- The Transmit Payload Data Input block
- The Transmit Overhead Data Input block
- The Transmit HDLC Controller block
- The Internal Overhead Data Generator

The manner in how the Transmit DS3 Framer block handles data from each of these sources is described below.

Handling of data from the Transmit Payload Data Input Interface

For DS3 applications, all data that is input to the Transmit Payload Data Input Interface will be inserted into the payload bit positions within the outbound DS3 frames.

Handling of data from the Internal Overhead Bit Generator

By default, the Transmit DS3 Framer block will internally generate the overhead bits. However, if the Terminal Equipment inserts its own values for the overhead bits (via the Transmit Overhead Data Input Interface) or, if the user enables and employs the Transmit DS3 HDLC Controller block, then these internally generated overhead bits will be overwritten.

Handling of data from the Transmit Overhead Data Input Interface

For DS3 applications, the Transmit DS3 Framer block automatically generates and inserts the framing alignment bits (e.g., the F and M bits) into the outbound DS3 frames. Further, the Transmit DS3 Framer block will automatically compute and insert the P-bits into the outbound DS3 frames. Hence, the Transmit DS3 Framer block will not accept data from the Transmit OH Data Input Interface block for the F, M and P bits.

However, the Transmit DS3 Framer block will accept (and insert) data from the Transmit Overhead Data Input Interface for the following bit-fields.

- X-bits
- FEBE bits
- FEAC bits
- DL bits
- UDL bits
- CP bits

If the user's local Data Link Equipment activates the Transmit Overhead Data Input Interface block and writes data into this interface for these bits, then the Transmit DS3 Framer block will insert this data into the appropriate overhead bit-fields, within the outbound DS3 frames.

Handling of Data from the Transmit HDLC Controller block

The exact manner in how the Transmit DS3 Framer handles data from the Transmit HDLC Controller block depends upon whether the Transmit HDLC Controller is transmitting BOS (Bit Oriented Signaling) or MOS (Message Oriented Signaling) data.

If the Transmit DS3 HDLC Controller block is not activated, then the Transmit DS3 Framer block will insert a "1" into each FEAC and "DL" bit-field, within each outbound DS3 frame.

If the Transmit DS3 HDLC Controller block is activated, and is configured to transmit either a "BOS" or "MOS" type message, then data will be inserted into the FEAC and "DL" bit-fields as described in Section 5.2.3.

5.2.4.2 Detailed Functional Description of the Transmit DS3 Framer Block

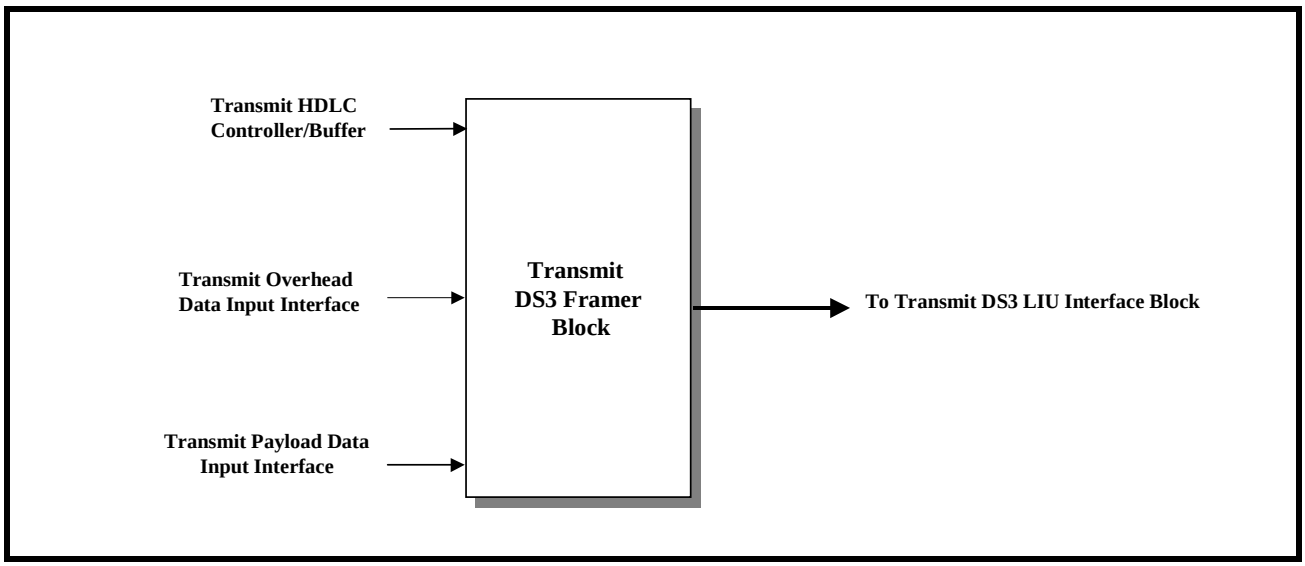
The Transmit DS3 Framer receives data from the following three sources and combines them together to form a DS3 data stream.

- The Transmit Payload Data Input Interface block.
- The Transmit Overhead Data Input Interface block
- The Transmit HDLC Controller block.

Afterwards, this DS3 data stream will be routed to the Transmit DS3 LIU Interface block, for further processing.

Figure 71 presents a simple illustration of the Transmit DS3 Framer block, along with the associated paths to the other functional blocks within the chip.

FIGURE 71. A SIMPLE ILLUSTRATION OF THE TRANSMIT DS3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO OTHER FUNCTIONAL BLOCKS



In addition to taking data from multiple sources and multiplexing them, in appropriate manner, to create the outbound DS3 frames, the Transmit DS3 Framer block has the following roles.

- Generating Alarm Conditions
- Generating Errored Frames (for testing purposes)
- Routing outbound DS3 frames to the Transmit DS3 LIU Interface block

Each of these additional roles are discussed below.

5.2.4.2.1 Generating Alarm Conditions

By writing the appropriate data into the on-chip registers, the Transmit DS3 Framer block permits the user to override the data that is being written into the Transmit Payload Data and Overhead Data Input Interfaces and transmit the following alarm conditions.

- Generate the Yellow Alarms (or FERF indicators)
- Manipulate the X-bit (set them to 1)
- Generate the AIS Pattern
- Generate the IDLE pattern
- Generate the LOS pattern
- Generate FERF (Yellow) Alarms, in response to detection of a Red Alarm condition (via the Receive Section of the XRT74L74).

- Generate and transmit a desired value for FEBE (Far-End-Block Error).

The procedure and results of generating any of these alarm conditions is presented below.

Each of these options can be exercised by writing the appropriate data to the Tx DS3 Configuration Register (Address = 0x30). The bit format of this register is presented below.

TX DS3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bit	Tx IDLE Pattern	Tx AIS Pattern	Tx LOS Pattern	FERF on LOS	FERF on OOF	FERF on AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	1

The role/function of each of these bit-fields within the register, are discussed below.

5.2.4.2.1.1 Transmit Yellow Alarm - Bit 7

This read/write bit field permits the user to force the transmission of a Yellow Alarm to the remote terminal equipment via software control. If the user opts to transmit a Yellow Alarm then both of the X-bits, within the outbound DS3 frames will be set to '0'. Table 43 relates the content of this bit field to the Transmit DS3 Framer block's action.

TABLE 43: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 7 (TX YELLOW ALARM) WITHIN THE TX DS3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT DS3 FRAMER BLOCK'S ACTION

BIT 7	TRANSMIT DS3 FRAMER'S ACTION
0	**Normal Operation:** The X-bits are generated by the Transmit DS3 Framer block based upon Near End Receiving Conditions (as detected by the Receive Section of the chip)
1	**Transmit Yellow Alarm:** The Transmit DS3 Framer block will overwrite the X-bits by setting them all to 0. The payload information is not modified and is transmitted as normal.

NOTE: *This bit is ignored when either the TxIDLE, TxAIS, or the TxLOS bit-fields are set.*

5.2.4.2.1.2 Transmit X-bit - Bit 6

This bit field functions as the logical complement to Bit 7 (e.g., Tx Yellow Alarm). This read/write bit field permits the user to force all of the X-bits, in the outbound DS3 frames, to "1" and transmit them to the remote terminal equipment. Table 44 relates the content of this bit field to the Transmit DS3 Framer Block's action.

TABLE 44: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 6 (TX X-BITS) WITHIN THE TX DS3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT DS3 FRAMER BLOCK'S ACTION

BIT 6	TRANSMIT DS3 FRAMER'S ACTION
0	**Normal Operation:** The X-bits are generated by the Transmit DS3 Framer block based upon Receiving Conditions (as detected by the Receive Section of the Framer chip).
1	**Set X-bits to 1:** The Transmit DS3 Framer will overwrite the X-bits by setting them to 1. Payload information is not modified and is transmitted as normal.

NOTE: *This bit is ignored when either the Transmit Yellow Alarm, Tx AIS, Tx IDLE, or TxLOS bit is set.*

5.2.4.2.1.3 Transmit Idle Pattern - Bit 5

This read/write bit field permits the user to transmit an Idle pattern to the remote terminal equipment upon software control. Table 45 relates the contents of this bit field to the Transmit DS3 Framer's action.

TABLE 45: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 5 (TX IDLE) WITHIN THE TX DS3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT DS3 FRAMER ACTION

BIT 5	TRANSMIT DS3 FRAMER'S ACTION
0	**Normal Operation:** The Overhead bits are either internally generated, or they are inserted via the Transmit Overhead Data Input Interface or the Transmit HDLC Controller blocks. The Payload bits are received from the Transmit Payload Data Input Interface.
1	**Transmit Idle Condition Pattern:** When this command is invoked, the Transmit DS3 Framer will do the following: • Set the X-bits to 1 • Set the CP-Bits (F-Frame #3) to 0 • Generate Valid M, F, and P bits Overwrite the data in the DS3 payload with a repeating 1100... pattern.

NOTE: *This bit is ignored when either the Tx AIS or the Tx LOS bit is set.*

5.2.4.2.1.4 Transmit AIS Pattern - Bit 4

This read/write bit field allows the user to transmit an AIS pattern to the remote terminal equipment, upon software control. Table 46 relates the contents of this bit field to the Transmit DS3 Framer block's action.

TABLE 46: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 4 (TX AIS PATTERN) WITHIN THE TX DS3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT DS3 FRAMER BLOCK'S ACTION

BIT 4	TRANSMIT DS3 FRAMER'S ACTION
0	**Normal Operation:** The Overhead bits are either internally generated, or they are inserted via the Transmit Overhead Data Input Interface or the Transmit HDLC Controller blocks. The Payload bits are received from the Transmit Payload Data Input Interface.
1	**Transmit AIS Pattern:** When this command is invoked, the Transmit DS3 Framer block will do the following. • Set the X-bits to 1 • Set all the C-bits to 0 • Generate valid M, F, and P bits Overwrite the data in the DS3 payload with a repeating 1010... pattern

NOTE: *This bit is ignored when the TxLOS bit is set.*

5.2.4.2.1.5 Transmit LOS Pattern - Bit 3

This read/write bit field allows the user to transmit an LOS (Loss of Signal) pattern to the remote terminal, upon software control. Table 47 relates the contents of this bit field to the Transmit DS3 Framer block's action.

TABLE 47: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 3 (TX LOS) WITHIN THE TX DS3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT DS3 FRAMER BLOCK'S ACTION

BIT 3	TRANSMIT DS3 FRAMER'S ACTION
0	**Normal Operation:** The Overhead bits are either internally generated, or they are inserted via the Transmit Overhead Data Input Interface or the Transmit HDLC Controller blocks. The Payload bits are received from the Transmit Payload Data Input Interface.
1	**Transmit LOS Pattern:** When this command is invoked the Transmit DS3 Framer will do the following. • Set all of the overhead bits to "0" (including the M, F, and P bits) Overwrite the DS3 payload bits with an all zeros pattern.

NOTE: *When this bit is set, it overrides all of the other bits in this register.*

5.2.4.2.1.6 FERF (Far-End Receive Failure) on LOS - Bit 2

This Read/Write bit-field allows the user to configure the Transmit DS3 Framer block to automatically generate a Yellow Alarm if the Near-End Receive Section (of the XRT74L74) detects a LOS (Loss of Signal) Condition.

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

5.2.4.2.1.7 FERF (Far-End Receive Failure) on OOF - Bit 1

This Read/Write bit-field allows the user to configure the Transmit DS3 Framer block to automatically generate a Yellow Alarm if the Near-End Receive Section (of the XRT74L74) detects an OOF (Out-of-Frame) Condition.

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

5.2.4.2.1.8 FERF (Far-End Receive Failure) on AIS - Bit 0

This Read/Write bit-field allows the user to configure the Transmit DS3 Framer block to automatically generate a Yellow Alarm if the Near-End Receive Section (of the XRT74L74) detects an AIS (Alarm Indication Signal) pattern.

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

5.2.4.2.1.9 Transmitting FEBE (Far-End Block Error) Values

By default, the Transmit DS3 Framer block will set the three (3) FEBE bit-fields to [1, 1, 1] if all of the following conditions are true.

- The Local Receive DS3 Framer block detects no P-Bit Errors.
- The Local Receive DS3 Framer block detects no CP-Bit Errors

Conversely, the Transmit DS3 Framer block will set the three (3) FEBE bit-fields to a value other than [1, 1, 1] if any one of the following conditions are true.

- The Local Receive DS3 Framer block detects a P-bit Error in the most recently received DS3 frame.
- The Local Receive DS3 Framer block detects a "CP" bit Error in the most recently received DS3 frame.

5.2.4.2.2 Generating Errored DS3 Frames

The Transmit DS3 Framer block permits the user to insert errors into the framing and error detection overhead bits (e.g., the P, M and F-bits) of the outbound DS3 data stream in order to support Far-End Equipment testing. This option can be exercised by writing data to any of the numerous Transmit DS3 Mask Registers. These Mask Registers and their comprising bit-fields are defined below.

TX DS3 M-BIT MASK REGISTER, ADDRESS = 0X35

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
TxFEBE DAT[2]	TxFEBE DAT[1]	TxFEBE DAT[0]	FEBE Reg Enable	TxErr PBit	MBit Mask(2)	MBit Mask(1)	MBit Mask(0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

The bit-fields of the Tx DS3 M-bit Mask Register, that are relevant to error-insertion are shaded. The remaining bit-fields pertain to the FEBE bit-fields, and are discussed in Section 5.2.4.2.1.9.

The Tx DS3 M-Bit Mask Register serves two purposes

1. It allows user values to be transmited for FEBE (3 bits) - please see Section 5.2.4.2.1.9.
2. It allows the user to transmit errored P-bits.
3. It allows the user to insert errors into the M-bit (framing bits) in order to support equipment testing.

Each of these bit-fields are discussed below.

Bit 3 - Tx Err (Transmit Errored) P-Bit

This bit-field allows the user to insert errors into the P-bits, of each outbound DS3 Frame, for equipment testing purposes. If this bit-field is 0, then the P-Bits are transmitted as calculated from the payload of the previous DS3 frames. However, if this bit-field is 1, then the P-bits are inverted (from their calculated value) prior to transmission.

Bits 2 - 0: M-Bit Mask[2:0]

The Transmit DS3 Framer will automatically perform an XOR operation with the M-bits (in the DS3 datastream) and the contents of the corresponding bit-field, within this register. The results of this operation will be written back into the M-bit positions within the outbound DS3 Frames. Therefore, to insure that no errors are inserted into the M-bits, make sure that the contents of the M-Bit Mask[2:0] bit-fields are 0.

F-Bit Error Insertion

The remaining mask registers (Tx DS3 F-Bit Mask1 through Mask4 registers) contain bit-fields which correspond to each of the 28 F-bits, within the DS3 frame. Prior to transmission, these bit-fields are automatically XORed with the contents of the corresponding bit fields within these Mask Registers. The result of this XOR operation is written back into the corresponding bit-field, within the outgoing DS3 frame, and is transmitted on the line. Therefore, if none of the bits are to be modified, then these registers must contain all 0s (the default value).

TX DS3 F-BIT MASK1 REGISTER, ADDRESS = 0X36

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Unused	Unused	Unused	Unused	FBit Mask(27)	FBit Mask(26)	FBit Mask(25)	FBit Mask(24)
RO	RO	RO	RO	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

TX DS3 F-BIT MASK2 REGISTER, ADDRESS = 0X37

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
FBit Mask(23)	FBit Mask(22)	FBit Mask(21)	FBit Mask(20)	FBit Mask(19)	FBit Mask(18)	FBit Mask(17)	FBit Mask(16)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

TX DS3 F-BIT MASK3 REGISTER, ADDRESS = 0X38

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
FBit Mask(15)	FBit Mask(14)	FBit Mask(13)	FBit Mask(12)	FBit Mask(11)	FBit Mask(10)	FBit Mask(9)	FBit Mask(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

TX DS3 F-BIT MASK4 REGISTER, ADDRESS = 0X39

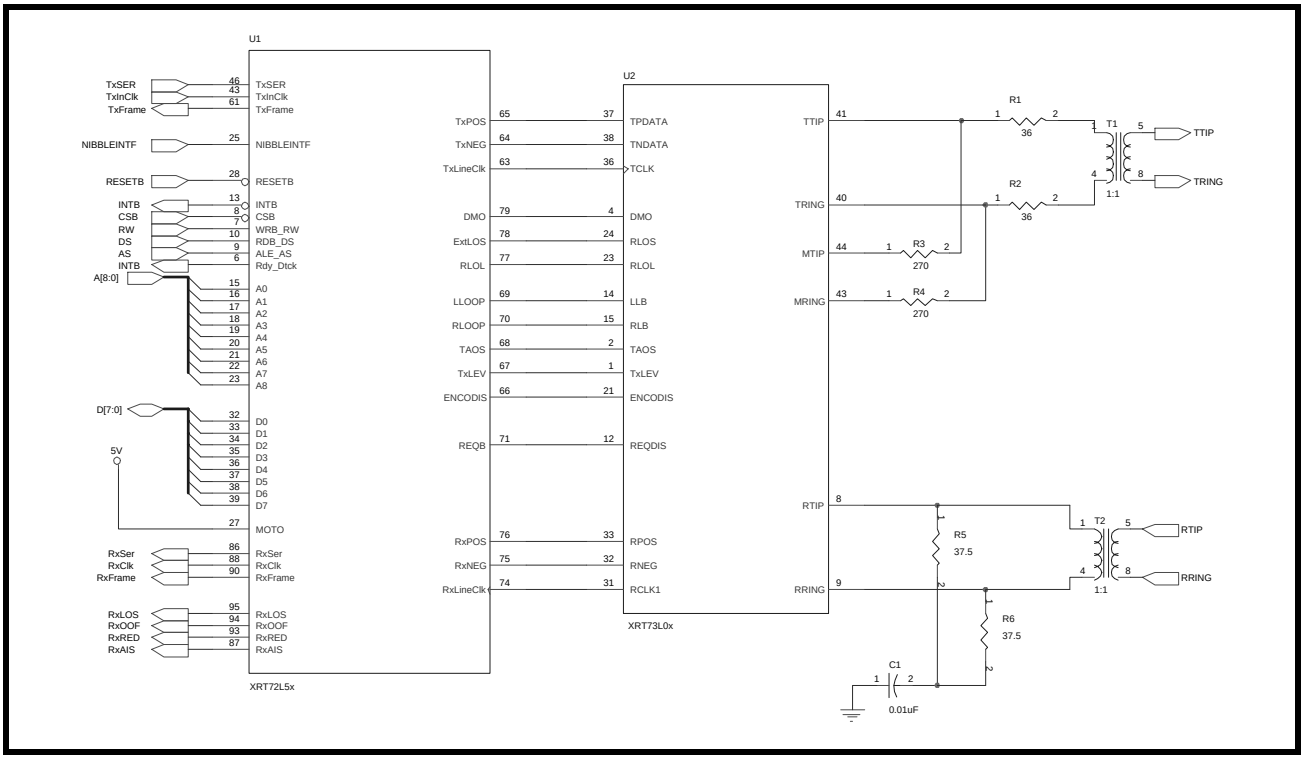
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
FBit Mask(7)	FBit Mask(6)	FBit Mask(5)	FBit Mask(4)	FBit Mask(3)	FBit Mask(2)	FBit Mask(1)	FBit Mask(0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

5.2.5 The Transmit DS3 Line Interface Block

The XRT74L74 Framer IC is a digital device that takes DS3 payload and overhead bit information from some terminal equipment, processes this data and ultimately, multiplexes this information into a series of outbound DS3 frames. However, for DS3 coaxial cable applications, the XRT74L74 Framer IC lacks the current drive capability to be able to directly transmit this DS3 data stream through some transformer-coupled coax cable with enough signal strength for it to comply with the Isolated Pulse Template requirements and be received by the remote receiver.

Therefore, in order to get around this problem, the Framer IC requires the use of an LIU (Line Interface Unit) 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 (1) comply with the DSX-3 Isolated Pulse Template requirements and (2) be reliably received by the Remote Terminal Equipment. Figure 72 presents a circuit drawing depicting the Framer IC interfacing to an LIU (XRT7300 DS3/E3/STS-1 Transmit LIU).

FIGURE 72. APPROACH TO INTERFACING THE XRT74L74 FRAMER IC TO THE XRT73L00 DS3/E3/STS-1 TRANSMITTER LIU (ONE CHANNEL SHOWN)

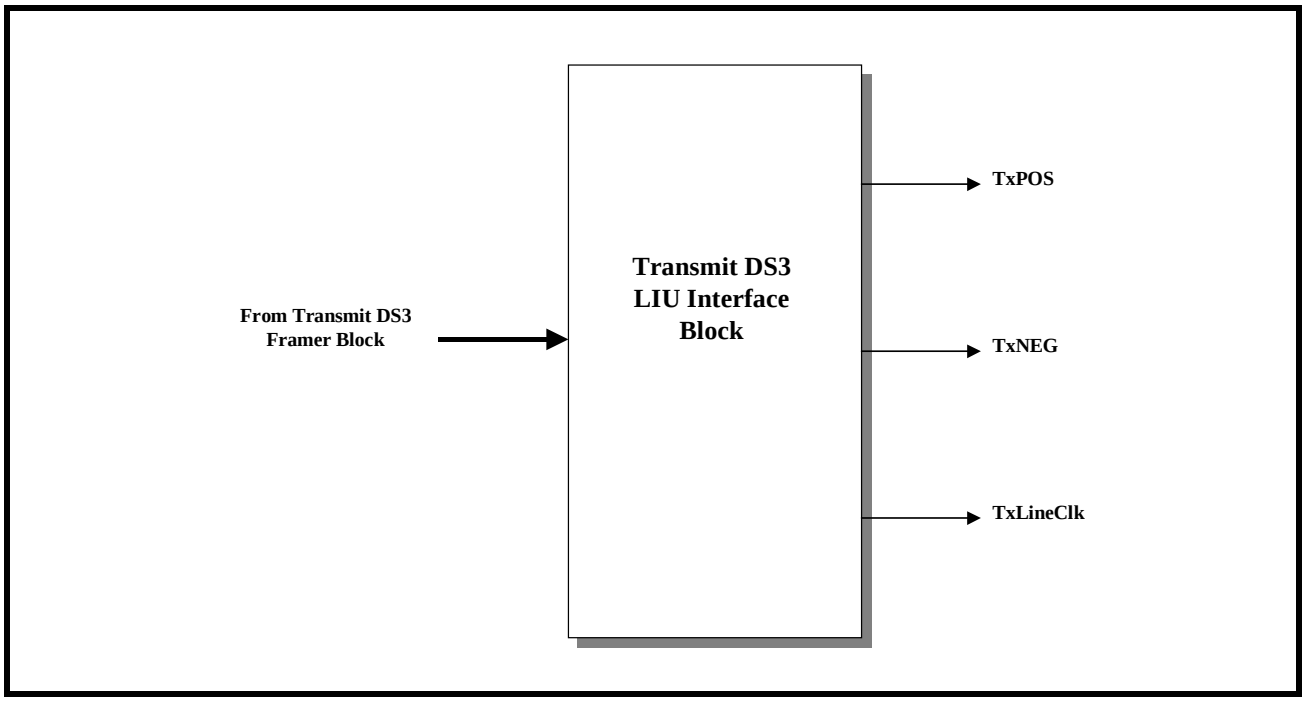


The Transmit Section of the XRT74L74 contains a block which is known as the Transmit DS3 LIU Interface block. The purpose of the Transmit DS3 LIU Interface block is to take the outbound DS3 data stream, from the Transmit DS3 Framer block, and to do the following:

1. Encode this data into one of the following line codes
   - **a.** Unipolar (e.g., Single-Rail)
   - **b.** AMI (Alternate Mark Inversion)
   - **c.** B3ZS (Bipolar 3 Zero Substitution)
2. And to transmit this data to the LIU IC.

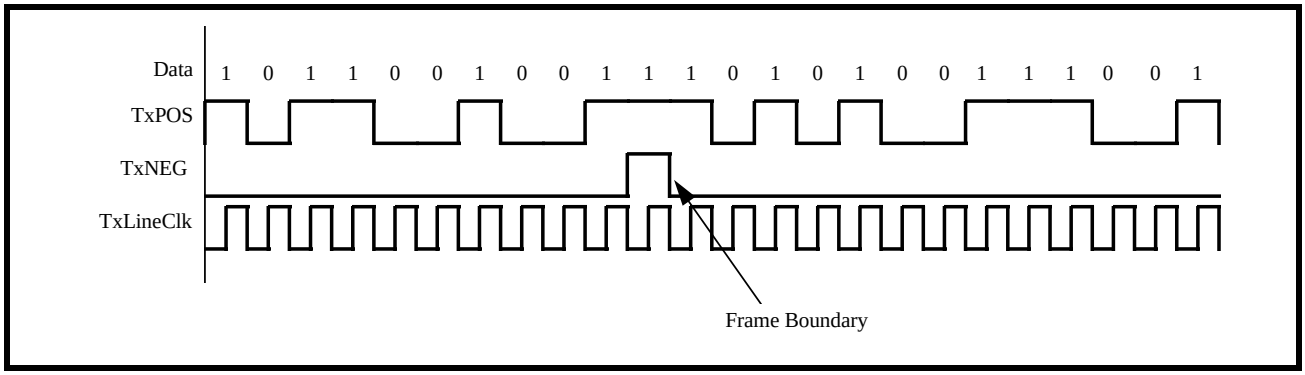
Figure 73 presents a simple illustration of the Transmit DS3 LIU Interface block.

FIGURE 73. A SIMPLE ILLUSTRATION OF THE TRANSMIT DS3 LIU INTERFACE BLOCK



The Transmit DS3 LIU Interface block can transmit data to the LIU IC or other external circuitry via two different output modes: Unipolar or Bipolar. If the Unipolar (or Single Rail) mode is selected, then the contents of the DS3 Frame is output, in a binary (NRZ manner) data stream via the TxPOS pin to the LIU IC. 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 DS3 frame, and will remain "Low" for the remainder of the frame. Figure 74 presents an illustration of the TxPOS and TxNEG signals during data transmission while the Transmit DS3 LIU Interface block is operating in the Unipolar mode. This mode is sometimes referred to as Single Rail mode because the data pulses only exist in one polarity: positive.

FIGURE 74. THE BEHAVIOR OF TXPOS AND TXNEG SIGNALS DURING DATA TRANSMISSION WHILE THE TRANSMIT DS3 LIU INTERFACE IS OPERATING IN THE UNIPOLAR MODE



When the Transmit DS3 LIU Interface block is operating in the Bipolar (or Dual Rail) mode, then the contents of the DS3 Frame is output via both the TxPOS and TxNEG pins. If the Bipolar mode is chosen, then the DS3 data to the LIU can be transmitted via one of two different line codes: Alternate Mark Inversion (AMI) or Binary - 3 Zero Substitution (B3ZS). Each one of these line codes will be discussed below. Bipolar mode is sometimes referred to as Dual Rail because the data pulses occur in two polarities: positive and negative. The role of the TxPOS, TxNEG and

TxLineClk output pins, for this mode are discussed below.

TxPOS - Transmit Positive Polarity Pulse: The Transmit DS3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a positive polarity pulse to the remote terminal equipment.

TxNEG - Transmit Negative Polarity Pulse: The Transmit DS3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a negative polarity pulse to the remote terminal equipment.

TxLineClk - Transmit Line Clock: The LIU IC uses this signal from the Transmit DS3 LIU Interface block 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 remote Receive DS3 Framer.

5.2.5.1 Selecting the various Line Codes

Either the Unipolar Mode or Bipolar Mode can be selected by writing the appropriate value to Bit 3 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 48 relates the value of this bit field to the Transmit DS3 LIU Interface Output Mode.

TABLE 48: THE RELATIONSHIP BETWEEN THE CONTENT OF BIT 3 (UNIPOLAR/BIPOLAR*) WITHIN THE UNI I/O CONTROL REGISTER AND THE TRANSMIT DS3 FRAMER LINE INTERFACE OUTPUT MODE

BIT 3	TRANSMIT DS3 FRAMER LIU INTERFACE OUTPUT MODE
0	**Bipolar Mode:** AMI or B3ZS Line Codes are Transmitted and Received
1	**Unipolar (Single Rail) Mode** of transmission and reception of DS3 data is selected.

NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the operation of the Receive DS3 LIU Interface block*

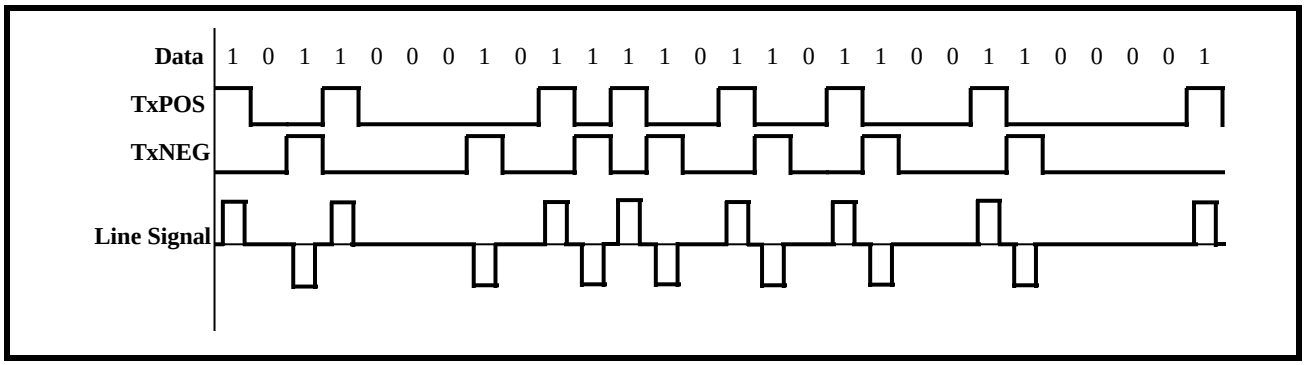
5.2.5.1.1 The Bipolar Mode Line Codes

If framer is to be operated in the Bipolar Mode, then the DS3 data-stream can be transmitted via the AMI (Alternate Mark Inversion) or the B3ZS Line Codes. The definition of AMI and B3ZS line codes follow.

5.2.5.1.1.1 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, and -1. +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. Figure 75 presents an illustration of the AMI Line Code as would appear at the TxPOS and TxNEG pins of the Framer, as well as the output signal on the line.

FIGURE 75. ILLUSTRATION OF AMI LINE CODE



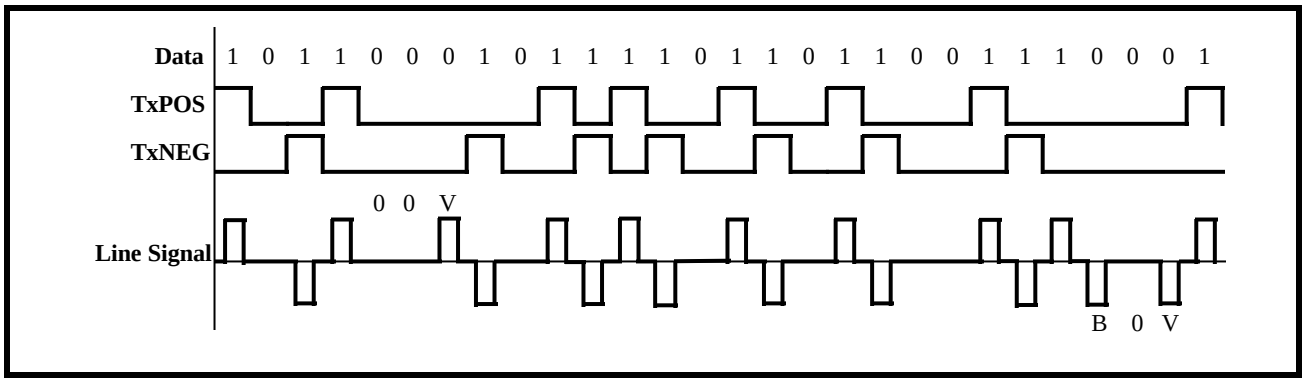
NOTE: One of the main reasons that the AMI Line Code has been chosen for driving transformer-coupled media is that this line code introduces no dc component, thereby minimizing dc distortion in the line.

5.2.5.1.1.2 The B3ZS Line Code

The Transmit DS3 Framer and the associated LIU IC combine the data and timing information (originating from the TxLineClk signal) into the line signal that is transmitted to the far-end receiver. The far-end receiver has the task of recovering this data and timing information from the incoming DS3 data stream. Many clock and data recovery schemes rely on the use of Phase Locked Loop technology. Phase-Locked-Loop (PLL) technology for clock recovery relies on transitions in the line signal, in order to maintain lock with the incoming DS3 data stream. However, PLL-based clock recovery scheme, are vulnerable to the occurrence of a long stream of consecutive zeros (e.g., the absence of transitions). This scenario can cause the PLL to lose lock with the incoming DS3 data, thereby causing the clock and data recovery process of the receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is B3ZS encoding. B3ZS (or Bipolar 3 Zero Substitution) is a form of AMI line coding that implements the following rule.

In general the B3ZS line code behaves just like AMI with the exception of the case when a long string of consecutive zeros occur on the line. Any string of 3 consecutive zeros will be replaced with either a 00V or a B0V 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 00V or a B0V is made to insure that an odd number of Bipolar (B) pulses exist between any two Bipolar Violation (V) pulses. Figure 76 presents a timing diagram that illustrates examples of B3ZS encoding.

FIGURE 76. ILLUSTRATION OF TWO EXAMPLES OF B3ZS ENCODING



The user chooses between AMI or B3ZS line coding by writing to bit 4 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 49 relates the content of this bit-field to the Bipolar Line Code that DS3 Data will be transmitted and received at.

TABLE 49: THE RELATIONSHIP BETWEEN BIT 4 (AMI/B3ZS*) WITHIN THE I/O CONTROL REGISTER AND THE BIPOLAR LINE CODE THAT IS OUTPUT BY THE TRANSMIT DS3 LIU INTERFACE BLOCK

BIT 4	BIPOLAR LINE CODE
0	B3ZS
1	AMI

NOTES:

1. *This bit is ignored if the Unipolar mode is selected.*
2. *This selection also effects the operation of the Receive DS3 LIU Interface block*

5.2.5.2 TxLineClk Clock Edge Selection

The Framer also allows the user to specify whether the DS3 output data (via TxPOS and/or TxNEG output pins) is to be updated on the rising or falling edges of the TxLineClk signal. The purpose of this feature is to insure that the Framer will always be able to output data to the LIU IC, in such a way that the LIU set-up and hold time requirements can always be met. This selection is made by writing to bit 2 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	X	X	0

Table 50 relates the contents of this bit field to the clock edge of TxClk that DS3 Data is output on the TxPOS and/or TxNEG output pins.

TABLE 50: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TXLINECLK INV) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG 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 77 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 78 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 77. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE RISING EDGE OF TXLINECLK

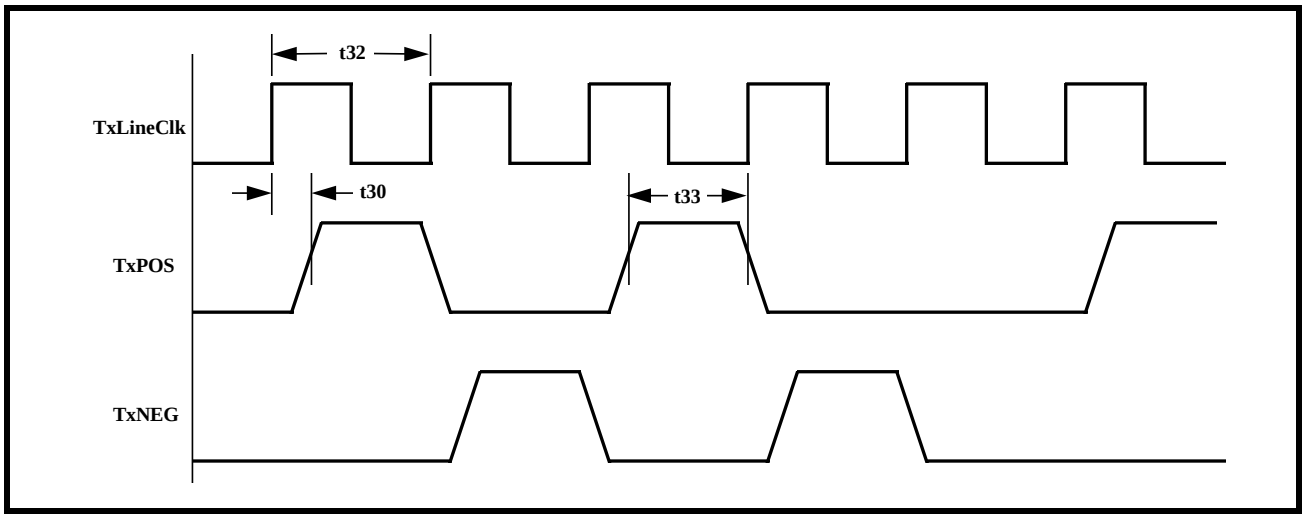
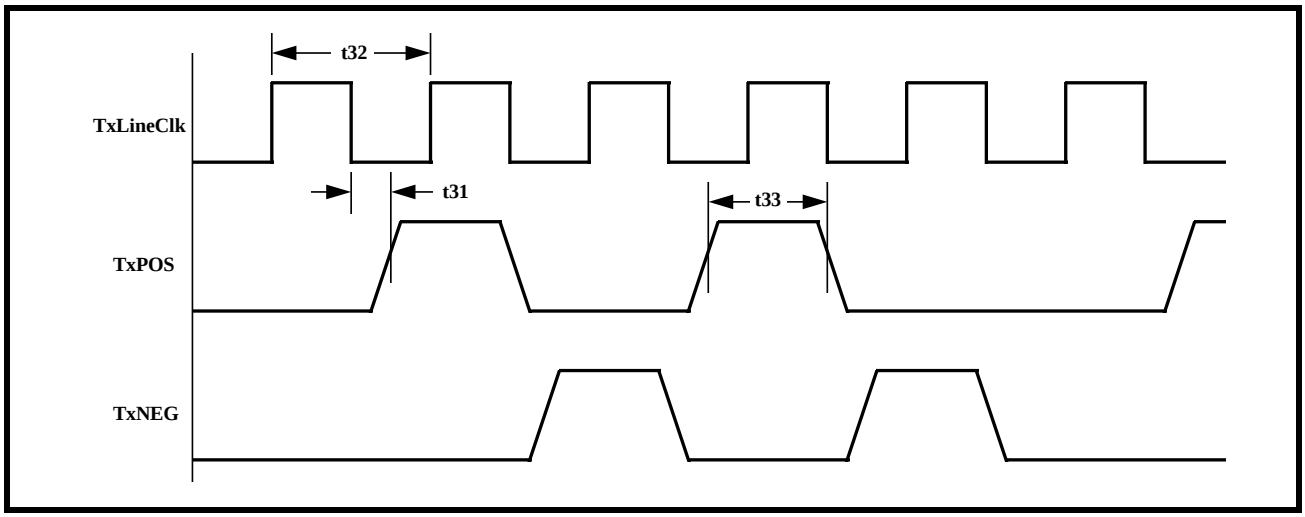


FIGURE 78. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE FALLING EDGE OF TXLINECLK



5.2.6 Transmit Section Interrupt Processing

The Transmit Section of the XRT74L74 can generate an interrupt to the Microcontroller/Microprocessor for the following two reasons.

- Completion of Transmission of FEAC Message
- Completion of Transmission of LAPD Message

5.2.6.1 Enabling Transmit Section Interrupts

The Interrupt Structure, within the XRT74L74 contains two hierarchical levels:

- Block Level
- Source Level

The Block Level

The Enable State of the Block Level for the Transmit Section Interrupts dictates whether or not interrupts (if enabled at the source level), are actually enabled.

These Transmit Section interrupts can be enabled or disabled at the Block Level, by writing the appropriate data into Bit 1 (Tx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables the Transmit Section (at the Block Level) for Interrupt Generation. Conversely, setting this bit-field to “0” disables the Transmit Section for interrupt generation.

What does it mean for the Transmit Section Interrupts to be enabled or disabled at the Block Level?

If the Transmit Section is disabled (for interrupt generation) at the Block Level, then ALL Transmit Section interrupts are disabled, independent of the interrupt enable/disable state of the source level interrupts.

If the Transmit Section is enabled (for interrupt generation) at the block level, then a given interrupt will be enabled at the source level. Conversely, if the Transmit Section is enabled (for interrupt generation) at the Block level, then a given interrupt will still be disabled, if it is disabled at the source level.

As mentioned earlier, the Transmit Section of the XRT74L74 Framer IC contains the following two interrupts

- Completion of Transmission of FEAC Message Interrupt.
- Completion of Transmission of LAPD Message Interrupt.

The Enabling/Disabling and Servicing of each of these interrupts is described below.

5.2.6.1.1 The Completion of Transmission of FEAC Message Interrupt.

If the Transmit Section interrupts have been enabled at the Block level, then the Completion of Transmission of a FEAC Message Interrupt can be enabled or disabled by writing the appropriate value into Bit 4 (Tx FEAC Interrupt Enable) within the Transmit DS3 FEAC Configuration & Status Register (Address = 0x31) as illustrated below.

TRANSMIT DS3 FEAC CONFIGURATION & STATUS REGISTER (ADDRESS = 0X31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			Tx FEAC Interrupt Enable	TxFEAC Interrupt Status	TxFEAC Enable	TxFEAC GO	TxFEAC Busy
RO	RO	RO	R/W	RUR	R/W	R/W	RO
0	0	0	X	0	0	0	0

Setting this bit-field to “1” enables the Completion of Transmission of a FEAC Message Interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

5.2.6.1.2 Servicing the Completion of Transmission of a FEAC Message Interrupt

As mentioned earlier, once the user commands the Transmit FEAC Processor to begin its transmission of a FEAC Message, it will do the following.

1. It will read in the six-bit contents of the Tx DS3 FEAC Register (Address = 0x32) and encapsulate these 6 bits into a 16-bit data structure.
2. The Transmit FEAC Processor will then begin to transmit this 16-bit data structure (to the Remote Terminal Equipment) repeatedly for 10 consecutive times.
3. Upon completion of the 10th transmission, the XRT74L74 Framer IC will generate the Completion of Transmission of a FEAC Message Interrupt to the Microcontroller/Microprocessor. Once the XRT74L74 Framer IC generates this interrupt, it will do the following.

- Assert the Interrupt Output pin (INT) by toggling it "Low".

- Set Bit 3 (Tx FEAC Interrupt Status) within the Tx DS3 FEAC Configuration & Status Register, as illustrated below.

TRANSMIT DS3 FEAC CONFIGURATION & STATUS REGISTER (ADDRESS = 0X31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			Tx FEAC Interrupt Enable	TxFEAC Interrupt Status	TxFEAC Enable	TxFEAC GO	TxFEAC Busy
RO	RO	RO	R/W	RUR	R/W	R/W	RO
0	0	0	1	1	0	0	0

The purpose of this interrupt is to alert the Microcontroller/Microprocessor that the Transmit FEAC Processor has completed its transmission of a given FEAC message and is now ready to transmit the next FEAC Message, to the Remote Terminal Equipment.

5.2.6.1.3 The Completion of Transmission of the LAPD Message Interrupt

If the Transmit Section interrupts have been enabled at the Block level, then the Completion of Transmission of a LAPD Message Interrupt can be enabled or disabled by writing the appropriate value into Bit 1 (TxLAPD Interrupt Enable) within the Tx DS3 LAPD Status & Interrupt Register (Address = 0x34), as illustrated below.

TXDS3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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

Setting this bit-field to “1’ enables the Completion of Transmission of a LAPD Message Interrupt. Conversely, setting this bit-field to “0” disables the Completion of Transmission of a LAPD Message interrupt.

5.2.6.1.4 Servicing the Completion of Transmission of a LAPD Message Interrupt

As mentioned previously, once the user commands the LAPD Transmitter to begin its transmission of a LAPD Message, it will do the following.

1. It will parse through the contents of the Transmit LAPD Message Buffer (located at address locations 0x86 through 0xDD) and search for a string of five (5) consecutive “1’s”. If the LAPD Transmitter finds a string of five consecutive “1’s” (within the content of the LAPD Message Buffer, then it will insert a “0” immediately after this string.
2. It will compute the FCS (Frame Check Sequence) value and append this value to the back-end of the user-message.
3. It will read out of the content of the user (zero-stuffed) message and will encapsulate this data into a LAPD Message frame.
4. Finally, it will begin transmitting the contents of this LAPD Message frame via the “DL” bits, within each outbound DS3 frame.
5. Once the LAPD Transmitter has completed its transmission of this LAPD Message frame (to the Remote Terminal Equipment), the XRT74L74 Framer IC will generate the Completion of Transmission of a LAPD Message Interrupt to the Microcontroller/Microprocessor. Once the XRT74L74 Framer IC generates this interrupt, it will do the following.

- Assert the Interrupt Output pin (INT) by toggling it "Low".
- Set Bit 0 (TxLAPD Interrupt Status) within the TxDS3 LAPD Status and Interrupt Register, as illustrated below.

TXDS3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	1

The purpose of this interrupt is to alert the Microcontroller/MIcroprocessor that the LAPD Transmitter has completed its transmission of a given LAPD (or PMDL) Message, and is now ready to transmit the next PMDL Message, to the Remote Terminal Equipment.

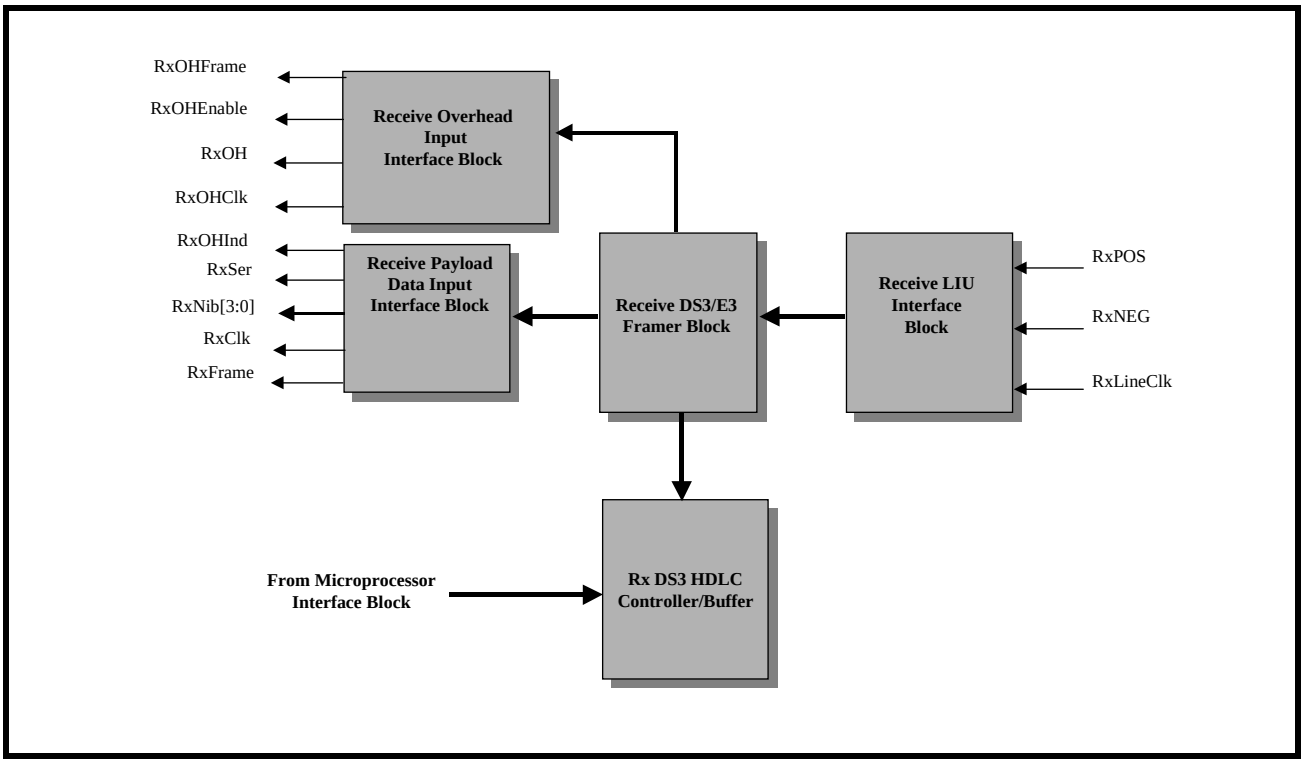
5.3 THE RECEIVE SECTION OF THE XRT74L74 (DS3 MODE OPERATION)

When the XRT74L74 has been configured to operate in the DS3 Mode, the Receive Section of the XRT74L74 consists of the following functional blocks.

- Receive LIU Interface block
- Receive HDLC Controller block
- Receive DS3 Framer block
- Receive Overhead Data Output Interface block
- Receive Payload Data Output Interface block

Figure 79 presents a simple illustration of the Receive Section of the XRT74L74 Framer IC.

FIGURE 79. A SIMPLE ILLUSTRATION OF THE RECEIVE SECTION OF THE XRT74L74, WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE DS3 MODE



Each of these functional blocks will be discussed in detail in this document.

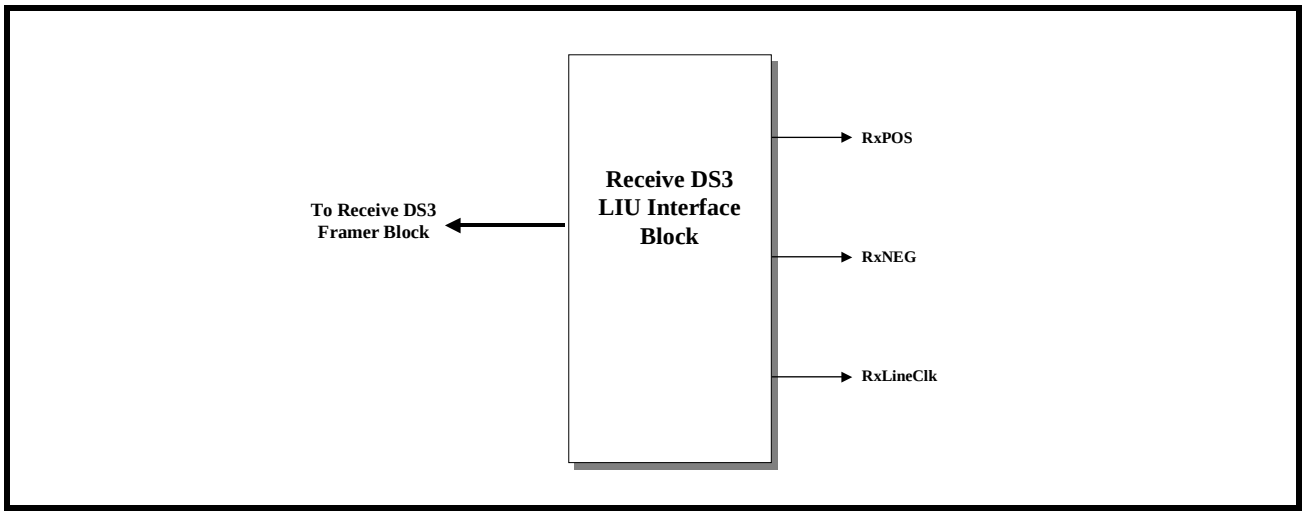
5.3.1 The Receive DS3 LIU Interface Block

The purpose of the Receive DS3 LIU Interface block is two-fold:

1. To receive encoded digital data from the DS3 LIU IC.
2. To decode this data, convert it into a binary data stream and to route this data to the Receive DS3 Framer block.

Figure 80 presents a simple illustration of the Receive DS3 LIU Interface block.

FIGURE 80. A SIMPLE ILLUSTRATION OF THE RECEIVE DS3 LIU INTERFACE BLOCK



The Receive Section of the XRT74L74 will via the Receive DS3 LIU Interface Block receive timing and data information from the incoming DS3 data stream. The DS3 Timing information will be received via the RxLineClk input pin and the DS3 data information will be received via the RxPOS and RxNEG input pins. The Receive DS3 LIU Interface block is capable of receiving DS3 data pulses in unipolar or bipolar format. If the Receive DS3 framer is operating in the bipolar format, then it can be configured to decode either AMI or B3ZS line code data. Each of these input formats and line codes will be discussed in detail, below.

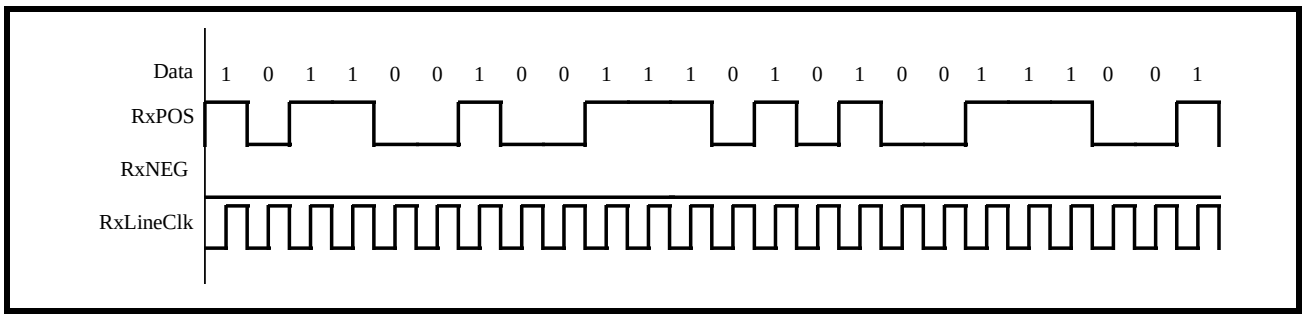
5.3.1.1 Unipolar Decoding

If the Receive DS3 LIU Interface block is operating in the Unipolar (single-rail) mode, then it will receive the Single Rail NRZ DS3 data pulses via the RxPOS input pin. The Receive DS3 LIU Interface block will also receive its timing signal via the RxLineClk signal.

NOTE: *The RxLineClk signal will function as the timing source for the entire Receive Section of the XRT74L74.*

No data pulses will be applied to the RxNEG input pin. The Receive DS3 LIU Interface block receives a logic "1" when a logic "1" level signal is present at the RxPOS 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 81 presents an illustration of the behavior of the RxPOS, RxNEG and RxLineClk input pins when the Receive DS3 LIU Interface block is operating in the Unipolar mode.

FIGURE 81. BEHAVIOR OF THE RXPOS, RXNEG AND RXLINECLK SIGNALS DURING DATA RECEPTION OF UNIPOLAR DATA



The user can configure the Receive DS3 LIU Interface block to operate in either the Unipolar or the Bipolar Mode by writing the appropriate data to the I/O Control Register, as depicted below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 51 relates the value of this bit-field to the Receive DS3 LIU Interface Input Mode.

TABLE 51: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 3 (UNIPOLAR/BIPOLAR) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG ARE UPDATED ON

BIT 3	RECEIVE DS3 LIU INTERFACE INPUT MODE
0	.**Bipolar Mode (Dual Rail):** AMI or B3ZS Line Codes are Transmitted and Received.
1	**Unipolar Mode (Single Rail) Mode** of transmission and reception of DS3 data is selected.

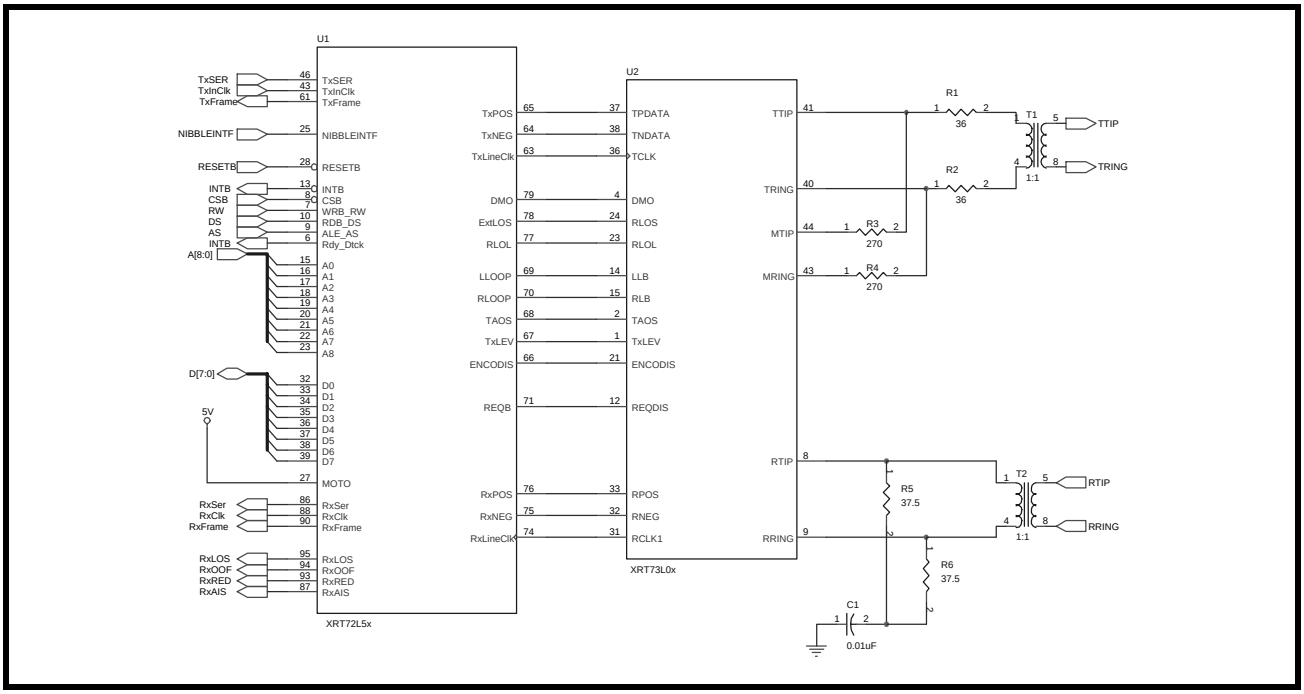
NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the Transmit DS3 Framer Line Interface Output Mode*

5.3.1.2 Bipolar Decoding

If the Receive DS3 LIU Interface block is operating in the Bipolar Mode, then it will receive the DS3 data pulses via both the RxPOS, RxNEG, and the RxLineClk input pins. Figure 82 presents a circuit diagram illustrating how the Receive DS3 LIU Interface block interfaces to the Line Interface Unit while the Framer is operating in Bipolar mode. The Receive DS3 LIU Interface block can be configured to decode the incoming data from either the AMI or B3ZS line codes.

FIGURE 82. ILLUSTRATION ON HOW THE RECEIVE DS3 FRAMER (WITHIN THE XRT74L74 FRAMER IC) BEING INTERFACED TO THEXRT73L00 LIU, WHILE THE FRAMER IS OPERATING IN BIPOLAR MODE (ONE CHANNEL SHOWN)

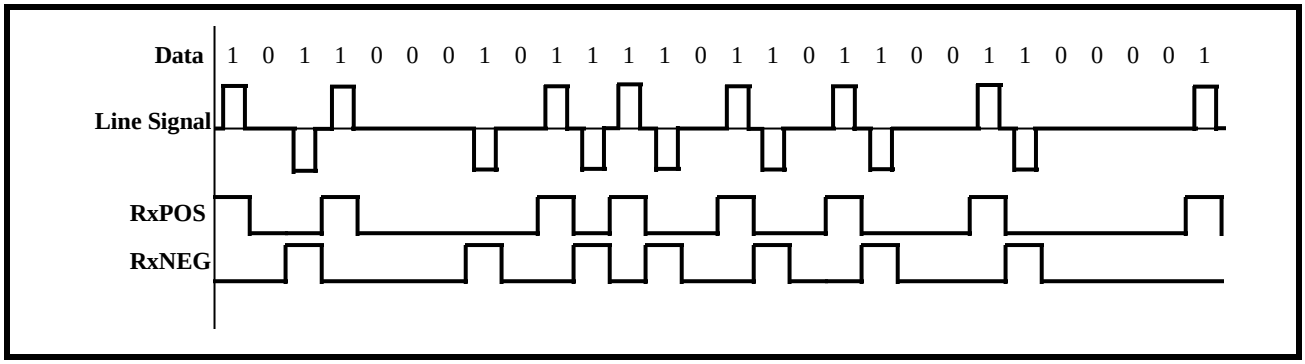


5.3.1.2.1 AMI Decoding

AMI or Alternate Mark Inversion, means that consecutive one's pulses (or marks) will be of opposite polar-

ity 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 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 the AMI line code 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. Figure 83 presents an illustration of the AMI Line Code as would appear at the RxPOS and RxNEG input pins of the Framer, as well as the corresponding output signal on the line.

FIGURE 83. ILLUSTRATION OF AMI LINE CODE



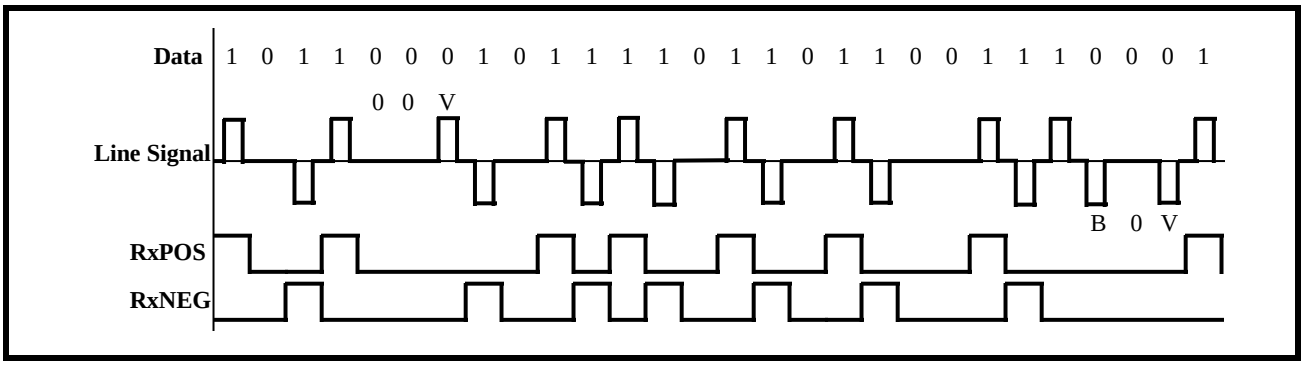
NOTE: *One of the reasons that the AMI Line Code has been chosen for driving copper medium, isolated via transformers, is that this line code has no dc component, thereby eliminating dc distortion in the line.*

5.3.1.2.2 B3ZS Decoding

The Transmit DS3 LIU Interface block and the associated LIU embed and combine the data and clocking information into the line signal that is transmitted to the remote terminal equipment. The remote terminal equipment has the task of recovering this data and timing information from the incoming DS3 data stream. Most clock and data recovery schemes rely on the use of Phase-Locked-Loop technology. One of the problems of using Phase-Locked-Loop (PLL) technology for clock recovery is that it relies on transitions in the line signal, in order to maintain lock with the incoming DS3 data-stream. Therefore, these clock recovery scheme, 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 DS3 data, thereby causing the clock and data recovery process of the receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is B3ZS (or Bipolar 3 Zero Substitution) encoding.

In general the B3ZS line code behaves just like AMI with the exception of the case when a long string of consecutive zeros occurs on the line. Any 3 consecutive zeros will be replaced with either a 00V or a B0V where B refers to a Bipolar pulse (e.g., a pulse with a polarity that is compliant with the alternating polarity scheme of 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 00V or a B0V is made to insure that an odd number of Bipolar (B) pulses exist between any two Bipolar Violation (V) pulses. The Receive DS3 Framer, when operating with the B3ZS Line Code is responsible for decoding the B3ZS-encoded data back into a unipolar (binary-format). For instance, if the Receive DS3 Framer detects a 00V or a B0V pattern in the incoming pattern, the Receive DS3 Framer will replace it with three consecutive zeros. Figure 84 presents a timing diagram that illustrates examples of B3ZS decoding.

FIGURE 84. ILLUSTRATION OF TWO EXAMPLES OF B3ZS DECODING



5.3.1.2.3 Line Code Violations

The Receive DS3 LIU Interface block will also check the incoming DS3 data stream for line code violations. For example, when the Receive DS3 LIU Interface block detects a valid bipolar violation (e.g., in B3ZS line code), it will substitute three zeros into the binary data stream. However, if the bipolar violation is invalid, then an LCV (Line Code Violation) is flagged and the PMON LCV Event Count Register (Address = 0x50 and 0x51) will also be incremented. Additionally, the LCV-One Second Accumulation Registers (Address = 0x6E and 0x6F) will be incremented. For example: If the incoming DS3 data is B3ZS encoded, the Receive DS3 LIU Interface block will also increment the LCV One Second Accumulation Register if three (or more) consecutive zeros are received.

5.3.1.2.4 RxLineClk Clock Edge Selection

The incoming unipolar or bipolar data, applied to the RxPOS and the RxNEG input pins are clocked into the Receive DS3 LIU Interface block via the RxLineClk signal. The Framer IC allows the user to specify which edge (e.g, rising or falling) of the RxLineClk signal will sample and latch the signal at the RxPOS and RxNEG input signals into the Framer IC. This feature was included in the XRT74L74 design to insure that the user can always meet the RxPOS and RxNEG to RxLineClk set-up and hold time requirements. This selection is made by writing the appropriate data to bit 1 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 52 depicts the relationship between the value of this bit-field to the sampling clock edge of RxLineClk.

TABLE 52: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (RXLINECLK INV) OF THE I/O CONTROL REGISTER, AND THE SAMPLING EDGE OF THE RXLINECLK SIGNAL

RXCLKINV (BIT 1)	RESULT
0	**Rising Edge:** RxPOS and RxNEG are sampled at the rising edge of RxLineClk. See Figure 85 for timing relationship between RxLineClk, RxPOS, and RxNEG.
1	**Falling Edge:** RxPOS and RxNEG are sampled at the falling edge of RxLineClk. See Figure 86 for timing relationship between RxLineClk, RxPOS, and RxNEG.

Figure 85 and Figure 86 present the Waveform and Timing Relationships between RxLineClk, RxPOS and RxNEG for each of these configurations.

FIGURE 85. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE RISING EDGE OF RXLINECLK

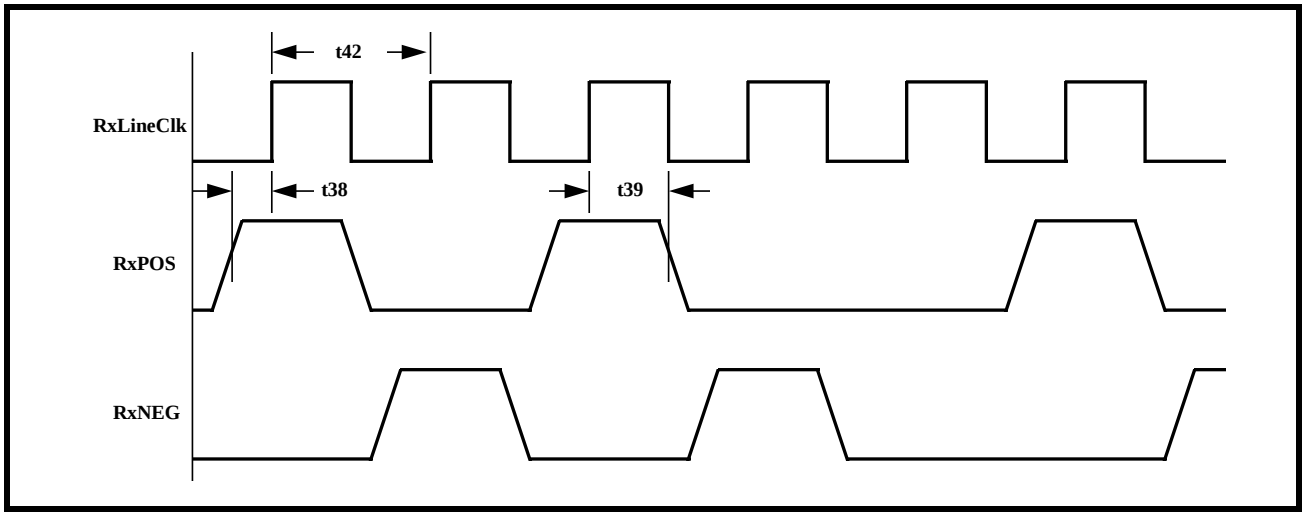
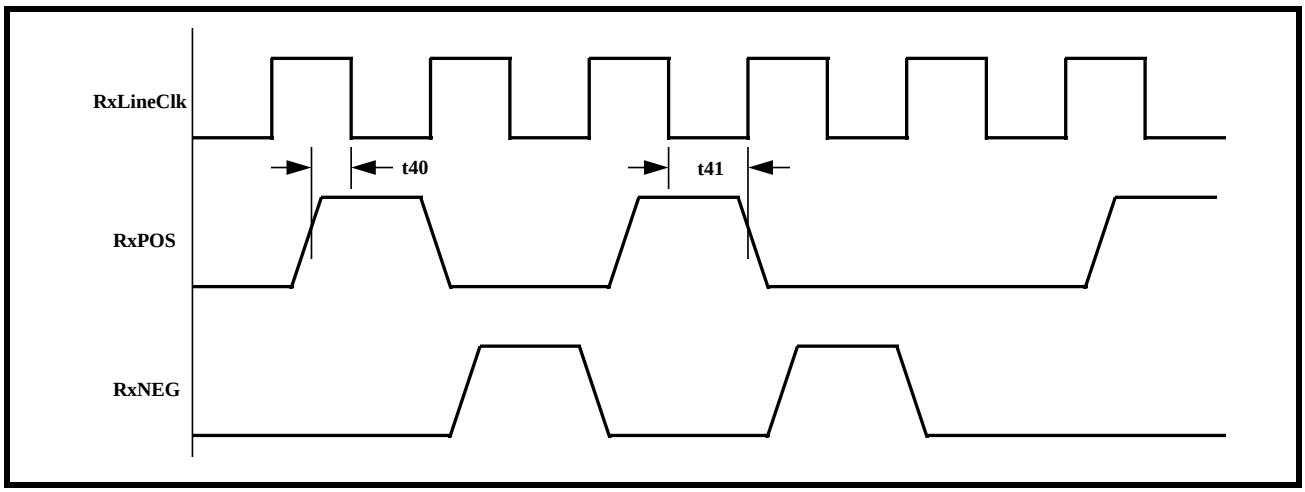


FIGURE 86. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE FALLING EDGE OF RXLINECLK



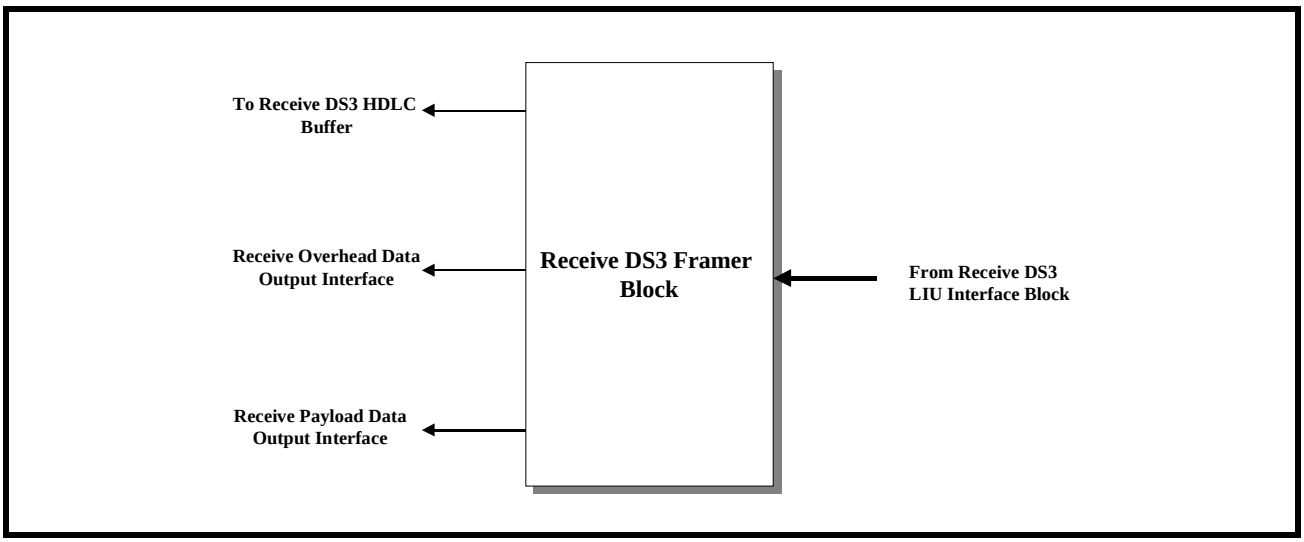
5.3.2 The Receive DS3 Framer Block

The Receive DS3 Framer block accepts decoded DS3 data from the Receive DS3 LIU Interface block, and routes data to the following destinations.

- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block.
- The Receive DS3 HDLC Controller Block

Figure 87 presents a simple illustration of the Receive DS3 Framer block along with the associated paths to the other functional blocks within the Framer chip.

FIGURE 87. A SIMPLE ILLUSTRATION OF THE RECEIVE DS3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO THE OTHER FUNCTIONAL BLOCKS

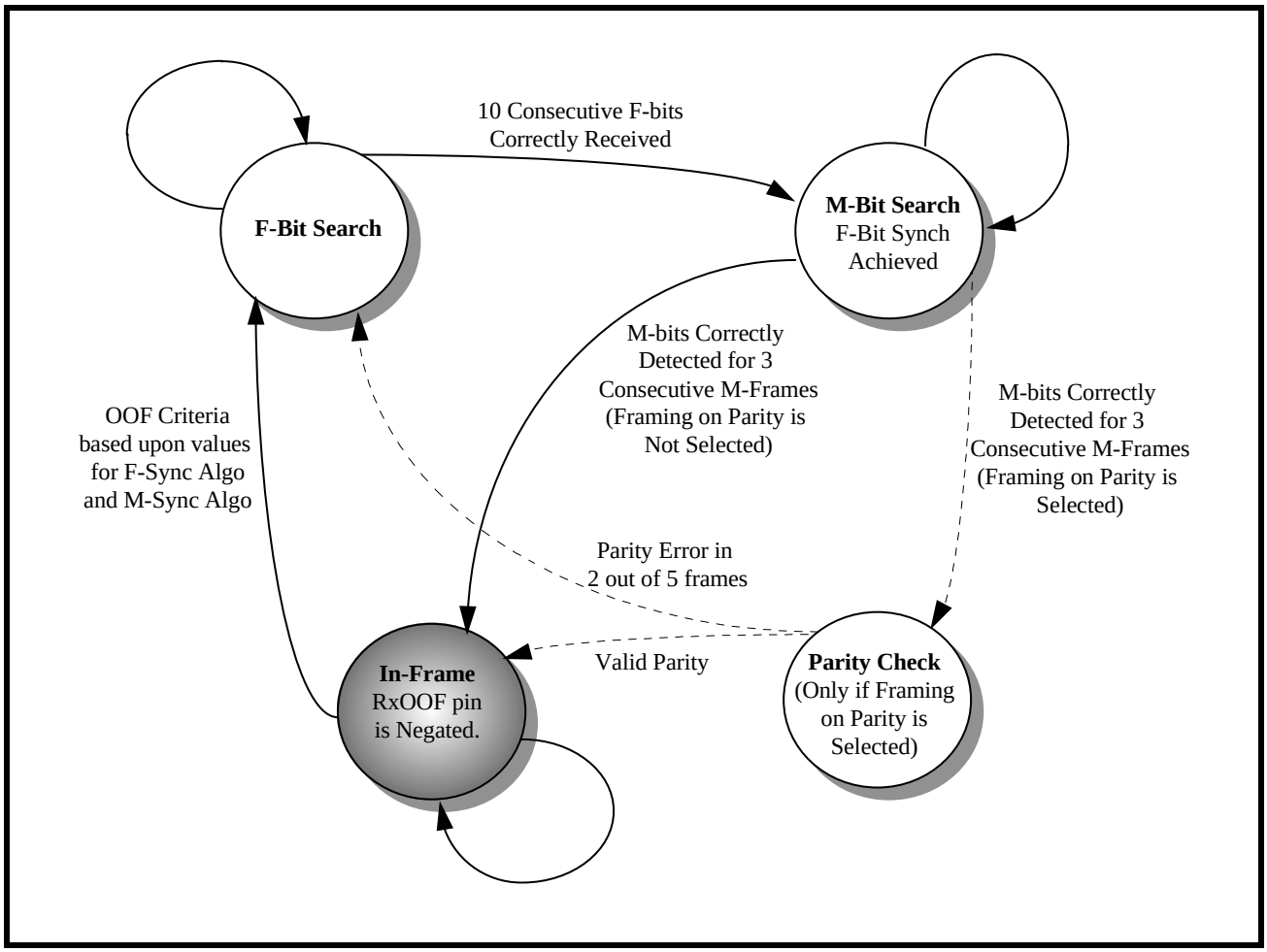


Once the B3ZS (or AMI) encoded data has been decoded into a binary data-stream, the Receive DS3 Framer block will use portions of this data-stream in order to synchronize itself to the remote terminal equipment. At any given time, the Receive DS3 Framer block will be operating in one of two modes.

- **The Frame Acquisition Mode:** In this mode, the Receive DS3 Framer block is trying to acquire synchronization with the incoming DS3 frames, or
- **The Frame Maintenance Mode:** In this mode, the Receive DS3 Framer block is trying to maintain frame synchronization with the incoming DS3 Frames.

Figure 88 presents a State Machine diagram that depicts the Receive DS3 Framer block's DS3 Frame Acquisition/Maintenance Algorithm.

FIGURE 88. THE STATE MACHINE DIAGRAM FOR THE RECEIVE DS3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM



5.3.2.1 Frame Acquisition Mode Operation

The Receive DS3 Framer block will be performing Frame Acquisition operation while it is operating in any of the following states (per the DS3 Frame Acquisition/Maintenance algorithm State Machine diagram, as depicted in Figure 88.)

- The F-bit Search state
- The M-bit Search state
- The P-Bit Check state (optional)

Once the Receive DS3 Framer block enters the In-Frame state (per Figure 88), then it will begin Frame Maintenance operation.

When the Receive DS3 Framer block is in the frame-acquisition mode, it will begin to look for valid DS3 frames by first searching for the F-bits in the incoming DS3 data stream. At this initial point the Receive DS3 Framer block will be operating in the F-Bit Search state within the DS3 Frame Acquisition/Maintenance algorithm state machine diagram (see Figure 88). Recall from the discussion in Section 5.1, that each DS3 F-frame consists of four (4) F-bits that occur in a repeating 1001 pattern. The Receive DS3 Framer block will attempt to locate this F-bit pattern by performing five (5) different searches in parallel. The F-bit search has been declared successful if at least 10 consecutive F-bits are detected. After the F-bit match has been declared, the Receive DS3 Framer block will then transition into the M-Bit Search state within the DS3 Frame Acquisition/Maintenance algorithm (per Figure 88). When the Receive DS3 Framer block reaches this state, it will begin searching for valid M-bits. Recall from the discussion in Section 5.1 that each DS3 M-frame consists of three (3) M-bits that occur in a repeating 010 pattern. The M-bit search is declared successful if three consecutive M-frames (or 21 F-frames) are detected correctly. Once this occurs an M-frame lock is declared, and the Receive DS3 Framer block will then transition to the In-Frame state. At this point, the Receive DS3 Framer

block will declare itself in the In-Frame condition, and will begin Frame Maintenance operations. The Receive DS3 Framer block will then indicate that it has transitioned from the OOF condition into the In-Frame condition by doing the following.

- Generate a Change in OOF Condition interrupt to the local µP.
- Negate the RxOOF output pin (e.g., toggle it "Low").
- Negate the RxOOF bit-field (Bit 4) within the Receive DS3 Configuration and Status Register.

The Receive DS3 Framer can be configured to operate such that 'valid parity' (P-bits) must also be detected before the Receive DS3 Framer can declare itself In Frame. This configuration is set by writing the appropriate data to the Rx DS3 Configuration and Status Register, as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

Table 53 relates the contents of this bit field to the framing acquisition criteria.

TABLE 53: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (FRAMING ON PARITY) WITHIN THE RX DS3 CONFIGURATION AND STATUS REGISTER, AND THE RESULTING FRAMING ACQUISITION CRITERIA

FRAMING ON PARITY (BIT 2)	FRAMING ACQUISITION CRITERIA
0	The In-frame is declared after F-bit synchronization (10 F-bit matches) followed by M-bit synchronization (M-bit matches for 3 DS3 M-frames)
1	The In-frame condition is declared after F-bit synchronization, followed by M-bit synchronization, with valid parity over the frames. Also, the occurrence of parity errors in 2 or more out of 5 frames starts a frame search

Once the Receive DS3 Framer block is operating in the In-Frame condition, normal data recovery and processing of the DS3 data stream begins. The maximum average reframing time is less than 1.5 ms.

5.3.2.2 Frame Maintenance Mode Operation

When the Receive DS3 Framer block is operating in the In-Frame state (per Figure 88), it will then begin to perform Frame Maintenance operations, where it will continue to verify that the F- and M-bits are present, at their proper locations. While the Receive DS3 Framer block is operating in the Frame Maintenance mode, it will declare an Out-of-Frame (OOF) condition if 3 or 6 F-bits (depending upon user selection) out of 16 consecutive F-bits are in error. This selection for the OOF Declaration criteria is made by writing the appropriate value to bit 1 (F-Sync Algo) of the Rx DS3 Configuration and Status Register, as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

Table 54 relates the contents of this bit-field to the OOF Declaration criteria

TABLE 54: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (F-SYNC ALGO) WITHIN THE RX DS3 CONFIGURATION AND STATUS REGISTER, AND THE RESULTING F-BIT OOF DECLARATION CRITERIA USED BY THE RECEIVE DS3 FRAMER BLOCK

F-SYNC ALGO (BIT 1)	OOF DECLARATION CRITERIA
0	OOF is declared when 6 out of 16 consecutive F-bits are in error.
1	OOF is declared when 3 out of 16 consecutive F-bits are in error.

NOTE: *Once the Receive DS3 Framer block has declared an OOF condition, it will transition back to the F-Bit Search state within the DS3 Frame Acquisition/Maintenance algorithm (per Figure 88).*

In addition to selecting an OOF Declaration criteria for the F-bits, the following options exist for configuring the OOF Declaration criteria based upon M-bits.

1. M-bit errors do not cause a OOF Declaration, or
2. OOF will be declared if 3 out of 4 consecutive M-bits are in error.

The selection between these two options is made by writing the appropriate value to Bit 0 (M-Sync Algo) within the Receive DS3 Configuration and Status Register, as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

Table 55 relates the contents of this Bit Field to the M-Bit Error criteria for Declaration of OOF.

TABLE 55: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 0 (M-SYNC ALGO) WITHIN THE RX DS3 CONFIGURATION AND STATUS REGISTER, AND THE RESULTING M-BIT OOF DECLARATION CRITERIA USED BY THE RECEIVE DS3 FRAMER BLOCK

MSYNC ALGO (BIT 0)	OOF DECLARATION CRITERIA
0	M-Bit Errors do not result in the declaration of OOF
1	OOF is declared when 3 out of 4 M-bits are in error.

The Framing on Parity Criteria for OOF Declaration

Finally, the Framer IC offers the Framing on Parity option, which also effects the OOF Declaration criteria. As was mentioned earlier, the Framer IC allows the user to configure the Receive DS3 Framer block to detect 'valid-parity' before declaring itself In-Frame. This same selection also configures the Receive DS3 Framer block to also declare an OOF Condition if a P-bit error is detected in 2 of the last 5 M-frames.

Whenever the Receive DS3 Framer block declares OOF after being in the In-Frame State the following will happen.

- The Receive DS3 Framer will assert the RxOOF output pin (e.g., toggles it "High").
- Bit 4 (RxOOF) within the Rx DS3 Configuration and Status Register will be set to "1" as depicted below.

Rx DS3 Configuration and Status Register, (Address = 0x10)

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

- The Receive DS3 Framer block will also issue a Change in OOF Status interrupt request, anytime there is a change in the OOF status.

5.3.2.3 Forcing a Reframe via Software Command

The Framer IC permits the user to force a reframe procedure of the Receive DS3 Framer block via software command. If a "1" is written into Bit 0 of the I/O Control Register, as depicted below, then the Receive DS3 Framer will be forced into the Frame Acquisition Mode, (or more specifically, in the F-Bit Search State per Figure 88). Afterwards, the Receive DS3 Framer block will begin its search for valid F-Bits. The Framer IC will also respond to this command by asserting the RxOOF output pin, and generating a Change in OOF Status interrupt.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	1

5.3.2.4 Performance Monitoring of the Receive DS3 Framer block

The user can monitor the number of framing bit errors (M and F bits) that have been detected by the Receive DS3 Framer block. This is accomplished by periodically reading the PMON Framing Bit Error Count Registers (Address = 0x52 and 0x53), as depicted below.

PMON FRAMING BIT ERROR EVENT COUNT REGISTER - MSB (ADDRESS = 0X52)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
F-Bit Error Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	1	0	0	0	0	0

PMON FRAMING BIT ERROR EVENT COUNT REGISTER - LSB (ADDRESS = 0X53)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
F-Bit Error Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

When the µP/µC reads these registers, it will read in the number of framing bit errors that have been detected since the last read of these two registers. These registers are reset upon read.

5.3.2.5 DS3 Receive Alarms

The Receive DS3 Framer block is capable of detecting any of the following alarm conditions.

- LOS (Loss of Signal)
- AIS (Alarm Indication Signal)
- The Idle Pattern.
- FERF (Far-End Receive Failure) of Yellow Alarm condition.
- FEBE (Far-End-Block Error)
- Change in AIC State

The methods by which the Receive DS3 Framer block uses to detect and declare each of these alarm conditions are described below.

5.3.2.5.1 The Loss of Signal (LOS) Alarm

The Receive DS3 Framer block will declare a Loss of Signal (LOS) state when it detects 180 consecutive

incoming “0s” via the RxPOS and RxNEG input pins or if the RLOS input pin (from the XRT7300 DS3 LIU or the XRT7295 Line Receiver IC) is asserted (e.g., driven "High"). The Receive DS3 Framer block will indicate the occurrence of an LOS condition by:

1. Asserting the RxLOS output pin (e.g., toggles it "High").
2. Setting Bit 6 (RxLOS) within the Rx DS3 Configuration and Status Register to 1, as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
0	1	0	1	x	x	x	x

3. The Receive DS3 Framer block will generate a Change in LOS Status interrupt request.

NOTE: *The Receive DS3 Framer will also declare an OOF condition and perform all of the notification procedures as described in Section 5.3.2.2.*

4. Force the on-chip Transmit Section to transmit a FERF (Far-End Receive Failure) indicator back out to the remote terminal.

The Receive DS3 Framer block will clear the LOS condition when at least 60 out of 180 consecutive received bits are 1.

NOTE: *The Receive DS3 Framer block will also generate the Change in LOS Condition interrupt, when it clears the LOS Condition.*

The Framer chip allows the user to modify the LOS Declaration criteria such that an LOS condition is declared only if the RLOS input pin (from the XRT7300 DS3/E3/STS-1 LIU IC) is asserted. In this case, the internally-generated LOS criteria of 180 consecutive 0s will be disabled. This can be accomplished by writing a "1" to bit 3 (Int LOS Disable) of the Rx DS3 Configuration and Status Register, as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
X	X	X	X	1	X	X	X

NOTE: *For more information on the RLOS input pin, please see Section 2.1.*

5.3.2.5.2 The Alarm Indication Signal (AIS)

The Receive DS3 Framer block will identify and declare an AIS condition if it detects all of the following conditions in the incoming DS3 Data Stream:

- Valid M-bits, F-bits and P-bits
- All C-bits are zeros.
- X-bits are set to 1
- The Payload portion of the DS3 Frame exhibits a repeating 1010... pattern.

The Receive DS3 Framer block contains, within its circuitry, an Up/Down Counter that supports the assertion and negation of the AIS condition. This counter begins with the value of 0x00 upon power up or reset. The counter is then incremented anytime the Receive DS3 Framer block detects an AIS Type M-frame. This counter is then decremented, or kept at zero value, when the Receive DS3 Framer block detects a non-AIS type M-frame. The Receive DS3 Framer block will declare an AIS Condition if this counter reaches the value of 63 M-frames or greater. Explained another way, the AIS condition is declared if the number of AIS-type M-frames is detected, such that it meets the following conditions:

NAIS - NVALID $\geq$ 63

where:

NAIS = the number of M-frames containing the AIS pattern.

NVALID = the number of M-frames not containing the AIS pattern

If at anytime, the contents of this Up/Down counter exceeds 63 M-frames, then the Receive DS3 Framer block will:

1. Assert the RxAIS output pin by toggling it "High".
2. Set Bit 7 (RxAIS) within the Rx DS3 Configuration and Status Register, to "1" as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
RO	RO	RO	RO	R/W	R/W	R/W	R/W
1	X	X	X	X	X	X	X

3. Generate a Change in AIS Status Interrupt Request to the µP/µC.
4. Force the Transmit Section to transmit a FERF indication back to the remote terminal.

The Receive DS3 Framer block will clear the AIS condition when the following expression is true.

NAIS - NVALID $\leq$ 0.

In other words, once the Receive DS3 Framer block has detected a sufficient number of normal (or Non-AIS) M-frames, such that this Up/Down counter reaches zero, then the Receive DS3 Framer block will clear the AIS Condition indicators. The Receive DS3 Framer block will inform the µC/µP of this negation of the AIS Status by generating a Change in AIS Status interrupt.

5.3.2.5.3 The Idle (Condition) Alarm

The Receive DS3 Framer block will identify and declare an Idle Condition if it receives a sufficient number of M-Frames that meets all of the following conditions.

- Valid M-bits, F-bits, and P-bits
- The 3 CP-bits (in F-Frame #3) are zeros.
- The X-bits are set to 1
- The payload portion of the DS3 Frame exhibits a repeating 1100... pattern.

The Receive DS3 Framer block circuitry includes an Up/Down Counter that is used to track the number of M-frames that have been identified as exhibiting the Idle Condition by the Receive DS3 Framer block. The contents of this counter are set to zero upon reset or power up. This counter is then incremented whenever the Receive DS3 Framer block detects an Idle-type M-frame. The counter is decremented, or kept at zero if a non-Idle M-frame is detected. If the Receive DS3 Framer block detects a sufficient number of Idle-type M-frames, such that the counter reaches the number 63, then the Receive DS3 Framer block will declare the Idle Condition. Explained another way, the Receive DS3 Framer block will declare an Idle Condition if the number of Idle-Pattern M-frames is detected such that it meets the following conditions.

NIDLE - NVALID $\geq$ 63,

where:

NIDLE = the number of M-frames containing the Idle Pattern

NVALID = the number of M-frames not exhibit the Idle Pattern

Anytime the contents of this Up/Down Counter reaches the number 63, then the Receive DS3 Framer block will:

1. Set Bit 5 (RxIdle) within the Rx DS3 Configuration and Status Register, to "1" as depicted below.

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

RX DS3 CONFIGURATION AND STATUS REGISTER, (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Int LOS Disable	Framing on Parity	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
X	X	1	X	X	X	X	X

2. Generate a Change in Idle Status Interrupt Request to the local µP/µC.

The Receive DS3 Framer block will clear the Idle Condition if it has detected a sufficient number of Non-Idle M-frames, such that this Up/Down Counter reaches the value 0.

5.3.2.5.4 The Detection of (FERF) or Yellow Alarm Condition

The Receive DS3 Framer block will identify and declare a Yellow Alarm condition or a Far-End Receive Failure (FERF) condition, if it starts to receive DS3 frames with both of its X-bits set to 0.

When the Receive DS3 Framer block detects a FERF condition in the incoming DS3 frames, then it will then do the following.

1. It will assert the RxFERF (bit-field 4) within the Rx DS3 Status Register, as depicted below.

RX DS3 STATUS REGISTER (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used			Rx FERF	RxAIC	RxFEBE [2]	RxFEBE [1]	RxFEBE [0]
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	1	X	X	X	X

This bit-field will remain asserted for the duration that the Yellow Alarm condition exists.

2. The Receive DS3 Framer block will also generate a Change in FERF Status interrupt to the µP/µC. Consequently, the Receive DS3 Framer block will also assert the FERF Interrupt Status bit, within the Rx DS3 Interrupt Status Register, as depicted below.

RX DS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Cp Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	IDLE Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	X	X	X	1	X	X	X

The Receive DS3 Framer block will clear the FERF condition, when it starts to receive Receive DS3 Frames that have its X bits set to 1.

NOTE: *The FERF indicator is frequently referred to as the Yellow Alarm.*

5.3.2.5.5 The Detection of the FEBE Events

As described in Section 5.2.4.2.1.9, a given Terminal Equipment will set the three FEBE (Far-End Block Error) bit-fields to the value [1, 1, 1] (e.g., all of the FEBE bits are set to "1") within the outbound DS3 frames if, all of the following conditions are true about the incoming DS3 line signal.

- The Receive Circuitry (within the Terminal Equipment) detects no P-Bit Errors.
- The Receive Circuitry (within the Terminal Equipment) detects no CP-Bit Errors.

If the Receive Section of the Terminal Equipment detects any P or CP bit errors, then the Transmit Section of the Terminal Equipment will set the three FEBE bits (within the outbound DS3 data stream) to a value other than [1, 1, 1].

How does the Receive DS3 Framer block (within the XRT74L74) respond when it receives a DS3 frame with all three (3) of its FEBE bit-fields set to "1"?

As mentioned above, the Terminal Equipment will transmit DS3 frames, with the FEBE bits set to [1, 1, 1], during un-erred conditions. Hence, if the Receive DS3 Framer block (within the XRT74L74 Framer IC) receives DS3 frames with the FEBE bits set to [1, 1, 1] it will interpret this event as an un-erred event, and will continue normal operation.

However, if the Receive DS3 Framer block receives a DS3 frame with the FEBE bits set to a value other than [1, 1, 1], then it will increment the PMON FEBE Event Count Registers (which are located at address locations 0x58 and 0x59 within the Framer Address space).

5.3.2.5.6 Detection of Change in the AIC State

Section 5.1 indicates that the AIC (Application Identification Channel) bit-field is the third overhead bit, within F-Frame # 1. This particular bit-field is set to "1" for the C-Bit Parity Framing Format, and is set to "0" for the M13 Framing Format.

Hence, a given Terminal Equipment receiving a DS3 data stream can identify the framing format of this DS3 data stream, by reading the value fo the AIC bit-field. The Receive DS3 Framer block permits the user's Microcontroller/MIcroprocessor to determine the state of the AIC bit-field (within the incoming DS3 data stream) by writing the value of the AIC bit-field, within the most recently received DS3 frame, into bit 3 (RxAIC) within the Rx DS3 Status Register (Address = 0x11), as illustrated below.

RXDS3 STATUS REGISTER (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved			RxFERF	RxAIC	RxFEBE[2:0]		
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

The Receive DS3 Framer block will also generate an interrupt if it detects a change of state in the AIC bit-field (within the incoming DS3 data stream). If this occurs, then the Receive DS3 Framer block will set Bit 2 (AIC Interrupt Status) within the Rx DS3 Interrupt Stauts Register (Address = 0x13) to “1” as illustrated below.r

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

5.3.2.6 Performance Monitoring of the DS3 Transport Medium

The DS3 Frame consists of some overhead bits that are used to support performance monitoring of the DS3 Transmission Link. These bits are the P-Bits and the CP-Bits.

5.3.2.6.1 P-Bit Checking/Options

The remote Transmit DS3 Framer will compute the even parity of the payload portion of an outbound DS3 Frame and will place the resulting parity bit value in the 2 P-bit-fields within the very next outbound DS3 Frame. The value of these two bits fields is expected to be the identical.

The Receive DS3 Framer block, while receiving each of these DS3 Frames (from the remote Transmit DS3 Framer), will compute the even-parity of the payload portion of the frame. The Receive DS3 Framer block will then compare this locally computed parity value to that of the P-bit fields within the very next DS3 Frame. If the Receive DS3 Framer block detects a parity error, then two things will happen:

1. The Receive DS3 Framer block will inform the µP/µC of this occurrence by generating a Detection of P-Bit Error interrupt,
2. The Receive DS3 Framer block will alter the value of the FEBE bits, (to a pattern other than 111) that the Near-End Transmit DS3 Framer will be transmitting back to the remote Terminal.
3. The XRT74L74 Framer IC will increment the PMON Parity Error Event Count Registers (Address = 0x54 and 0x55) for each detected parity error, in the incoming DS3 data stream. The bit-format of these two registers follows.

PMON PARITY ERROR EVENT COUNT REGISTER - MSB (ADDRESS = 0X54)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Parity Error Count - High Byte							
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

PMON PARITY ERROR EVENT COUNT REGISTER - LSB (ADDRESS = 0X55)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Parity Error Count - "Low" Byte							
RO	RO	RO	RO	RO	RO	RO	RO

PMON PARITY ERROR EVENT COUNT REGISTER - LSB (ADDRESS = 0X55)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
0	0	0	0	0	0	0	0

When the µP reads these registers, it will read in the number of parity-bit errors that have been detected by the Receive DS3 Framer block, since the last time these registers were read. These registers are reset upon read.

NOTE: When the Framing with Parity option is selected, the Receive DS3 Framer block will declared an OOF condition if P-bit errors were detected in two out of 5 consecutive DS3 M-frames.

5.3.2.6.2 CP-Bit Checking/Options

CP-bits are very similar to P-bits except for the following.

1. CP-bits are used to permit performance monitoring over an entire DS3 path (e.g., from the source terminal) through any number of mid-network terminals to the sink terminal).
2. P-bits are used to permit performance monitoring of a DS3 data stream, as it is transmitted from one terminal to an adjacent terminal.

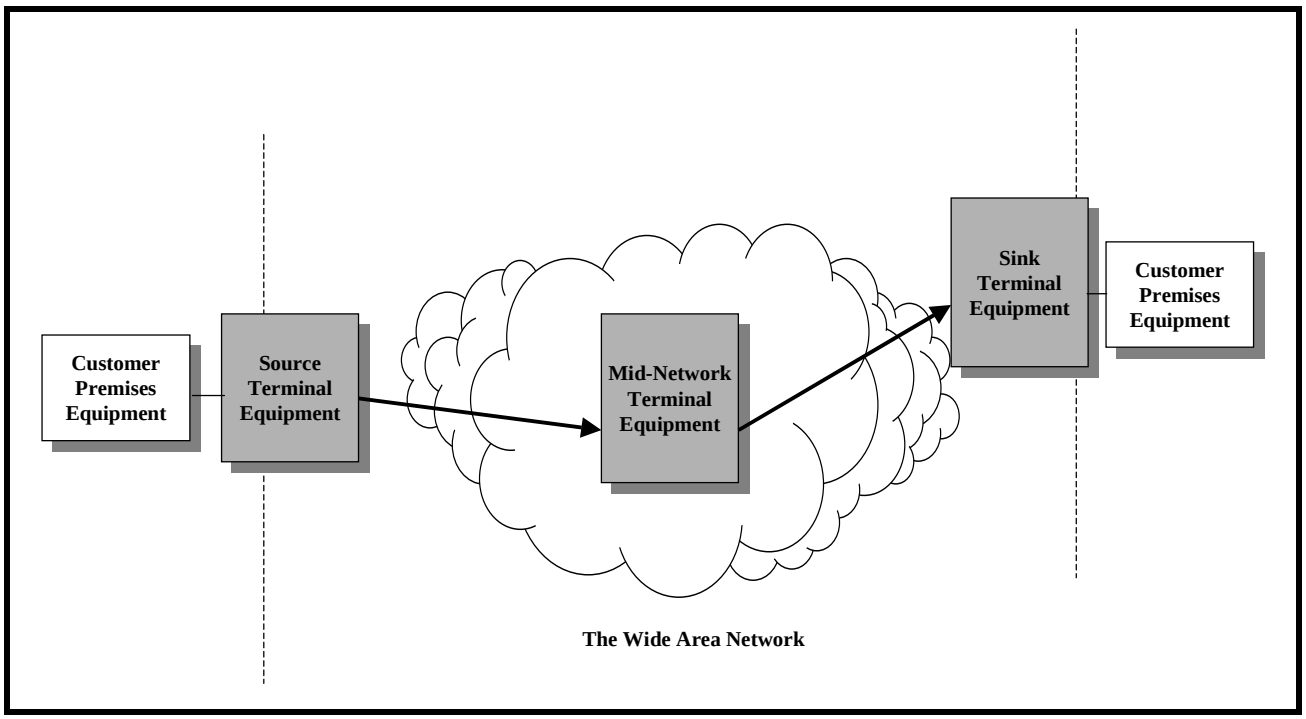
How CP-Bits are Processed

The following section describes how the CP-bits are processed at three locations.

- The Source Terminal Equipment
- The Mid-Network Terminal Equipment
- The Sink Terminal Equipment

Figure_62 presents a simple illustration of the locations of these three types of Terminal Equipment, within the Wide-Area Network.

FIGURE 89. A SIMPLE ILLUSTRATION OF THE LOCATIONS OF THE SOURCE, MID-NETWORK AND SINK TERMINAL EQUIPMENT (FOR CP-BIT PROCESSING)



NOTE: The use of the terms Source and Sink Terminal Equipment are used to simplify this discussion of CP-Bit Processing. In reality, the Source Terminal Equipment (in Figure_62) will also function as the Sink Terminal Equipment (for DS3 traffic traveling in the opposite direction). Likewise, the Sink Terminal Equipment (in Figure_62) will also function as the Source Terminal Equipment.

Processing at the Source Terminal Equipment

The Source Terminal Equipment (located at one edge of the wide-area network) will typically receive its DS3 payload data from some Customer Premise Equipment (CPE). As the Source Terminal Equipment receives this data from the CPE, it will compute the even-parity value over all bits within a given outbound DS3 frame. The Terminal Equipment will then insert this even parity value into both of the P-bit fields and

both of the CP-bits fields, within the very next outbound DS3 frame.

Hence, both the P-bit values and CP-bit values will originate at the Source Terminal Equipment.

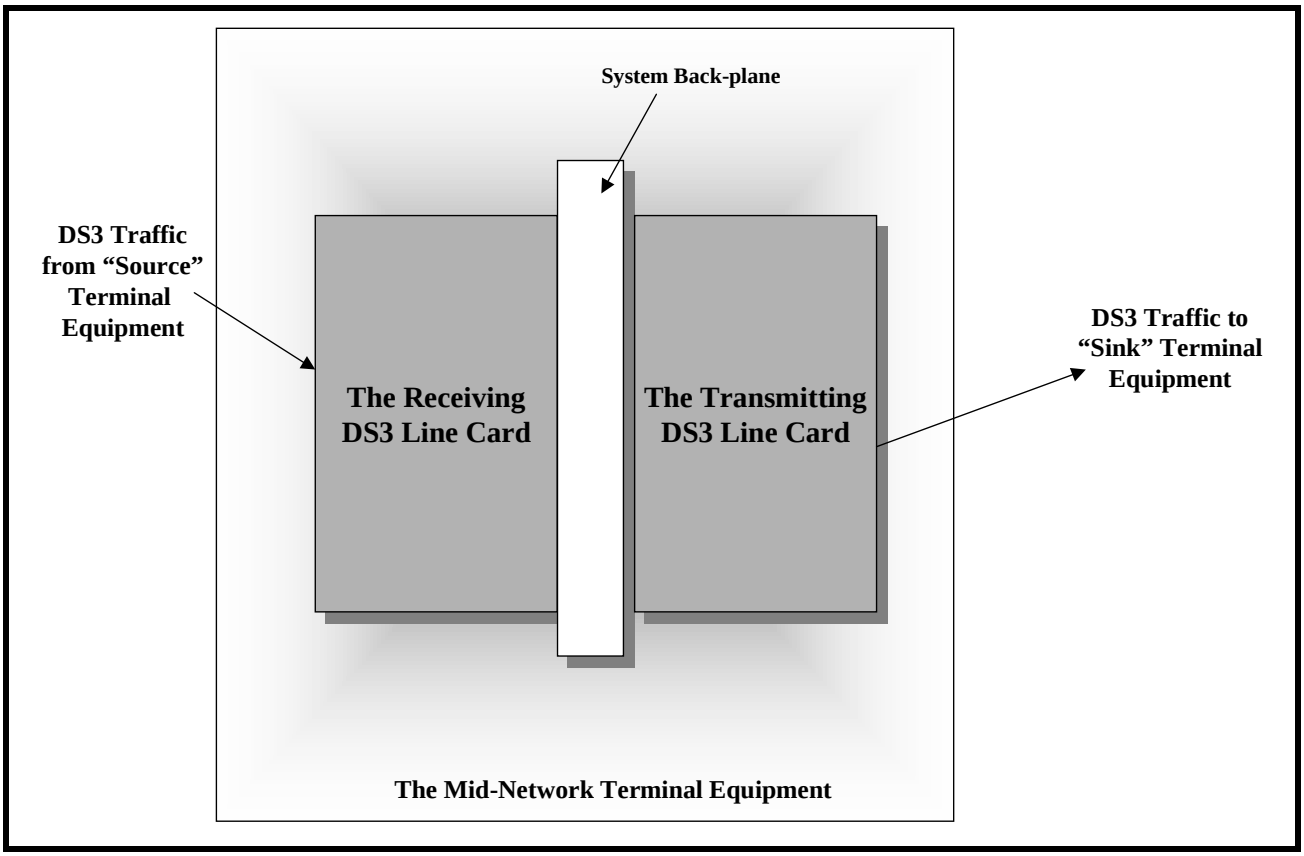
Processing at the Mid-Network Terminal Equipment

The Mid-Network Terminal Equipment has the task of doing the following.

- Receiving a DS3 data stream, via the Receive WAN Interface Line Card.
- Transmitting this same DS3 data stream (out to another Remote Terminal Equipment) via the Transmit WAN Interface Line Card.

Figure 90 presents an illustration of the basic architecture of the Mid-Network Terminal Equipment.

FIGURE 90. ILLUSTRATION OF THE PRESUMED CONFIGURATION OF THE MID-NETWORK TERMINAL EQUIPMENT



Operation of the Receive WAN Interface Line Card

The Receive WAN Interface line card receives a DS3 data stream from some remote Terminal Equipment. As the Receive WAN Interface card does this, it will also do the following:

1. Compute and verify the "P-Bits" of each inbound DS3 frame.
2. Compute and verify the "CP-Bits" of each inbound DS3 frame.
3. Output both the payload and overhead bits to the system back-plane.

Operation of the Transmit WAN Interface Line Card

The Transmit WAN Interface Line Card receives the outbound DS3 data stream from the system back-plane. As the Transmit WAN Interface Line Card receives this data it will also do the following.

1. Extract out the "CP-bit" values, from the Receive WAN Interface line card (via the system back-plane) and insert these values into the CP-bit fields, within the outbound DS3 data stream, via the Transmit Overhead Data Input Interface block of the XRT74L74 Framer IC.
2. Compute the even-parity over all bits, within a given outbound DS3 frame, and insert this value into the "P" bits within the very next outbound DS3 frame.
3. Transmit this resulting DS3 data stream to the remote terminal equipment.

Processing at the Sink Terminal

The Sink Terminal Equipment (located at the opposite edge of the wide-area-network, from the Source Terminal Equipment) will receive and terminate this DS3 data stream. As the Sink Terminal Equipment receives this DS3 data stream it will also do the following.

1. Compute and verify the “P”-bits within each inbound DS3 frame.
2. Compute and verify the “CP” bits within each inbound DS3 frame.

5.3.3 The Receive HDLC Controller Block

The Receive DS3 HDLC Controller block can be used to receive either bit-oriented signaling (BOS) or message-oriented signaling (MOS) type data link messages. The Receive DS3 HDLC Controller block can also be configured to receive both types of message from the remote terminal equipment.

Both BOS and MOS types of HDLC message processing are discussed in detail below.

5.3.3.1 Bit-Oriented Signaling (or FEAC) Processing via the Receive DS3 HDLC Controller.

The Receive DS3 HDLC Controller block consists of two major sub-blocks

- The Receive FEAC Processor
- The LAPD Receiver

This section describes how to operate the Receive FEAC Processor.

If the Receive DS3 Framer block is operating in the C-bit Parity Framing format, then the FEAC bit-field within the DS3 Frame can be used to receive FEAC (Far End Alarm and Control) messages (See Figure 91). Each FEAC code word is actually six bits in length. However, this six bit FEAC Code word is encapsulated with 10 framing bits to form a 16 bit message of the form:

	FEAC CODE WORD						FRAMING								
0	d5	d4	d3	d2	d1	d0	0	1	1	1	1	1	1	1	1

Where, [d5, d4, d3, d2, d1, d0] is the FEAC Code word. The rightmost bit of the 16-bit data structure (e.g., a 1) will be received first. Since each DS3 Frame contains only 1 FEAC bit-field, 16 DS3 Frames are required to transmit the 16 bit FEAC code message. The six bits, labeled “d5” through “d0” can represent 64 distinct messages, of which 43 have been defined in the standards.

The Receive FEAC Processor frames and validates the incoming FEAC data from the remote Transmit FEAC Processor via the received FEAC channel. Additionally, the Receive FEAC Processor will write the Received FEAC code words into an 8 bit Rx-FEAC register. Framing is performed by looking for two 0s spaced 6 bits apart preceded by 8 1s. The Receive DS3 HDLC Controller contains two registers that support FEAC Message Reception.

- Rx DS3 FEAC Register (Address = 0x16)
- Rx DS3 FEAC Interrupt Enable/Status Register (Address = 0x17)

The Receive FEAC Processor generates an interrupt upon validation and removal of the incoming FEAC Code words.

Operation of the Receive DS3 FEAC Processor

The Receive FEAC Processor will validate or remove FEAC code words that it receives from the remote Transmit FEAC Processor. The FEAC Code Validation and Removal functions are described below.

FEAC Code Validation

When the remote terminal equipment wishes to send a FEAC message to the Local Receive FEAC Processor, it (the remote terminal equipment) will transmit this 16 bit message, repeatedly for a total of 10 times. The Receive FEAC Processor will frame to this incoming FEAC Code Message, and will attempt to validate this message. Once the Receive FEAC Processor has received the same FEAC code word in at least 8 out of the last 10 received codes, it will validate this code word by writing this 6 bit code word into the Receive DS3 FEAC Register. The Receive FEAC Processor will then inform the µC/µP of this Receive FEAC validation event by generating a Rx FEAC Valid interrupt and asserting the FEAC Valid and the RxFEAC Valid Interrupt Status Bits in the Rx DS3 Interrupt Enable/Status Register, as depicted below. The Bit Format of the Rx DS3 FEAC Register is presented below.

RX DS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	Not Used	Not Used	FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
X	X	X	1	X	0	1	1

The bit-format of the Rx DS3 FEAC register is presented below. It is important to note that the last validated FEAC code word will be written into the shaded bit-fields below.

RX DS3 FEAC REGISTER (ADDRESS = 0X16)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	RxFEAC [5]	RxFEAC [4]	RxFEAC [3]	RxFEAC [2]	RxFEAC [1]	RxFEAC [0]	Not Used
RO	RO	RO	RO	RO	RO	RO	RO
0	d5	d4	d3	d2	d1	d0	0

The purpose of generating an interrupt to the µP, upon FEAC Code Word Validation is to inform the local µP that the Framer has a newly received FEAC message that needs to be read. The local µP would read-in this FEAC code word from the Rx DS3 FEAC Register (Address = 0x16).

FEAC Code Removal

After the 10th transmission of a given FEAC code word, the remote terminal equipment may proceed to transmit a different FEAC code word. When the Receive FEAC processor detects this occurrence, it must Remove the FEAC codeword that is presently residing in the Rx DS3 FEAC Register. The Receive FEAC Processor will remove the existing FEAC code word when it detects that 3 (or more) out of the last 10 received FEAC codes are different from the latest validated FEAC code word. The Receive FEAC Processor will inform the local µP/µC of this removal event by generating a Rx FEAC Removal interrupt, and asserting the RxFEAC Remove Interrupt Status bit in the Rx DS3 Interrupt Enable/Status Register, as depicted below.

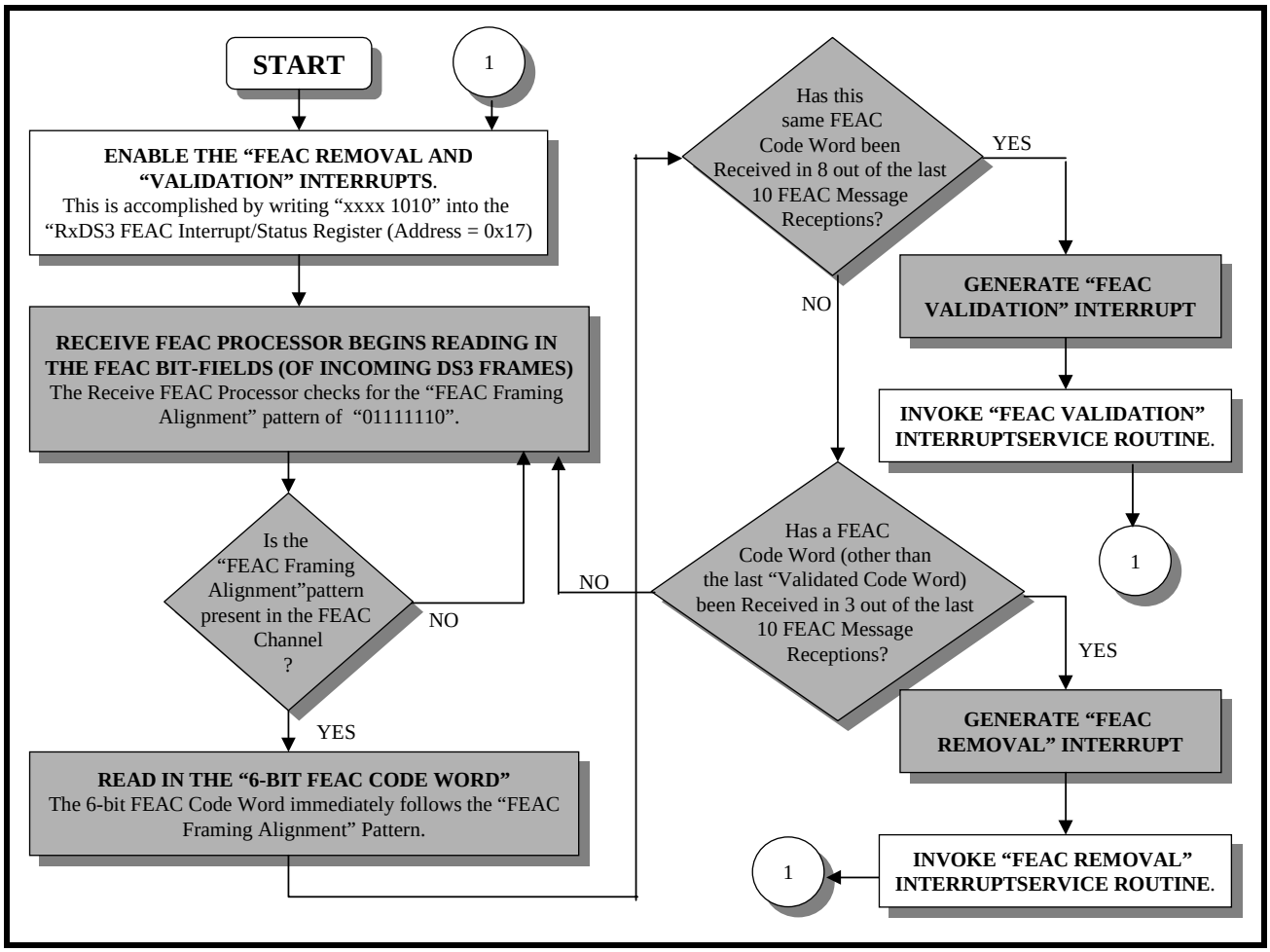
RX DS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	Not Used	Not Used	FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
X	X	X	0	1	1	X	0

Additionally, the Receive FEAC processor will also denote the removal event by setting the FEAC Valid bit-field (Bit 4), within the Rx DS3 FEAC Interrupt Enable/Status Register to 0, as depicted above.

The description of Bits 0 through 3 within this register, all support Interrupt Processing, and will therefore be presented in Section 5.3.6. Figure 91 presents a flow diagram depicting how the Receive FEAC Processor functions.

FIGURE 91. FLOW DIAGRAM DEPICTING HOW THE RECEIVE FEAC PROCESSOR FUNCTIONS



NOTES:

1. *The white (e.g., unshaded) boxes reflect tasks that the user's system must perform in order to configure the Receive FEAC Processor to receive FEAC messages.*
2. *A brief description of the steps that must exist within the FEAC Validation and FEAC Removal Interrupt Service Routines exists in Section 5.3.3*

5.3.3.2 The Message Oriented Signaling (e.g., LAP-D) Processing via the Receive DS3 HDLC Controller block

The LAPD Receiver (within the Receive DS3 HDLC Controller block) allows the user to receive PMDL messages from the remote terminal equipment, via the inbound DS3 frames. In this case, the inbound message bits will be carried by the 3 DL bit-fields of F-Frame 5, within each DS3 M-Frame. The remote LAPD Transmitter will transmit a LAPD Message to the Near-End Receiver via these three bits within each DS3 Frame. The LAPD Receiver will receive and store the information portion of the received LAPD frame into the Receive LAPD Message Buffer, which is located at addresses: 0xDE through 0x135 within the on-chip RAM. The LAPD Receiver has the following responsibilities.

- Framing to the incoming LAPD Messages
- Filtering out stuffed 0s (within the information payload)
- Storing the Frame Message into the Receive LAPD Message Buffer
- Perform Frame Check Sequence (FCS) Verification
- Provide status indicators for
  - End of Message (EOM)
  - Flag Sequence Byte detected
  - Abort Sequence detected
  - Message Type
  - C/R Type
  - The occurrence of FCS Errors

The LAPD receiver's actions are facilitated via the following two registers.

- Rx DS3 LAPD Control Register
- Rx DS3 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 (0x7E), as depicted in Figure 92.

FIGURE 92. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The 16 bit FCS is calculated using CRC-16, x16 + x12 + x5 + 1

The microprocessor/microcontroller (at the remote terminal), while assembling the LAPD Message frame, will insert an additional byte at the beginning of the information (payload) field. This first byte of the information field indicates the type and size of the message being transferred. The value of this information field and the corresponding message type/size follow:

CL Path Identification = 0x32 (76 bytes)

IDLE Signal Identification = 0x34 (76 bytes)

Test Signal Identification = 0x38 (76 bytes)

ITU-T Path Identification = 0x3F (82 bytes)

The LAPD Receiver must be enabled before it can begin receiving any LAPD messages. The LAPD Receiver can be enabled by writing a "1" into Bit 2 (RxLAPD Enable) within the Rx DS3 LAPD Control Register. The bit format of this register is depicted below.

RX DS3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	Not Used	Not Used	Not Used	Not Used	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	1	X	X

Once the LAPD Receiver has been enabled, it will begin searching for the Flag Sequence octets (0x7E), in the DL bit-fields, within the incoming DS3 frames. When the LAPD Receiver finds the flag sequence byte, it will assert the Flag Present bit (Bit 0) within the Rx DS3 LAPD Status Register, as depicted below.

RX DS3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used	RxAbort	RxLAPD Type[1, 0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO

RX DS3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
X	X	X	X	X	X	X	1

The receipt of the Flag Sequence octet can mean one of two things.

1. The Flag Sequence byte marks the beginning or end of an incoming LAPD Message.
2. The received Flag Sequence octet could be just one of many Flag Sequence octets that are transmitted via the DS3 Transport Medium, during idle periods between the transmission of LAPD Messages.

The LAPD Receiver will clear the Flag Present bit as soon as it has received an octet that is something other than the Flag Sequence octet. At this point, 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 DS3 LAPD Status Register. However, if this next octet is Octet #2 of an incoming LAPD Message, then the Rx DS3 LAPD Status Register will begin to present some additional status information on this incoming message. Each of these indicators is presented below in sequential order.

Bit 3 - RxCR Type - C/R (Command/Response) Type

This bit-field reflects the contents of the C/R bit-field within octet #2 of the LAPD Frame Header. When this bit is "0" it means that this message is originating from a customer installation. When this bit is "1" it means that this message is originating from a network terminal.

Bit 4,5 - RxLAPD Type[1, 0] - LAPD Message Type

The combination of these two bit fields indicate the Message Type and the Message Size of the incoming LAPD Message frame. Table 56 relates the values of Bits 4 and 5 to the Incoming LAPD Message Type/Size.

TABLE 56: THE RELATIONSHIP BETWEEN RXLAPDTYPE[1:0] AND THE RESULTING LAPD MESSAGE TYPE AND SIZE

RXLAPD TYPE[1, 0]	MESSAGE TYPE	MESSAGE SIZE
00	CL Path Identification	76 bytes
01	Idle Signal Identification	76 bytes
10	Test Signal Identification	76 bytes
11	TU-T Path Identification	82 bytes

NOTE: *The Message Size pertains to the size of the Information portion of the LAPD Message Frame (as presented in Figure 92).*

Bit 3 - Flag Present

The LAPD Receiver should receive another Flag Sequence octet, which marks the End of the Message. Therefore, this bit field should be asserted once again.

Bit 1 - EndOfMessage - End of LAPD Message Frame

Upon receipt of the closing Flag Sequence octet, this bit-field should be asserted. The assertion of this bit-field indicates that a LAPD Message Frame has been completely received. Additionally, if this newly received LAPD Message is different from the previous message, then the LAPD Receiver will inform the µC/µP of the EndOfMessage event by generating an interrupt.

Bit 2 - RxFCSErr - Frame Check Sequence Error Indicator

The LAPD Receiver will take the incoming LAPD Message and compute its own version of the Frame Check Sequence (FCS) word. Afterwards, the LAPD Receiver will compare its computed value with that it has received from the remote LAPD Transmitter. If these two values match, then the LAPD Receiver will presume that the LAPD Message has been properly received and the contents of the Received LAPD Message (payload portion) will be retained at locations 0xDE through 0x135 in on-chip RAM. The LAPD Receiver will indicate an error-free reception of the LAPD Message by keeping this bit field negated (Bit 2 = 0). However, if these two FCS values do not match, then the received LAPD Message is corrupted and the user is advised not to process this erroneous information. The LAPD Receiver will indicate an erred receipt of this message by setting this bit-field to 1.

NOTE: *The Receive DS3 HDLC Controller block will not generate an interrupt to the µP due to the detection of an FCS error. Therefore, the user is advised to validate each*

and every received LAPD message by checking this bit-field prior to processing the LAPD message.

Removal of Stuff Bits from the Payload Portion of the incoming LAPD Message

While the LAPD Receiver is receiving a LAPD Message, it has the responsibility of removing all of the "0" stuff bits from the Payload Portion of the incoming LAPD Message Frame. Recall that the text in Section *5.2.3.2* indicated that the LAPD Transmitter (at the remote terminal) will insert a "0" immediately following a string of 5 consecutive “1s” within the payload portion of the LAPD Message frame. The LAPD Transmitter performs this bit-stuffing procedure in order to prevent the user data from mimicking the Flag Sequence octet (0x7E) or the ABORT sequence. Therefore, in order to recover the user data to its original content (prior to the bit-stuffing), the LAPD Receiver will remove the "0" that immediately follows a string of 5 consecutive 1s.

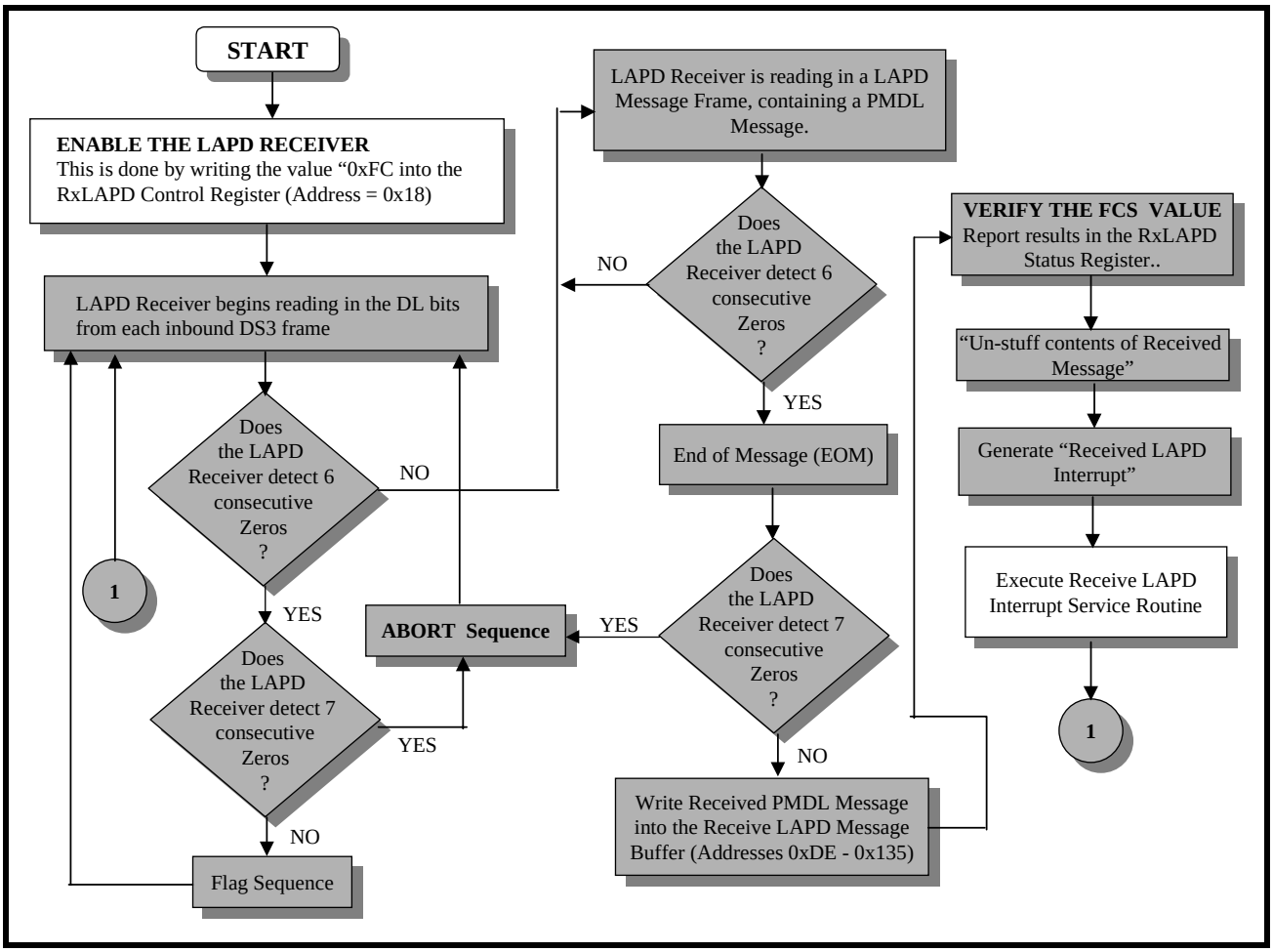
Writing the Incoming LAPD Message into the Receive LAPD Message Buffer

The LAPD receiver will obtain the LAPD Message frame from the incoming DS3 data-stream. In addition to processing the framing overhead octets, performing error checking (via FCS) and removing the stuffed 0s from the user payload data. The LAPD Receiver will also write the payload portion of the LAPD Frame into the Receive LAPD Message buffer at locations 0xDE through 0x135 in on-chip RAM.

Therefore, the local µP/µC must read this location when it wishes to process this newly received LAPD Message.

Figure 93 presents a flow chart depicting how the LAPD Receiver works.

FIGURE 93. FLOW CHART DEPICTING THE FUNCTIONALITY OF THE LAPD RECEIVER



NOTES:

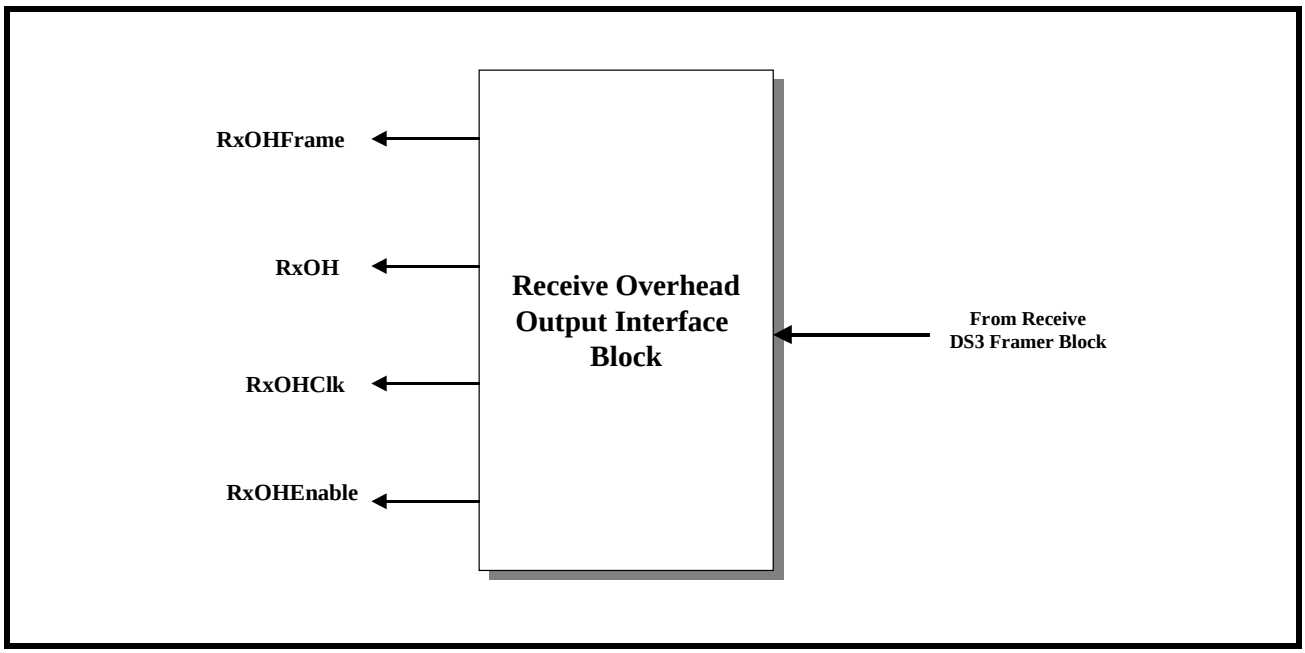
1. *The white (e.g., unshaded) boxes reflect tasks that the user’s system must perform in order to configure the LAPD Receiver to receive LAPD Messages.*

2. *A brief description of the steps that must exist within the Receive LAPD Interrupt Service routine exists in Section* 5.3.6

5.3.4 The Receive Overhead Data Output Interface

Figure 94 presents a simple illustration of the Receive Overhead Data Output Interface block within the XRT74L74.

FIGURE 94. A SIMPLE ILLUSTRATION OF THE RECEIVE OVERHEAD OUTPUT INTERFACE BLOCK



The DS3 frame consists of 4760 bits. Of these bits, 4704 bits are payload bits and the remaining 56 bits are overhead bits. The XRT74L74 has been designed to handle and process both the payload type and overhead type bits for each DS3 frame.

The Receive Payload Data Output Interface block, within the Receive Section of the XRT74L74, has been designed to handle the payload bits. Likewise, the Receive Overhead Data Output Interface block has been designed to handle and process the overhead bits.

The Receive Overhead Data Output Interface block unconditionally outputs the contents of all overhead bits within the incoming DS3 data stream. The XRT74L74 does not offer the user a means to shut off this transmission of data. However, the Receive Overhead Output Interface block does provide the user with the appropriate output signals for external Data Link Layer equipment to sample and process these overhead bits, via the following two methods.

- Method 1- Using the RxOHClk clock signal.
- Method 2 - Using the RxClk and RxOHEnable output signals.

Each of these methods are described below.

5.3.4.1 Method 1 - Using the RxOHClk Clock signal

The Receive Overhead Data Output Interface block consists of four (4) signals. Of these four signals, the following three signals are to be used when sampling the DS3 overhead bits via Method 1.

- RxOH
- RxOHClk
- RxOHFrame

Each of these signals are listed and described below in Table 57.

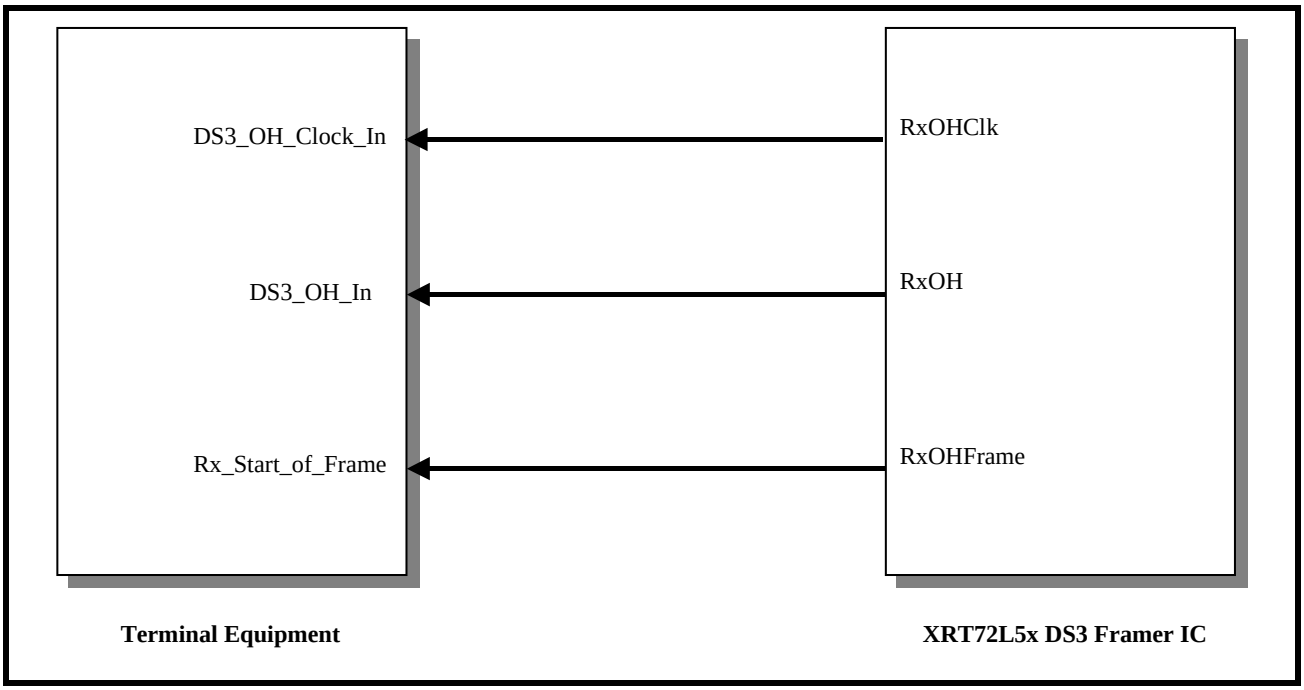
TABLE 57: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming DS3 frames, via this pin. The Receive Overhead Data Output Interface block will output a given overhead bit, upon the falling edge of RxOHClk. Hence, the external data link equipment should sample the data, at this pin, upon the rising edge of RxOHClk. The XRT74L74 will always output the DS3 Overhead bits via this output pin. There are no external input pins or register bit settings available that will disable this output pin.
RxOHClk	Output	**Receive Overhead Data Output Interface Clock Signal:** The XRT74L74 will output the Overhead bits (within the incoming DS3 frames), via the RxOH output pin, upon the falling edge of this clock signal. As a consequence, the user's data link equipment should use the rising edge of this clock signal to sample the data on both the RxOH and RxOHFrame output pins. This clock signal is always active.
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin "High" (for one period of the RxOHClk signal), whenever the first overhead bit within a given DS3 frame is being driven onto the RxOH output pin.

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 1)

Figure 95 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment when using Method 1 to sample and process the overhead bits from the inbound DS3 data stream.

FIGURE 95. ILLUSTRATION OF HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (FOR METHOD 1).



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the inbound DS3 data stream (via

the Receive Overhead Data Output Interface block) then it is expected to do the following:

1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input signal) on the rising edge of the RxOHClk (e.g., the DS3_OH_Clock_In) signal.
2. Keep track of the number of rising clock edges that have occurred in the RxOHClk (e.g., the DS3_OH_Clock_In) signal, since the last time the RxOHFrame signal was sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.

Table 58 relates the number of rising clock edges (in the RxOHClk signal, since the RxOHFrame signal was last sampled "High") to the DS3 Overhead bit that is being output via the RxOH output pin.

TABLE 58: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

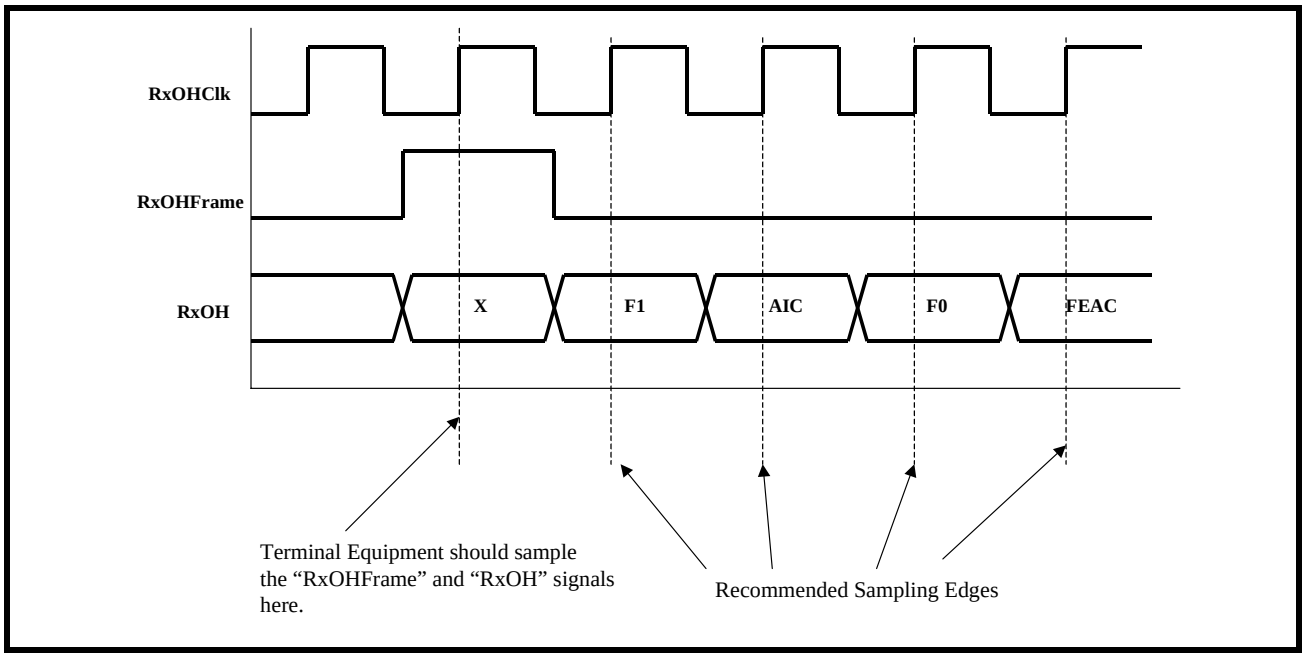
NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (Clock edge is coincident with RxOHFrame being detected "High")	X
1	F1
2	AIC
3	F0
4	NA
5	F0
6	FEAC
7	F1
8	X
9	F1
10	UDL
11	F0
12	UDL
13	F0
14	UDL
15	F1
16	P
17	F1
18	CP
19	F0
20	CP
21	F0
22	CP
23	F1
24	P
25	F1
26	FEBE
27	F0
28	FEBE
29	F0

TABLE 58: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
30	FEBE
31	F1
32	M0
33	F1
34	DL
35	F0
36	DL
37	F0
38	DL
39	F1
40	M1
41	F1
42	UDL
43	FO
44	UDL
45	FO
46	UDL
47	F1
48	M0
49	F1
50	UDL
51	F0
52	UDL
53	F0
54	UDL
55	F1

Figure 96 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 1 is being used to sample the incoming DS3 overhead bits.

FIGURE 96. ILLUSTRATION OF THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD OUTPUT INTERFACE (FOR METHOD 1).



Method 2 - Using RxOutClk and the RxOHEnable signals

Method 1 requires that the Terminal Equipment be able to handle an additional clock signal, RxOHClk. However, there may be a situation in which the Terminal Equipment circuitry does not have the means to accommodate and process this extra clock signal, in order to use the Receive Overhead Data Output Interface. Hence, Method 2 is available. Method 2 involves the use of the following signals.

- RxOH
- RxOutClk
- RxOHEnable
- RxOHFrame

Each of these signals are listed and described below in Table 59.

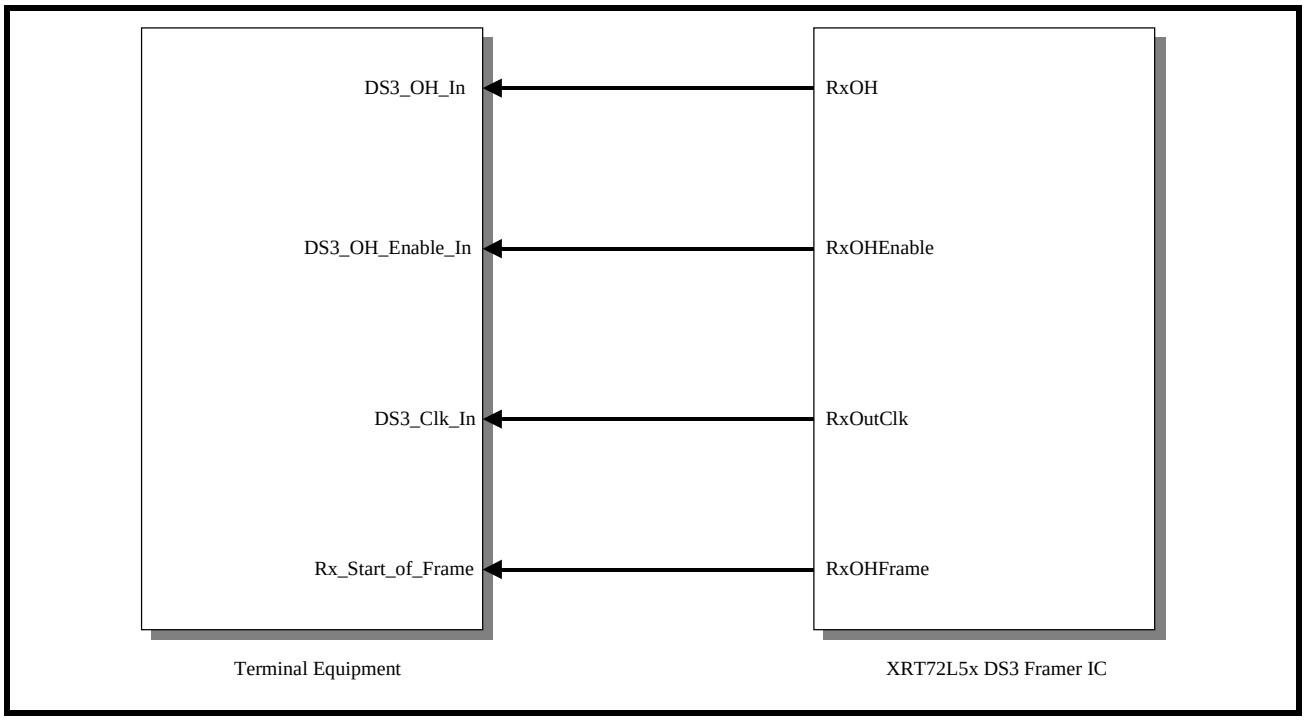
TABLE 59: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (METHOD 2)

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming DS3 frames, via this pin. The Receive Overhead Output Interface will pulse the RxOHEnable output pin (for one RxOutClk period) at approximately the middle of the RxOH bit period. The user is advised to design the Terminal Equipment to latch the contents of the RxOH output pin, whenever the RxOHEnable output pin is sampled "High" on the falling edge of RxOutClk.
RxOHEnable	Output	**Receive Overhead Data Output Enable - Output pin:** The XRT74L74 will assert this output signal for one RxOutClk period when it is safe for the Terminal Equipment to sample the data on the RxOH output pin.
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin "High" (for one period of the RxOH signal), whenever the first overhead bit, within a given DS3 frame is being driven onto the RxOH output pin.
RxOutClk	Output	**Receive Section Output Clock Signal:** This clock signal is derived from the RxLineClk signal (from the LIU) for loop-timing applications, and the TxInClk signal (from a local oscillator) for local-timing applications. For DS3 applications, this clock signal will operate at 44.736MHz. The user is advised to design the Terminal Equipment to latch the contents of the RxOH pin, anytime the RxOHEnable output signal is sampled "High" on the falling edge of this clock signal.

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 2)

Figure 97 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment, when using Method 2 to sample and process the overhead bits from the inbound DS3 data stream.

FIGURE 97. ILLUSTRATION OF HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (FOR METHOD 2).



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the inbound DS3 data stream (via the Receive Overhead Data Output Interface), then it is expected to do the following.

1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input) on the falling edge of the RxOutClk clock signal, whenever the RxOHEnable output signal is also sampled "High".
2. Keep track of the number of times that the RxOHEnable signal has been sampled "High" since the last time the RxOHFrame was also sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.
3. Table 60 relates the number of RxOHEnable output pulses (that have occurred since both the RxOHFrame and the RxOHEnable pins were both sampled "High") to the DS3 overhead bit that is being output via the RxOH output pin.

TABLE 60: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES ((SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

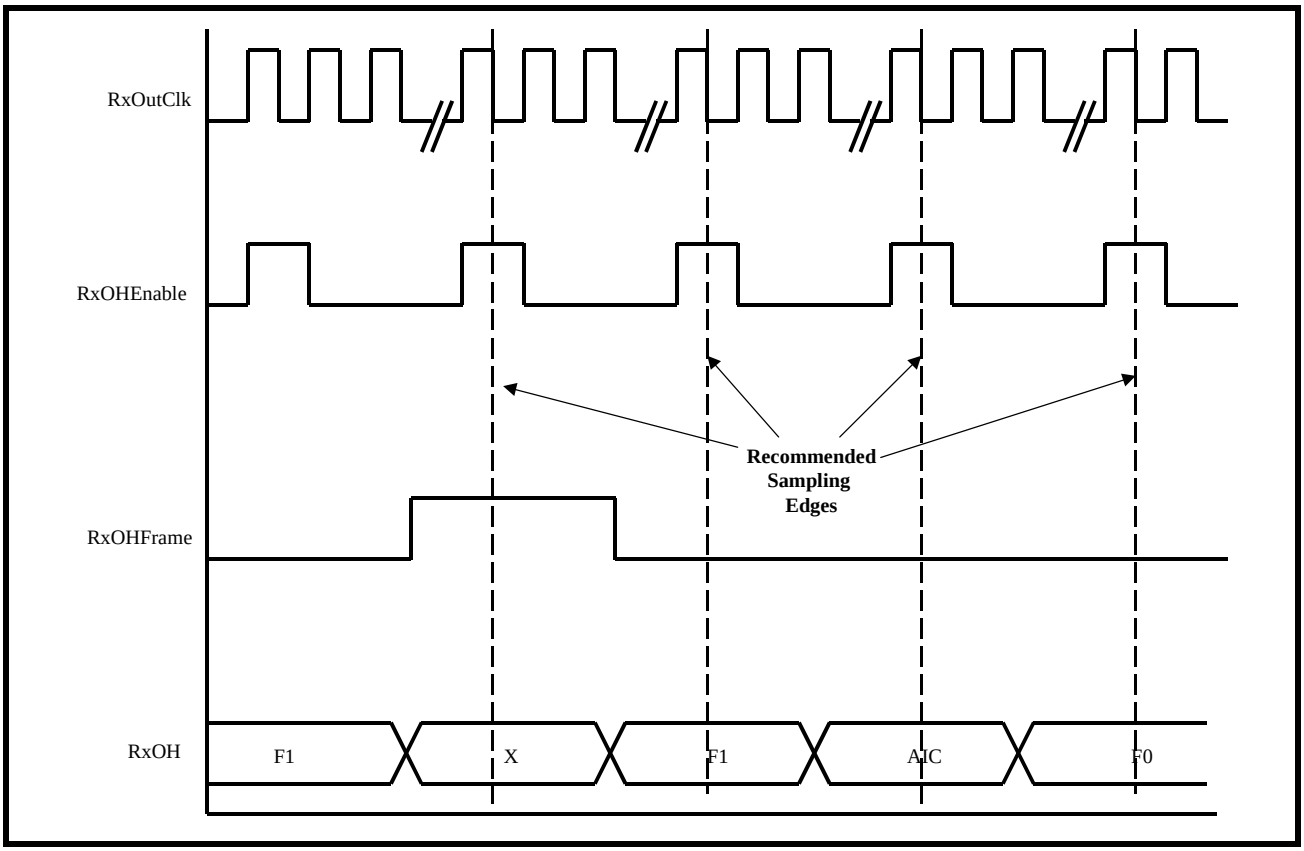
NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (The RxOHEnable and RxOHFrame signals are both sampled "High")	X
1	F1
2	AIC
3	F0
4	NA
5	F0
6	FEAC
7	F1
8	X
9	F1
10	UDL
11	F0
12	UDL
13	F0
14	UDL
15	F1
16	P
17	F1
18	CP
19	F0
20	CP
21	F0
22	CP
23	F1
24	P
25	F1
26	FEBE
27	F0
28	FEBE
29	F0
30	FEBE
31	F1
32	M0
33	F1
34	DL
35	F0
36	DL
37	F0
38	DL

TABLE 60: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES ((SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE DS3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
39	F1
40	M1
41	F1
42	UDL
43	FO
44	UDL
45	FO
46	UDL
47	F1
48	M0
49	F1
50	UDL
51	F0
52	UDL
53	F0
54	UDL
55	F1

Figure 98 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 2 is being used to sample the incoming DS3 overhead bits.

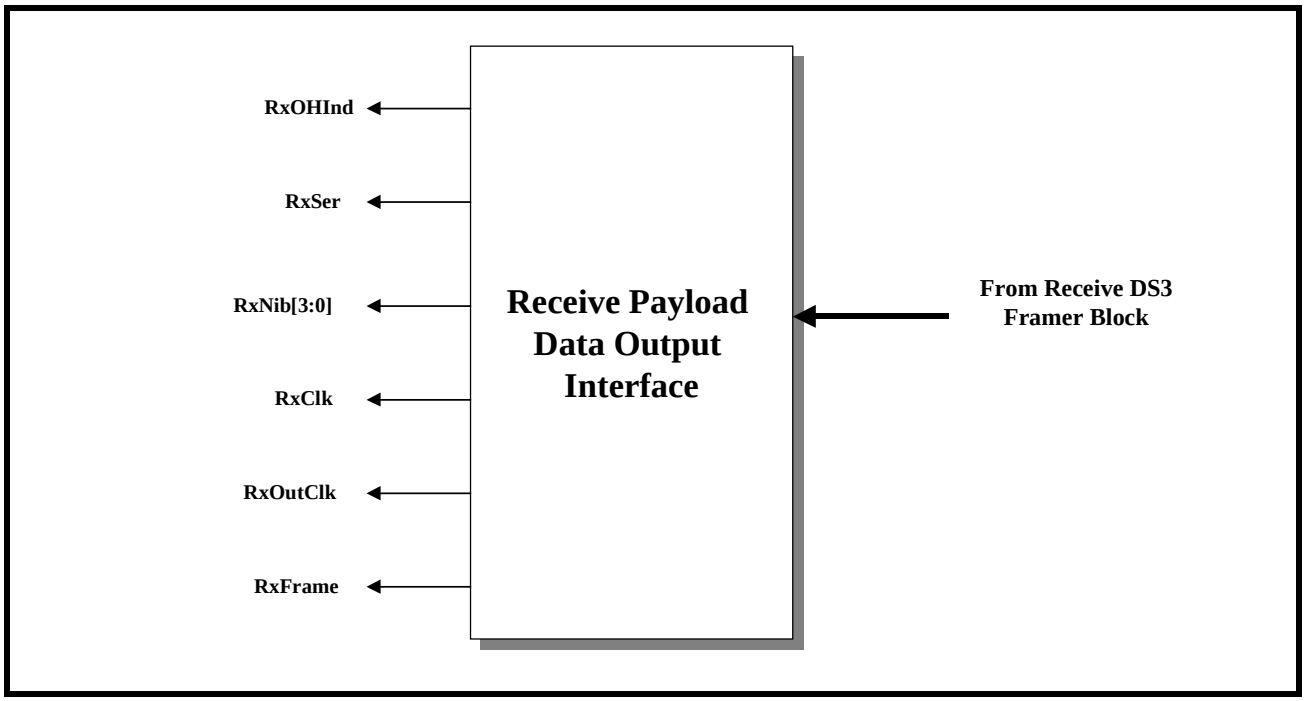
FIGURE 98. ILLUSTRATION OF THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (FOR METHOD 2).



5.3.5 The Receive Payload Data Output Interface

Figure 99 presents a simple illustration of the Receive Payload Data Output Interface block.

FIGURE 99. A SIMPLE ILLUSTRATION OF THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK



Each of the output pins of the Receive Payload Data Output Interface block are listed in Table 61 and described below. The exact role that each of these output pins assume, for a variety of operating scenarios are described throughout this section.

TABLE 61: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	TYPE	DESCRIPTION
RxSer	Output	**Receive Serial Payload Data Output pin:** If the XRT74L74 is operated in the serial mode, then the chip will output the payload data, of the incoming DS3 frames, via this pin. The XRT74L74 will output this data upon the rising edge of RxClk. The user is advised to design the Terminal Equipment such that it will sample this data on the falling edge of RxClk. This signal is only active if the NibInt input pin is pulled "Low".
RxNib[3:0]	Output	**Receive Nibble-Parallel Payload Data Output pins:** If the XRT74L74 is operated in the nibble-parallel mode, then the chip will output the payload data, of the incoming DS3 frames, via these pins. The XRT74L74 will output data via these pins, upon the falling edge of the RxClk output pin. The user is advised to design the Terminal Equipment such that it will sample this data upon the rising edge of RxClk. These pins are only active if the NibInt input pin is pulled "High".
RxClk	Output	**Receive Payload Data Output Clock pin:** The exact behavior of this signal depends upon whether the XRT74L74 is operating in the Serial or in the Nibble-Parallel-Mode. **Serial Mode Operation** In the serial mode, this signal is a 44.736MHz clock output signal. The Receive Payload Data Output Interface will update the data via the RxSer output pin, upon the rising edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxSer pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation** In this Nibble-Parallel Mode, the XRT74L74 will derive this clock signal, from the RxLineClk signal. The XRT74L74 will pulse this clock 1176 times for each inbound DS3 frame. The Receive Payload Data Output Interface will update the data, on the RxNib[3:0] output pins upon the falling edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxNib[3:0] output pins, upon the rising edge of this clock signal
RxOHInd	Output	**Receive Overhead Bit Indicator Output:** This output pin will pulse "High" whenever the Receive Payload Data Output Interface outputs an overhead bit via the RxSer output pin. The purpose of this output pin is to alert the Terminal Equipment that the current bit, (which is now residing on the RxSer output pin), is an overhead bit and should not be processed by the Terminal Equipment. The XRT74L74 will update this signal, upon the rising edge of the RxClk signal. The user is advised to design (or configure) the Terminal Equipment to sample this signal (along with the data on the RxSer output pin) on the falling edge of the RxClk signal. For DS3 applications, this output pin is only active if the XRT74L74 is operating in the Serial Mode. This output pin will be "Low" if the device is operating in the Nibble-Parallel Mode.
RxFrame	Output	**Receive Start of Frame Output Indicator:** The exact behavior of this pin, depends upon whether the XRT74L74 has been configured to operate in the Serial Mode or the Nibble-Parallel Mode. **Serial Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" (for one bit period) when the Receive Payload Data Output Interface block is driving the very first bit of a given DS3 frame, onto the RxSer output pin. **Nibble-Parallel Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" (for one nibble period), when the Receive Payload Data Output Interface is driving the very first nibble of a given DS3 frame, onto the RxNib[3:0] output pins.

Operation of the Receive Payload Data Output Interface block

The Receive Payload Data Output Interface permits the user to read out the payload data of inbound DS3 frames, via either of the following modes.

- Serial Mode
- Nibble-Parallel Mode

Each of these modes are described in detail, below.

5.3.5.1 Serial Mode Operation

Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in the Serial mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data, of the incoming DS3 frames via the RxSer output, upon the rising edge of RxClk.

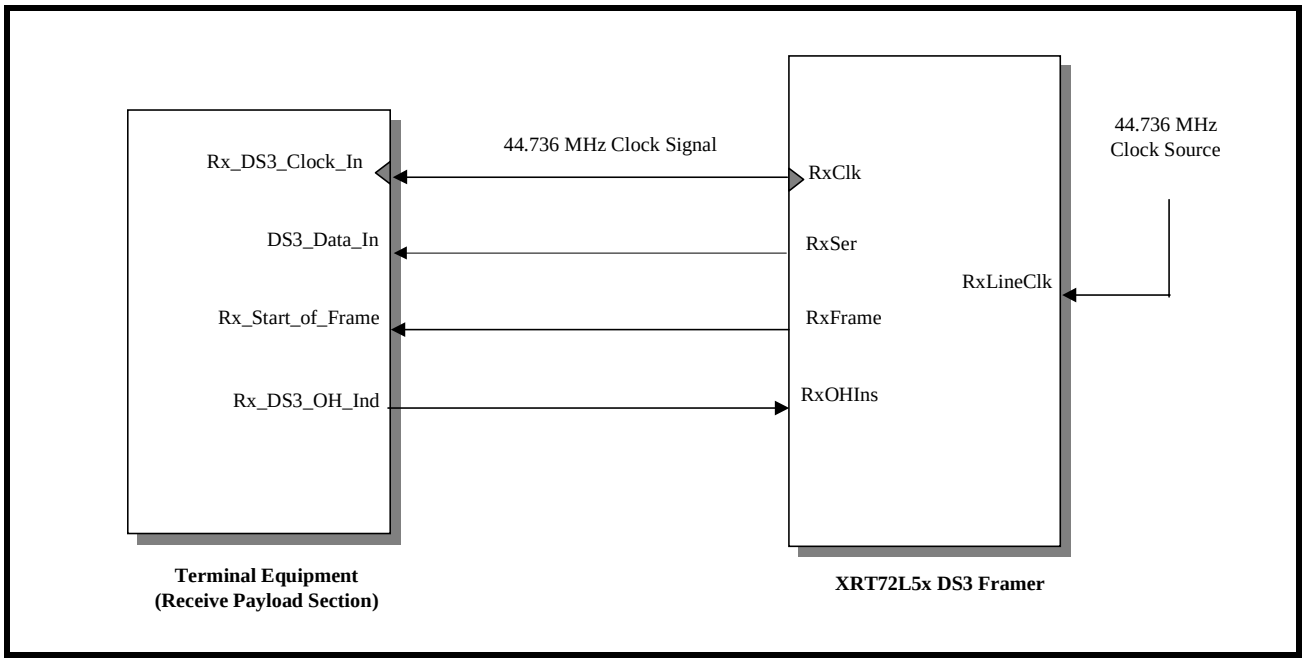
Delineation of inbound DS3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one bit-period, coincident with it driving the first bit within a given DS3 frame, via the RxSer output pin.

Interfacing the XRT74L74 to the Receive Terminal Equipment

Figure 100 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 100. ILLUSTRATION OF THE XRT74L74 DS3/E3 FRAMER IC BEING INTERFACED TO THE RECEIVE TERMINAL EQUIPMENT (SERIAL MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxSer output pin, upon the rising edge of RxClk. However, because the rising edge of RxClk to data delay is between 14ns to 16ns, the Terminal Equipment should sample the data on the RxSer output pin (or the DS3_Data_In pin at the Terminal Equipment) upon the rising edge of RxClk. This will still permit the Terminal Equipment with a RxSer to RxClk set-up time of approximately 6ns and a hold time of 14 to 16ns. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the following signals.

- RxFrame
- RxOHInd

The Need for sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first bit of a given DS3 frame onto the RxSer output pin. If knowledge of the DS3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

The Need for sampling RxOHInd

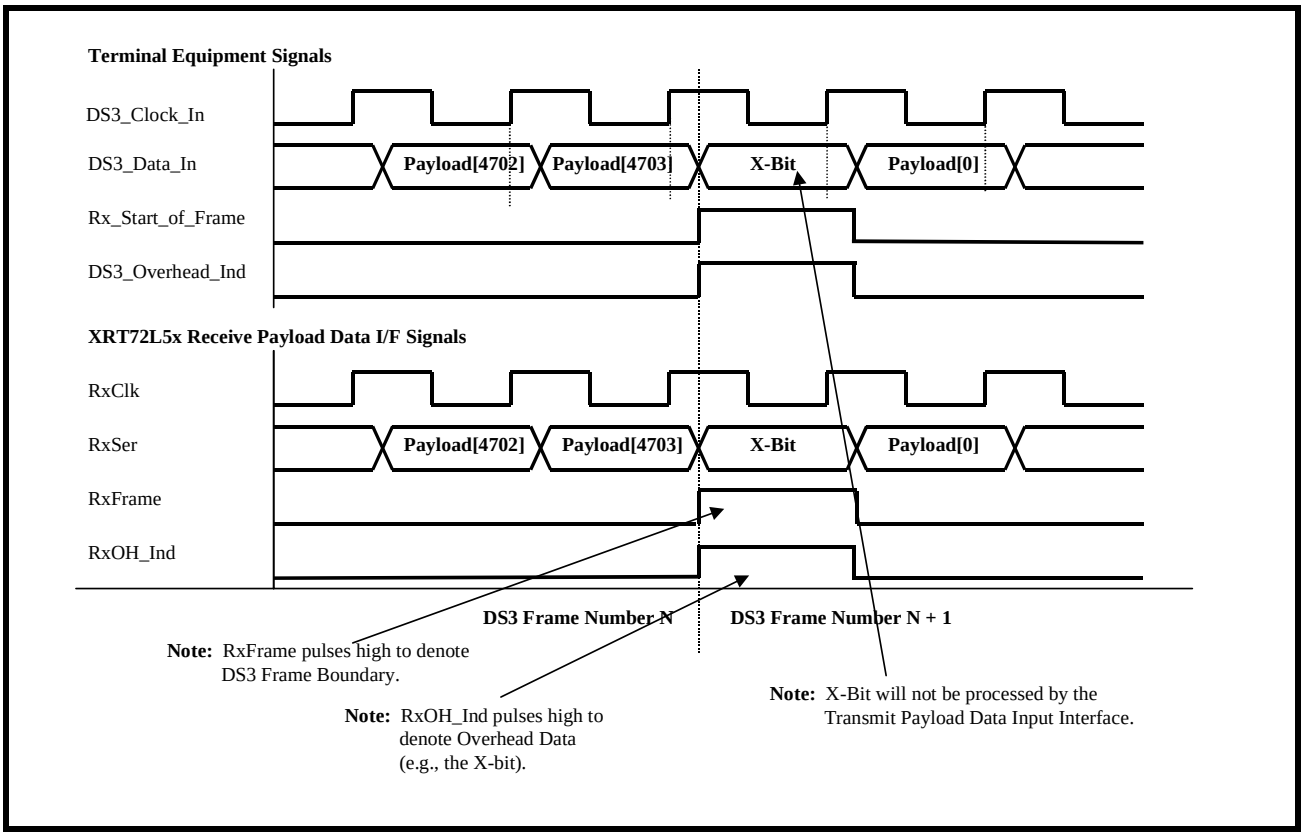
The XRT74L74 will indicate that it is currently driving an overhead bit onto the RxSer output pin, by pulsing the RxOHInd output pin "High". If the Terminal Equipment samples this signal "High", then it should know

that the bit, that it is currently sampling via the RxSer pin is an overhead bit and should not be processed.

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Serial Mode Operation is illustrated in Figure 101.

FIGURE 101. AN ILLUSTRATION OF THE BEHAVIOR OF THE SIGNALS BETWEEN THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT (SERIAL MODE OPERATION)



5.3.5.2 Nibble-Parallel Mode Operation

Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in the Nibble-Parallel Mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data of the incoming DS3 frames, via the RxNib[3:0] output pins, upon the falling edge of RxClk.

NOTES:

1. *In this case, RxClk will function as the Nibble Clock signal between the XRT74L74 the Terminal Equipment. The XRT74L74 will pulse the RxClk output signal "High" 1176 times, for each inbound DS3 frame.*
2. *Unlike Serial Mode operation, the duty cycle of RxClk, in Nibble-Parallel Mode operation is approximately 25%.*

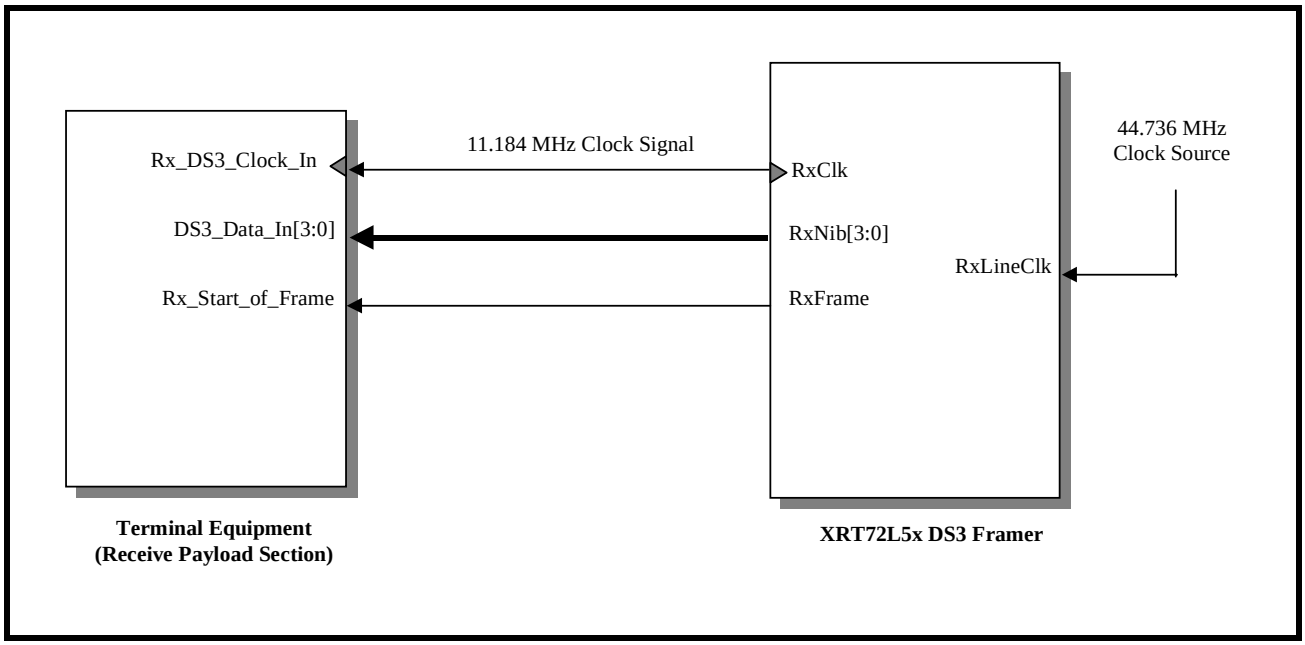
Delineation of Inbound DS3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one nibble-period coincident with it driving the very first nibble, within a given inbound DS3 frame, via the RxNib[3:0] output pins.

Interfacing the XRT74L74 the Terminal Equipment.

Figure 102 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 102. ILLUSTRATION OF THE XRT74L74 DS3/E3 FRAMER IC BEING INTERFACED TO THE RECEIVE SECTION OF THE TERMINAL EQUIPMENT (NIBBLE-MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxNib[3:0] line, upon the falling edge of RxClk. Hence, the Terminal Equipment should sample the data on the RxNib[3:0] output pins (or the DS3_Data_In[3:0] input pins at the Terminal Equipment) upon the rising edge of RxClk. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the RxFrame signal.

The Need for Sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first nibble of a given DS3 frame, onto the RxNib[3:0] output pins.

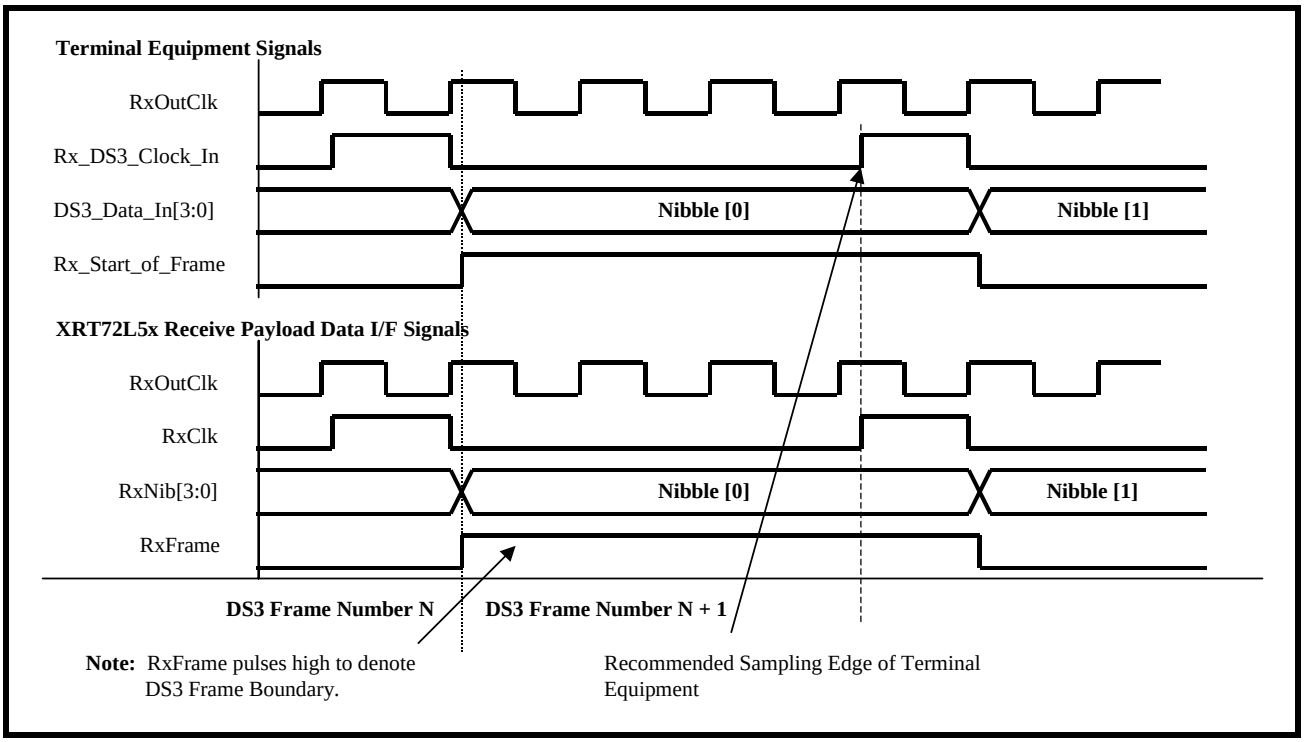
If knowledge of the DS3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

NOTE: *For DS3/Nibble-Parallel Mode Operation, none of the Overhead bits will be output via the RxNib[3:0] output pins. Hence, the RxOH_Ind output pin will be in-active in this mode.*

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for DS3 Nibble-Mode operation is illustrated in Figure 103.

FIGURE 103. AN ILLUSTRATION OF THE BEHAVIOR OF THE SIGNALS BETWEEN THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT (NIBBLE-MODE OPERATION).



5.3.6 Receive Section Interrupt Processing

The Receive Section of the XRT74L74 can generate an interrupt to the Microcontroller/Microprocessor for the following reasons.

- Change of State of Receive LOS (Loss of Signal) condition
- Change of State of Receive OOF (Out of Frame) condition
- Change of State of Receive AIS (Alarm Indicator Signal) condition
- Change of State of Receive Idle Condition.
- Change of State of Receive FERF (Far-End Receive Failure) condition.
- Change of State of AIC (Application Identification Channel) bit.
- Detection of P-Bit Error in a DS3 frame
- Detection of CP-Bit Error in a DS3 frame
- The Receive FEAC Message - Validation Interrupt
- The Receive FEAC Message - Removal Interrupt
- Completion of Reception of a LAPD Message

5.3.6.1 Enabling Receive Section Interrupts

The Interrupt Structure, within the XRT74L74 contains two hierarchical levels.

- Block Level
- Source Level

The Block Level

The Enable state of the Block level for the Receive Section Interrupts dictates whether or not interrupts (if enabled at the source level), are actually enabled.

These Receive Section interrupts can be enabled or disabled at the Block Level, by writing the appropriate data into Bit 7 (Rx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
X	0	0	0	0	0	0	0

Setting this bit-field to “1” enables the Receive Section (at the Block Level) for interrupt generation. Conversely, setting this bit-field to “0” disables the Receive Section for interrupt generation.

5.3.6.2 Enabling/Disabling and Servicing Receive Section Interrupts

The Receive Section of the XRT74L74 Framer IC contains numerous interrupts. The Enabling/Disabling and Servicing of each of these interrupts is described below.

5.3.6.2.1 The Change of State on Receive LOS Interrupt

If the Change of State on Receive LOS (Loss of Signal) Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an LOS (Loss of Signal) condition, and
2. When the XRT74L74 Framer IC clears the LOS (Loss of Signal) condition.

Conditions causing the XRT74L74 Framer IC to declare an LOS condition

- If the XRT7300 LIU IC declares an LOS condition, and drives the RLOS input pin (of the XRT74L74 Framer IC) "High".
- If the XRT74L74 Framer IC detects a 180 consecutive “0’s”, via the RxPOS and RxNEG input pins.

Conditions causing the XRT74L74 Framer IC to clear the LOS condition.

- When the XRT7300 LIU IC ceases declaring an LOS condition and drives the RLOS input pin (of the XRT74L74 Framer IC) "Low".
- When the XRT74L74 Framer IC detects at least 60 marks (via the RxPOS and RxNEG input pins) out of 180 bit-periods.

Enabling and Disabling the Change of State on Receive LOS Interrupt:

The Change of State on Receive LOS Interrupt can be enabled or disabled by writing the appropriate value into Bit 6 (LOS Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive LOS Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving this pin "Low".

- It will set Bit 6 (LOS Interrupt Status) within the RxDS3 Interrupt Status register to “1”, as illustrated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

Whenever the user’s system encounters the Change of LOS on Receive Interrupt, then it should do the following.

1. It should determine the current state of the LOS condition. Recall, that this interrupt can generated, whenever the XRT74L74 Framer declares or clears the LOS defects. Hence, the current state of the LOS defect can be determined by reading the state of Bit 6 (RxLOS), within the RxDS3 Configuration & Status Registers, as illustrated below.

RXDS3 CONFIGURATION & STATUS REGISTER (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Reserved	Framing On Parity	FSync Algo	MSync Algo
RO	RO	RO	RO	RO	R/W	R/W	R/W
0	1	0	0	0	0	0	0

If the LOS State is TRUE

1. It should transmit a FERF (Far-End Receive Failure) to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-LOS feature.
2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE), to the Remote Terminal, indicating that a Loss of Signal condition has been declared.

If the LOS State is FALSE

1. It should cease transmitting a FERF indicator to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-LOS feature.
2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE), to the Remote Terminal Equipment, indicating that the Loss of Signal condition has been cleared.

5.3.6.2.2 The Change of State on Receive OOF Interrupt

If the Change of State on Receive OOF (Out-of-Frame) Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an OOF (Out of Frame) condition, and
2. When the XRT74L74 Framer IC clears the OOF (Out of Frame) condition.

Conditions causing the XRT74L74 Framer IC to declare an OOF condition

- If the Receive DS3 Framer block (within the XRT74L74 Framer IC) detects at least either 3 or 6 F-bit errors, in the last 16 F-bits.

Conditions causing the XRT74L74 Framer IC to clear the OOF condition.

- Whenever, the Receive DS3 Framer block transitions from the M-Bit Search into the In-Frame state (within the Frame Acquisition/Maintenance State Machine Diagram).

Enabling and Disabling the Change of State on Receive OOF Interrupt:

The Change of State on Receive OOF Interrupt can be enabled or disabled by writing the appropriate val-

ue into Bit 1 (OOF Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive OOF Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving this pin "Low".
- It will set Bit 1 (OOF Interrupt Status), within the RxDS3 Interrupt Status Register to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

Whenever the Terminal Equipment encounters a Change in OOF on Receive Interrupt, then it should do the following.

1. It should determine the current state of the OOF condition. Recall, that this interrupt can generated, whenever the XRT74L74 Framer declares or clears the OOF defects. Hence, the current state of the OOF defect can be determined by reading the state of Bit 4 (RxOOF), within the RxDS3 Configuration & Status Registers, as illustrated below.

RXDS3 CONFIGURATION & STATUS REGISTER (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Reserved	Framing On Parity	FSync Algo	MSync Algo
RO	RO	RO	RO	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

If OOF is TRUE.

1. It should transmit a FERF (Far-End Receive Failure) to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-OOF feature.
2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE), to the Remote Terminal, indicating that a Service Affecting condition has been detected in the Local Terminal Equipment.

if OOF is FALSE

1. It should cease transmitting a FERF (Far-End Receive Failure) indicator to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-OOF feature.

2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE), to the Remote Terminal Equipment, indicating that the Service Affecting condition has been cleared.

5.3.6.2.3 The Change of State of Receive AIS Interrupt

If the Change of State on Receive AIS (Alarm Indication Signal) Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC detects an AIS pattern, in the incoming DS3 data stream, and
2. When the XRT74L74 Framer IC no longer detects the AIS pattern in the incoming DS3 data stream.

Conditions causing the XRT74L74 Framer IC to declare an AIS condition

- If the Receive DS3 Framer block (within the XRT74L74 Framer IC) detects at least 63 DS3 frames, which contains the AIS pattern.

Conditions causing the XRT74L74 Framer IC to clear the AIS condition.

- Whenever, the Receive DS3 Framer block detects 63 DS3 frames, which do not contain the AIS pattern.

Enabling and Disabling the Change of State on Receive AIS Interrupt:

The Change of State on Receive AIS Interrupt can be enabled or disabled by writing the appropriate value into Bit 5 (AIS Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive AIS Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 5 (AIS Interrupt Status) within the RxDS3 Interrupt Status Register, to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

Whenever the Terminal Equipment encounters a Change in AIS on Receive interrupt, it should do the following.

1. It should determine the current state of the AIS condition. Recall, that this interrupt can generated, whenever the XRT74L74 Framer declares or clears the AIS defects. Hence, the current state of the AIS defect can be determined by reading the state of Bit 7 (RxAIS), within the RxDS3 Configuration & Status Registers, as illustrated below

RXDS3 CONFIGURATION & STATUS REGISTER (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Reserved	Framing On Parity	FSync Algo	MSync Algo
RO	RO	RO	RO	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

If the AIS Condition is TRUE

1. The Local Terminal Equipment should transmit a FERF (Far-End Receive Failure) to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-AIS feature.
2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE), to the Remote Terminal, indicating that a Service Affecting condition has been detected in the Local Terminal Equipment.

If the AIS Condition is FALSE

1. The Local Terminal Equipment should cease transmitting a FERF (Far-End Receive Failure) indicator to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-AIS feature.
2. It should transmit the appropriate FEAC Message (per Bellcore GR-499-CORE) to the Remote Terminal, indicates that the Service Affecting condition no longer exists.

5.3.6.2.4 The Change of State of Receive Idle Interrupt

If the Change of State on Receive Idle Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC detects an Idle pattern, in the incoming DS3 data stream, and
2. When the XRT74L74 Framer IC no longer detects the Idle pattern in the incoming DS3 data stream.

Conditions causing the XRT74L74 Framer IC to declare an Idle condition

- If the Receive DS3 Framer block (within the XRT74L74 Framer IC) detects at least 63 DS3 frames, which contains the Idle pattern.

Conditions causing the XRT74L74 Framer IC to clear the Idle condition.

- Whenever, the Receive DS3 Framer block detects 63 DS3 frames, which do not contain the Idle pattern.

Enabling and Disabling the Change of State on Receive Idle Interrupt:

To enable or disable the Change of State on Receive Idle Interrupt, write the appropriate value into Bit 4 (Idle Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive Idle Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request Output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 4 (Idle Interrupt Status), within the Rx DS3 Interrupt Status Register to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

Whenever the Terminal Equipment encounters the Change in Idle Condition Receive Interrupt, it should do the following.

1. It should determine the current state of the Idle condition. Recall, that this interrupt can generated, whenever the XRT74L74 Framer declares or clears the Idle condition. Hence, the current state of the Idle condition can be determined by reading the state of Bit 5 (RxIdle), within the RxDS3 Configuration & Status Registers, as illustrated below

RXDS3 CONFIGURATION & STATUS REGISTER (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS	RxLOS	RxIdle	RxOOF	Reserved	Framing On Parity	FSync Algo	MSync Algo
RO	RO	RO	RO	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

5.3.6.2.5 The Change of State of Receive FERF Interrupt

If the Change of State on Receive FERF Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC detects the FERF indicator, in the incoming DS3 data stream, and
2. When the XRT74L74 Framer IC no longer detects the FERF indicator, in the incoming DS3 data stream.

Conditions causing the XRT74L74 Framer IC to declare an FERF (Far-End-Receive Failure) condition

- If the Receive DS3 Framer block (within the XRT74L74 Framer IC) detects some incoming DS3 frames with both of the “X” bits set to “0”.

Conditions causing the XRT74L74 Framer IC to clear the FERF condition.

- Whenever, the Receive DS3 Framer block starts to detect some incoming DS3 frames, in which the “X” bits are not set to “0”.

Enabling and Disabling the Change of State on Receive FERF Interrupt:

To enable or disable the Change of State on Receive FERF Interrupt, write the appropriate value into Bit 3 (FERF Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive FERF Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request Output pin ($\overline{INT}$) by driving it "High".
- It will set Bit 3 (FERF Interrupt Status), within the Rx DS3 Interrupt Status Register, to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Whenever the Terminal Equipment encounters a Change in FERF Condition on Receive Interrupt, it should do the following.

1. It should determine the current state of the FERF condition. Recall, that this interrupt can generated, whenever the XRT74L74 Framer declares or clears the FERF condition. Hence, to determine the current state of the FERF condition read the state of Bit 4 (RxFERF), within the RxDS3 Status Registers, as illustrated below

RXDS3 STATUS REGISTER (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved			RxFERF	RxAIC	RxFEBE[2:0]		
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

5.3.6.2.6 The Change of State of Receive AIC Interrupt

If the Change of State of Receive AIC Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive DS3 Framer block has detected a change in the value of the AIC bit, within the incoming DS3 data stream.

Enabling and Disabling the Change of State of Receive AIC Interrupt:

To enable or disable the Change of State on Receive AIC Interrupt, write the appropriate value into Bit 2 (AIC Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change of State on Receive AIC Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request Output pin ($\overline{INT}$) by driving it "High".

- It will set Bit 2 (AIC Interrupt Status), within the Rx DS3 Interrupt Status Register, to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

Whenever the Terminal Equipment encounters this interrupt, it should do the following.

- It should continue to check the state of the AIC bit, in order to see if this change is constant.
- If this change is constant, then the user should configure the XRT74L74 Framer IC to operate in the M13 framing format, if the AIC bit-field is “0”.
- Conversely, if the AIC bit-field is “1”, then the user should configure the XRT74L74 Framer IC to operate in the C-bit Parity framing format.

5.3.6.2.7 The Detection of P-Bit Error Interrupt

If the Detection of P-Bit Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive DS3 Framer block has detected a P-bit error, within the incoming DS3 data stream.

Enabling and Disabling the Detection of P-Bit Error Interrupt:

The Detection of P-Bit Error Interrupt can be enabled or disabled by writing the appropriate value into Bit 0 (P-Bit Error Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Detection of P-Bit Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "High".
- It will set Bit 0 (P-Bit Error Interrupt Status) within the Rx DS3 Interrupt Status Register, to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

Whenever the Terminal Equipment encounters the Detection of P-bit Error Interrupt, It should read the contents of PMON Parity Error Count Register (located at 0x54 and 0x55), in order to determine the number of P-bit errors recently received.

5.3.6.2.8 The Detection of CP-Bit Error Interrupt

If the Detection of CP-Bit Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive DS3 Framer block has detected a CP-bit error, within the incoming DS3 data stream.

Enabling and Disabling the Detection of CP-Bit Error Interrupt:

To enable or disable the Detection of CP-Bit Error Interrupt, write the appropriate value into Bit 7 (CP-Bit Error Interrupt Enable) within the RxDS3 Interrupt Enable Register, as illustrated below.

RXDS3 INTERRUPT ENABLE REGISTER (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP Bit Error Interrupt Enable	LOS Interrupt Enable	AIS Interrupt Enable	Idle Interrupt Enable	FERF Interrupt Enable	AIC Interrupt Enable	OOF Interrupt Enable	P-Bit Error Interrupt Enable
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Detection of CP-Bit Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "High".
- It will set Bit 7 (CP-Bit Error Interrupt Status) within the Rx DS3 Interrupt Status Register, to “1”, as indicated below.

RXDS3 INTERRUPT STATUS REGISTER (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CP-Bit Error Interrupt Status	LOS Interrupt Status	AIS Interrupt Status	Idle Interrupt Status	FERF Interrupt Status	AIC Interrupt Status	OOF Interrupt Status	P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	1

Whenever the Terminal Equipment encounters the Detection of CP-bit Error Interrupt, it should do the following.

- It should read contents of PMON Frame CP-Bit Error Count Register (located at 0x72 and 0x73), in order to determine the number of CP-bit errors recently received.

5.3.6.2.9 The Receive FEAC Message - Validation Interrupt

If the Receive FEAC Message - Validation Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt any time the Receive FEAC Processor validates a new FEAC (Far-End Alarm & Control) Message.

In particular, the Receive FEAC Processor will validate a FEAC Message, it that same FEAC Message has been received in 8 of the last 10 FEAC Message receptions.

Enabling/Disabling the Receive FEAC Message - Validation Interrupt

To enable or disable the Receive FEAC Message - Validation Interrupt, write the appropriate data into Bit 1 (RxFEAC Valid Interrupt Enable) within the RxDS3 FEAC Interrupt Enable/Status Register, as indicated below.

RXDS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Receive FEAC Message - Validation Interrupt.

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 0 (RxFEAC Valid Interrupt Status), within the RxDS3 FEAC Interrupt Enable/Status Register to “1”, as indicated below.

RXDS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
0	0	0	0	0	0	1	1

- It will write the contents of this validated FEAC Message into the Rx DS3 FEAC Register, as indicated below.

RXDS3 FEAC REGISTER (ADDRESS = 0X16)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxFEAC[5:0]						Not Used
RO	RO	RO	RO	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Whenever the Terminal Equipment encounters the Receive FEAC Message - Validation Interrupt, then it should do the following.

- It should read the contents of the High RxDS3 FEAC Register, and respond accordingly.

5.3.6.2.10 The Receive FEAC Message - Removal Interrupt

if the Receive FEAC Message - Removal Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt any time the High Receive FEAC Processor removes a new FEAC (Far-End Alarm & Control) Message.

In particular, the Receive FEAC Processor will remove a FEAC Message, it has received a different FEAC Message (from the most recently validated message) in 3 of the last 10 FEAC Message receptions.

Enabling/Disabling the Receive FEAC Message - Removal Interrupt

To enable or disable the Receive FEAC Message - Removal Interrupt, write the appropriate data into Bit 3 (RxFEAC Remove Interrupt Enable) within the RxDS3 FEAC Interrupt Enable/Status Register, as indicated below.

RXDS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
0	0	0	0	X	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Receive FEAC Message - Validation Interrupt.

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 2 (RxFEAC Remove Interrupt Status), within the RxDS3 FEAC Interrupt Enable/Status Register to “1”, as indicated below.

RXDS3 FEAC INTERRUPT ENABLE/STATUS REGISTER (ADDRESS = 0X17)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
RO	RO	RO	RO	R/W	RUR	R/W	RUR
0	0	0	0	0	0	1	1

- It will write the delete contents of the most recently validated FEAC Message from the Rx DS3 FEAC Register, as indicated below.

RXDS3 FEAC REGISTER (ADDRESS = 0X16)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxFEAC[5:0]						Not Used
RO	RO	RO	RO	R/O	R/O	R/O	R/O
0	X	X	X	X	X	X	0

5.3.6.2.11 The Completion of Reception of a LAPD Message Interrupt

If the Completion of Reception of a LAPD Message interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt anytime the Receive HDLC Controller block has received a new LAPD Message buffer, from the Remote Terminal Equipment, and has stored the contents of this message in the Receive LAPD Message Buffer.

Enabling/Disable the Receive LAPD Message Interrupt

To enable or disable the Receive LAPD Message Interrupt, write the appropriate data into Bit 1 (RxLAPD

Interrupt Enable) within the RxDS3 LAPD Control Register, as indicated below.

RXDS3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	X	0

Writing a “1” into this bit-field enables the Receive LAPD Message Interrupt. Conversely, writing a “0” into this bit-field disables the Receive LAPD Message interrupt.

Servicing the Receive LAPD Message Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 0 (RxLAPD Interrupt Status), within the Rx DS3 LAPD Control Register to “1”, as indicated below.

RXDS3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	1	1

- It will write the contents of this newly Received LAPD Message into the Receive LAPD Message buffer (located at 0xDE through 0x135).

Whenever the Terminal Equipment encounters the Receive LAPD Interrupt, then it should read out the contents of the Receive LAPD Message buffer, and respond accordingly.

6.0 E3/ITU-T G.751 OPERATION OF THE XRT74L74

Configuring the XRT74L74 to Operate in the E3, ITU-T G.751 Mode

The XRT74L74 can be configured to operate in the E3/ITU-T G.751 Mode by writing a "0" into bit-field 6 and a "0" into bit-field 2, within the Framer Operating Mode register, as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	0	x	0	x	0	x	x

Prior to describing the functional blocks within the Transmit and Receive Sections of the XRT74L74, it is important to describe the E3, ITU-T G.751 framing format.

6.1 DESCRIPTION OF THE E3, ITU-T G.751 FRAMES AND ASSOCIATED OVERHEAD BITS

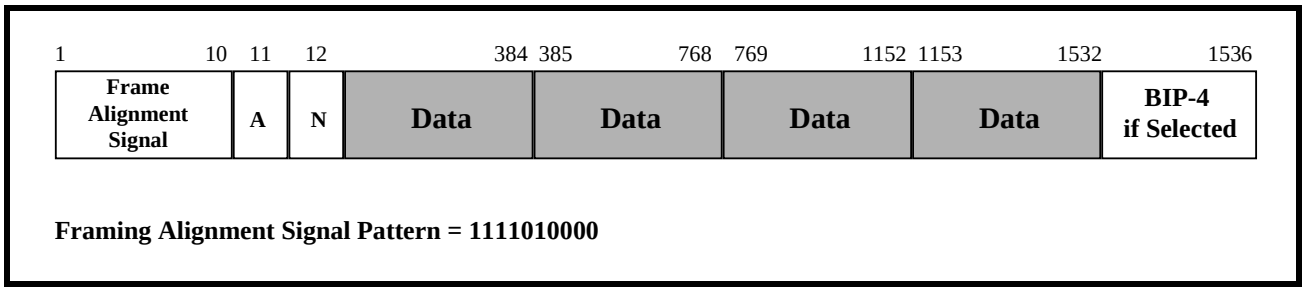
The role of the various overhead bytes are best described by discussing the E3, ITU-T G.751 Frame Format as a whole. The E3, ITU-T G.751 Frame contains 1536 bits, of which 12 bits are overhead and the remaining 1524 bits are payload bits.

Each E3, ITU-T G.751 Frame consists of the following 12 overhead bits.

- A 10 bit FAS (Framing Alignment Signal) pattern. This pattern is assigned the constant pattern of "1111010000", and is used by the Receive E3 Framer block to acquire and maintain Frame Synchronization with the incoming E3 frames.
- The "A" (or Alarm) Bit.
- The "N" (or National) Bit.
- The BIP-4 Bits (if configured).

The frame repetition rate for this type of E3 frame is 22375 times per second, thereby resulting in the standard E3 bit rate of 34.368 Mbps. Figure 104 presents an illustration of the E3, ITU-T G.751 Frame Format.

FIGURE 104. THE E3, ITU-T G.751 FRAMING FORMAT.



6.1.1 Definition of the Overhead Bits

Each of these Overhead Bits are further defined below.Frame Alignment Signaling (FAS) Pattern Bits

The first 10 bits, within each E3, ITU-T G.751 frame are known as the FAS (or Framing Alignment Signaling) bits. The Receive E3 Framer block, while trying to acquire or maintain framing synchronization with its incoming E3 frames, will attempt to locate the FAS bits. The FAS pattern is assigned the value "1111010000".

6.1.1.1 The "A" (Alarm) Bit

The "A" bit typically functions as a FERF (Far-End Receive Failure) indicator bit. However, if the user configures the XRT74L74 Framer IC to transmit and receive E3 frames which are carrying the BIP-4 value (located at the end of a given E3 frame), then this bit will also function as the FEBE indicator bit. A detailed discussion on the practical use of the "A" is presented in Section 4.2.2. Each of these roles of the "A" bit are briefly discussed below.

The "A" Bit Functioning as the FERF bit-field

If the Receive E3 Framer block (at a Local Terminal) is experiencing problems receiving E3 frame data from a Remote Terminal (e.g., an LOS, OOF or AIS

condition), then it will inform the Remote Terminal Equipment of this fact by commanding the Local Transmit E3 Framer block to set the “A” bit-field, within the next outbound E3 frame, to “1”. The Local Transmit E3 Framer block will continue to set the “A” bit-field (within the subsequent outbound E3 frames) to “1” until the Receive E3 Framer block no longer experiences problems in receiving the E3 frame data. If the Remote Terminal Equipment receives a certain number of consecutive E3 frames, with the “A” bit-field set to “1”, then the Remote Terminal Equipment will interpret this signaling as an indication of a Far-End Receive Failure (e.g., a problem with the Local Terminal Equipment).

Conversely, if the Receive E3 Framer block (at a Local Terminal Equipment) is not experiencing any problems receiving E3 frame data from a Remote Terminal Equipment, then it will also inform the Remote Terminal Equipment of this fact by commanding the Local Transmit E3 Framer block to set the “A” bit-field within an outbound E3 frame (which is destined for the Remote Terminal) to “0”. The Remote Terminal Equipment will interpret this form of signaling as an indication of a normal operation.

A detailed discussion into the practical use of the A bit-field is presented in Section 4.2.2.

6.1.1.2 The “N” Bit

The “N” bit is typically used to transport PMDL (Path Maintenance Data Link) information, from one terminal to the next. However, the “N” bit-field can also be used to transport a proprietary data link, if configured according.

A detailed discussion into the practical use of the N-bit field is presented in Section 4.2.2.

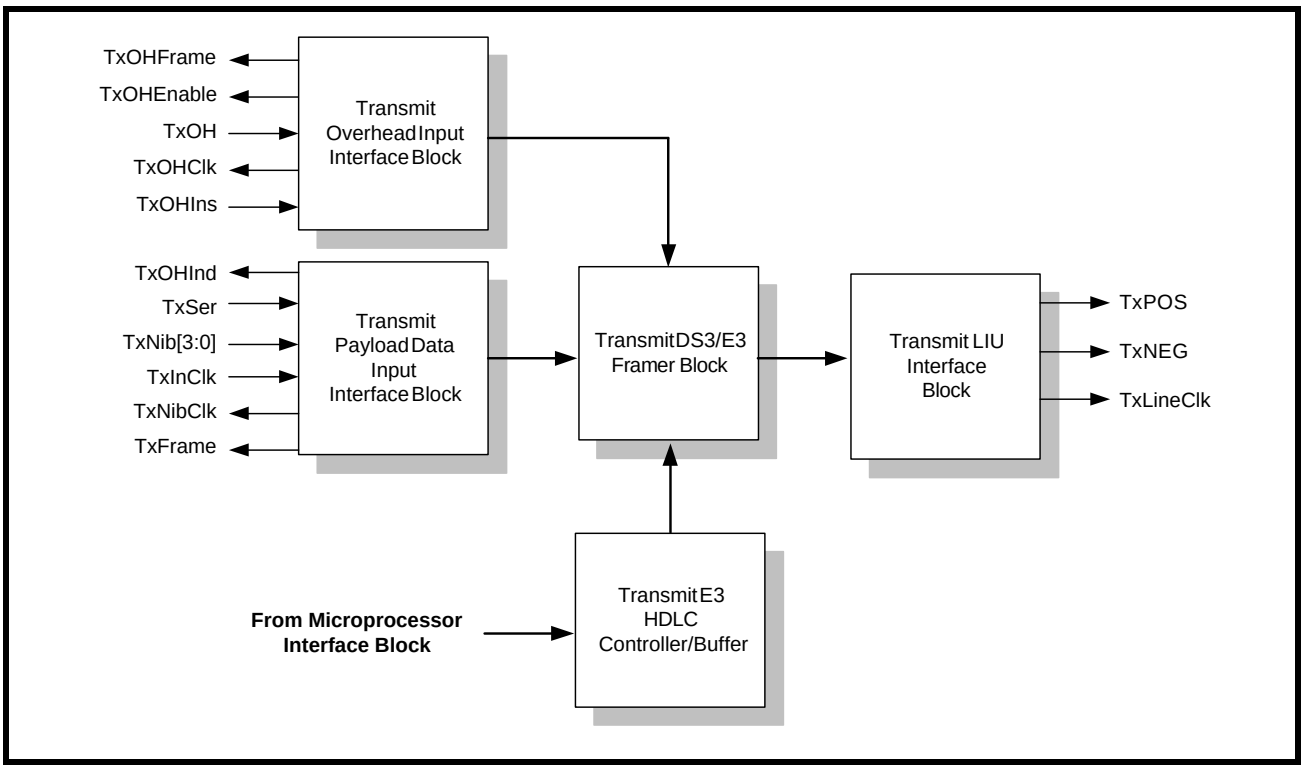
6.2 The Transmit Section of the XRT74L74 (E3, ITU-T G.751 Mode Operation)

When the XRT74L74 has been configured to operate in the E3, ITU-T G.751 Mode, the Transmit Section of the XRT74L74 consists of the following functional blocks.

- Transmit Payload Data Input Interface block
- Transmit Overhead Data Input Interface block
- Transmit E3 Framer block
- Transmit HDLC Controller block
- Transmit LIU Interface block

Figure 105 presents a simple illustration of the Transmit Section of the XRT74L74 Framer IC.

Figure 105. The XRT74L74 Transmit Section when it has been configured to operate in the E3 Mode

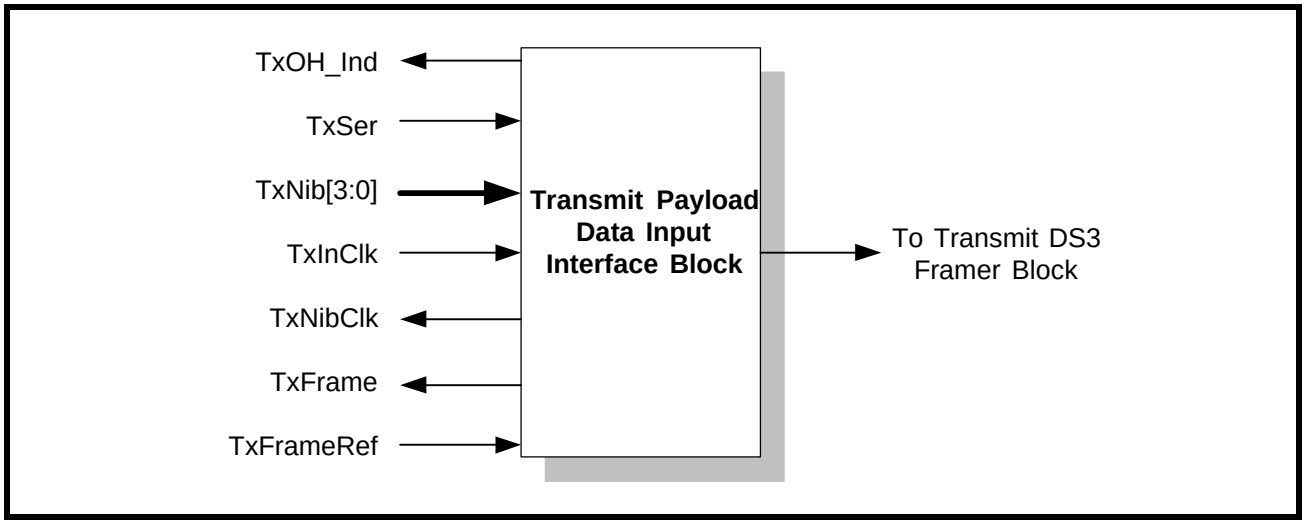


Each of these functional blocks will be discussed in detail in this document.

6.2.1 The Transmit Payload Data Input Interface Block

Figure 106 presents a simple illustration of the Transmit Payload Data Input Interface block.

FIGURE 106. THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK



Each of the input and output pins of the Transmit Payload Data Input Interface are listed in Table 62 and described below. The exact role that each of these inputs and output pins assume, for a variety of operating scenarios are described throughout this section.

TABLE 62: LISTING AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE

SIGNAL NAME	TYPE	DESCRIPTION
TxSer	Input	**Transmit Serial Payload Data Input Pin:** If the user opts to operate the XRT74L74 in the serial mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the outbound E3 data stream) to this input pin. The XRT74L74 will sample the data that is at this input pin upon the rising edge either the RxOutClk or the TxInClk signal (whichever is appropriate). ***NOTE:*** *This signal is only active if the NibInt input pin is pulled "Low".*
TxNib[3:0]	Input	**Transmit Nibble-Parallel Payload Data Input pins:** If the user opts to operate the XRT74L74 in the Nibble-Parallel mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the outbound E3 data stream) to these input pins. The XRT74L74 will sample the data that is at these input pins upon the rising edge of the TxNibClk signal. ***NOTE:*** *These pins are only active if the NibInt input pin is pulled "High".*
TxInClk	Input	**Transmit Section Timing Reference Clock Input pin:** The Transmit Section of the XRT74L74 can be configured to use this clock signal as the Timing Reference. If the user has made this configuration selection, then the XRT74L74 will use this clock signal to sample the data on the TxSer input pin. ***NOTE:*** *If this configuration is selected, then a 34.368 MHz clock signal must be applied to this input pin.*
TxNibClk	Output	**Transmit Nibble Mode Output** If the user opts to operate the XRT74L74 in the Nibble-Parallel mode, then the XRT74L74 will derive this clock signal from the selected Timing Reference for the Transmit Section of the chip (e.g., either the TxInClk or the RxLineClk signals). The XRT74L74 will use this signal to sample the data on the TxNib[3:0] input pins.
TxOHInd	Output	**Transmit Overhead Bit Indicator Output:** This output pin will pulse "High" one-bit period prior to the time that the Transmit Section of the XRT74L74 will be processing an Overhead bit. The purpose of this output pin is to warn the Terminal Equipment that, during the very next bit-period, the XRT74L74 is going to be processing an Overhead bit and will be ignoring any data that is applied to the TxSer input pin.
TxFrame	Output	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT74L74 will pulse this output pin "High" (for one bit-period), when the Transmit Payload Data Input Interface is processing the last bit of a given E3 frame. The purpose of this output pin is to alert the Terminal Equipment that it needs to begin transmission of a new E3 frame to the XRT74L74 (e.g., to permit the XRT74L74 to maintain Transmit E3 framing alignment control over the Terminal Equipment).
TxFrameRef	Input	**Transmit Frame Reference Input:** The XRT74L74 permits the user to configure the Transmit Section to use this input pin as a frame reference. If the user makes this configuration selection, then the Transmit Section will initiate its transmission of a new E3 frame, upon the rising edge of this signal. The purpose of this input pin is to permit the Terminal Equipment to maintain Transmit E3 Framing alignment control over the XRT74L74.
RxOutClk	Output	**Loop-Timed Timing Reference Clock Output pin:** The Transmit Section of the XRT74L74 can be configured to use the RxLineClk signal as the Timing Reference (e.g., loop-timing). If the user has made this configuration selection, then the XRT74L74 will: • Output a 34.368 MHz clock signal via this pin, to the Terminal Equipment. • Sample the data on the TxSer input pin, upon the rising edge of this clock signal.

Operation of the Transmit Payload Data Input Interface

The Transmit Terminal Input Interface is extremely flexible, in that it permits the user to make the following configuration options.

- The Serial or the Nibble-Parallel Interface Mode
- The Loop-Timing or the TxInClk (Local Timing) Mode

Further, if the XRT74L74 has been configured to operate in the Local-Timing mode, then the user has two additional options.

- The XRT74L74 is the Frame Master (e.g., it dictates when the Terminal Equipment will initiate the transmission of data within a new E3 frame).
- The XRT74L74 is the Frame Slave (e.g., the Terminal Equipment will dictate when the XRT74L74 initiates the transmission of a new E3 frame).

Given these three set of options, the Transmit Terminal Input Interface can be configured to operate in one of the six (6) following modes.

- Mode 1 - Serial/Loop-Timed Mode
- Mode 2 - Serial/Local-Timed/Frame Slave Mode
- Mode 3 - Serial/Local-Timed/Frame Master Mode
- Mode 4 - Nibble/Loop-Timed Mode
- Mode 5 - Nibble/Local-Timed/Frame Slave Mode
- Mode 6 - Nibble/Local-Timed/Frame Master Mode

Each of these modes are described, in detail, below.

6.2.1.1 Mode 1 - The Serial/Loop-Timing Mode The Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Loop-Timing (Uses the RxLineClk signal as the Timing Reference)

Since the XRT74L74 is configured to operate in the loop-timed mode, the Transmit Section of the XRT74L74 will use the RxLineClk input clock signal (e.g., the Recovered Clock signal, from the LIU) as its timing source. When the XRT74L74 is operating in this mode it will do the following.

1. It will ignore any signal at the TxInClk input pin.
2. The XRT74L74 will output a 34.368MHz clock signal via the RxOutClk output pin. This clock signal functions as the Transmit Payload Data Input Interface block clock signal.
3. The XRT74L74 will use the rising edge of the RxOutClk signal to latch in the data residing on the TxSer input pin.

B. Serial Mode

The XRT74L74 will accept the E3 payload data from the Terminal Equipment, in a serial-manner, via the TxSer input pin The Transmit Payload Data Input Interface will latch this data into its circuitry, on the rising edge of the RxOutClk output clock signal.

C. Delineation of outbound E3 frames

The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period coincident with the XRT74L74 processing the last bit of a given E3 frame.

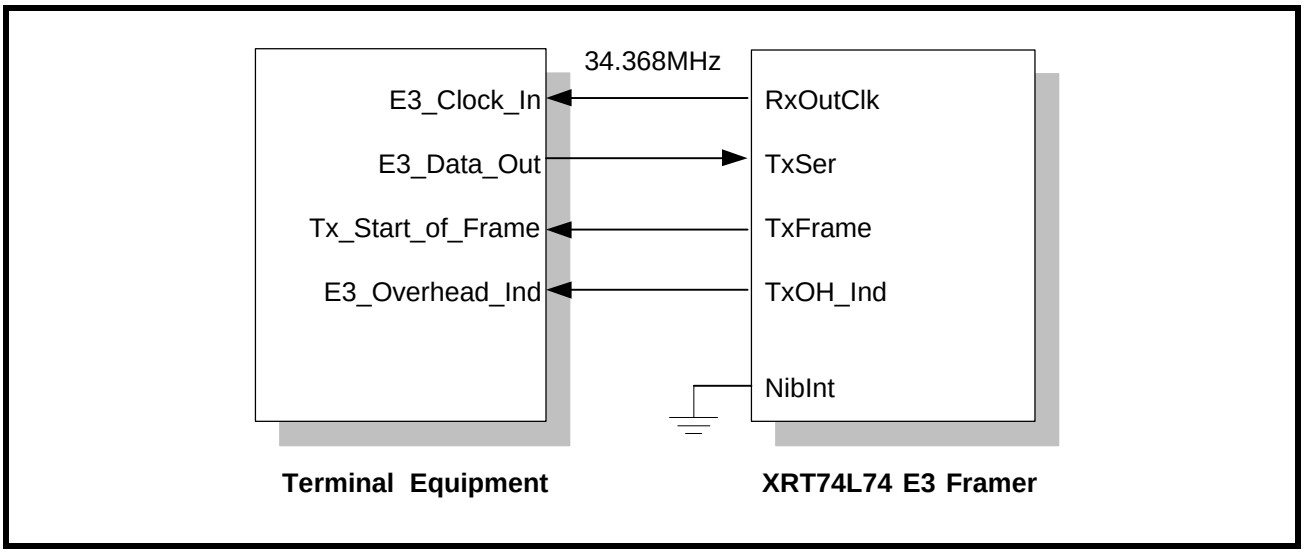
D. Sampling of Payload Data, from the Terminal Equipment

In Mode 1, the XRT74L74 will sample the data at the TxSer input, on the rising edge of RxOutClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 1 Operation

Figure 107 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 1 operation.

FIGURE 107. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 1 (SERIAL/LOOP-TIMED) OPERATION



Mode 1 Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode, it will function as the source of the 34.368MHz clock signal. This clock signal will be used as the Terminal Equipment Interface clock by both the XRT74L74 IC and the Terminal Equipment.

The Terminal Equipment will serially output the payload data of the outbound E3 data stream via its E3_Data_Out pin. The Terminal Equipment will update the data on the E3_Data_Out pin upon the rising edge of the 34.368 MHz clock signal, at its E3_Clock_In input pin (as depicted in Figure 107 and Figure 108).

The XRT74L74 will latch the outbound E3 data stream (from the Terminal Equipment) on the rising edge of the RxOutClk signal.

The XRT74L74 will indicate that it is processing the last bit, within a given outbound E3 frame, by pulsing its TxFrame output pin "High" for one bit-period. When the Terminal Equipment detects this pulse at its Tx_Start_of_Frame input, it is expected to begin transmission of the very next outbound E3 frame to the XRT74L74 via the E3_Data_Out (or TxSer pin).

Finally, the XRT74L74 will indicate that it is about to process an overhead bit by pulsing the TxOH_Ind output pin "High" one bit period prior to its processing of an OH (Overhead) bit. In Figure 107 , the TxOH_Ind output pin is connected to the E3_Overhead_Ind input pin of the Terminal Equipment. Whenever the E3_Overhead_Ind pin is pulsed "High" the Terminal Equipment is expected to not transmit a E3 payload bit upon the very next clock edge. Instead, the Terminal Equipment is expected to delay its transmission of the very next payload bit, by one clock cycle.

The behavior of the signals, between the XRT74L74 and the Terminal Equipment, for E3 Mode 1 operation is illustrated in Figure 108 .

Inserting the A and N bits into the outbound E3 frames via the Transmit Payload Data Input Interface block

The XRT74L74 DS3/E3 Framer permits the Terminal Equipment to insert its own values for the “A” and/or “N” bits, into the outbound E3 frame, via the Transmit Payload Data Input Interface block. If the user desires to do this, the XRT74L74 Framer IC must be configured to accept the Terminal Equipment’s value for the “A” and “N” bits, by writing to appropriate data into the TxASourceSel[1:0] and TxNSourceSel[1:0] bit-fields, within the TxE3 Configuration Register (Address =0x30), as illustrated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	X	X	X	X	0	0	0

Configuring the Transmit Payload Data Input Interface block to accept the "A Bits" from the Terminal Equipment

If the user wishes to configure the Transmit Payload Data Input Interface block to accept the "A" bits from the Terminal Equipment, then the user must write the value "10" into the TxASourceSel[1:0] bit-fields. Once the user does this, then any value, which resides on the TxSer input pin, when the "A" bit is being processed by the Transmit Section will be inserted into the "A" bit-field within the very next outbound E3 frame.

For completeness, the relationship between the contents of the TxASourceSel[1:0] bits and the resulting source of the "A" bit is listed below.

Bit 6, 5, TxASourceSel[1:0]

These two Read/Write bit-fields combine to specify the source of the A-bit, within each outbound E3 frame. The relationship between these two bit-fields and the resulting source of the A Bit is tabulated below.

TXASOURCESEL[1:0]	SOURCE OF A BIT
00	TxE3 Service Bits Register (Address = 0x35)
01	Transmit Overhead Data Input Interface
10	Transmit Payload Data Input Interface
11	Functions as a FEBE (Far-End-Block Error) bit-field. This bit-field is set to "0", if the Near-End Receive Section (within this chip) detects no BIP-4 Errors within the incoming E3 frames. This bit-field is set to "1", if the Near-End Receive Section (within this chip) detects a BIP-4 Error within the incoming E3 frame.

Configuring the Transmit Payload Data Input Interface block to accept the "N" Bits from the Terminal Equipment, then the user must write the value "11" into the TxNSourceSel[1:0] bit-fields. Once the user does this, then any value, which resides on the TxSer input pin, when the "N" bit is being processed by the Transmit Section will be inserted into the "N" bit-field within the very next outbound E3 frame.

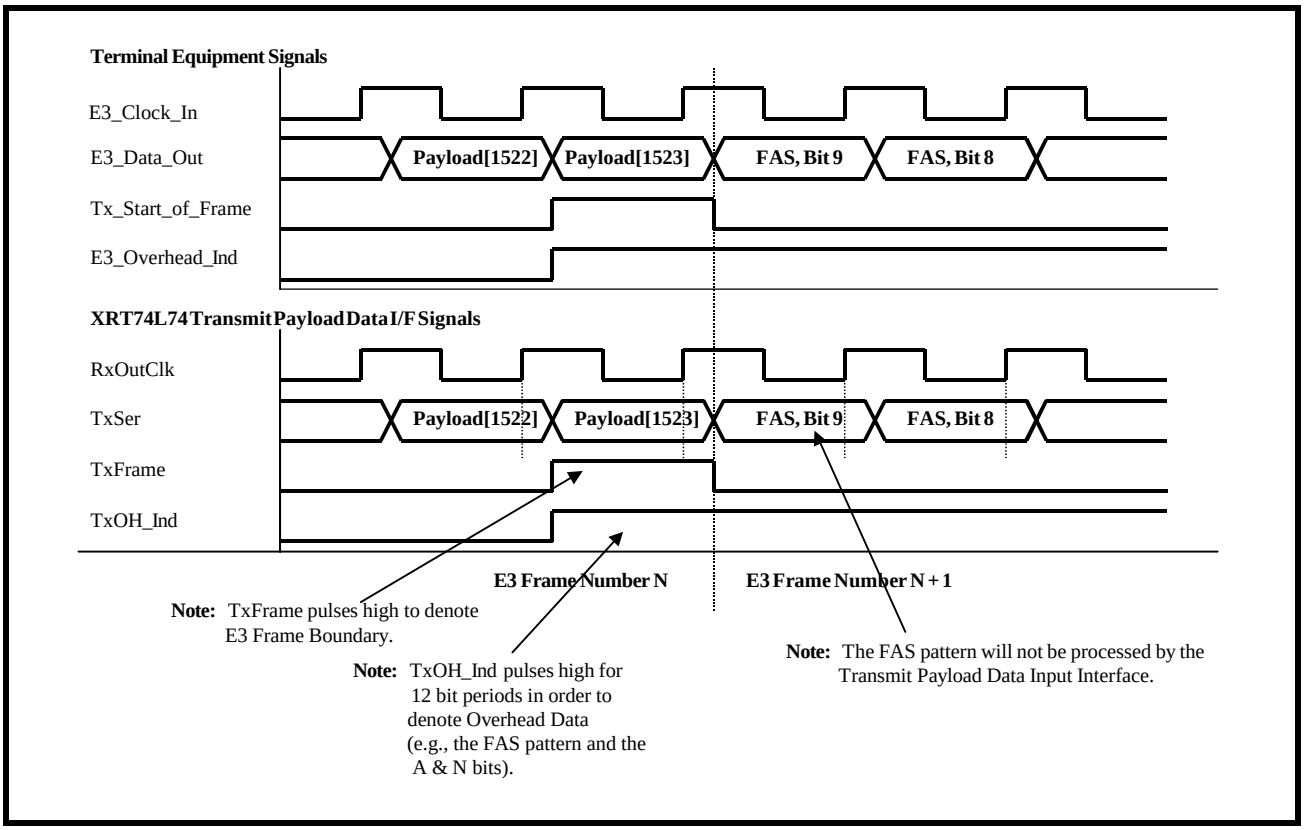
For completeness, the relationship between the contents of the TxNSourceSel[1:0] bits and the resulting source of the "N" bit is listed below.

Bits 4, 3, TxNSourceSel[1:0]

These two Read/Write bit-fields combine to specify the source of the N-bit, within each outbound E3 frame. The relationship between these two bit-fields and the resulting source of the N Bit is tabulated below.

TXNSOURCESEL[1:0]	SOURCE OF N BIT
00	TxE3 Service Bits Register (Address = 0x35)
01	Transmit Overhead Data Input Interface
10	Transmit LAPD Controller
11	Transmit Payload Data Input Interface.

FIGURE 108. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK AND THE TERMINAL EQUIPMENT (FOR MODE 1 OPERATION)



How to configure the XRT74L74 into the Serial/ Loop-Timed/Non-Overhead Interface Mode

1. Set the NibIntf input pin "Low".
2. Set the TimRefSel[1:0] bit fields (within the Framer Operating Mode Register) to "00", as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 107 .

6.2.1.2 Mode 2 - The Serial/Local-Timed/ Frame-Slave Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local-Timed - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the E3 payload data, in a serial manner, via the TxSer input pin. The Transmit Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of outbound E3 frames (Frame Slave Mode)

The Transmit Section of the XRT74L74 will use the TxInClk input as its timing reference, and will use the TxFrameRef input signal as its framing reference. In

other words, the Transmit Section of the XRT74L74 will initiate frame generation upon the rising edge of the TxFrameRef input signal).

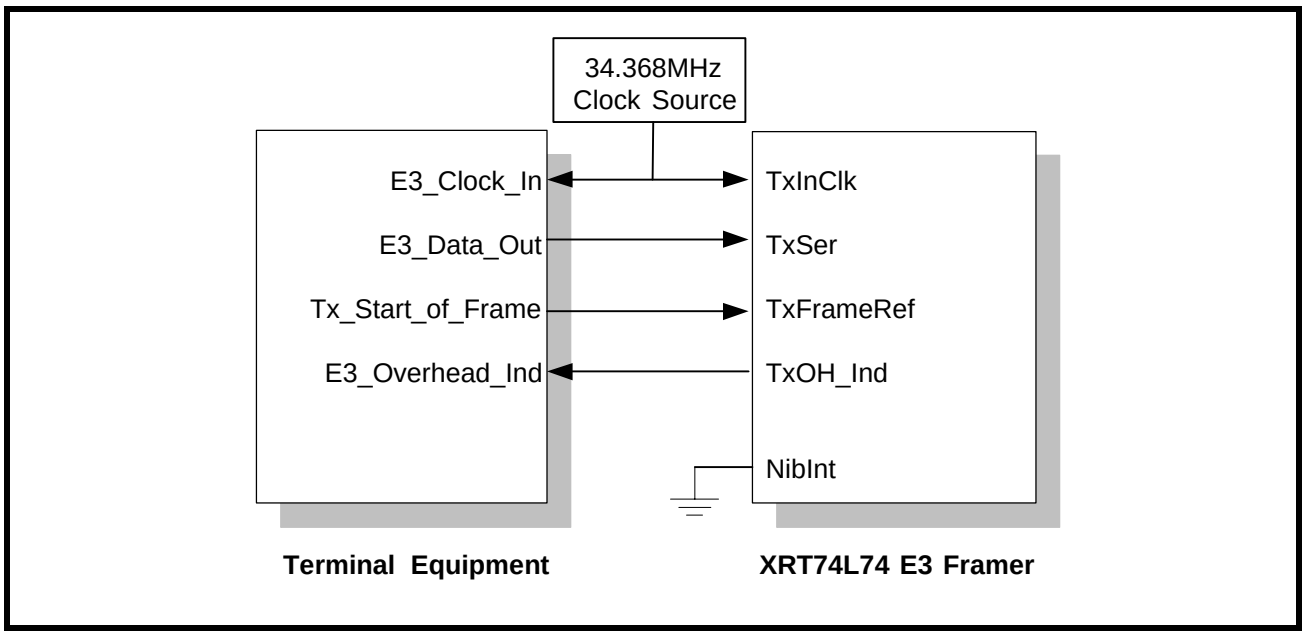
D. Sampling of payload data, from the Terminal Equipment

In Mode 2, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 2 Operation

Figure 109 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 2 operation.

FIGURE 109. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 2 (SERIAL/LOCAL-TIMED/FRAME-SLAVE) OPERATION



Mode 2 Operation of the Terminal Equipment

As shown in Figure 109 , both the Terminal Equipment and the XRT74L74 will be driven by an external 34.368MHz clock signal. The Terminal Equipment will receive the 34.368MHz clock signal via its E3_Clock_In input pin, and the XRT74L74 Framer IC will receive the 34.368MHz clock signal via the TxInClk input pin.

The Terminal Equipment will serially output the payload data of the outbound E3 data stream, via the E3_Data_Out output pin, upon the rising edge of the signal at the E3_Clock_In input pin.

NOTE: *The E3_Data_Out output pin of the Terminal Equipment is electrically connected to the TxSer input pin*

The XRT74L74 Framer IC will latch the data, residing on the TxSer input line, on the rising edge of the TxInClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing its Tx_Start_of_Frame output signal (and in turn, the TxFrameRef input pin of the XRT74L74), "High" for one-bit period, coincident with the first bit of a new E3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new E3 frame.

NOTES:

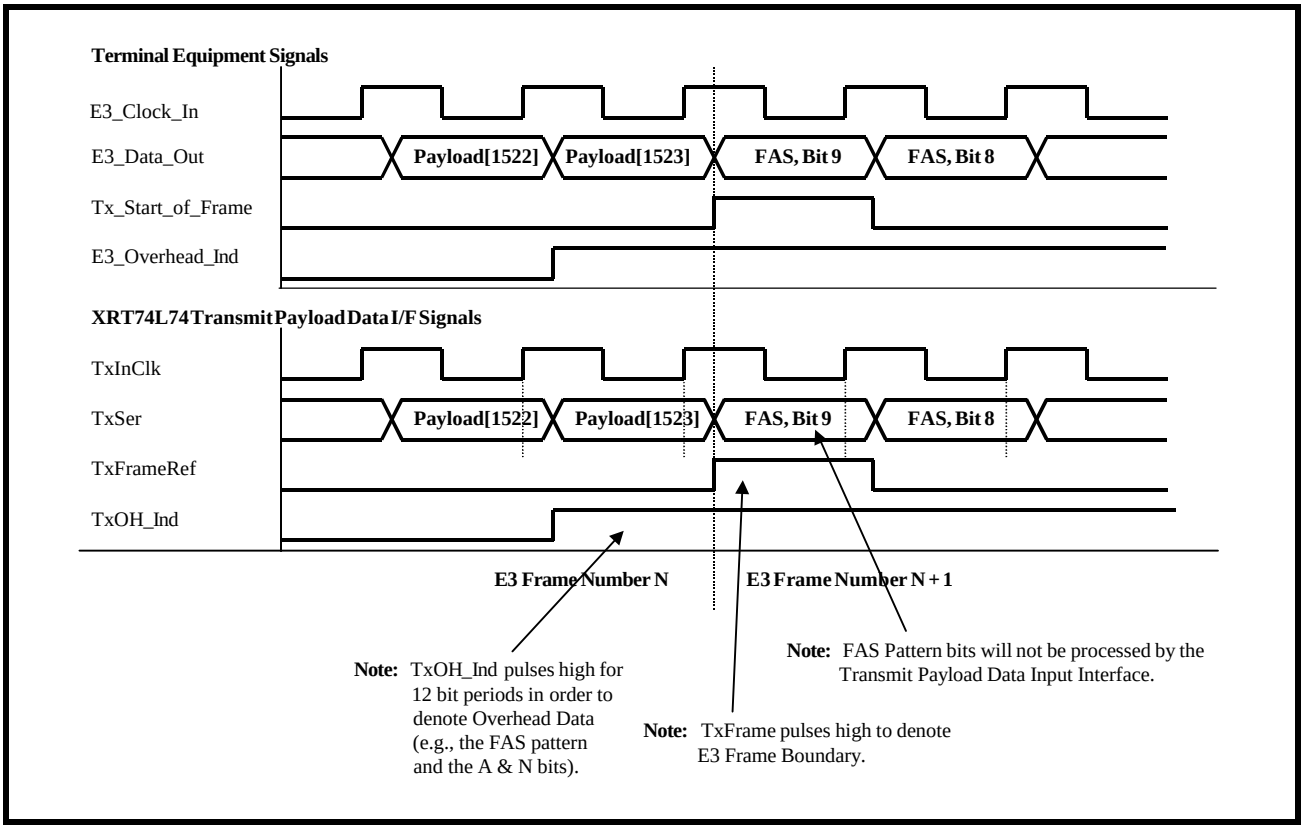
1. *In this case, the Terminal Equipment is controlling the start of Frame Generation, and is therefore referred to as the Frame Master. Conversely, since the XRT74L74 does not control the generation of a new E3 frame, but is rather driven by the Terminal Equipment, the XRT74L74 is referred to as the Frame Slave.*
2. *If the user opts to configure the XRT74L74 to operate in Mode 2, it is imperative that the Tx_Start_of_Frame (or TxFrameRef) signal is synchronized to the TxInClk input clock signal.*

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT74L74 is electrically connected to the E3_Overhead_Ind whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the E3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it

should delay transmission of the very next E3 frame payload bit by one clock cycle.

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 2 Operation is illustrated in Figure 110 .

FIGURE 110. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 2 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 109 .

6.2.1.3 Mode 3 - The Serial/Local-Timed/ Frame-Master Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local-Timed - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the E3 payload data, in a serial manner, via the TxSer input pin. The Transmit

Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of outbound DS3 frames (Frame Master Mode)

The Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference, and will initiate E3 frame generation, asynchronously with respect to any externally applied signal. The XRT74L74 will pulse its TxFrame output pin "High" whenever its it processing the very last bit-field within a given E3 frame.

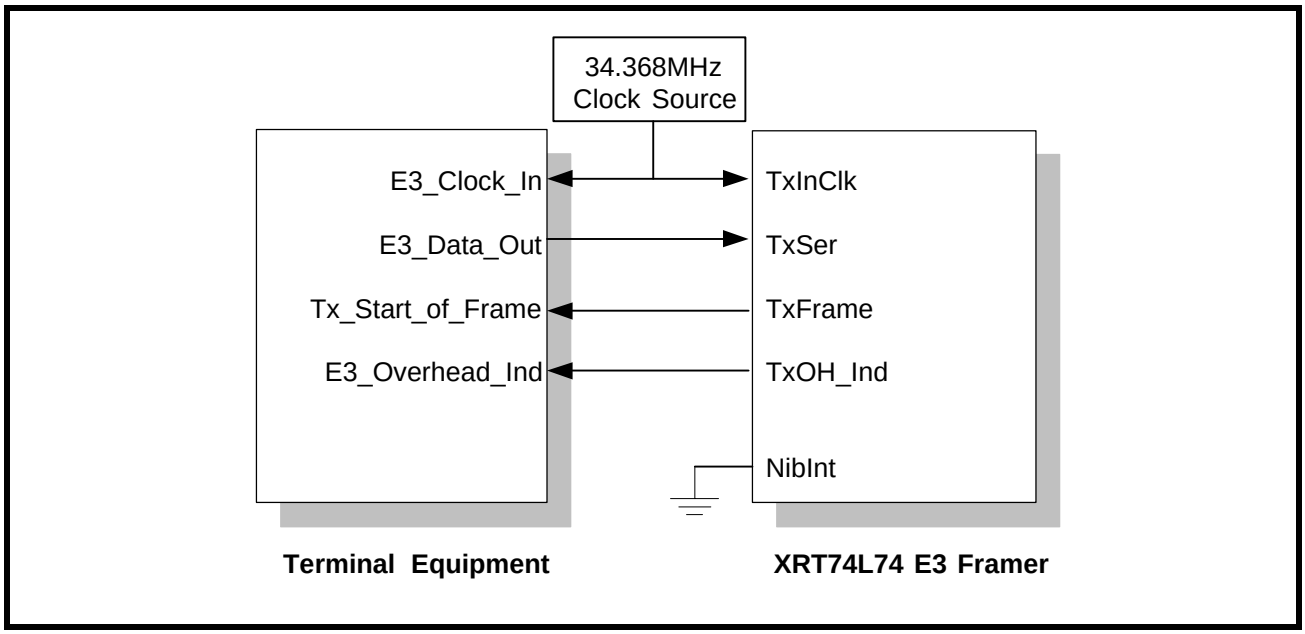
D. Sampling of payload data, from the Terminal Equipment

In Mode 3, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 3 Operation

Figure 111 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 3 operation.

FIGURE 111. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 3 (SERIAL/LOCAL-TIME/FRAME-MASTER) OPERATION



Mode 3 Operation of the Terminal Equipment

In Figure 111 , both the Terminal Equipment and the XRT74L74 are driven by an external 34.368 MHz clock signal. This clock signal is connected to the E3_Clock_In input of the Terminal Equipment and the TxInClk input pin of the XRT74L74.

The Terminal Equipment will serially output the payload data on its E3_Data_Out output pin, upon the rising edge of the signal at the E3_Clock_In input pin. Similarly, the XRT74L74 will latch the data, residing on the TxSer input pin, on the rising edge of TxInClk.

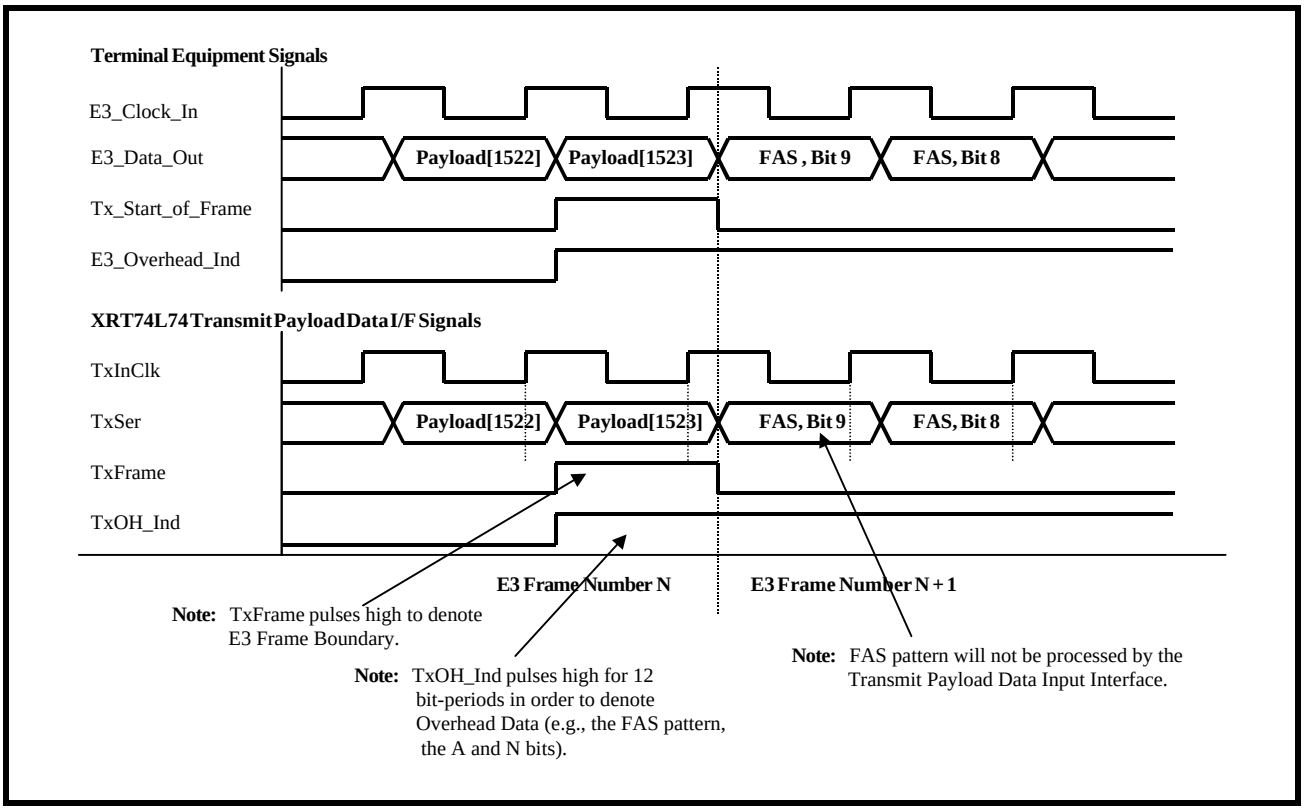
The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period, coincident while it is processing the last bit-field within a given outbound E3 frame. The Terminal Equipment is expected to monitor the TxFrame signal (from the XRT74L74) and to place the first bit, within the very next outbound E3 frame on the TxSer input pin.

NOTE: *In this case, the XRT74L74 dictates exactly when the very next E3 frame will be generated. The Terminal Equipment is expected to respond appropriately by providing the XRT74L74 with the first bit of the new E3 frame, upon demand. Hence, in this mode, the XRT74L74 is referred to as the Frame Master and the Terminal Equipment is referred to as the Frame Slave.*

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the outbound E3 frame. Since the TxOH_Ind output pin (of the XRT74L74) is electrically connected to the E3_Overhead_Ind whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the E3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it should delay transmission of the very next DS3 frame payload bit by one clock cycle.

The behavior of the signal between the XRT74L74 and the Terminal Equipment for E3 Mode 3 Operation is illustrated in Figure 112 .

FIGURE 112. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3 MODE 3 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 111 .

6.2.1.4 Mode 4 - The Nibble-Parallel/Loop-Timed Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Looped Timing (Uses the RxLineClk as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the RxLineClk signal as its timing reference. When the XRT74L74 is operating in the Nibble-Mode, it will internally divide the RxLineClk signal, by a factor of four (4) and will output this signal via the TxNibClk output pin.

B. Nibble-Parallel Mode

The XRT74L74 will accept the E3 payload data, from the Terminal Equipment in a nibble-parallel manner,

via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface block will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of the outbound E3 frames

The XRT74L74 will pulse the TxNibFrame output pin "High" for one bit-period coincident with the XRT74L74 processing the last nibble of a given E3 frame.

D. Sampling of payload data, from the Terminal Equipment

In Mode 4, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the RxOutClk clock signal, following a pulse in the TxNibClk signal (see Figure 114).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 8.592 MHz (e.g., 34.368 MHz divided by 4).*

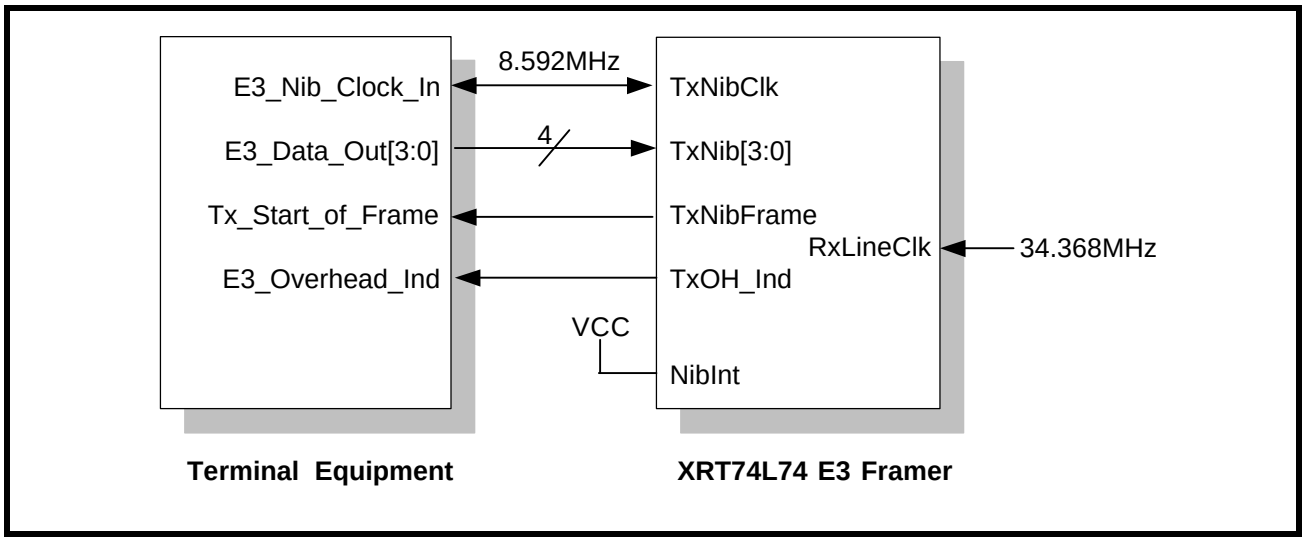
The E3 Frame consists of 1536 bits or 384 nibbles. Therefore, the XRT74L74 will supply 384 TxNibClk pulses between the rising edges of two consecutive TxNibFrame pulses. The E3 Frame repetition rate is 22.375kHz. Hence, 384 TxNibClk pulses for each E3 frame period amounts to TxNibClk running at approximately 8.592 MHz. The method by which the 384 TxNibClk pulses are distributed throughout the E3 frame period is presented below.

Nominally, the Transmit Section within the XRT74L74 will generate a TxNibClk pulse for every 4 RxOutClk (or TxInClk) periods.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 4 Operation

Figure 113 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 4 Operation.

FIGURE 113. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMED) OPERATION



Mode 4 Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode, it will function as the source of the 8.592MHz (e.g., the 34.368MHz clock signal divided by 4) clock signal, that will be used as the Terminal Equipment Interface clock by both the XRT74L74 and the Terminal Equipment.

The Terminal Equipment will output the payload data of the outbound E3 data stream via its E3_Data_Out[3:0] pins on the rising edge of the 8.592MHz clock signal at the E3_Nib_Clock_In input pin.

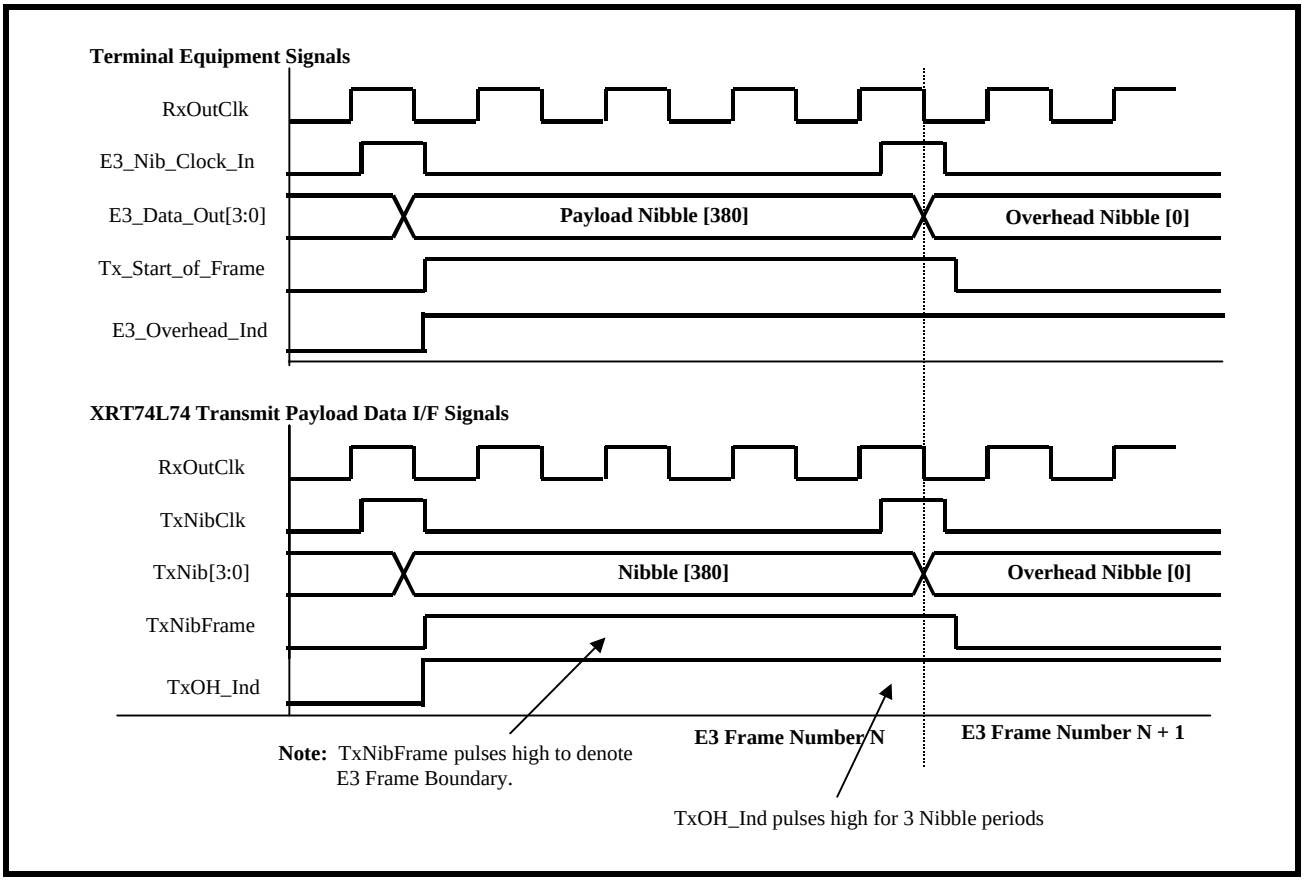
The XRT74L74 will latch the outbound E3 data stream (from the Terminal Equipment) on the rising edge of the TxNibClk output clock signal. The XRT74L74 will indicate that it is processing the last nibble, within a given E3 frame, by pulsing its TxNibFrame output pin "High" for one TxNibClk clock period. When the Terminal Equipment detects a pulse at its Tx_Start_of_Frame input pin, it is expected to transmit the first nibble, of the very next outbound E3 frame to the XRT74L74 via the E3_Data_Out[3:0] (or TxNib[3:0] pins).

Finally, for the Nibble-Parallel Mode operation, the XRT74L74 will pulse the TxOHInd output pin "High"

for 3 nibble-periods (e.g., the 3 nibbles consisting of the 10 bit FAS pattern, the “A” and the “N” bits). The TxOHInd output pin will remain "Low” for the remainder of the frame period. The TxOHInd output pin will toggle "High” one-nibble period before the Transmit Section (of the Framer IC) processes the first four bits of the FAS pattern.

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 4 Operation is illustrated in Figure 114 .

FIGURE 114. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 4 OPERATION)



How to configure the XRT74L74 into Mode 4

1. Set the NibIntf input pin "High".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "00" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 113 .

NOTE: *The XRT74L74 Framer IC cannot support the Framer Local Loop-back Mode of operation, while operating in Mode 4. The user must configure the XRT74L74 Framer IC into any of the following modes prior to configuring the Framer Local Loop-back Mode operation.*

- Mode 2 - Serial/Local-Timed/Frame-Slave Mode
- Mode 3 - Serial/Local-Timed/Frame-Master Mode

- Mode 5 - Nibble-Parallel/Local-Timed/Frame-Slave Mode
- Mode 6 - Nibble-Parallel/Local-Timed/Frame-Master Mode.

For more detailed information on the Framer Local Loop-back Mode, please see Section 6.0.

6.2.1.5 Mode 5 - The Nibble-Parallel/Local-Timed/Frame-Slave Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local-Timed - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the E3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of outbound E3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will use the TxFrameRef input signal as its Framing Reference (e.g., the Transmit Section of the XRT74L74 initiates frame generation upon the rising edge of the TxFrameRef signal).

D. Sampling of payload data, from the Terminal Equipment

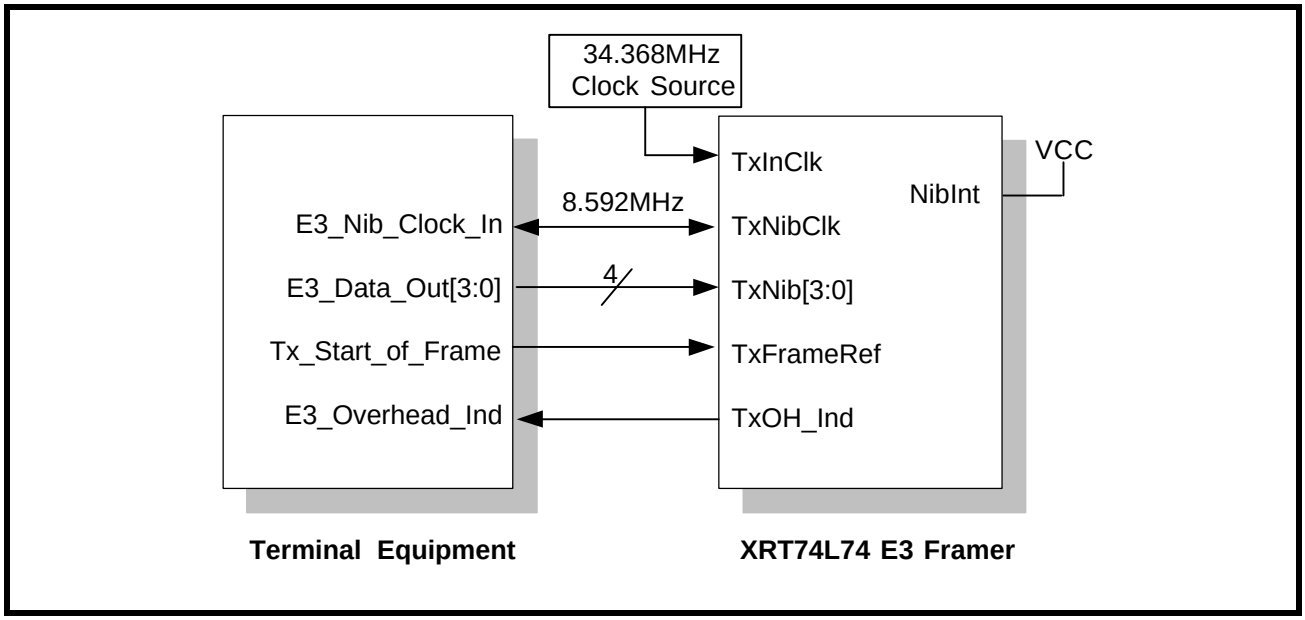
In Mode 5, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 115).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 8.592 MHz (e.g., 34.368 MHz divided by 4).*

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 5 Operation

Figure 115 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 5 Operation.

FIGURE 115. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMED/FRAME-SLAVE) OPERATION



Mode 5 Operation of the Terminal Equipment

In Figure 115 both the Terminal Equipment and the XRT74L74 will be driven by an external 8.592MHz clock signal. The Terminal Equipment will receive the 8.592MHz clock signal via the E3_Nib_Clock_In input pin. The XRT74L74 will output the 8.592MHz clock signal via the TxNibClk output pin.

The Terminal Equipment will serially output the data on the E3_Data_Out[3:0] pins, upon the rising edge of the signal at the E3_Clock_In input pin.

NOTE: *The E3_Data_Out[3:0] output pins of the Terminal Equipment is electrically connected to the TxNib[3:0] input pins.*

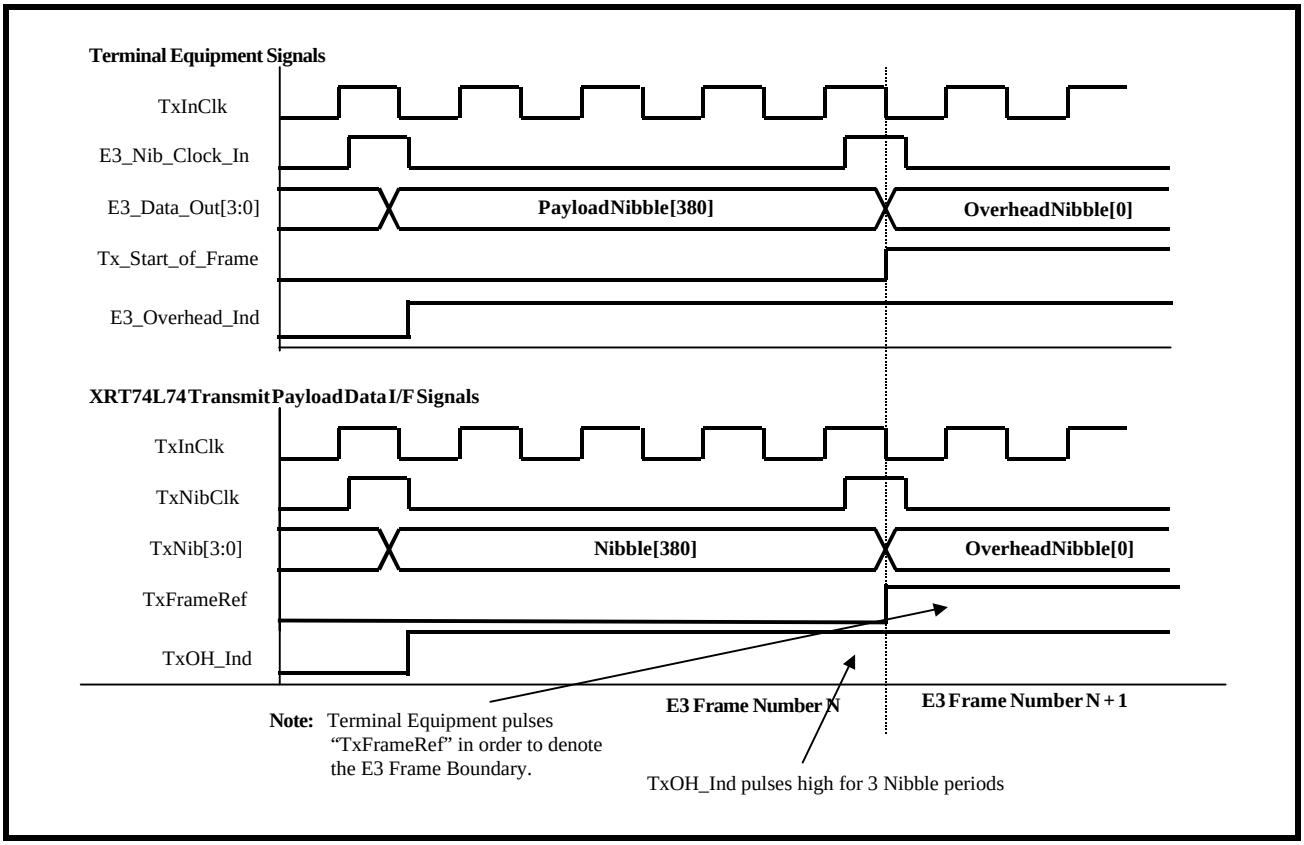
The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing the Tx_Start_of_Frame output pin (and in turn, the TxFrameRef input pin of the XRT74L74) "High" for one bit-period, coincident with the first bit of a new E3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new E3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the E3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 5 Operation is illustrated in Figure 116 .

FIGURE 116. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3, MODE 5 OPERATION)



How to configure the XRT74L74 into Mode 5

1. Set the NibIntf input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 115 .

6.2.1.6 4.2.1.6 Mode 6 - The Nibble-Parallel/ Local-Timed/Frame-Master Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local-Timed - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the E3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of outbound E3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will initiate the generation of E3 frames, asynchronous with respect to any external signal. The XRT74L74 will pulse the TxFrame output pin "High" whenever it is processing the last bit, within a given outbound E3 frame.

D. Sampling of payload data, from the Terminal Equipment

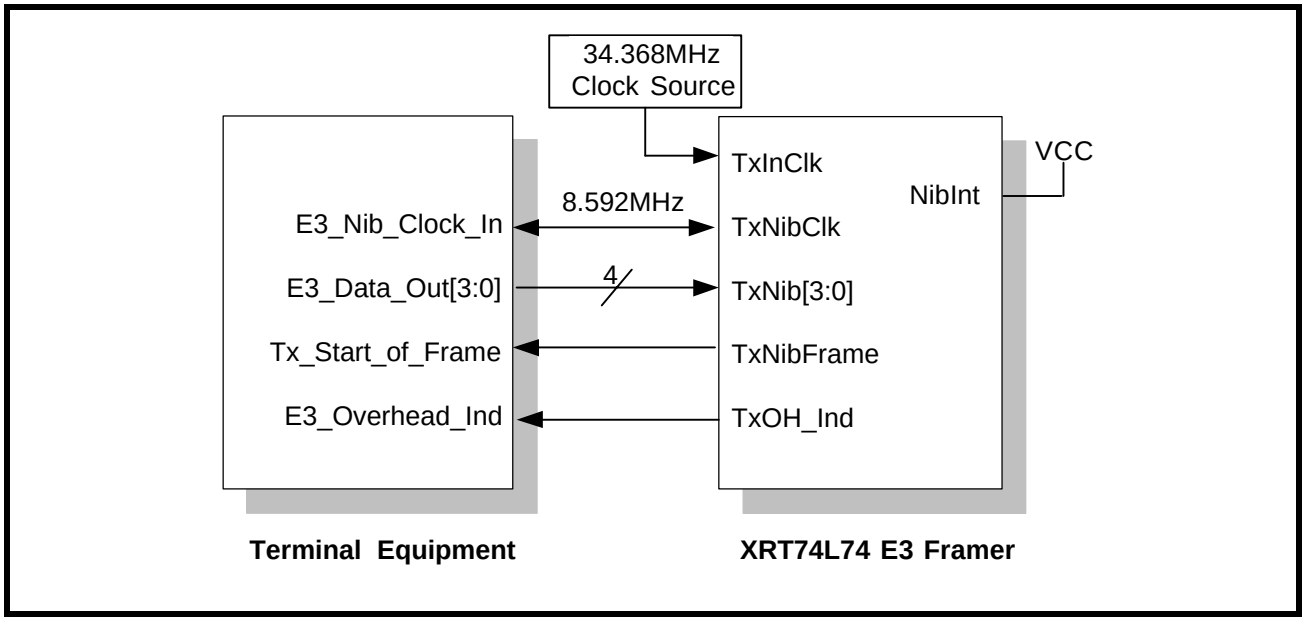
In Mode 6, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 118).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 8.592 MHz (e.g., 34.368 MHz divided by 4).*

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 6 Operation

Figure 117 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 6 Operation.

FIGURE 117. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMED/FRAME-MASTER) OPERATION



Mode 6 Operation of the Terminal Equipment

In Figure 117 both the Terminal Equipment and the XRT74L74 will be driven by an external 8.592MHz clock signal. The Terminal Equipment will receive the 8.592MHz clock signal via the E3_Nib_Clock_In input pin. The XRT74L74 will output the 8.592MHz clock signal via the TxNibClk output pin.

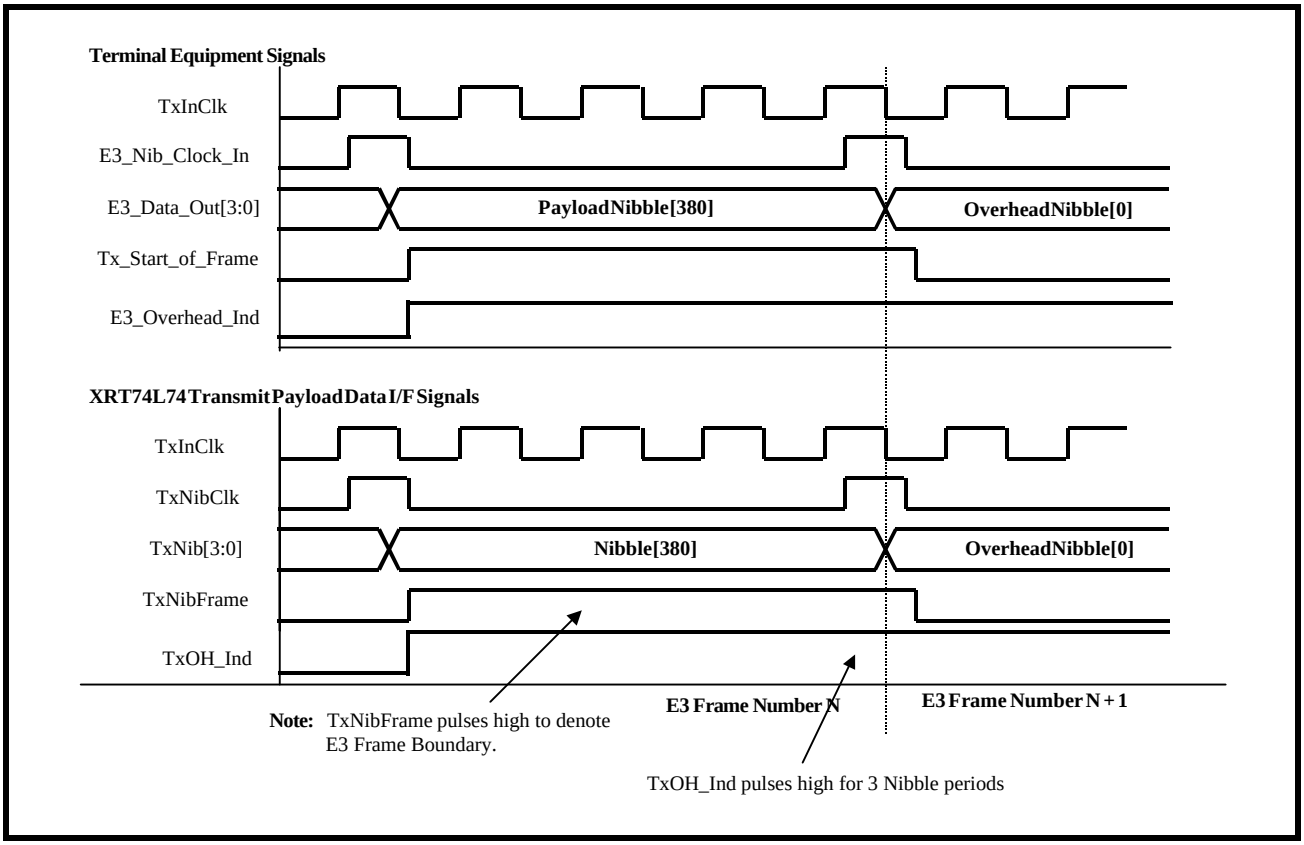
The Terminal Equipment will serially output the data on the E3_Data_Out[3:0] pins upon the rising edge of the signal at the E3_Clock_In input pin. The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case the XRT74L74 has the responsibility of providing the framing reference signal by pulsing the TxFrame output pin (and in turn the Tx_Start_of_Frame input pin of the Terminal Equipment) "High" for one bit-period, coincident with the last bit within a given E3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the E3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 6 Operation is illustrated in Figure 118 .

FIGURE 118. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3 MODE 6 OPERATION)



How to configure the XRT74L74 into Mode 6

1. Set the NibInt input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "1X" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

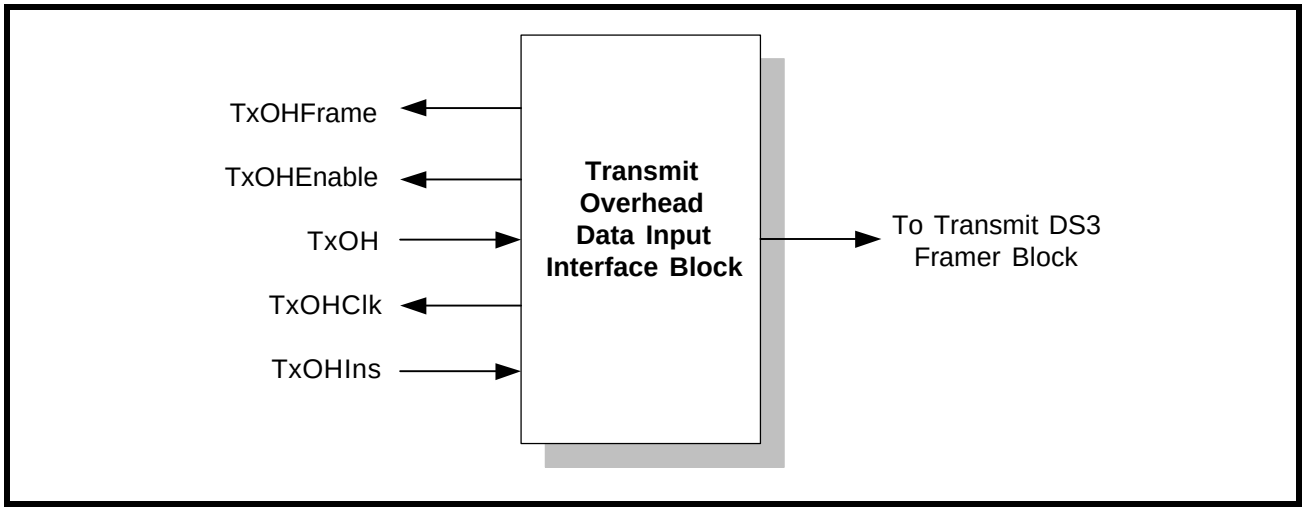
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	x

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 117 .

6.2.2 The Transmit Overhead Data Input Interface

Figure 119 presents a simple illustration of the Transmit Overhead Data Input Interface block within the XRT74L74.

FIGURE 119. THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK



The E3, ITU-T G.751 Frame consists of 1536 bits. Of these bits, 1524 are payload bits and the remaining 12 are overhead bits. The XRT74L74 has been designed to handle and process both the payload type and overhead type bits for each E3 frame. Within the Transmit Section within the XRT74L74, the Transmit Payload Data Input Interface has been designed to handle the payload data. Likewise, the Transmit Overhead Input Interface has been designed to handle and process the overhead bits.

The Transmit Section of the XRT74L74 generates or processes the various overhead bits within the E3 frame, in the following manner.

The Frame Alignment Signaling (FAS) Overhead Bits

The FAS (Framing Alignment Signaling) bits are always internally generated by the Transmit Section of the XRT74L74. Hence, the user cannot insert his/her value for the FAS bits into the outbound E3 data stream, via the Transmit Overhead Data Input Interface.

The "A" (Alarm) Overhead bit

The "A" bit is used to transport the FERF (Far-End Receive Failure) condition. This bit-field can be either internally generated by the Transmit Section within the XRT74L74, or can be externally generated and inserted into the outbound E3 data stream, via the Transmit Overhead Data Input Interface. The Data Link Related Overhead Bits

The "N" (National) Overhead bit

The E3 frame structure also contains the N bit which can be used to transport a proprietary User Data Link information and or Path Maintenance Data Link information. The UDL (User Data Link) bits are only accessible via the Transmit Overhead Data Input Interface. The Path Maintenance Data Link (PMDL) bits can either be sourced from the Transmit LAPD Controller/Buffer or via the Transmit Overhead Data Input Interface.

Table 63 lists the Overhead Bits within the E3 frame. In addition, this table also indicates whether or not these overhead bits can be sourced by the Transmit Overhead Data Input Interface.

TABLE 63: A LISTING OF THE OVERHEAD BITS WITHIN THE E3 FRAME, AND THEIR POTENTIAL SOURCES

OVERHEAD BIT	INTERNALLY GENERATED	ACCESSIBLE VIA THE TRANSMIT OVERHEAD DATA INPUT INTERFACE	BUFFER/REGISTER ACCESSIBLE
FAS Signal - Bit 9	Yes	Yes	Yes*
FAS Signal - Bit 8	Yes	Yes	Yes
FAS Signal - Bit 7	Yes	Yes	Yes*
FAS Signal - Bit 6	Yes	Yes	Yes*
FAS Signal - Bit 5	Yes	Yes	Yes
FAS Signal - Bit 4	Yes	Yes	Yes
FAS Signal - Bit 3	Yes	Yes	Yes
FAS Signal - Bit 2	Yes	Yes	Yes
FAS Signal - Bit 1	Yes	Yes	Yes
FAS Signal - Bit 0	Yes	Yes	Yes
A Bit	Yes	Yes	Yes
N Bit	Yes	Yes	Yes

NOTES:

1. *The XRT74L74 contains mask register bits that permit the user to alter the state of the internally generated value for these bits.*
2. *The Transmit LAPD Controller/Buffer can be configured to be the source of the N bits, within the outbound E3 data stream.*

The Transmit Overhead Data Input Interface permits the user to insert overhead data into the outbound E3 frames via the following two different methods.

- Method 1 - Using the TxOHClk clock signal
- Method 2 - Using the TxInClk and the TxOHEnable signals.

Each of these methods are described below.

6.2.2.1 Method 1 - Using the TxOHClk Clock Signal

The Transmit Overhead Data Input Interface consists of the five signals. Of these five (5) signals, the following four (4) signals are to be used when implementing Method 1.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

Each of these signals are listed and described below.

Table 64 .

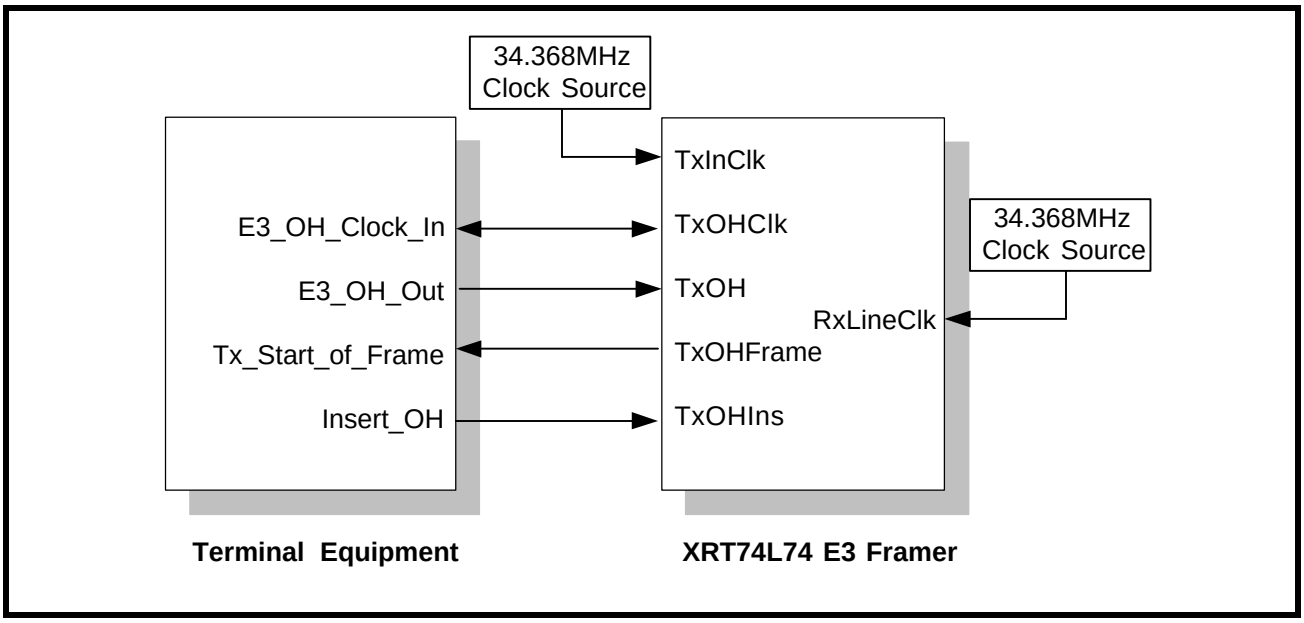
TABLE 64: DESCRIPTION OF METHOD 1 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next outbound E3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.
TxOHClk	Output	**Transmit Overhead Input Interface Clock Output signal:** This output signal serves two purposes: 1. The Transmit Overhead Data Input Interface will provide a rising clock edge on this signal, one bit-period prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit. 2. The Transmit Overhead Data Input Interface will sample the data at the TxOH input, on the falling edge of this clock signal (provided that the TxOHIns input pin is "High"). ***NOTE:*** *The Transmit Overhead Data Input Interface will supply a clock edge for all overhead bits within the DS3 frame (via the TxOHClk output signal). This includes those overhead bits that the Transmit Overhead Data Input Interface will not accept from the Terminal Equipment.*
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given E3 frame. The purpose of this output signal is to alert the Terminal Equipment that the Transmit Overhead Data Input Interface block is about to begin processing the overhead bits for a new E3 frame.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment.

Figure 120 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment, when using Method 1.

FIGURE 120. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 1)



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the outbound E3 data stream, (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of the TxOHFrame signal (e.g., the Tx_Start_of_Frame input signal) on the rising edge of the TxOHClk (e.g., the E3_OH_Clock_In signal).
2. To keep track of the number of rising clock edges that have occurred, via the TxOHClk (e.g., the E3_OH_Clock_In signal) since the last time the TxOHFrame signal was sampled "High". By doing this the Terminal Equipment will be able to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface block at any given time. When the Terminal Equipment knows which overhead bit is being processed, at a given TxOHClk period, it will know when to insert a desired overhead bit value into the outbound E3 data stream. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin (of the XRT74L74).

Table 65 relates the number of rising clock edges (in the TxOHClk signal, since TxOHFrame was sampled "High") to the E3 Overhead Bit, that is being processed.

TABLE 65: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN TXOHCLK, (SINCE TXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED

NUMBER OF RISING CLOCK EDGES IN TXOHCLK	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (Clock edge is coincident with TxOHFrame being detected "High")	FAS Signal - Bit 9	Yes
1	FAS Signal - Bit 8	Yes
2	FAS Signal - Bit 7	Yes
3	FAS Signal - Bit 6	Yes
4	FAS Signal - Bit 5	Yes
5	FAS Signal - Bit 4	Yes
6	FAS Signal - Bit 3	Yes
7	FAS Signal - Bit 2	Yes
8	FAS Signal - Bit 1	Yes
9	FAS Signal - Bit 0	Yes
10	A Bit	Yes
11	N Bit	Yes

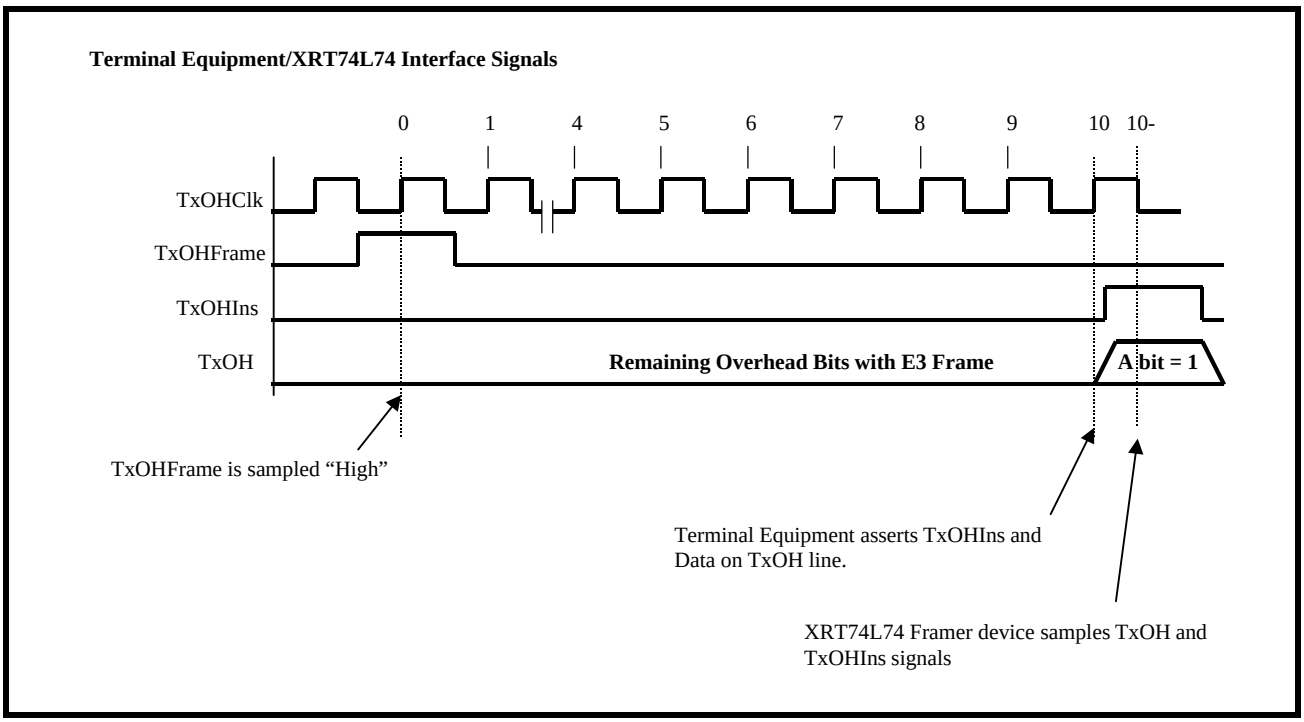
3. After the Terminal Equipment has waited the appropriate number of clock edges (from the TxOHFrame signal being sampled "High"), it should assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal, stable until the next rising edge of TxOHClk is detected.

Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using Method 1) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this example, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface, such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for E3, ITU-T G.751 Applications, a Yellow Alarm is transmitted by setting the "A" bit to "1".

If one assumes that the connection between the Terminal Equipment and the XRT74L74 are as illustrated in Figure 120 then Figure 121 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 121. ILLUSTRATION OF THE SIGNAL THAT MUST OCCUR BETWEEN THE TERMINAL EQUIPMENT AND THE XRT74L74 IN ORDER TO CONFIGURE THE XRT74L74 TO TRANSMIT A YELLOW ALARM TO THE REMOTE TERMINAL EQUIPMENT



In Figure 121 the Terminal Equipment samples the TxOHFrame signal being "High" at rising clock edge # 0. From this point, the Terminal Equipment will wait until it has detected the 10th rising edge of the TxOHClk signal. At this point, the Terminal Equipment knows that the XRT74L74 is just about to process the “A” bit within a given outbound E3 frame. Additionally, according to Table 65 , the 10th overhead bit to be processed is the "A" bit. In order to facilitate the transmission of the Yellow Alarm, the Terminal Equipment must set this "A" bit to "1". Hence, the Terminal Equipment starts this process by implementing the following steps concurrently.

a. Assert the TxOHIns input pin by setting it "High".

b. Set the TxOH input pin to "1".

After the Terminal Equipment has applied these signals, the XRT74L74 will sample the data on both the TxOHIns and TxOH signals upon the very next falling edge of TxOHClk (designated as "10-" in Figure 121). Once the XRT74L74 has sampled this data, it will then insert a "1" into the "A" bit position, in the outbound E3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as clock edge 1 in Figure 121 , the Terminal Equipment will negate the TxOHIns signal (e.g., toggles it "Low") and will cease inserting data into the Transmit Overhead Data Input Interface.

After the Terminal Equipment has performed this insertion procedure, it leaves the remaining overhead bits (within this particular outbound E3 frame) in-tact, by terminating this Overhead Bit Insertion procedure. The Terminal Equipment should now terminate this overhead bit insertion, by doing the following.

a. Assert the TxOHIns input pin by setting it "High".

b. Set the TxOH input to "0".

If the Terminal Equipment wishes to continue its transmission of the Yellow Alarm condition to the Remote Terminal Equipment, then it should resume the Overhead Bit Insertion procedure (as described above), at the beginning of each outbound E3 frame (or each time TxOHFrame is sampled "High”).

6.2.2.2 Method 2 - Using the TxInClk and TxOHEnable Signals

Method 1 requires the use of an additional clock signal, TxOHClk. However, there may be a situation in which the user does not wish to add this extra clock signal to their design, in order to use the Transmit Overhead Data Input Interface. Hence, Method 2 is available. When using Method 2, either the TxInClk or RxOutClk signal is used to sample the overhead bits and signals which are input to the Transmit Over-

head Data Input Interface. Method 2 involves the use of the following signals:

- TxOH
- TxInClk
- TxOHFrame
- TxOHEnable

Each of these signals are listed and described in Table 66 .

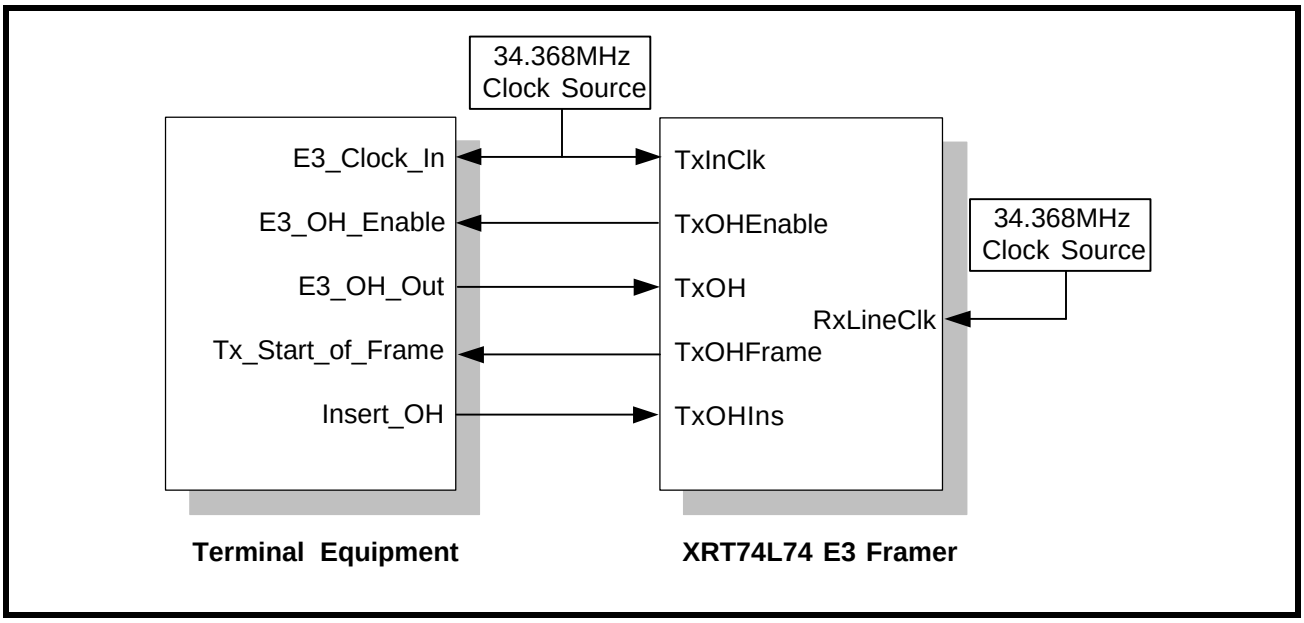
TABLE 66: DESCRIPTION OF METHOD 2 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHEnable	Output	**Transmit Overhead Data Enable Output pin** The XRT74L74 will assert this signal, for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit.
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given DS3 frame.
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxInClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next outbound DS3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment

Figure 122 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment when using Method 2.

FIGURE 122. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 2)



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the outbound E3 data stream (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of both the TxOHFrame and the TxOHEnable input signals, via the E3_Clock_In (e.g., either the TxInClk or the RxOutClk signal of the XRT74L74) signal. If the Terminal Equipment samples the TxOHEnable signal "High", then it knows that the XRT74L74 is about to process an overhead bit. Further, if the Terminal Equipment samples both the TxOHFrame and the TxOHEnable pins "High" (at the same time) then the Terminal Equipment knows that the XRT74L74 is about to process the first overhead bit, within a new E3 frame.
2. To keep track of the number of times that the TxOHEnable signal has been sampled "High" since the last time both the TxOHFrame and the TxOHEnable signals were sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit the Transmit Overhead Data Input Interface is about ready to process. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pins of the XRT74L74.

Table 67 also relates the number of TxOHEnable output pulses (that have occurred since both the TxOHFrame and TxOHEnable pins were sampled "High") to the E3 overhead bit, that is being processed.

TABLE 67: THE RELATIONSHIP BETWEEN THE NUMBER OF TXOHENABLE PULSES (SINCE THE LAST OCCURRENCE OF THE TXOHFRAME PULSE) TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED BY THE XRT74L74

NUMBER OF TXOHENABLE PULSES	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (Clock edge is coincident with TxOHFrame being detected "High")	FAS Signal - Bit 9	Yes
1	FAS Signal - Bit 8	Yes
2	FAS Signal - Bit 7	Yes
3	FAS Signal - Bit 6	Yes
4	FAS Signal - Bit 5	Yes
5	FAS Signal - Bit 4	Yes
6	FAS Signal - Bit 3	Yes
7	FAS Signal - Bit 2	Yes
8	FAS Signal - Bit 1	Yes
9	FAS Signal - Bit 0	Yes
10	A Bit	Yes
11	N Bit	Yes

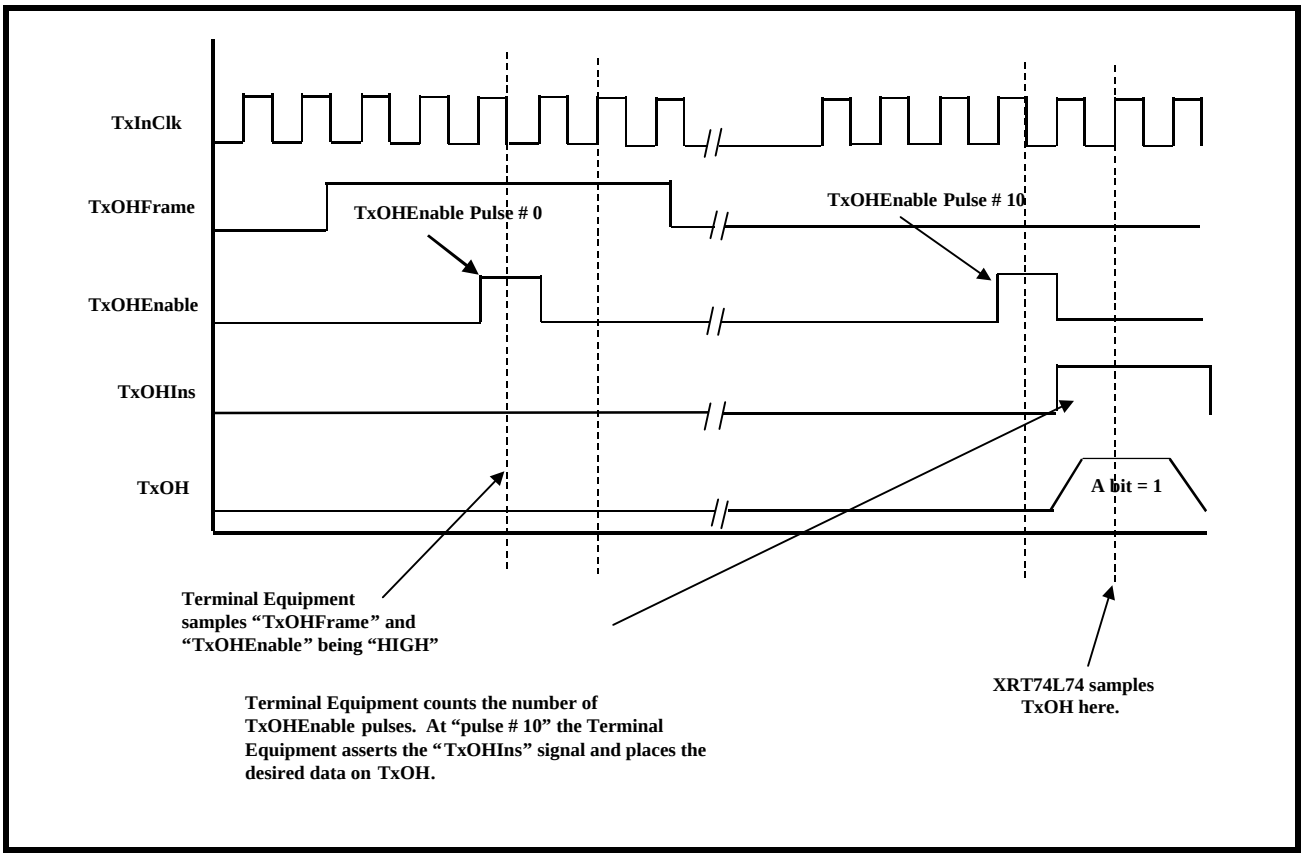
3. After the Terminal Equipment has waited through the appropriate number of pulses via the TxOHEnable pin, it should then assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal stable, until the next TxOHEnable pulse is detected.

Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using Method 2) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this case, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for E3, ITU-T G.751 applications, a Yellow Alarm is transmitted by setting the “A” bit to “1".

If one assumes that the connection between the Terminal Equipment and the XRT74L74 is as illustrated in Figure 122 then, Figure 123 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 123. BEHAVIOR OF TRANSMIT OVERHEAD DATA INPUT INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (FOR METHOD 2)



6.2.3 The Transmit E3 HDLC Controller

The Transmit E3 HDLC Controller block can be used to transport Message-Oriented Signaling (MOS) type messages to the remote terminal equipment as discussed in detail below.

6.2.3.1 Message-Oriented Signaling (e.g., LAP-D) processing via the Transmit DS3 HDLC Controller

The LAPD Transmitter (within the Transmit E3 HDLC Controller Block) allows the user to transmit path maintenance data link (PMDL) messages to the remote terminal via the outbound E3 Frames. In this case the message bits are inserted into and carried by the "N" bit, within the outbound E3 frames. The on-chip LAPD transmitter supports both the 76 byte and 82 byte length message formats, and the Framer IC allocates 88 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer) to store the message to be transmitted. The message format complies with ITU-T Q.921 (LAP-D) protocol with different addresses and is presented below in Figure 124 .

FIGURE 124. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following sections defines each of these bit/byte-fields within the LAPD Message Frame Format.

Flag Sequence Byte

The Flag Sequence byte is of the value 0x7E, 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 (decimal).

TEI - Terminal Endpoint Identifier

The TEI bit-fields are assigned the value of 0x00. The TEI field is used in N-ISDN systems to identify a terminal out of multiple possible terminal. However, since the Framer IC 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 Framer assigned the Control byte the value 03h. Hence, the Framer 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 0x86 through 0xDD).

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 = 0x86, within the Framer). The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. Table 68 presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT74L74 Framer device and the corresponding octet value that the user must write into to the first octet position within the Transmit LAPD Message buffer.

TABLE 68: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD

LAPD MESSAGE TYPE	VALUE OF FIRST BYTE, WITHIN INFORMATION PAYLOAD OF MESSAGE	MESSAGE SIZE
CL Path Identification	0x38	76 bytes
IDLE Signal Identification	0x34	76 bytes
Test Signal Identification	0x32	76 bytes
ITU-T Path Identification	0x3F	82 bytes

Frame Check Sequence Bytes

The 16 bit FCS (Frame Check Sequence) is calculated over the LAPD Message Header and Information Payload bytes, by using the CRC-16 polynomial, $x^{16} + x^{12} + x^5 + 1$.

Operation of the LAPD Transmitter

If the user wishes to transmit a message via the LAPD Transmitter, the information portion (or the body) of the message must be written into the Transmit LAPD Message Buffer, which is located at 0x86 through 0xDD in on-chip RAM via the Microprocessor Interface. Afterwards, the user must do five things:

1. Configure the source of the "N" bit (within each outbound E3 frame, to be the LAPD Transmitter.
2. Specify the length of LAPD message to be transmitted.
3. Specify whether the LAPD Transmitter should transmit this LAPD Message frame only once, or an indefinite number of times at One-Second intervals.
4. Enable the LAPD Transmitter.
5. Initiate the Transmission of the PMDL Message.

Each of these steps will be discussed in detail.

STEP 1 - Configure the source of the "N" bit (within each outbound E3 frame, to be the LAPD Transmitter.

This is accomplished by writing the appropriate data into the TxNSourceSel[1:0] bit-fields, within the TxE3 Configuration Register, as illustrated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Setting TxNSourceSel[1:0] to "10" configures the Transmit E3 Framer block to use the LAPD Transmitter as the data source for the "N" bits. Hence, the "N" bit, (within each outbound E3 frame) is now carrying LAPD Messages to the remote terminal equipment.

STEP 2 - 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 Buffer (e.g., at Address Location 0x86 within the Framer IC). 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 that is desired to be transmitted. This octet will ultimately be used by the Remote Terminal Equipment in order to help it identify the type of LAPD message frame that it is receiving. Table 69 lists these octets and the corresponding LAPD Message types.

STEP 3 - 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 0x87 through 0x135 within the Framer IC).

STEP 4 - Specifying the Length of the LAPD Message

One of two different sizes of LAPD Messages can be transmitted. This can be accomplished by writing the appropriate data to bit 1 within the Tx E3 LAPD Configuration Register. The bit-format of this register is presented below.

TRANSMIT E3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	X

The relationship between the contents of bit-fields 1 and the LAPD Message size is given in Table 69 .

TABLE 69: RELATIONSHIP BETWEEN TXLAPD MSG LENGTH AND THE LAPD MESSAGE SIZE

TXLAPD MESSAGE LENGTH	LAPD MESSAGE LENGTH
0	LAPD Message size is 76 bytes
1	LAPD Message size is 82 bytes

NOTE: *The Message Type selected must correspond with the contents of the first byte of the Information (Payload) portion, as presented in Table 68 .*

STEP 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 HDLC Control block 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 3 (Auto Retransmit) within the Tx E3 LAPD Configuration Register (Address = 0x33), as depicted below.

TXE3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
RO	RO	RO	RO	R/W	RO	R/W	R/W
0	0	0	0	1	0	0	0

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.

STEP 5 - Enabling the LAPD Transmitter

Prior to the transmission of any data via the LAPD Transmitter, the LAPD Transmitter must be enabled. This is accomplished by writing a "1" to bit 0 (TxLAPD Enable) of the Tx E3 LAPD Configuration Register, as depicted below.

TRANSMIT E3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	E/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	1

If the user writes a “0” into this bit-field, then the LAPD Transmitter will be enabled, and the LAPD Transmitter will immediately begin to transmit a continuous stream of Flag Sequence octets (0x7E), via the “N” bit-field of each outbound E3 frame.

Conversely, if the user writes a “1” into this bit-field, then the LAPD Transmitter will be disabled. The Transmit E3 Framer block will automatically insert a “1” into the “N” bit-field, within each outbound E3 frame. No transmission of PMDL data will occur.

STEP 7 - Initiate the Transmission

At this point, the user should have written the PMDL message into the on-chip Transmit LAPD Message buffer and the type of LAPD Message that is desired to be transmitted should have been specified. Finally, the user should have enabled the LAPD Transmitter. The only remaining to do is initiate the transmission of this message. This process is initiated by writing a “1” to Bit 3 (Tx DL Start) within the Tx E3 LAPD Status and Interrupt Register (Address = 0x34), as depicted below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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

A “0” to “1” transition in Bit 3 (Tx DL 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 “1’s” in the PMDL Message then the LAPD Transmitter will insert a “0” immediately following these strings of consecutive “1’s”. 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 octet (e.g., six consecutive “1’s”) or the ABORT Sequence octet (e.g., seven consecutive “1’s”). 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 bits and will begin to transport these bits via the N bit-field within each outbound E3 frame.

While the LAPD Transmitter is transmitting this LAPD Message frame, the TxDL Busy bit-field (Bit 2) within the Tx E3 LAPD Status and Interrupt Register, will be set to “1”. 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, then this bit-field will toggle back to “0”.

The user can configure the LAPD Transmitter to interrupt the local Microprocessor/Microcontroller upon completion of transmission of the LAPD Message frame, by setting bit-field “1” (TxLAPD Interrupt Enable) within the Tx E3 LAPD Status and Interrupt register (Address = 0x34). to “1” as depicted below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	1	X

'The purpose of t his interrupt is to let the Microprocessor/Microcontroller know that the LAPD Transmitter is available and ready to transmit a LAPD Message frame (which contains a new PMDL Message) to the remote terminal equipment. Bit 0 (Tx LAPD Interrupt Status) within the Tx E3 LAPD Status and Interrupt Register will reflect the status for the Transmit LAPD 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.

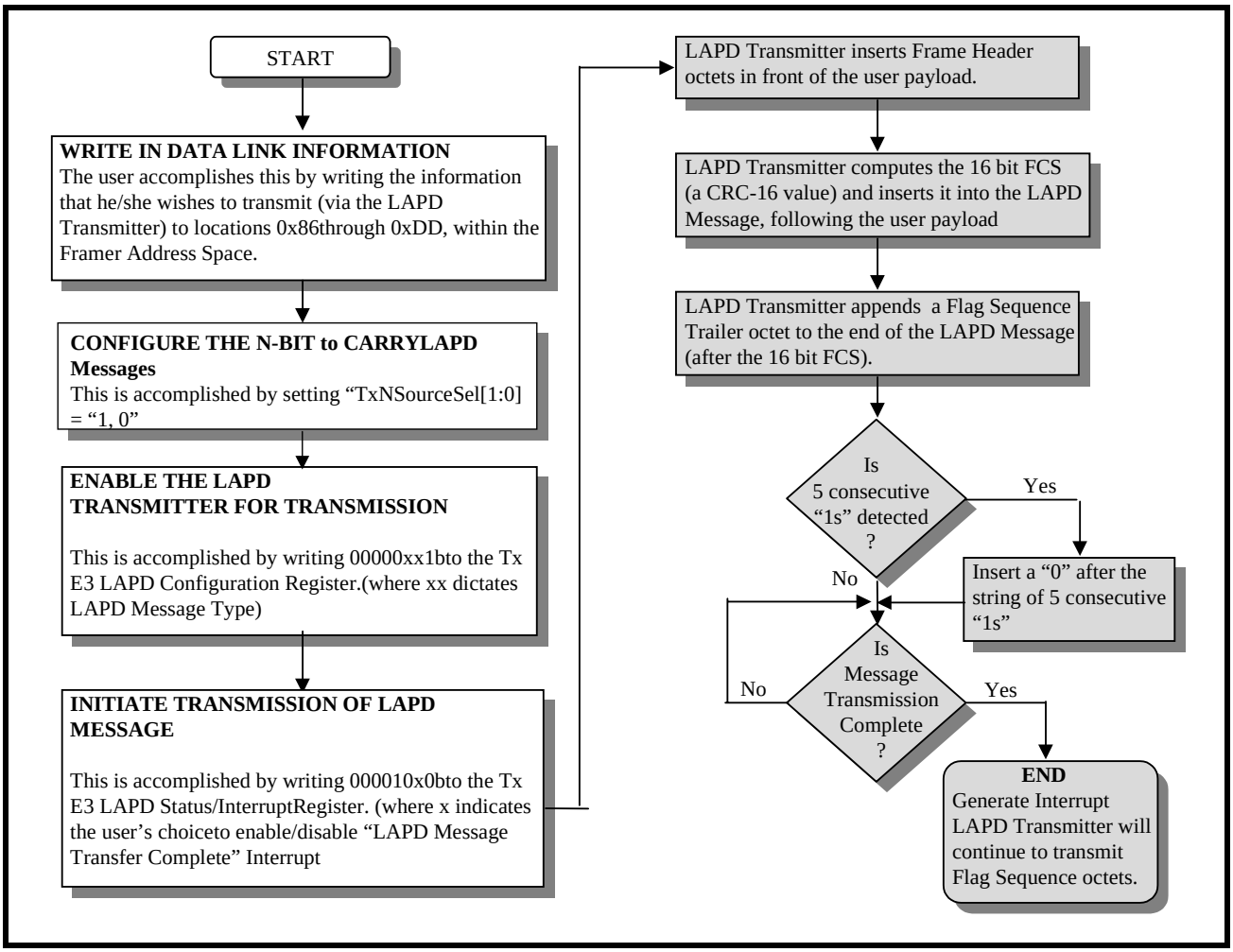
- Generate the four octets of the LAPD Message frame header (e.g., the Flag Sequence, SAPI, TEI, Control, etc.,) and insert them into the header byte positions within the LAPD Message frame.
- It will read in the contents of the Transmit LAPD Message buffer (e.g., the PMDL Message data) and insert it into the Information Payload portion of the LAPD Message frame.
- Compute the 16-bit Frame Check Sequence (FCS) value of the LAPD Message frame (e.g, of the LAPD Message header and Payload bytes) 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 octets).
- Fragment the resulting LAPD Message frame into bits and begin inserting these bits into the "N" bit-field within each outbound E3 frame.
- Complete the transmission of the overhead bytes, information payload byte, FCS value, and the trailing Flag Sequence octets via the Transmit E3 Framer block.

Once the LAPD Transmitter has completed its transmission of the LAPD Message frame, the Framer will generate an Interrupt to the MIcroprocessor/Microcontroller (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 transmissions of the LAPD Message frames, the LAPD Transmitter will be sending a 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 Data Link) within the Tx E3 Configuration register.

NOTE: *In order to prevent the user's data (e.g., the PMDL Message within the LAPD Message frame) from mimicking the Flag Sequence byte or an ABORT Sequence, the LAPD Transmitter will parse through the PMDL Message data and insert a "0" into this data, immediately following the detection of five (5) consecutive "1's" (this stuffing occurs while the PMDL message data is being read in from the Transmit LAPD Message frame. The Remote LAPD Receive (See Section 4.3.5) will have the responsibility of checking the newly received PMDL messages for a string of five (5) consecutive "1's" and removing the subsequent "0" from the payload portion of the incoming LAPD Message.*

Figure 125 presents a flow chart diagram. Figure 125 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 125. FLOW CHART DEPICTING HOW TO USE THE LAPD TRANSMITTER



NOTE: *In Figure 125 , the unshaded boxes depict the tasks that the user must perform. The shaded boxes present the resulting tasks that the Transmit HDLC Controller block will perform.*

The Mechanics of Transmitting a New LAPD Message frame, if the LAPD Transmitter has been configured to re-transmit the LAPD Message frame, repeatedly, at One-Second intervals.

If the LAPD Transmitter has been configured to re-transmit the LAPD Message frame 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 Remote Terminal Equipment.

If another (e.g., a different) PMDL Message is to be transmitted to the Remote Terminal Equipment, this new message will have to be written into the Transmit LAPD Message buffer, via the Microprocessor Interface block of the Framer IC. However, care must be taken when writing this new PMDL message. If this message is written 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 Remote Terminal Equipment. 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 Framer to automatically reset activated interrupts.

The user can do this by writing a "1" into Bit 3 within the Framer Operating Mode register (Address = 0x00), as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	1

This action will prevent the LAPD Transmitter from generating its own One-Second interrupt (following each transmission of the LAPD Message frame).

2. Enable the One-Second Interrupt

This can be done by writing a “1” into Bit 0 (One-Second Interrupt Enable) within the Block Interrupt Enable Register, as depicted below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	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 synchronizing 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 Remote LAPD Receiver.

6.2.4 The Transmit E3 Framer Block

6.2.4.1 Brief Description of the Transmit E3 Framer

The Transmit E3 Framer block accepts data from any of the following four sources, and uses it to form the E3 data stream.

- The Transmit Payload Data Input block
- The Transmit Overhead Data Input block
- The Transmit HDLC Controller block
- The Internal Overhead Data Generator

The manner in how the Transmit E3 Framer block handles data from each of these sources is described below.

Handling of data from the Transmit Payload Data Input Interface

For E3 applications, all data that is input to the Transmit Payload Data Input Interface will be inserted into the payload bit positions within the outbound E3 frames.

Handling of data from the Internal Overhead Bit Generator

By default, the Transmit E3 Framer block will internally generate the overhead bytes. However, if the Terminal Equipment inserts its own values for the overhead bits or bytes (via the Transmit Overhead Data Input Interface) or if the user enables and employs the Transmit E3 HDLC Controller block, then these internally generated overhead bytes will be overwritten.

Handling of data from the Transmit Overhead Data Input Interface

For E3 applications, the Transmit E3 Framer block automatically generates and inserts the framing alignment bytes (e.g., the 10 bit FAS framing alignment signal) into the outbound E3 frames. Hence, the Transmit E3 Framer block will not accept data from the Transmit OH Data Input Interface block for the FAS signal.

However, the Transmit E3 Framer block will accept (and insert) data from the Transmit Overhead Data Input Interface for both the “A” and “N” bit-fields.

If the user's local Data Link Equipment activates the Transmit Overhead Data Input Interface block and writes data into this interface for these bits or bytes, then the Transmit E3 Framer block will insert this data into the appropriate overhead bit/byte-fields, within the outbound E3 frames.

Handling of data from the Transmit HDLC Controller Block

The exact manner in how the Transmit E3 Framer handles data from the Transmit HDLC Controller block depends upon whether the Transmit HDLC Controller is activated or not. If the Transmit DS3 HDLC Controller block is not activated, then the Transmit E3 Framer block will insert a “1” into each “N” bit-field, within each outbound E3 frame.

If the Transmit E3 HDLC Controller block is activated, then data will be inserted into the “N” bit-fields as described in Section 4.2.3.

6.2.4.2 Detailed Functional Description of the Transmit E3 Framer Block

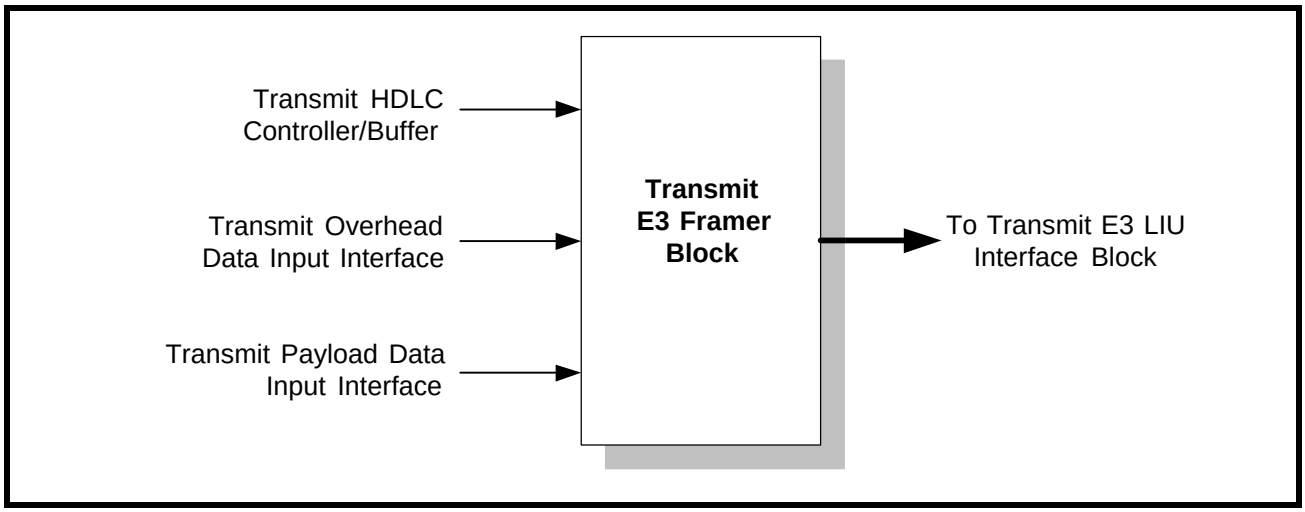
The Transmit E3 Framer receives data from the following three sources and combines them together to form the E3 data stream.

- The Transmit Payload Data Input Interface block.
- The Transmit Overhead Data Input Interface block
- The Transmit HDLC Controller block.
- The Internal Overhead Data Generator.

Afterwards, this E3 data stream will be routed to the Transmit E3 LIU Interface block, for further processing.

Figure 126 presents a simple illustration of the Transmit E3 Framer block, along with the associated paths to the other functional blocks within the chip.

FIGURE 126. THE TRANSMIT E3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO OTHER FUNCTIONAL BLOCKS



In addition to taking data from multiple sources and multiplexing them, in appropriate manner, to create the outbound E3 frames, the Transmit E3 Framer block has the following roles.

- Generating Alarm Conditions
- Generating Errored Frames (for testing purposes)
- Routing outbound E3 frames to the Transmit E3 LIU Interface block

Each of these additional roles are discussed below.

6.2.4.2.1 Generating Alarm Conditions

The Transmit E3 Framer block permits the user to, by writing the appropriate data into the on-chip registers, to override the data that is being written into the Transmit Payload Data and Overhead Data Input Interfaces and transmit the following alarm conditions.

- Generate the Yellow Alarms (or FERF indicators)
- Manipulate the A-bit, by forcing it to “0”.
- Generate the AIS Pattern
- Generate the LOS pattern
- Generate FERF (Yellow) Alarms, in response to detection of a Red Alarm condition (via the Receive Section of the XRT74L74).

The procedure and results of generating any of these alarm conditions is presented below.

The user can exercise each of these options by writing the appropriate data to the Tx E3 Configuration Register (Address = 0x30). The bit format of this register is presented below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit-fields 1 and 2 permit the user to transmit various alarm conditions to the remote terminal equipment. The role/function of each of these two bit-fields within the register, are discussed below.

6.2.4.2.1.1 Tx AIS Enable - Bit 2

This read/write bit field permits the user to force the transmission of an AIS (Alarm Indication Signal) pattern to the remote terminal equipment via software control. If the user opts to transmit an AIS pattern, then the Transmit Section of the Framer IC will begin to transmit an unframed all ones pattern to the remote terminal equipment. Table 70 presents the relationship between the contents of this bit-field, and the resulting Framer action.

TABLE 70: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TX AIS ENABLE) WITHIN THE TX E3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT E3 FRAMER BLOCK'S ACTION

BIT 2	TRANSMIT E3 FRAMER'S ACTION
0	**Normal Operation:** The Transmit Section of the XRT74L74 Framer IC will transmit E3 traffic based upon data that it accepts via the Transmit Payload Data Input Interface block, the Transmit Overhead Data Input Interface block, the Transmit HDLC Controller block and internally generated overhead bytes.
1	**Transmit AIS Pattern:** The Transmit E3 Framer block will overwrite the E3 traffic, within an Unframed "All Ones" pattern.

NOTE: *This bit is ignored whenever the TxLOS bit-field is set.*

6.2.4.2.1.2 Transmit LOS Enable - Bit 1

This read/write bit field allows the user to transmit an LOS (Loss of Signal) pattern to the remote terminal, upon software control. Table 71 relates the contents of this bit field to the Transmit E3 Framer block's action.

TABLE 71: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (TX LOS) WITHIN THE TX E3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT E3 FRAMER BLOCK'S ACTION

BIT 1	TRANSMIT E3 FRAMER'S ACTION
0	**Normal Operation:** The Overhead bits are either internally generated, or they are inserted via the Transmit Overhead Data Input Interface or the Transmit HDLC Controller blocks. The Payload bits are received from the Transmit Payload Data Input Interface.
1	**Transmit LOS Pattern:** When this command is invoked the Transmit E3 Framer will do the following. • Set all of the overhead bytes to "0" (including the FA1 and FA2 bytes) Overwrite the E3 payload bits with an "all zeros" pattern.

NOTE: *When this bit is set, it overrides all of the other bits in this register.*

6.2.4.2.1.3 Transmitting FERF (Far-End Receive Failure) Indicator or Yellow Alarm

The XRT74L74 Framer IC permits the user to control the state of the "A" bit-field, within each outbound E3 frame. This can be achieved by writing the appropriate data into the TxASource[1:0] bit-fields within the Tx E3 Configuration Register, as illustrated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	X	X	0	0	0	0	0

The following table presents the relationship between the contents of TxASource[1:0] and the resulting source of the "A" bit.

TxASourceSel[1:0]	Source of A Bit
00	TxE3 Service Bits Register (Address = 0x35)
01	Transmit Overhead Data Input Interface
10	Transmit Payload Data Input Interface
11	Functions as a FEBE (Far-End-Block Error) bit-field. This bit-field is set to "0", if the Near-End Receive Section (within this chip) detects no BIP-4 Errors within the incoming E3 frames. This bit-field is set to "1", if the Near-End Receive Section (within this chip) detects a BIP-4 Error within the incoming E3 frame.

Hence, if a Yellow Alarm condition needs to be transmitted to the Remote Terminal Equipment, this can be accomplished by executing the following steps.

STEP 1 - Write a “1” into Bit 1 (A Bit) within the Tx E3 Service Bits Register, as indicated below.

TXE3 SERVICE BITS REGISTER (ADDRESS = 0X35)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used						A Bit	N Bit
RO	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	1	0

STEP 2 - Write the value “00” into the TxASource[1:0] bit-fields within the Tx E3 Configuration Register, as indicated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	0	0	X	X	X	X	X

These two steps will cause the Transmit E3 Framer block to read in the contents of Bit 1 (within the Tx E3 Service Bit register) and insert it into the “A” bit-field within the outbound E3 data stream. Hence, the “A” bit will be set to “1”, which will be interpreted as an Alarm Condition, by the Remote Terminal Equipment.

6.2.4.2.2 Configuring the Transmit E3 Framer block to insert the BIP-4 nibble into each outbound E3 frame.

The XRT74L74 Framer IC permits the user to (1) configure the Transmit Section of the device to insert the BIP-4 value into each outbound E3 frame and (2) to configure the Receive Section of the device to compute and verify the BIP-4 value, within each inbound’ E3 frame.

These two configurations are accomplished by setting bit 7 (Tx BIP-4 Enable), within the Tx E3 Configuration Register, to “1”, as indicated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	X	X	X	X	X	X	X

Setting this bit-field to "1" accomplishes the following.

- It configures the Transmit E3 Framer block to compute the BIP-4 value of a given E3 frame, and insert in to the very last nibble, within the very next outbound E3 frame. (Hence, bits 1533 through 1536, within each E3 frame, will function as the BIP-4 value)
- It configures the Receive E3 Framer block to compute and verify the BIP-4 value of each incoming E3 frame.

6.2.4.2.3 Generating Errored E3 Frames

The Transmit E3 Framer block permits the user to insert errors into the framing and error detection overhead bites (e.g., the FAS pattern, and the BIP-4 nibble) of the outbound E3 data stream in order to support Remote Terminal Equipment testing. The user can exercise this option by writing data into any of the following registers.

- TxE3 FAS Error Mask Register - 0
- TxE3 FAS Error Mask Register - 1
- TxE3 BIP-4 Error Mask Register

Inserting Errors into the FAS pattern of the outbound' E3 frames.

The user can insert errors into the FAS pattern bits, of each outbound E3 frame, by writing the appropriate data into either the TxE3 FAS Error Mask Register - 0 or TxE3 FAS Error Mask Register - 1.

As the Transmit E3 Framer block formulates the outbound E3 frames, the contents of the FAS pattern bits are automatically XORed with the contents of these two registers. The results of this XOR operation is written back into the corresponding bit-field within the outbound E3 frame, and is transmitted to the Remote Terminal Equipment. Therefore, if the user does not wish to modify any of these bits, then these registers must contain all "0's" (the default value).

TXE3 FAS ERROR MASK REGISTER - 0 (ADDRESS = 0X48)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			TxFAS_Error_Mask_Upper[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

TXE3 FAS ERROR MASK REGISTER - 1 (ADDRESS = 0X49)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			TxFAS_Error_Mask_Lower[4:0]				
RO	RO	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	X	X	X

Inserting Errors into the BIP-4 nibble, within each outbound E3 frame.

The user can insert errors into the BIP-4 nibble, within each outbound E3 frame, by writing the appropriate data into the TxE3 BIP-4 Error Mask Register.

As the Transmit E3 Framer block formulates the outbound E3 frames, the contents of the BIP-4 bits are automatically XORed with the contents of this register. The results of this XOR operation is written back into the corresponding bit-field within the outbound E3 frame, and is transmitted to the Remote Terminal Equipment. Therefore, if the user does not wish to modify any of these bits, then this register must contain all “0’s” (the default value).

NOTE: *This register is only active if the XRT74L74 Framer IC has been configured to insert the BIP-4 nibble into each outbound E3 frame.*

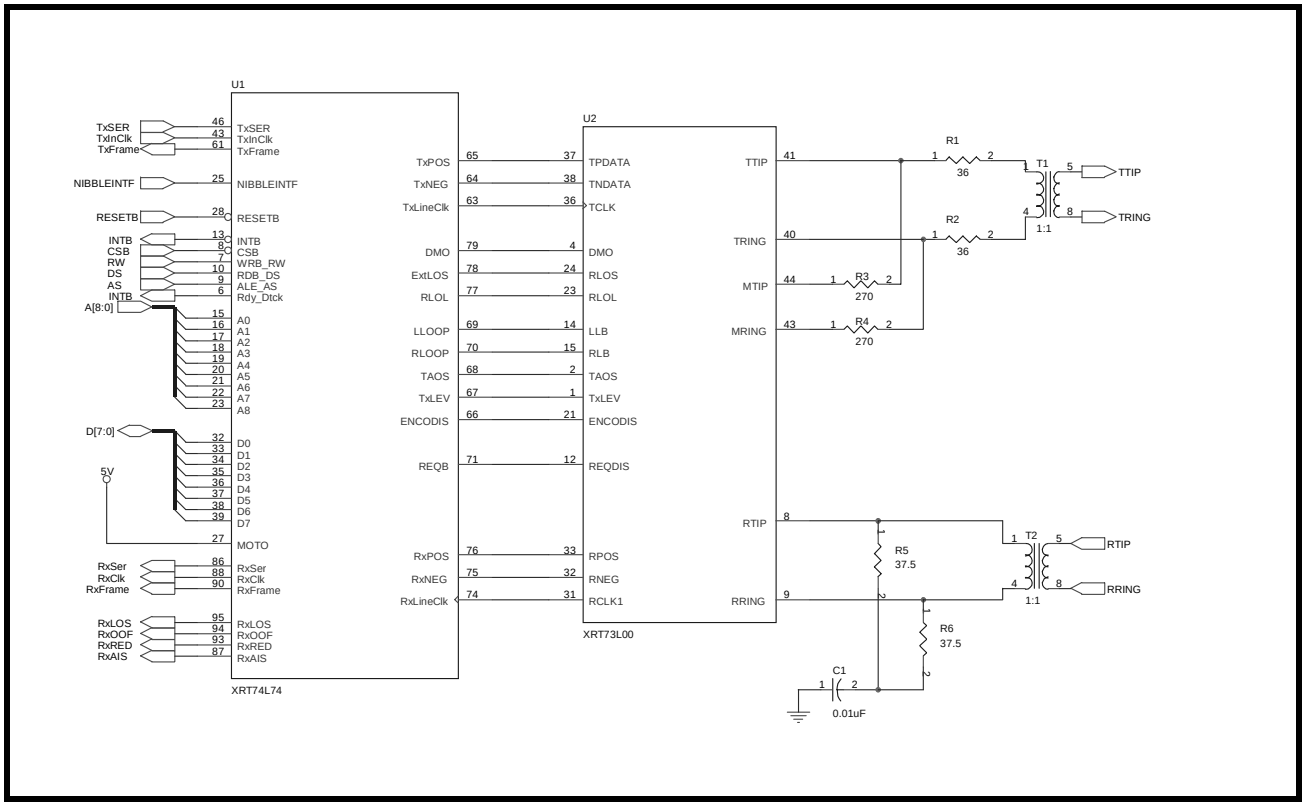
TXE3 BIP-4 ERROR MASK REGISTER (ADDRESS = 0X4A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				TxBIP-4 Mask[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

6.2.5 The Transmit E3 Line Interface Block

The XRT74L74 Framer IC is a digital device that takes E3 payload and overhead bit information from some terminal equipment, processes this data and ultimately, multiplexes this information into a series of outbound E3 frames. However, the XRT74L74 Framer IC lacks the current drive capability to be able to directly transmit this E3 data stream through some transformer-coupled coax cable with enough signal strength for it to be received by the remote receiver. Therefore, in order to get around this problem, the Framer IC requires the use of an LIU (Line Interface Unit) 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 127 presents a circuit drawing depicting the Framer IC interfacing to an LIU (XRT73L00 DS3/E3/STS-1 Transmit LIU).

FIGURE 127. APPROACH TO INTERFACING THE XRT74L74 FRAMER IC TO THE XRT73L00 DS3/E3/STS-1 LIU

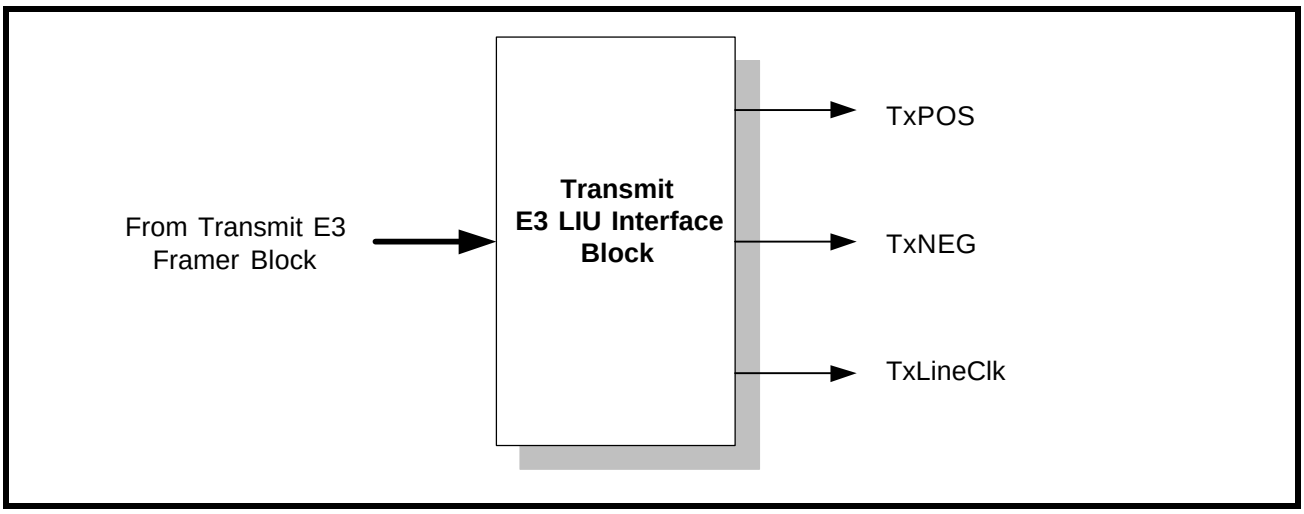


The Transmit Section of the XRT74L74 contains a block which is known as the Transmit E3 LIU Interface block. The purpose of the Transmit E3 LIU Interface block is to take the outbound E3 data stream, from the Transmit E3 Framer block, and to do the following:

1. Encode this data into one of the following line codes
   a. Unipolar (e.g., Single-Rail)
   b. AMI (Alternate Mark Inversion)
   c. HDB3 (High Density Bipolar - 3)
2. And to transmit this data to the LIU IC.

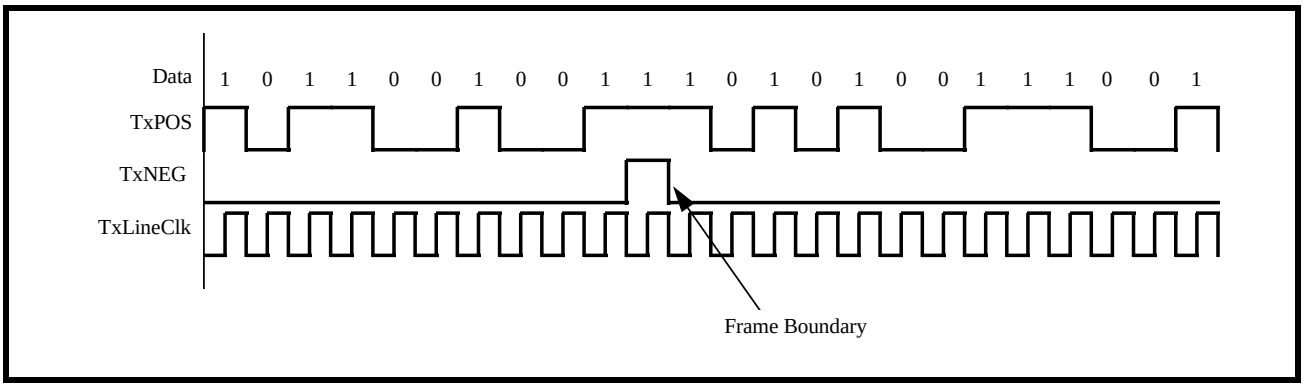
Figure 128 presents a simple illustration of the Transmit E3 LIU Interface block.

FIGURE 128. THE TRANSMIT E3 LIU INTERFACE BLOCK



The Transmit E3 LIU Interface block 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, in a binary (NRZ manner) data stream via the TxPOS pin to the LIU IC. 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. Figure 129 presents an illustration of the TxPOS and TxNEG signals during data transmission while the Transmit E3 LIU Interface block is operating in the Unipolar mode. This mode is sometimes referred to as Single Rail mode because the data pulses only exist in one polarity: positive.

FIGURE 129. THE BEHAVIOR OF TXPOS AND TXNEG SIGNALS DURING DATA TRANSMISSION WHILE THE TRANSMIT DS3 LIU INTERFACE IS OPERATING IN THE UNIPOLAR MODE



When the Transmit E3 LIU Interface block is operating in the Bipolar (or Dual Rail) mode, then the contents of the E3 Frame is output via both the TxPOS and TxNEG pins. If the Bipolar mode is chosen, then E3 data can be transmitted to the LIU 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. Bipolar mode is

sometimes referred to as Dual Rail because the data pulses occur in two polarities: positive and negative. The role of the TxPOS, TxNEG and TxLineClk output pins, for this mode are discussed below.

TxPOS - Transmit Positive Polarity Pulse: The Transmit E3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a positive polarity pulse to the remote terminal equipment.

TxNEG - Transmit Negative Polarity Pulse: The Transmit E3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a negative polarity pulse to the remote terminal equipment.

TxLineClk - Transmit Line Clock: The LIU IC uses this signal from the Transmit E3 LIU Interface block 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 remote Receive E3 Framer.

6.2.5.1 Selecting the various Line Codes

The user can select either the Unipolar Mode or Bipolar Mode by writing the appropriate value to Bit 3 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 72 relates the value of this bit field to the Transmit E3 LIU Interface Output Mode.

TABLE 72: THE RELATIONSHIP BETWEEN THE CONTENT OF BIT 3 (UNIPOLAR/BIPOLAR*) WITHIN THE UNI I/O CONTROL REGISTER AND THE TRANSMIT E3 FRAMER LINE INTERFACE OUTPUT MODE

BIT 3	TRANSMIT E3 FRAMER LIU INTERFACE OUTPUT MODE
0	**Bipolar Mode:** AMI or HDB3 Line Codes are Transmitted and Received
1	**Unipolar (Single Rail) Mode** of transmission and reception of E3 data is selected.

NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the operation of the Receive E3 LIU Interface block*

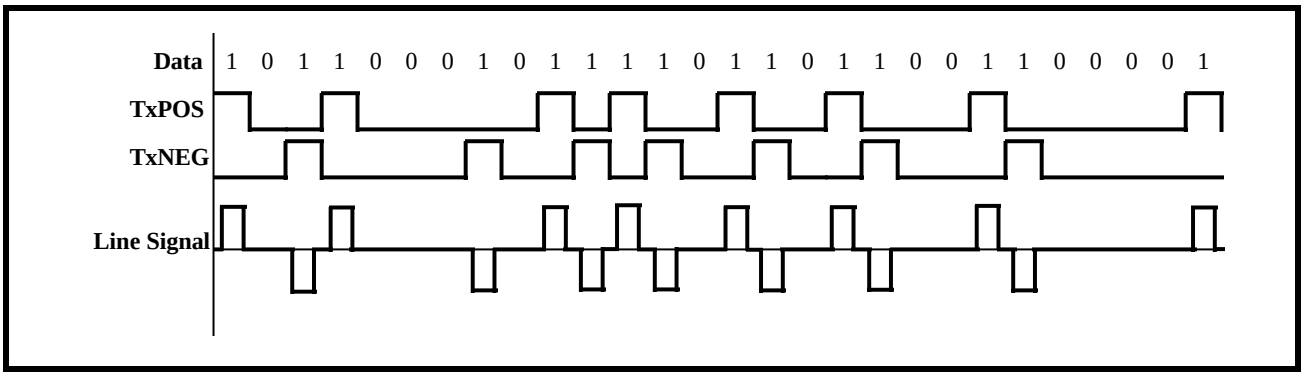
6.2.5.1.1 The Bipolar Mode Line Codes

If the Framer is choosen to operate in the Bipolar Mode, then the DS3 data-stream can be choosen to be transmitted via the AMI (Alternate Mark Inversion) or the HDB3 Line Codes. The definition of AMI and HDB3 line codes follow.

6.2.5.1.1.1 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, and -1. +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. Figure 130 presents an illustration of the AMI Line Code as would appear at the TxPOS and TxNEG pins of the Framer, as well as the output signal on the line.

FIGURE 130. AMI LINE CODE



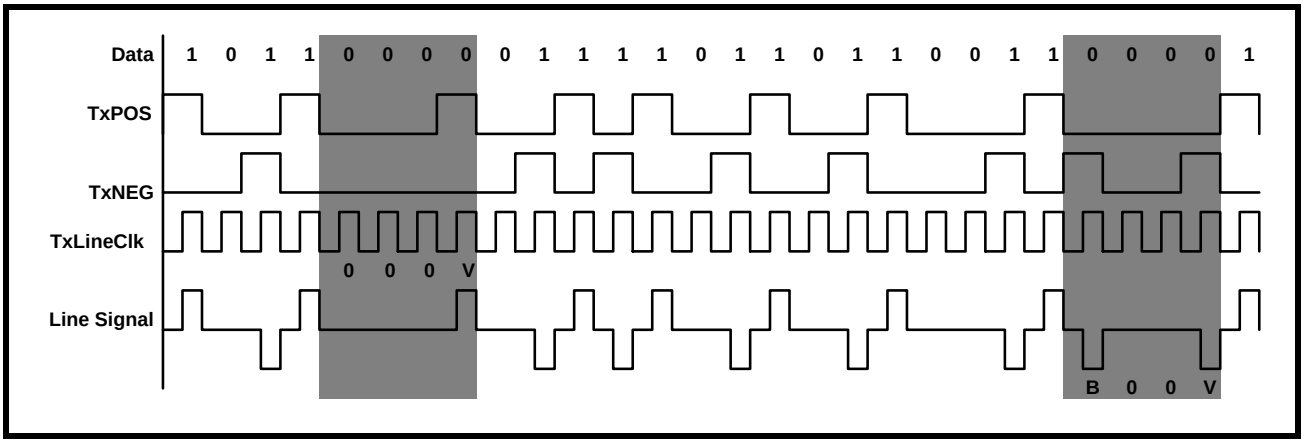
NOTE: *One of the main reasons that the AMI Line Code has been chosen for driving transformer-coupled media is that this line code introduces no dc component, thereby minimizing dc distortion in the line.*

6.2.5.1.1.2 The HDB3 Line Code

The Transmit E3 Framer and the associated LIU IC combine the data and timing information (originating from the TxLineClk signal) into the line signal that is transmitted to the remote receiver. The remote receiver has the task of recovering this data and timing information from the incoming E3 data stream. Many clock and data recovery schemes rely on the use of Phase Locked Loop technology. Phase-Locked-Loop (PLL) technology for clock recovery relies on transitions in the line signal, in order to maintain lock with the incoming E3 data stream. However, PLL-based clock recovery scheme, are vulnerable to the occurrence of a long stream of consecutive zeros (e.g., the absence of transitions). This scenario can cause the PLL to lose lock with the incoming E3 data, thereby causing the clock and data recovery process of the receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is HDB3 encoding. HDB3 (or High Density Bipolar - 3) is a form of AMI line coding that implements the following rule.

In general the HDB3 line code behaves just like AMI with the exception of the case when a long string of consecutive zeros occur on the line. 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. Figure 131 presents a timing diagram that illustrates examples of HDB3 encoding.

FIGURE 131. TWO EXAMPLES OF HDB3 ENCODING



The user chooses between AMI or HDB3 line coding by writing to bit 4 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 73 relates the content of this bit-field to the Bipolar Line Code that E3 Data will be transmitted and received at.

TABLE 73: THE RELATIONSHIP BETWEEN BIT 4 (AMI/ HDB3*) WITHIN THE I/O CONTROL REGISTER AND THE BIPOLAR LINE CODE THAT IS OUTPUT BY THE TRANSMIT E3 LIU INTERFACE BLOCK

BIT 4	BIPOLAR LINE CODE
0	HDB3
1	AMI

NOTES:

1. *This bit is ignored if the Unipolar mode is selected.*
2. *This selection also effects the operation of the Receive E3 LIU Interface block*

6.2.5.2 TxLineClk Clock Edge Selection

The Framer also allows the user to specify whether the E3 output data (via TxPOS and/or 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 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 74 relates the contents of this bit field to the clock edge of TxClk that E3 Data is output on the TxPOS and/or TxNEG output pins.

TABLE 74: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TXLINECLK INV) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG 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 132 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 133 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 132. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE RISING EDGE OF TXLINECLK

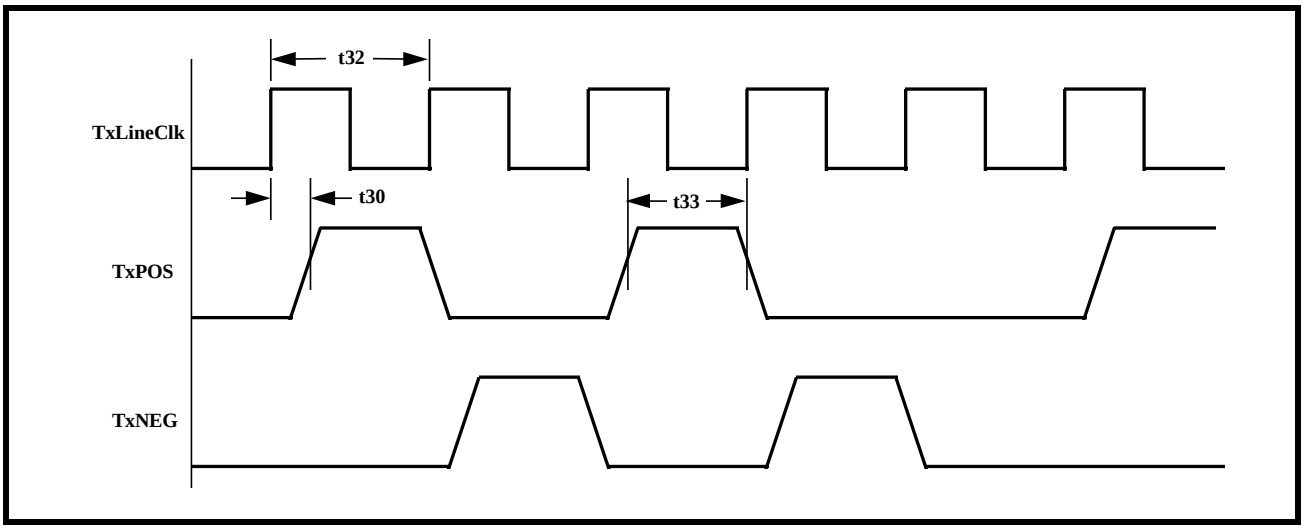
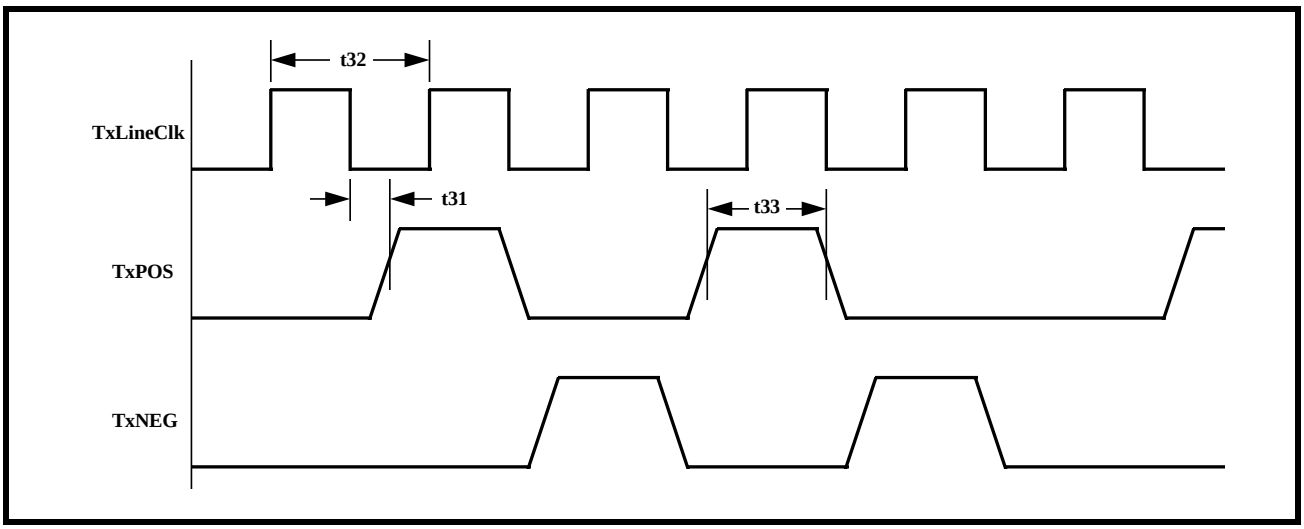


FIGURE 133. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE FALLING EDGE OF TXLINECLK



6.2.6 Transmit Section Interrupt Processing

The Transmit Section of the XRT74L74 can generate an interrupt to the Microprocessor/Microcontroller for the following reasons.

- Completion of Transmission of LAPD Message

6.2.6.1 Enabling Transmit Section Interrupts

The Interrupt Structure, within the XRT74L74 contains two hierarchical levels:

- Block Level
- Source Level

The Block Level

The Enable State of the Block Level for the Transmit Section Interrupts dictates whether or not interrupts (enabled) at the source level, are actually enabled.

The user can enable or disable these Transmit Section interrupts, at the Block Level by writing the appropriate data into Bit 1 (Tx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables the Transmit Section (at the Block Level) for Interrupt Generation. Conversely, setting this bit-field to “0” disables the Transmit Section for interrupt generation.

What does it mean for the Transmit Section Interrupts to be enabled or disabled at the Block Level?

If the Transmit Section is disabled (for interrupt generation) at the Block Level, then ALL Transmit Section interrupts are disabled, independent of the interrupt enable/disable state of the source level interrupts.

If the Transmit Section is enabled (for interrupt generation) at the block level, then a given interrupt will be enabled at the source level. Conversely, if the Transmit Section is enabled (for interrupt generation) at the Block level, then a given interrupt will still be disabled, if it is disabled at the source level.

As mentioned earlier, the Transmit Section of the XRT74L74 Framer IC contains the Completion of Transmission of LAPD Message Interrupt.

The Enabling/Disabling and Servicing of this interrupt is presented below.

6.2.6.1.1 The Completion of Transmission of the LAPD Message Interrupt

If the Transmit Section interrupts have been enabled at the Block level, then the user can enable or disable the Completion of Transmission of a LAPD Message Interrupt, by writing the appropriate value into Bit 1 (TxLAPD Interrupt Enable) within the Tx E3 LAPD Status & Interrupt Register (Address = 0x34), as illustrated below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	X	0

Setting this bit-field to “1’ enables the Completion of Transmission of a LAPD Message Interrupt. Conversely, setting this bit-field to “0” disables the Completion of Transmission of a LAPD Message interrupt.

6.2.6.1.2 Servicing the Completion of Transmission of a LAPD Message Interrupt

As mentioned previously, once the user commands the LAPD Transmitter to begin its transmission of a LAPD Message, it will do the following.

1. It will parse through the contents of the Transmit LAPD Message Buffer (located at address locations 0x86 through 0xDD) and search for a string of five (5) consecutive “1’s”. If the LAPD Transmitter finds a string of five consecutive “1’s” (within the content of the LAPD Message Buffer, then it will insert a “0” immediately after this string.
2. It will compute the FCS (Frame Check Sequence) value and append this value to the back-end of the user-message.
3. It will read out of the content of the user (zero-stuffed) message and will encapsulate this data into a LAPD Message frame.
4. Finally, it will begin transmitting the contents of this LAPD Message frame via the “N” bits, within each outbound E3 frame.
5. Once the LAPD Transmitter has completed its transmission of this LAPD Message frame (to the

Remote Terminal Equipment), the XRT74L74 Framer IC will generate the Completion of Transmission of a LAPD Message Interrupt to the Microcontroller/Microprocessor. Once the XRT74L74 Framer IC generates this interrupt, it will do the following.

- Assert the Interrupt Output pin (INT) by toggling it "Low".
- Set Bit 0 (TxLAPD Interrupt Status) within the TxE3 LAPD Status and Interrupt Register, to "1" as illustrated below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	1

The purpose of this interrupt is to alert the Microcontroller/MIcroprocessor that the LAPD Transmitter has completed its transmission of a given LAPD (or PMDL) Message, and is now ready to transmit the next PMDL Message, to the Remote Terminal Equipment.

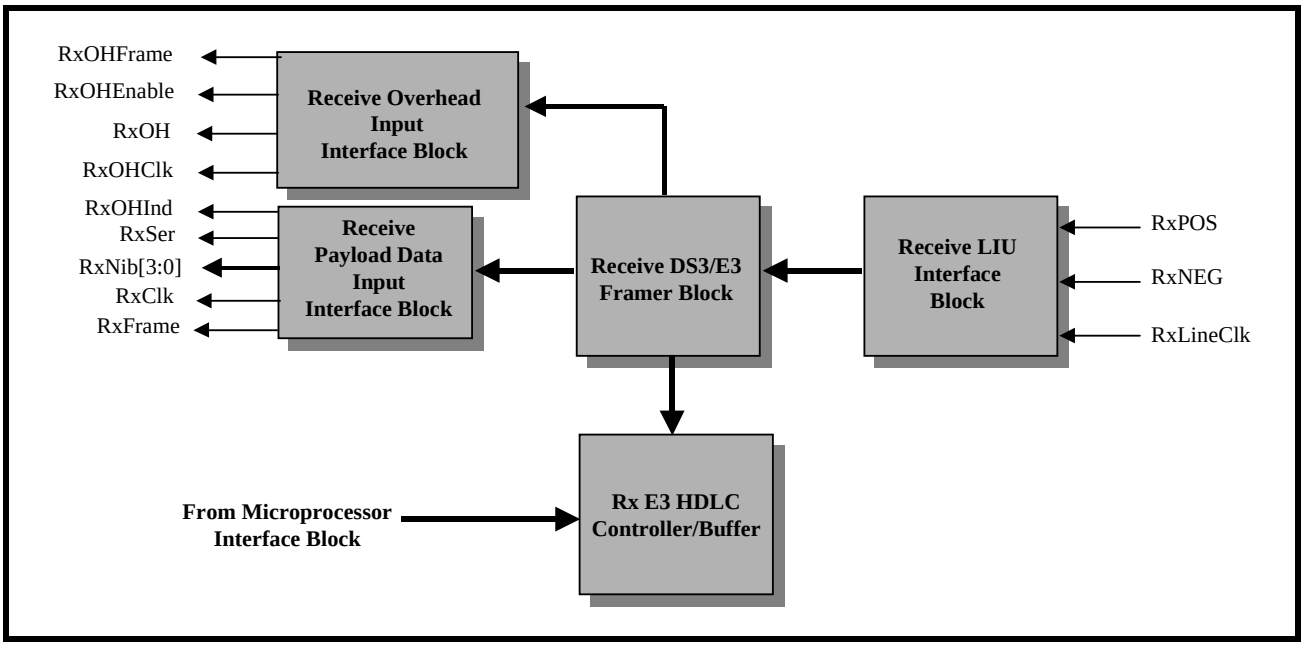
6.3 THE RECEIVE SECTION OF THE XRT74L74 (E3 MODE OPERATION)

When the XRT74L74 has been configured to operate in the E3 Mode, the Receive Section of the XRT74L74 consists of the following functional blocks.

- Receive LIU Interface block
- Receive HDLC Controller block
- Receive E3 Framer block
- Receive Overhead Data Output Interface block
- Receive Payload Data Output Interface block

Figure 134 presents a simple illustration of the Receive Section of the XRT74L74 Framer IC.

FIGURE 134. THE RECEIVE SECTION OF THE XRT74L74 CONFIGURED TO OPERATE IN THE E3 MODE



Each of these functional blocks will be discussed in detail in this document.

6.3.1 The Receive E3 LIU Interface Block

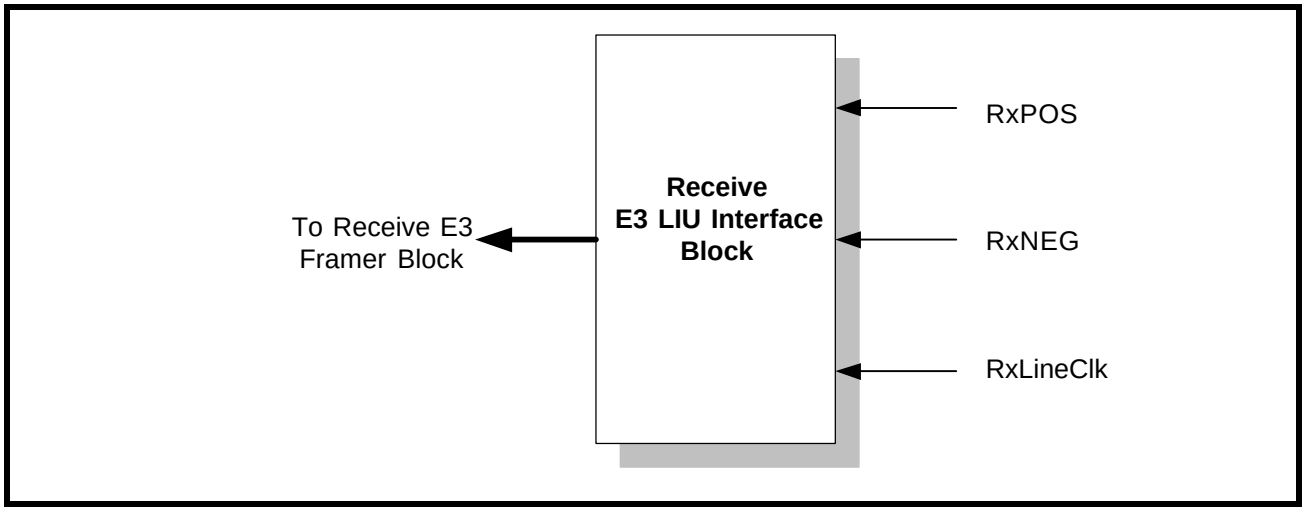
The purpose of the Receive E3 LIU Interface block is two-fold:

1. To receive encoded digital data from the E3 LIU IC.

2. To decode this data, convert it into a binary data stream and to route this data to the Receive E3 Framer block.

Figure 135 presents a simple illustration of the Receive E3 LIU Interface block.

FIGURE 135. THE RECEIVE E3 LIU INTERFACE BLOCK



The Receive Section of the XRT74L74 will via the Receive E3 LIU Interface Block receive 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 data information will be received via the RxPOS and RxNEG input pins. The Receive E3 LIU Interface block is capable of receiving E3 data pulses in unipolar or bipolar format. If the Receive E3 framer is operating in the bipolar format, 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.

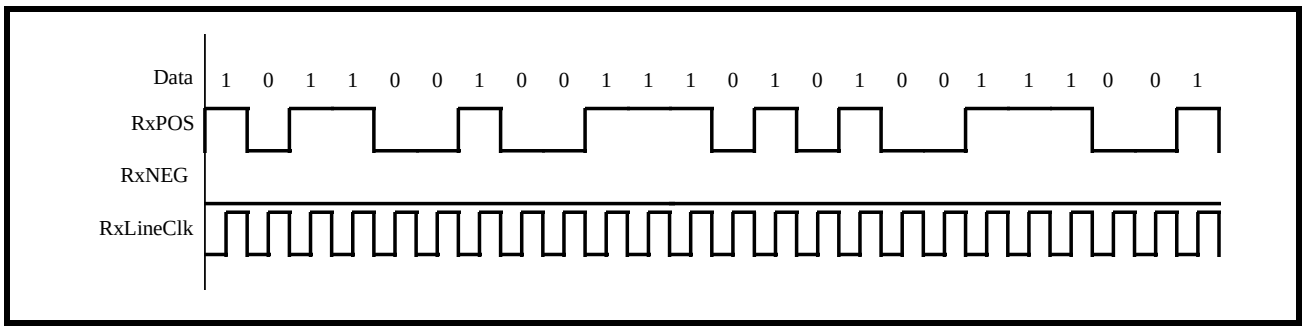
6.3.1.1 Unipolar Decoding

If the Receive E3 LIU Interface block is operating in the Unipolar (single-rail) mode, then it will receive the Single Rail NRZ DS3 data pulses via the RxPOS input pin. The Receive E3 LIU Interface block will also receive its timing signal via the RxLineClk signal.

NOTE: *The RxLineClk signal will function as the timing source for the entire Receive Section of the XRT74L74.*

No data pulses will be applied to the RxNEG input pin. The Receive E3 LIU Interface block receives a logic "1" when a logic "1" level signal is present at the RxPOS 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 136 presents an illustration of the behavior of the RxPOS, RxNEG and RxLineClk input pins when the Receive E3 LIU Interface block is operating in the Unipolar mode.

FIGURE 136. BEHAVIOR OF THE RXPOS, RXNEG AND RXLINECLK SIGNALS DURING DATA RECEPTION OF UNIPOLAR DATA



The user can configure the Receive E3 LIU Interface block to operate in either the Unipolar or the Bipolar Mode by writing the appropriate data to the I/O Control Register, as depicted below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 75 relates the value of this bit-field to the Receive E3 LIU Interface Input Mode.

TABLE 75: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TXLINECLK INV) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG ARE UPDATED ON

BIT 3	RECEIVE E3 LIU INTERFACE INPUT MODE
0	**Bipolar Mode (Dual Rail):** AMI or HDB3 Line Codes are Transmitted and Received.
1	**Unipolar Mode (Single Rail) Mode** of transmission and reception of E3 data is selected.

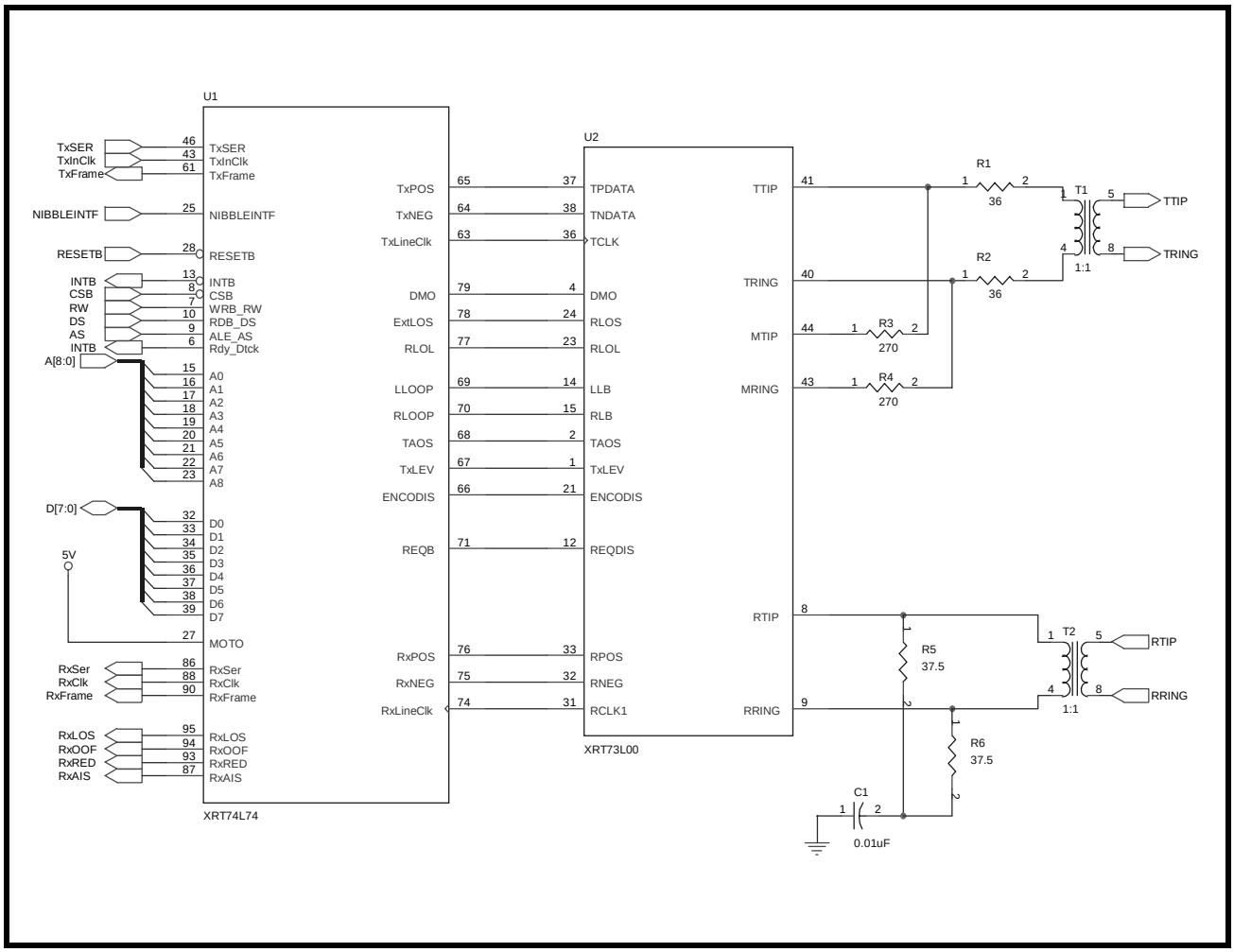
NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the Transmit E3 Framer Line Interface Output Mode*

6.3.1.2 Bipolar Decoding

If the Receive E3 LIU Interface block is operating in the Bipolar Mode, then it will receive the E3 data pulses via both the RxPOS, RxNEG, and the RxLineClk input pins. Figure 137 presents a circuit diagram illustrating how the Receive E3 LIU Interface block interfaces to the Line Interface Unit while the Framer is operating in Bipolar mode. The Receive E3 LIU Interface block can be configured to decode either the AMI or HDB3 line codes.

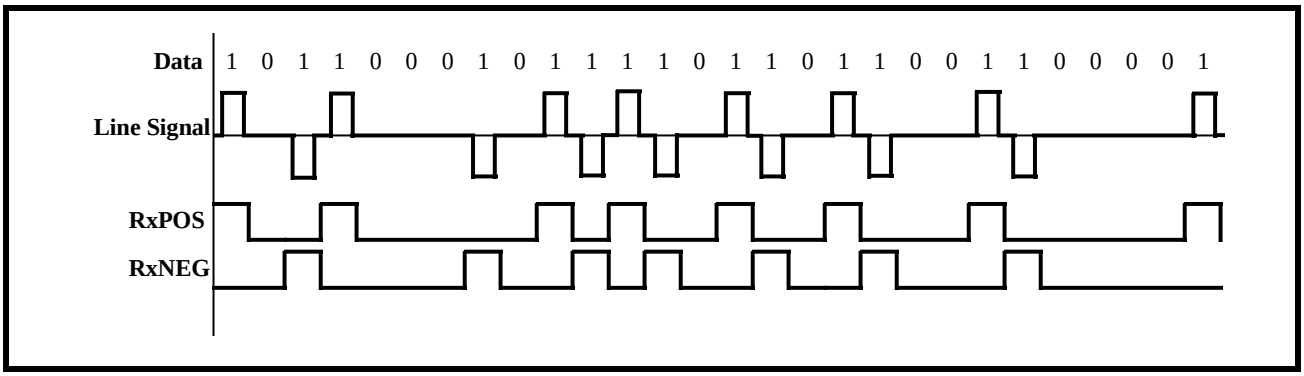
FIGURE 137. ILLUSTRATION ON HOW A CHANNEL OF THE RECEIVE E3 FRAMER (WITHIN THE XRT74L74 FRAMER IC) BEING INTERFACE TO THEXRT73L00 LINE INTERFACE UNIT, WHILE OPERATING IN BIPOLAR MODE



6.3.1.2.1 AMI Decoding

AMI or Alternate Mark Inversion, means that consecutive "one's" pulses (or marks) 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 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 exist between these two pulses. Figure 138 presents an illustration of the AMI Line Code as would appear at the RxPOS and RxNEG pins of the Framer, as well as the output signal on the line.

FIGURE 138. AMI LINE CODE



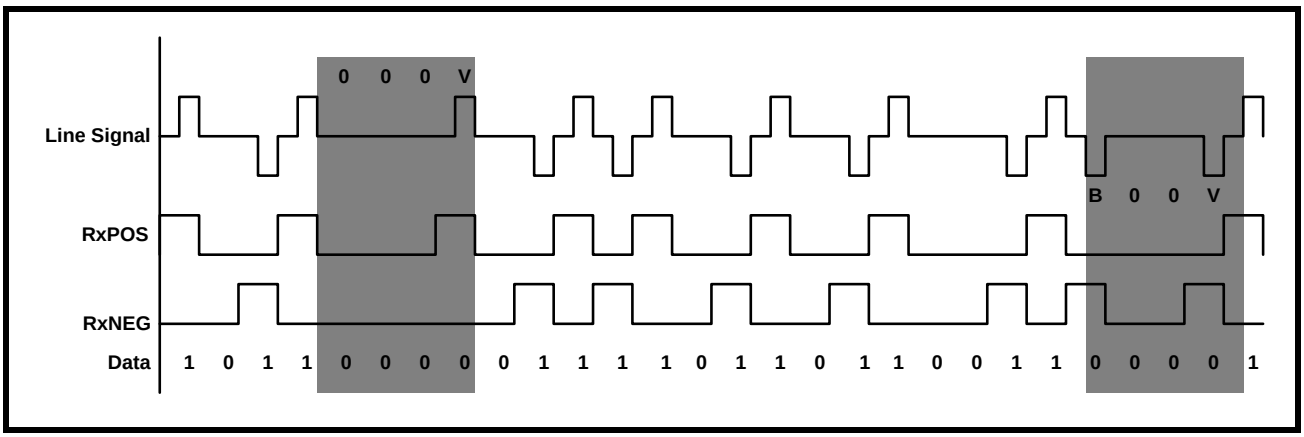
NOTE: One of the reasons that the AMI Line Code has been chosen for driving copper medium, isolated via transformers, is that this line code has no dc component, thereby eliminating dc distortion in the line.

6.3.1.2.2 HDB3 Decoding

The Transmit E3 LIU Interface block and the associated LIU embed and combine the data and clocking information into the line signal that is transmitted to the remote terminal equipment. The remote terminal equipment has the task of recovering this data and timing information from the incoming E3 data stream. Most clock and data recovery schemes rely on the use of Phase-Locked-Loop technology. One of the problems of using Phase-Locked-Loop (PLL) technology for clock recovery is that it relies on transitions in the line signal, in order to maintain lock with the incoming E3 data-stream. Therefore, these clock recovery scheme, 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 receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is HDB3 (or High Density Bipolar -3) 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. Any 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 LIU Interface block, when operating with the HDB3 Line Code is responsible for decoding the HD-encoded data back into a unipolar (binary-format). For instance, if the Receive E3 LIU Interface block detects a "000V" or a "B00V" pattern in the incoming pattern, the Receive E3 LIU Interface block will replace it with four (4) consecutive zeros. Figure 139 presents a timing diagram that illustrates examples of HDB3 decoding.

FIGURE 139. TWO EXAMPLES OF HDB3 DECODING



6.3.1.2.3 Line Code Violations

The Receive E3 LIU Interface block will also check the incoming E3 data stream for line code violations. For example, when the Receive E3 LIU Interface block detects a valid bipolar violation (e.g., in HDB3 line code), it will substitute four zeros into the binary data stream. However, if the bipolar violation is invalid, then an LCV (Line Code Violation) is flagged and the PMON LCV Event Count Register (Address = 0x50 and 0x51) will also be incremented. Additionally, the LCV-One-Second Accumulation Registers (Address = 0x6E and 0x6F) will be incremented. For example: If the incoming E3 data is HDB3 encoded, the Receive E3 LIU Interface block will also increment the LCV One-Second Accumulation Register if three (or more) consecutive zeros are received.

6.3.1.2.4 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 LIU Interface block via the RxLineClk signal. The Framer IC allows the user to specify which edge (e.g, rising or falling) of the RxLineClk signal will sample and latch the signal at the RxPOS and RxNEG input signals into the Framer IC. The user can make this selection by writing the appropriate data to bit 1 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 76 depicts the relationship between the value of this bit-field to the sampling clock edge of RxLineClk.

TABLE 76: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (RXLINECLK INV) OF THE I/O CONTROL REGISTER, AND THE SAMPLING EDGE OF THE RXLINECLK SIGNAL

RXCLKINV (BIT 1)	RESULT
0	**.Rising Edge:** RxPOS and RxNEG are sampled at the rising edge of RxLineClk. See Figure 140 for timing relationship between RxLineClk, RxPOS, and RxNEG.
1	**Falling Edge:** RxPOS and RxNEG are sampled at the falling edge of RxLineClk. See Figure 141 for timing relationship between RxLineClk, RxPOS, and RxNEG.

Figure 140 and Figure 141 present the Waveform and Timing Relationships between RxLineClk, RxPOS and RxNEG for each of these configurations.

FIGURE 140. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE RISING EDGE OF RXLINECLK

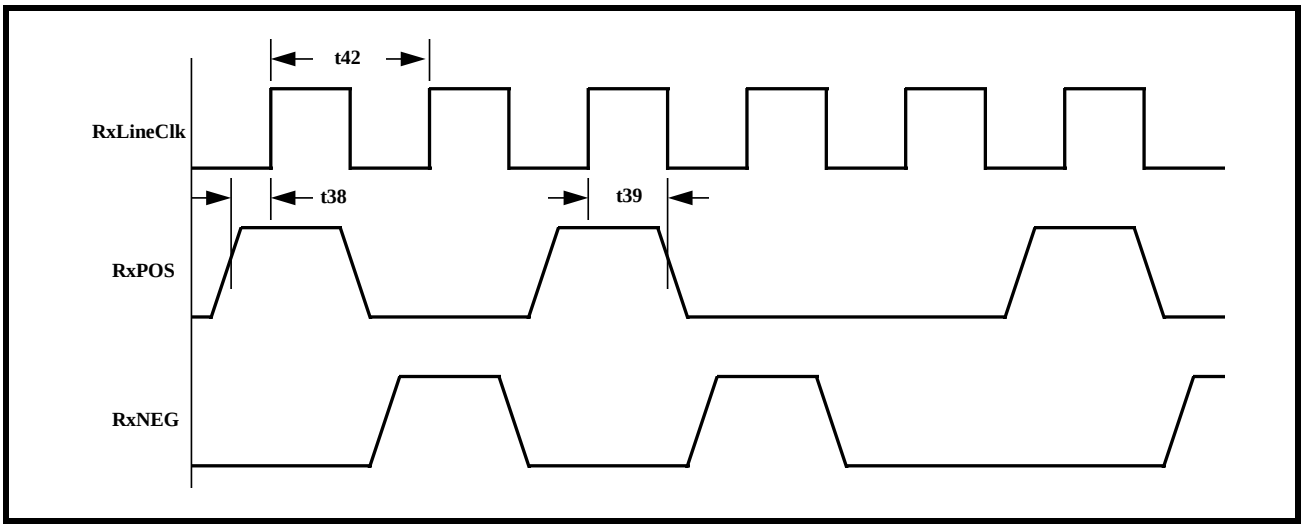
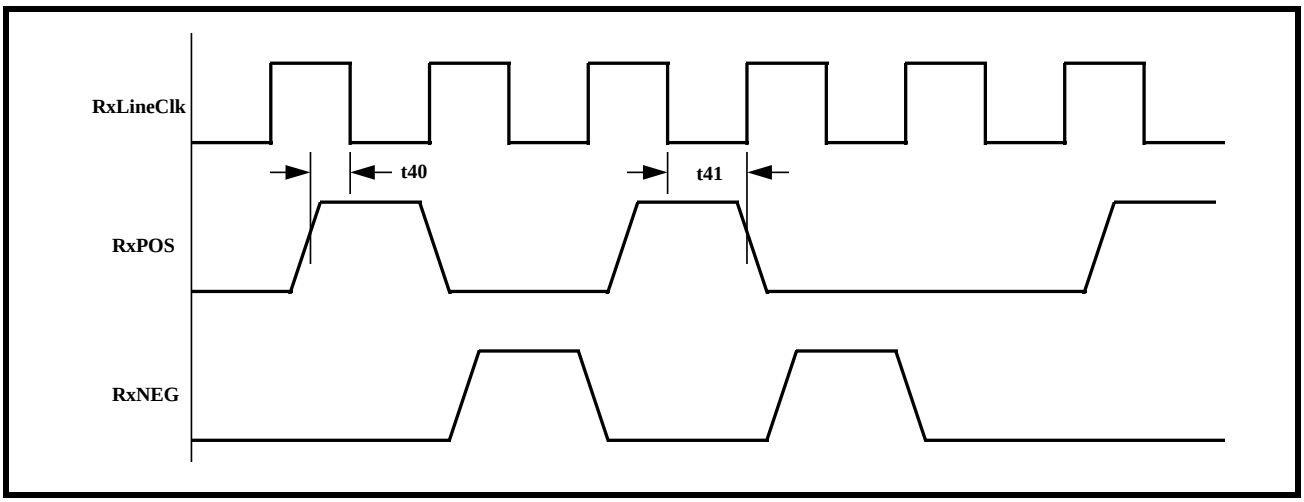


FIGURE 141. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE FALLING EDGE OF RXLINECLK



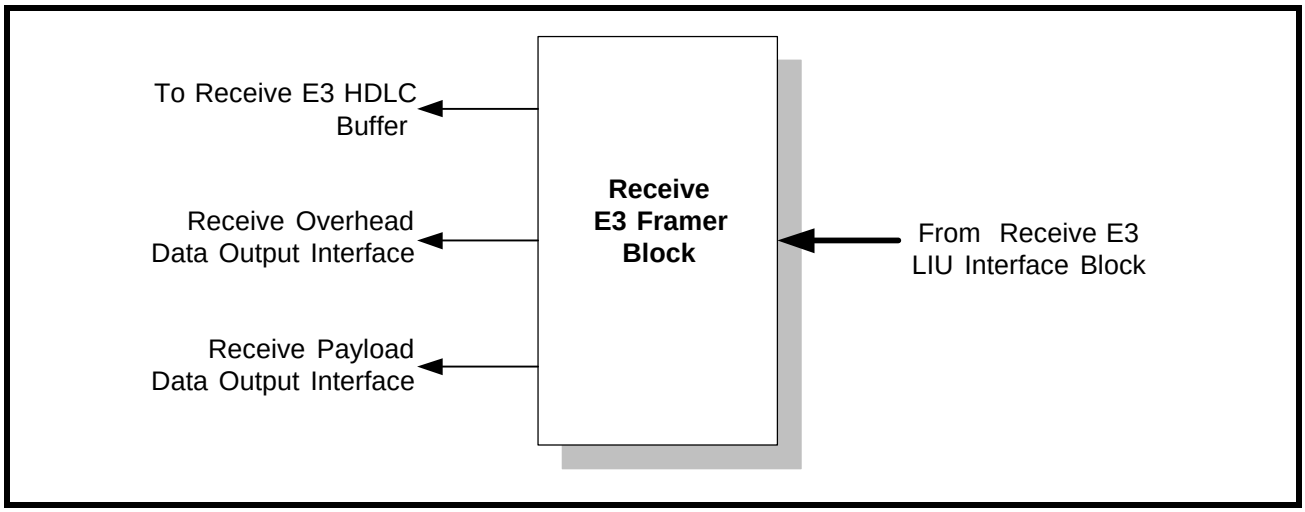
6.3.2 The Receive E3 Framer Block

The Receive E3 Framer block accepts decoded E3 data from the Receive E3 LIU Interface block, and routes data to the following destinations.

- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block.
- The Receive E3 HDLC Controller Block

Figure 142 presents a simple illustration of the Receive E3 Framer block along with the associated paths to the other functional blocks within the Framer chip.

FIGURE 142. THE RECEIVE E3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO THE OTHER FUNCTIONAL BLOCKS



Once the HDB3 (or AMI) encoded data has been decoded into a binary data-stream, the Receive E3 Framer block will use portions of this data-stream in order to synchronize itself to the remote terminal equipment. At any given time, the Receive E3 Framer block will be operating in one of two modes.

- **The Frame Acquisition Mode:** In this mode, the Receive E3 Framer block is trying to acquire synchronization with the incoming E3 frame, or
- **The Frame Maintenance Mode:** In this mode, the Receive E3 Framer block is trying to maintain frame synchronization with the incoming E3 Frames.

Figure 143 presents a State Machine diagram that depicts the Receive E3 Framer block's E3/ITU-T G.751 Frame Acquisition/Maintenance Algorithm.

6.3.2.1 The Framing Acquisition Mode

The Receive E3 Framer block 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 143 .

- FAS Pattern Search State
- FAS Pattern 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 FAS Pattern Search State

When the Receive E3 Framer block is first powered up, it will be operating in the FAS Pattern Search state. While the Receive E3 Framer is operating in this state, it will be performing a bit-by-bit search for the FAS (Framing Alignment Signal) pattern, of "1111010000". Figure 144 , which presents an illustration of the E3, ITU-T G.751 Framing Format, indicates that this framing alignment signal will occur at the beginning of each E3 frame.

FIGURE 143. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER E3 FRAME ACQUISITION/MAINTENANCE ALGORITHM

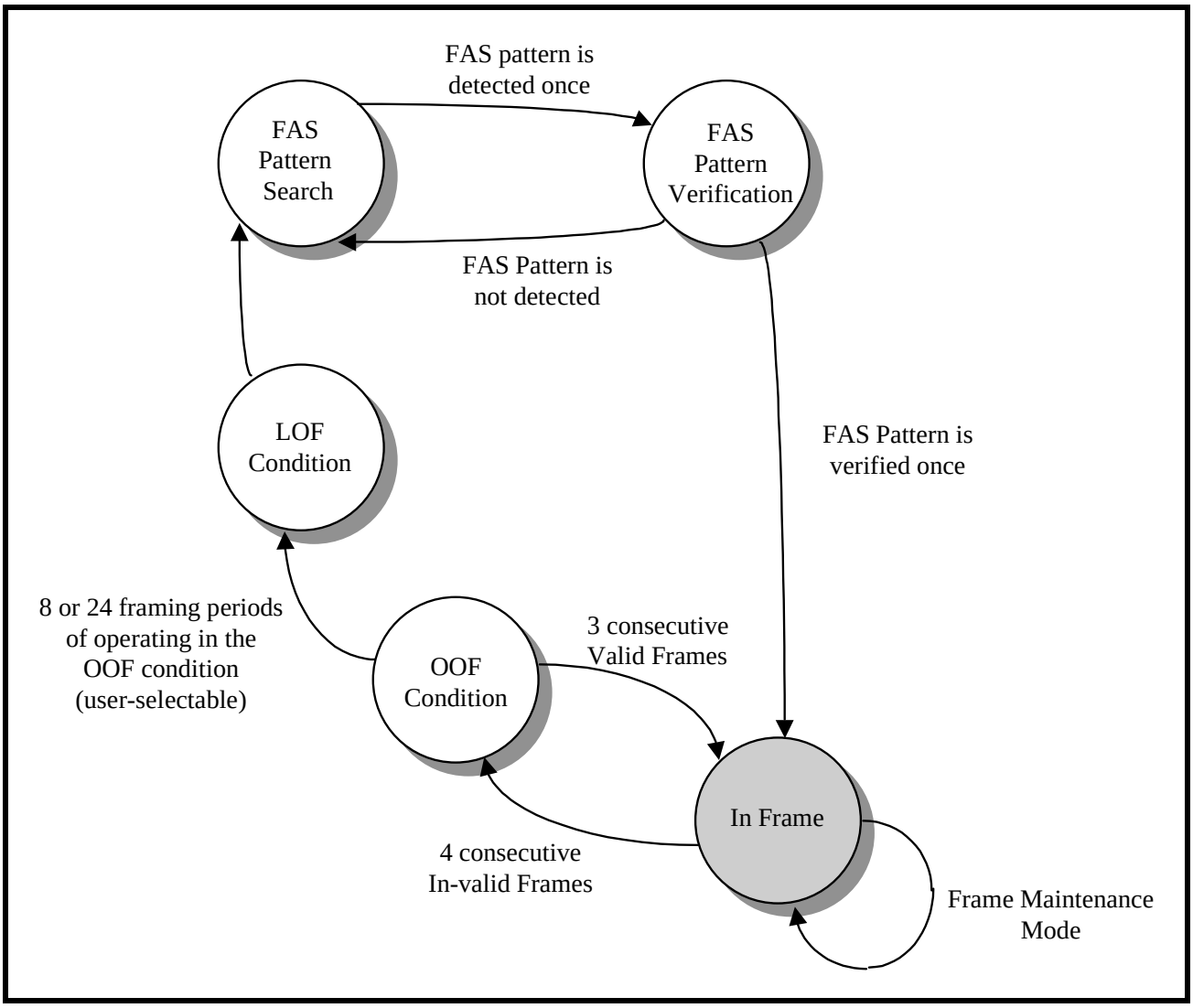
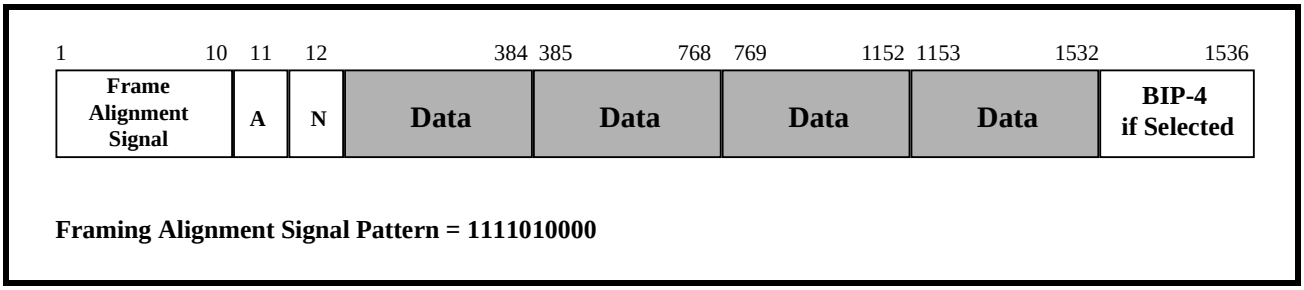


FIGURE 144. THE E3, ITU-T G.751 FRAMING FORMAT



When the Receive E3 Framer block detects the FAS pattern, it will then transition over to the FAS Pattern Verification state, per Figure 144 .

The FAS Pattern Verification State

Once the Receive E3 Framer block has detected an "1111010000" pattern, it must verify that this pattern is indeed the FAS pattern and not some other set of bits, within the E3 frame, mimicking the FAS Pattern.

Hence, the purpose of the FAS Pattern Verification state.

When the Receive E3 Framer block enters this state, it will then quit performing its bit-by-bit search for the Frame Alignment Signaling bits. Instead, the Receive E3 Framer block will read in the 10 bits that occur 1536 bit (e.g., one E3 frame period later) after the candidate FAS pattern was first detected. If these ten bits match the assigned values for the FAS Pattern octets, then the Receive E3 Framer block will conclude that it has found the FAS pattern and will then transition to the In-Frame state. However, if these two bytes do not match the assigned values for the FAS pattern then the Receive E3 Framer block will concluded that it has been fooled by data mimicking the Frame Alignment bytes, and will transition back to the FAS Pattern Search state.

In Frame State

Once the Receive E3 Framer block 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 block, while operating in the In-Frame state, can be found in Section 4.3.2.2 (The Framing Maintenance Mode).

OOF (Out of Frame) Condition State

If the Receive E3 Framer while operating in the In-Frame state detects four (4) consecutive frames, which do not have the valid Frame Alignment Signaling (FAS) patterns, then it will transition into the OOF Condition State. The Receive E3 Framer block's operation, while in the OOF condition state is a unique mix of Framing Maintenance and Framing Acquisition operation. The Receive E3 Framer block will exhibit some Framing Acquisition characteristics by attempting to locate (once again) the FAS pattern. However, the Receive E3 Framer block will also exhibit some Frame Maintenance behavior by still using the most recent frame synchronization for its overhead bits and payload bits processing.

The Receive E3 Framer block will inform the Microprocessor/Microcontroller 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.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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

The Receive E3 Framer block 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 block is capable of finding the FAS pattern within a user-selectable number of E3 frame periods, then it will transition back into the In-Frame state. The Receive E3 Framer block will then inform the Microprocessor/Microcontroller of its transition back into the In-Frame state by generating the Change in OOF Condition Interrupt.

However, if the Receive E3 Framer block 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 (Loss of Frame) Condition state.

The user can select this user-selectable number of E3 frame periods that the Receive E3 Framer block will remain in the OOF Condition state by writing the appropriate value into Bit 7 (RxLOF Algo) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

Writing a “0” into this bit-field causes the Receive E3 Framer block to reside in the OOF Condition state for at most 24 E3 frame periods. Writing a “1” into this bit-field causes the Receive E3 Framer block to reside in the OOF Condition state for at most 8 E3 frame periods.

LOF (Loss of Framing) Condition State

If the Receive E3 Framer block enters the LOF Condition state, then the following things will happen.

- The Receive E3 Framer block will discard the most recent frame synchronization and,
- The Receive E3 Framer block will make an unconditional transition to the FAS Pattern Search state.
- The Receive E3 Framer block will notify the Microprocessor/Microcontroller 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.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0	0

Finally, the Receive E3 Framer block will also inform the external circuitry of this transition to the LOF Condition state by toggling the RxLOF output pin "High”.

6.3.2.2 The Framing Maintenance Mode

Once the Receive E3 Framer block enters the In-Frame state, then it will notify the Microprocessor/Microcontroller of this fact by generating both the Change in OOF Condition and Change in LOF Condition Interrupts. When this happens, bits 2 and 3 (LOF Interrupt Status and OOF Interrupt Status) will be set to “1”, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	1	0	0

Additionally, the Receive E3 Framer block will inform the external circuitry of its transition to the In-Frame state by toggling both the RxOOF and RxLOF output pins "Low”.

Finally, the Receive E3 Framer block will negate both the RxOOF and the RxLOF bit-fields within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	1	1	1

When the Receive E3 Framer block 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 signal (FAS pattern) is present, and at its proper location. While the Receive E3 Framer block is operating in the Frame Maintenance Mode, it will declare an Out-of-Frame (OOF) Condition if it detects an invalid FAS pattern in four consecutive frames.

Since the Receive E3 Framer block requires the detection of an invalid FAS pattern in four consecutive frames, in order for it to transition to the OOF Condition state, it can tolerate some errors in the Framing Alignment bytes, and still remain in the In-Frame state. However, each time the Receive E3 Framer block detects an error in the FAS pattern, it will increment the PMON Framing Error Event Count Registers (Address = 0x52 and 0x53). The bit-format for these two registers are depicted below.

PMON FRAMING BIT/BYTE ERROR COUNT REGISTER - MSB (ADDRESS = 0X52)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framing Bit/Byte Error Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FRAMING BIT/BYTE ERROR COUNT REGISTER - LSB (ADDRESS = 0X53)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framing Bit/Byte Error Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

6.3.2.3 Forcing a Reframe via Software Command

The XRT74L74 Framer IC permits the user to command a reframe procedure with the Receive E3 Framer block via software command. If the user writes a “1” into Bit 0 (Reframe) within the I/O Control Register (Address = 0x01), as depicted below, then the Receive E3 Framer block will be forced into the FAS Pattern Search state, per Figure 145 ., and will begin its search for the FAS Pattern.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ ZeroSup*	Unipolar/ Bipolar*	TxLine Clk Invert	RxLine Clk Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	1

The Framer IC will 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 Microprocessor.
3. Asserting both the RxLOF and RxOOF bit-fields within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	0

6.3.2.4 Performance Monitoring of the Frame Synchronization Section, within the Receive E3 Framer block

The user can monitor the number of FAS pattern errors that have been detected by the Receive E3 Framer block. This is accomplished by periodically reading the PMON Framing Bit/Byte Error Event Count Registers (Address = 0x52 and 0x53). The byte format of these registers are presented below.

6.3.2.5 The RxOOF and RxLOF output pin.

The user can roughly determine the current framing state that the Receive E3 Framer block is operating in by reading the logic state of the RxOOF and the RxLOF output pins. Table 77 presents the relationship between the state of the RxOOF and RxLOF output pins, and the Framing State of the Receive E3 Framer block.

TABLE 77: THE RELATIONSHIP BETWEEN THE LOGIC STATE OF THE RXOOF AND RXLOF OUTPUT PINS, AND THE FRAMING STATE OF THE RECEIVE E3 FRAMER BLOCK

RXLOF	RXOOF	FRAMING STATE OF THE RECEIVE E3 FRAMER BLOCK
0	0	In Frame
0	1	OOF Condition (The Receive E3 Framer block is operating in the 3ms OOF period).
1	0	Invalid
1	1	LOF Condition

6.3.2.6 E3 Receive Alarms

6.3.2.7 The Loss of Signal (LOS) Alarm

Declaring an LOS Condition

The Receive E3 Framer block will declare a Loss of Signal (LOS) Condition, when it detects 32 consecutive incoming “0’s” via the RxPOS and RxNEG input pins or if the ExtLOS input pin (from the XRT7300 DS3/E3/STS-1 LIU IC) is asserted. The Receive E3 Framer block will indicate that it is declaring an LOS condition by.

- Asserting the RxLOS output pin (e.g., toggling it "High").
- Setting Bit 4 (RxLOS) of the Rx E3 Configuration & Status Register to “1” as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	1	0	0	0	0

- The Receive E3 Framer block will generate a Change in LOS Condition interrupt request. Upon generating this interrupt request, the Receive E3 Framer block will assert Bit 1 (LOS Interrupt Status within the Rx E3 Framer Interrupt Status Register - 1, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Clearing the LOS Condition

The Receive E3 Framer block will clear 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 block clears the LOS condition, then it will notify the Microprocessor and the external circuitry of this occurrence by:

- Generating the Change in LOS Condition Interrupt to the Microprocessor.
- Clearing Bit 4 (RxLOS) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

- Clear the RxLOS output pin (e.g., toggle it "Low").

6.3.2.8 The AIS (Alarm Indication Status) Condition

Declaring the AIS Condition

The Receive E3 Framer block 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 block will declare an AIS Condition if 7 or less “0’s” are detected in each of 2 consecutive E3 frames.

If the Receive E3 Framer block declares an AIS Condition, then it will do the following.

- Generate the Change in AIS Condition Interrupt to the Microprocessor. Hence, the Receive E3 Framer block will assert Bit 0 (AIS Interrupt Status) within the Rx E3 Framer Interrupt Status register - 1, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1

- Assert the RxAIS output pin.
- Set Bit 3 (RxAIS) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	1	1	1	1

Clearing the AIS Condition

The Receive E3 Framer block will clear the AIS condition when it detects two consecutive E3 frames, with eight or more “zeros” in the incoming data stream. The Receive E3 Framer block will inform the Microprocessor that the AIS Condition has been cleared by:

- Generating the Change in AIS Condition Interrupt to the Microprocessor. Hence, the Receive E3 Framer block will assert Bit 0 (AIS Interrupt Status) within the Rx E3 Framer Interrupt Status Register - 1.
- Clearing 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.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	1	0	0	X

6.3.2.9 The Far-End-Receive Failure (FERF) Condition

Declaring the FERF Condition

The Receive E3 Framer block will declare a Far-End Receive Failure (FERF) condition if it detects a user-selectable number of consecutive incoming E3 frames, with the “A” bit-field set to “1”.

This User-selectable number of E3 frames is either 3 or 5, depending upon the value that has been written into Bit 4 (RxFERF Algo) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 1 G.751 (ADDRESS = 0X10)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved			RxFERF Algo	Reserved			RxBIP4
RO	RO	RO	R/W	RO	RO	RO	R/W
0	0	0	0	0	0	0	0

Writing a “0” into this bit-field causes the Receive E3 Framer block to declare a FERF condition, if it detects 3 consecutive incoming E3 frames, that have the “A” bit set to “1”.

Writing a “1” into this bit-field causes the Receive E3 Framer block to declare a FERF condition, if it detects 5 consecutive incoming E3 frames, that have the “A” bit set to “1”.

Whenever the Receive E3 Framer block declares a FERF condition, then it will do the following.

- Generate a Change in FERF Condition interrupt to the MIcroprocessor. Hence, the Receive E3 Framer block will assert Bit 3 (FERF Interrupt Status) within the Rx E3 Framer Interrupt Status register - 2, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Status	BIP-4 Error Interrupt Status	Framing Error Interrupt Status	Not Used
RO	RO	RO	RO	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.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	0	0	0

Clearing the FERF Condition

The Receive E3 Framer block will clear the FERF condition once it has received a User-Selectable number of E3 frames with the “A” bit-field being set to “0” (e.g., no FERF condition). This User-Selectable number of E3 frames is either 3 or 5 depending upon the value that has been written into Bit 4 (RxFERF Algo) of the Rx E3 Configuration/Status Register, as discussed above.

Whenever the Receive E3 Framer clears the FERF status, then it will do the following:

1. Generate a Change in the FERF Status Interrupt to the Microprocessor.
2. Clear the Bit 0 (RxFERF) within the Rx E3 Configuration & Status register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

6.3.2.10 Error Checking of the Incoming E3 Frames

The Receive E3 Framer block can be configured to performs error-checking on the incoming E3 frame data that it receives from the Remote Terminal Equipment. If configured accordingly, the Receive E3 Framer block will performs this error-checking by computing the BIP-4 value of an incoming E3 frame. Once the Receive E3 Framer block has obtained this value, it will compare this value with that of the BIP-4 value that it receives, within the very next E3 frame. If the locally computed BIP-4 value matches the EM byte of the corresponding E3 frame, then the Receive E3 Framer block will conclude that this particular frame has been properly received. The Receive E3 Framer block will then inform the Remote Terminal Equipment of this fact by having the Local Terminal Equipment Transmit E3 Framer block send the Remote Terminal an E3 frame, with the “A” bit-field, set to “0”.

This procedure is illustrated in Figure 145 and Figure 146 , below.

Figure 145 illustrates the Local Receive E3 Framer receiving an error-free E3 frame. In this figure, the locally computed BIP-4 value of “0xA” matches that received from the Remote Terminal, within the EM byte-field. Figure 146 illustrates the subsequent action of the Local Transmit E3 Framer block, which will transmit an E3 frame, with the A bit-field set to “0”, to the Remote Terminal. This signaling indicates that the Local Receive E3 Framer has received an error-free E3 frame.

FIGURE 145. THE LOCAL RECEIVE E3 FRAMER BLOCK, RECEIVING AN E3 FRAME FROM THE REMOTE TERMINAL WITH A CORRECT BIP-4 VALUE.

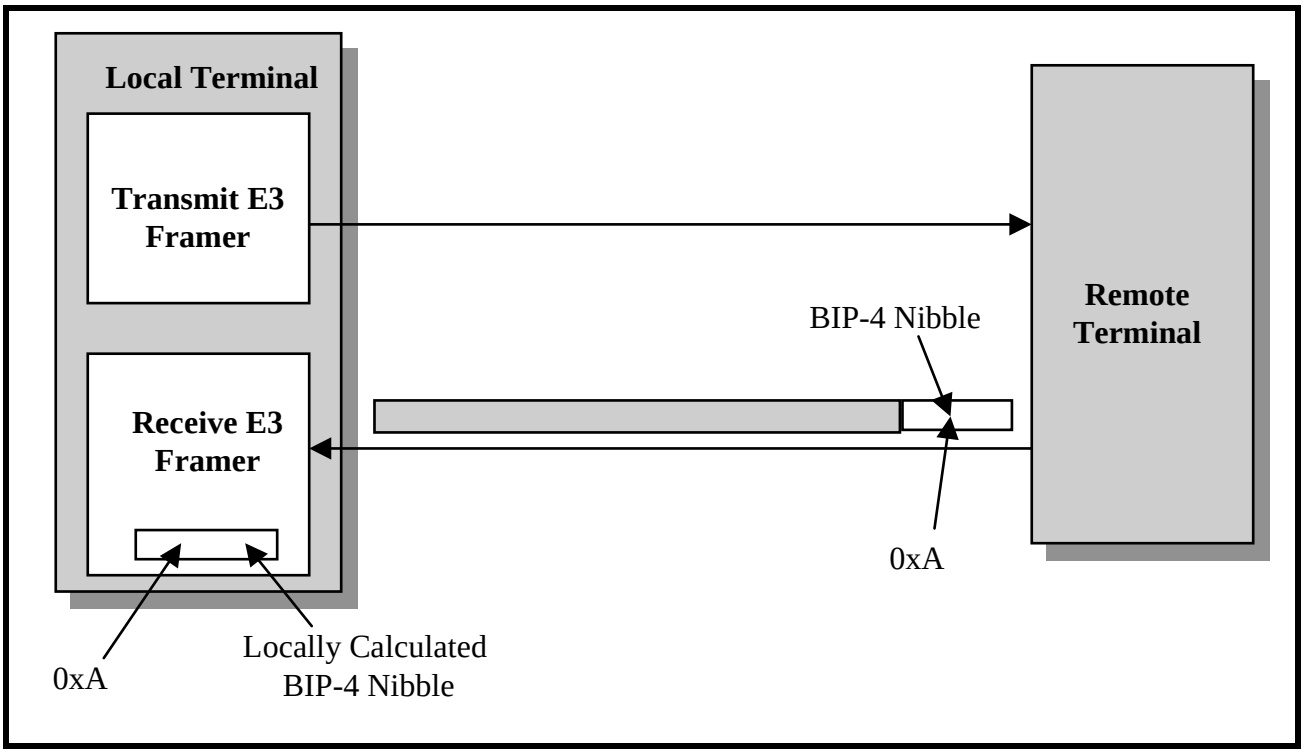
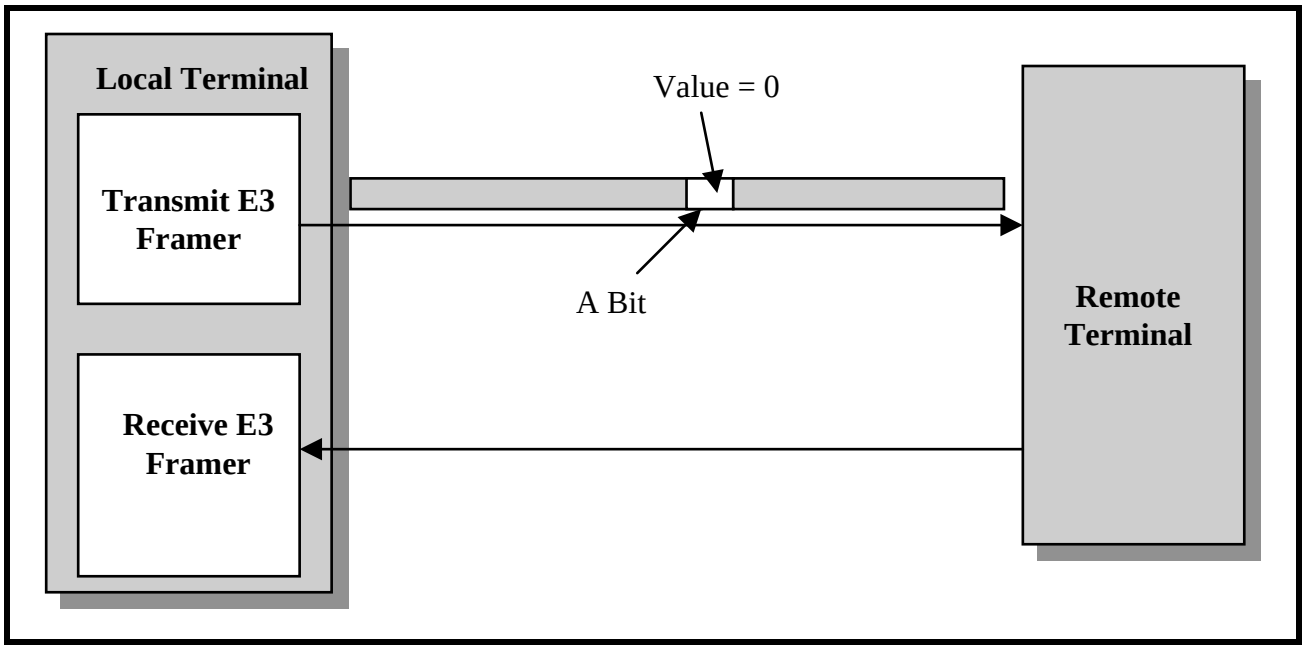


FIGURE 146. THE LOCAL RECEIVE E3 FRAMER BLOCK, TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL WITH THE "A" BIT SET TO "0"



However, if the locally computed BIP-4 value does not match the BIP-4 value of the corresponding E3 frame, then the Receive E3 Framer block will do the following.

- It will inform the Remote Terminal of this fact by having the Local Transmit E3 Framer block send the Remote Terminal an E3 frame, with the "A" bit-field set to "1". This phenomenon is illustrated below in Figure 147 and Figure 148 .

Figure 147 illustrates the Local Receive E3 Framer receiving an errored E3 frame. In this figure, the Local Receive E3 Frame block is receiving an E3 frame with an BIP-4 containing the value "0xA". This value does not match the locally computed BIP-4 value of "0xB". Consequently, there is an error in the previous E3 frame.

Figure 148 illustrates the subsequent action of the Local Transmit E3 Framer block, which will transmit an E3 frame, with the A bit-field set to "1" to the Remote Terminal. This signaling indicates that the Local Receive E3 Framer block has received an errored E3 frame.

FIGURE 147. THE LOCAL RECEIVE E3 FRAMER BLOCK, RECEIVING AN E3 FRAME FROM THE REMOTE TERMINAL WITH AN INCORRECT BIP-4 VALUE.

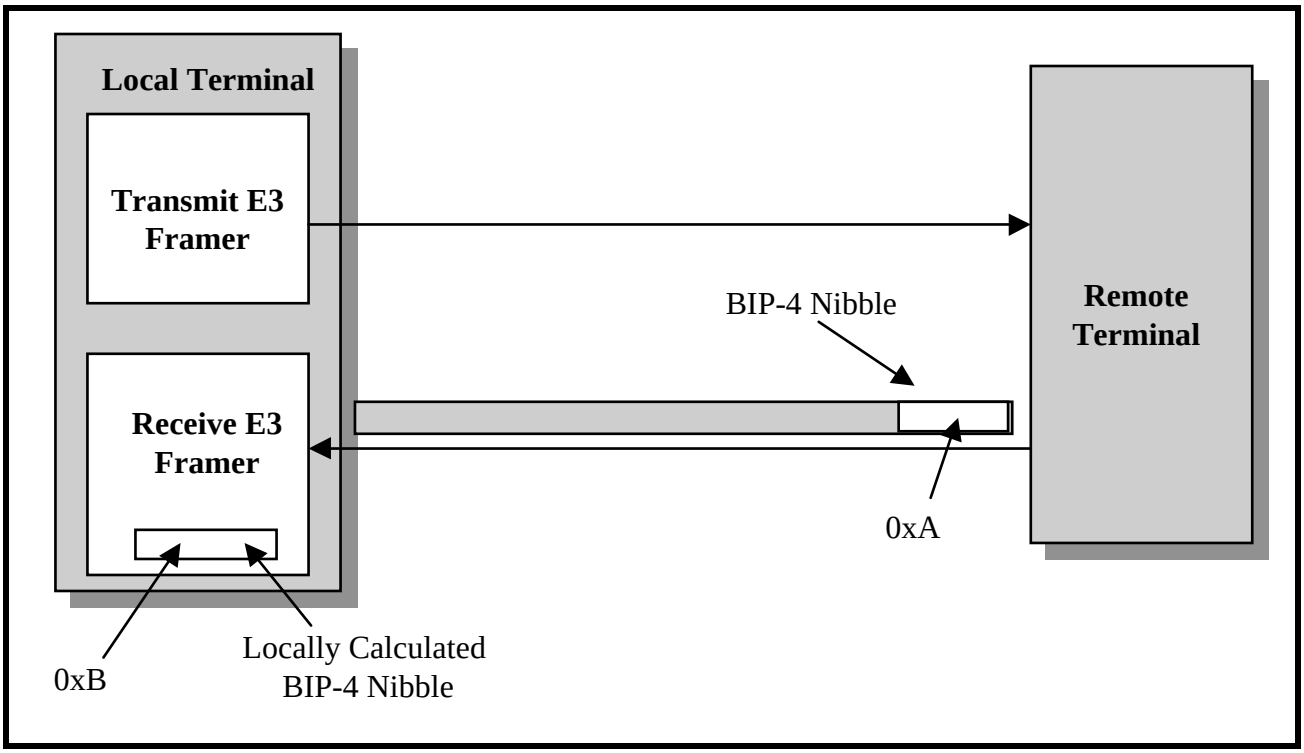
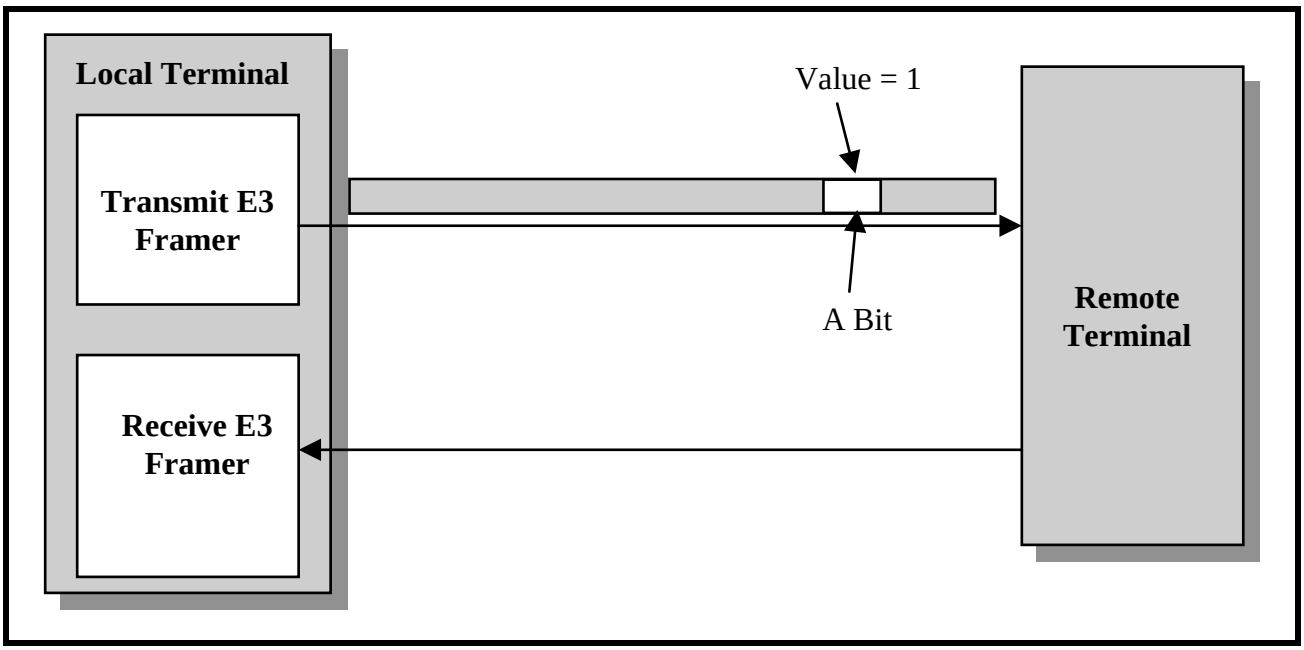


FIGURE 148. THE LOCAL RECEIVE E3 FRAMER BLOCK, TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL WITH THE "A" BIT-FIELD SET TO "1"



In additional to the FEBE bit-field signaling, the Receive E3 Framer block will generate the BIP-4 Error Interrupt to the Microprocessor. Hence, it will set bit 2 (BIP-8 Error Interrupt Status) to "1", as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Status	BIP-4 Error Interrupt Status	Framing Error Interrupt Status	Not Used
RO	RO	RO	RO	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

Finally, the Receive E3 Framer block will increment the PMON Parity Error Count registers. The byte format of these registers are presented below.

PMON PARITY ERROR COUNT REGISTER - MSB (ADDRESS = 0X54)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Parity Error Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON PARITY ERROR COUNT REGISTER - LSB (ADDRESS = 0X55)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Parity Error Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The user can determine the number of BIP-4 Errors that have been detected by the Receive E3 Framer block, since the last read of these registers. These registers are reset-upon-read.

Configuring the XRT74L74 Framer IC to support BIP-4 Error Detection

In order to perform BIP-4 checking of each E3 frame, the user must configure the XRT74L74 Framer IC accordingly, by executing the following steps.

1. Configure the Transmit Section (of the XRT74L74 Framer IC) to insert the BIP-4 value into the outbound E3 frames. This is accomplished by writing a "1" into bit-field 7 (Tx BIP-4 Enable) within the TxE3 Configuration Register, as illustrated below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx BIP-4 Enable	TxASourceSel[1:0]		TxNSourceSel[1:0]		Tx AIS Enable	Tx LOS Enable	Tx FAS Source Select
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	0	0	0

2. Enable the BIP-4 Error Interrupt. This is accomplished by writing a “1” into bit-field 2 (BIP-4 Error Interrupt Enable) within the RxE3 Interrupt Enable Register, as illustrated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Enable	BIP-4 Error Interrupt Enable	Framing Error Interrupt Enable	Not Used
R/W	RO	RO	RO	R/W	R/W	R/W	RO
0	0	0	0	0	1	0	0

After doing this, the XRT74L74 Framer IC will generate an interrupt to the Microprocessor/Microcontroller anytime the Receive Section detects a BIP-4 error.

6.3.3 The Receive HDLC Controller Block

The Receive E3 HDLC Controller block can be used to receive message-oriented signaling (MOS) type data link messages from the remote terminal equipment.

The MOS types of HDLC message processing is discussed in detail below.

The Message Oriented Signaling (e.g., LAP-D) Processing via the Receive E3 HDLC Controller block

The LAPD Receiver (within the Receive E3 HDLC Controller block) allows the user to receive PMDL messages from the remote terminal equipment, via the inbound E3 frames. In this case, the inbound message bits will be carried by the “N” bit-field within each inbound E3 Frame. The remote LAPD Transmitter will transmit a LAPD Message to the Local Receiver via either the “N” bit within each E3 Frame. The LAPD Receiver will receive and store the information portion of the received LAPD frame into the Receive LAPD Message Buffer, which is located at addresses: 0xDE through 0x135 within the on-chip RAM. The LAPD Receiver has the following responsibilities.

- Framing to the incoming LAPD Messages
- Filtering out stuffed "0’s" (within the information payload)
- Storing the Frame Message into the Receive LAPD Message Buffer
- Perform Frame Check Sequence (FCS) Verification
- Provide status indicators for
  - End of Message (EOM)
  - Flag Sequence Byte detected
  - Abort Sequence detected
  - Message Type
  - C/R Type
  - The occurrence of FCS Errors

The LAPD receiver’s actions are facilitated 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 (0x7E), as depicted in Figure 149 .

FIGURE 149. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The 16 bit FCS is calculated using CRC-16, x16 + x12 + x5 + 1

The local µP (at the remote terminal), while assembling the LAPD Message frame, will insert an additional byte at the beginning of the information (payload) field. This first byte of the information field indicates the type and size of the message being transferred. The value of this information field and the corresponding message type/size follow:

CL Path Identification = 0x38 (76 bytes)

IDLE Signal Identification = 0x34 (76 bytes)

Test Signal Identification = 0x32 (76 bytes)

ITU-T Path Identification = 0x3F (82 bytes)

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. 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 2 (RxLAPD Enable) of the Rx E3 LAPD Control Register, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Enable
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	0	0

Once the LAPD Receiver has been enabled, it will begin searching for the Flag Sequence octet (0x7E), in the “N” bit-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.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	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 or end 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 "1's"). If this next set of data is an ABORT Sequence, then the LAPD Receiver will assert the RxABORT bit-field (Bit 6) within the Rx E3 LAPD Status Register, as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	1	0	0	0	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 "N" bit-fields within each incoming E3 frame. Secondly, it is reassembling these bits 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 Framer IC.

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., "0's" that were inserted into the PMDL Message by the Remote 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 "1's".*

NOTE: *For more information on how the LAPD Transmitter inserted these stuff bits, please see Section 4.2.3.1.*

2. 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 locally-computed CRC-16 value, then the LAPD Receiver will conclude that it has received this LAPD Message frame in an error-free manner.

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 erred.

The LAPD Receiver will indicate the results of this FCS Verification process by setting Bit 2 (RxFCS Error) within the Rx E3 LAPD Status Register, to the appropriate value as tabulated below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	1	0	0

If the LAPD Receiver detects an error in the FCS value, then it will set the RxFCS Error bit-field to "1". Conversely, if the LAPD Receiver does not detect an error in the FCS value, then it will clear the RxFCS Error bit-field to "0".

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 is read in from the Receive LAPD Message buffer, by first checking the state of this bit-field.*

3. Check and Report the State of the "C/R" Bit-field

After 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) within the Rx E3 LAPD Status Register, as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	1	0	0	0

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.

4. 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. 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 Remote LAPD Receiver. This Message-Type Identification octet is transported to the Remote LAPD Receiver, along with the rest of the LAPD frame. From this Message Type Identification octet, the LAPD Receiver will know the type of 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.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Table 78 presents the relationship between the contents of RxLAPDType[1:0] and the type of message received by the LAPD Receiver.

TABLE 78: 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.*

5. Inform the Local Microprocessor/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 Microprocessor that a LAPD Message frame has been received and is ready for user-system handling. The LAPD Receiver will inform the Microprocessor/Microcontroller and the external circuitry by:

- Generating a LAPD Message Frame Received interrupt to the Microprocessor. The purpose of this interrupt is to let the Microprocessor 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 0 (RxLAPD Interrupt Status) within the Rx E3 LAPD Control Register to “1” as depicted below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	0	1

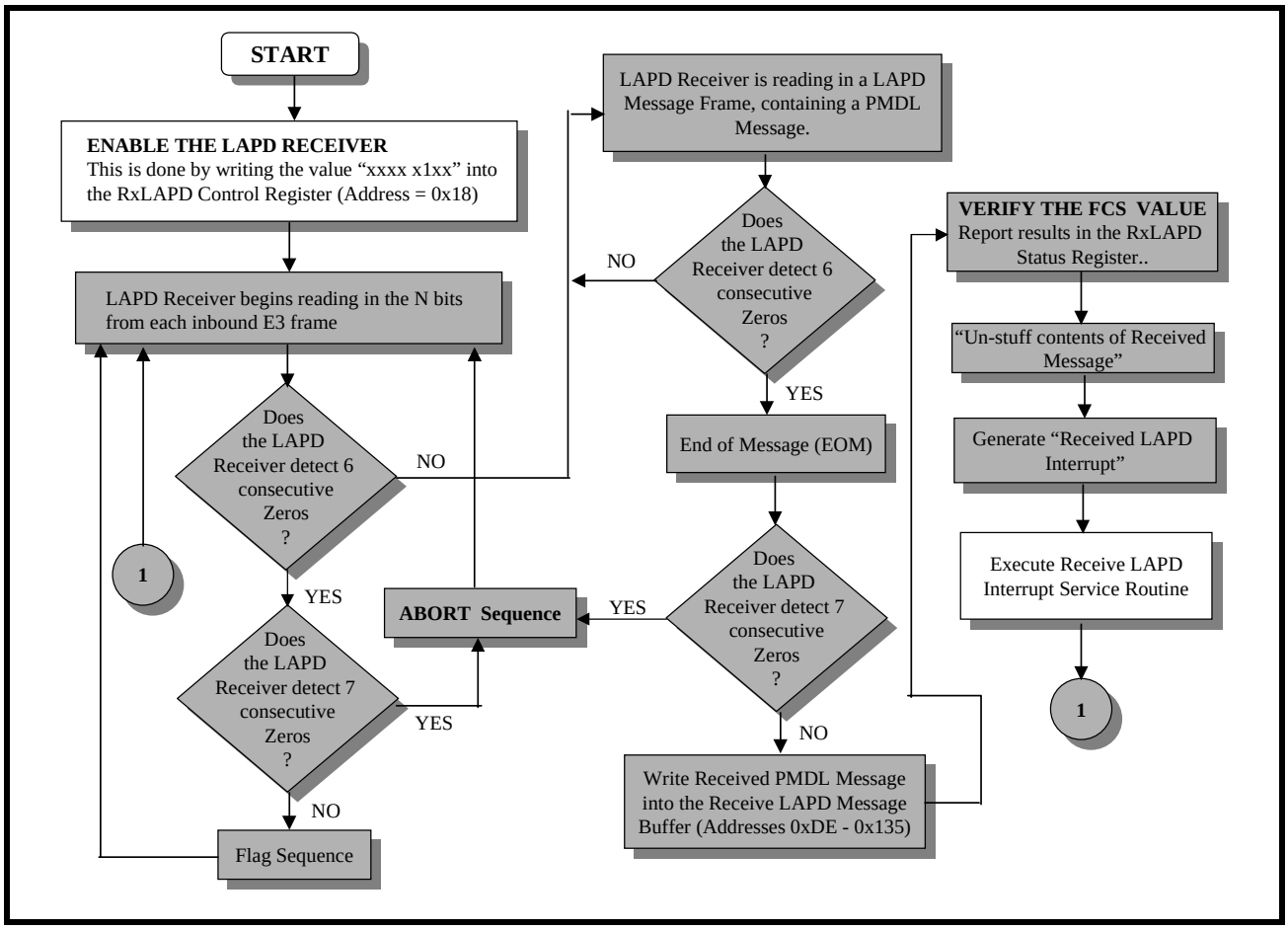
- Setting Bit 1 (End of Message) within the Rx E3 LAPD Status Register, to “1” as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	1	0

In summary, Figure 150 presents a flow chart depicting how the LAPD Receiver functions.

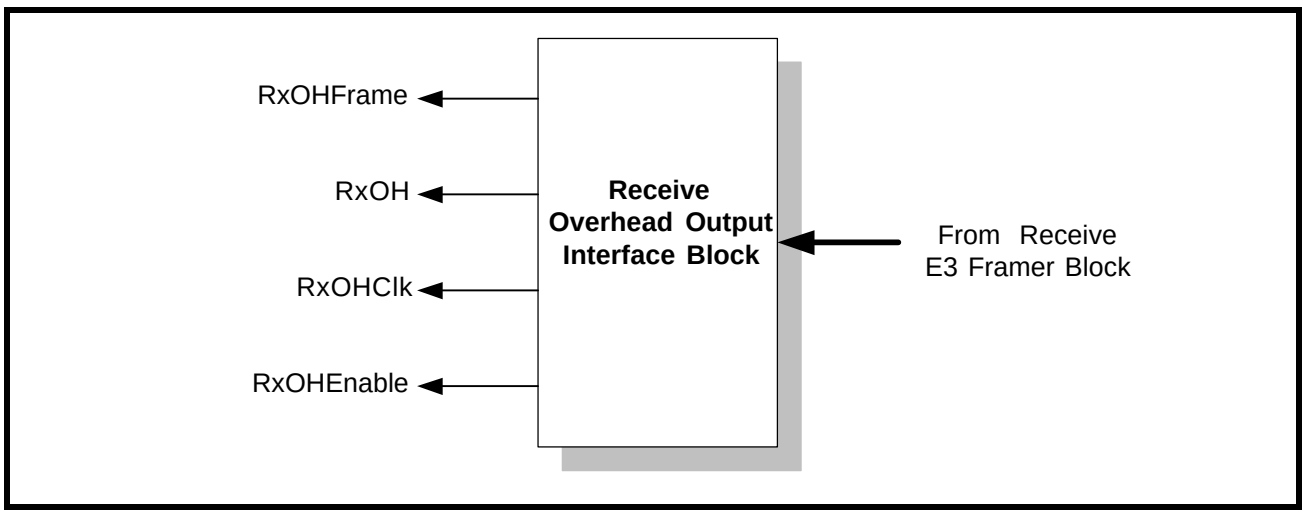
FIGURE 150. FLOW CHART DEPICTING THE FUNCTIONALITY OF THE LAPD RECEIVER



6.3.4 The Receive Overhead Data Output Interface

Figure 151 presents a simple illustration of the Receive Overhead Data Output Interface block within the XRT74L74.

FIGURE 151. THE RECEIVE OVERHEAD OUTPUT INTERFACE BLOCK



The E3, ITU-T G.751 frame consists of 1536 bits. Of these bytes, 1524 bits are payload bits and the remaining 12 bits are overhead bits. The XRT74L74 has been designed to handle and process both the payload type and overhead type bits for each E3 frame.

Within the Receive Section of the XRT74L74, the Receive Payload Data Output Interface block has been designed to handle the payload bits. Likewise, the Receive Overhead Data Output Interface block has been designed to handle and process the overhead bits.

The Receive Overhead Data Output Interface block unconditionally outputs the contents of all overhead bits. The XRT74L74 does not offer the user a means to shut off this transmission of data. However, the Receive Overhead Output Interface block does provide the user with the appropriate output signals for external Data Link Layer equipment to sample and process these overhead bits, via the following two methods.

- Method 1- Using the RxOHClk clock signal.
- Method 2 - Using the RxClk and RxOHEnable output signals.

Each of these methods are described below.

6.3.4.1 Method 1 - Using the RxOHClk Clock signal

The Receive Overhead Data Output Interface block consists of four (4) signals. Of these four signals, the following three signals are to be used when sampling the E3 overhead bits via Method 1.

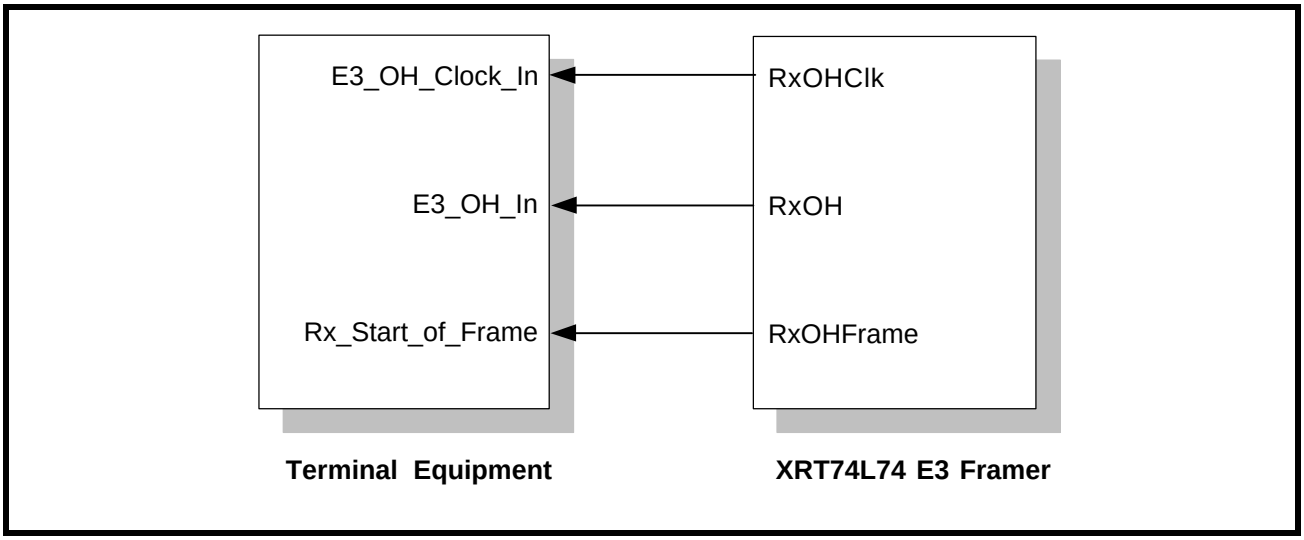
- RxOH
- RxOHClk
- RxOHFrame

Each of these signals are listed and described below in Table 79 .

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 1)

Figure 152 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment when using Method 1 to sample and process the overhead bits from the inbound E3 data stream.

FIGURE 152. HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 1



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the inbound E3 data stream (via the Receive Overhead Data Output Interface block) then it is expected to do the following:

1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input signal) on the rising edge of the RxOHClk (e.g., the E3_OH_Clock_In) signal.
2. Keep track of the number of rising clock edges that have occurred in the RxOHClk (e.g., the E3_OH_Clock_In) signal, since the last time the RxOHFrame signal was sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead byte is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.

TABLE 79: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 1

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming E3 frames, via this pin. The Receive Overhead Data Output Interface block will output a given overhead bit, upon the falling edge of RxOHClk. Hence, the external data link equipment should sample the data, at this pin, upon the rising edge of RxOHClk. ***NOTE:*** *The XRT74L74 will always output the E3 Overhead bits via this output pin. There are no external input pins or register bit settings available that will disable this output pin.*
RxOHClk	Output	**Receive Overhead Data Output Interface Clock Signal:** The XRT74L74 will output the Overhead bits (within the incoming E3 frames), via the RxOH output pin, upon the falling edge of this clock signal. As a consequence, the user's data link equipment should use the rising edge of this clock signal to sample the data on both the RxOH and RxOHFrame output pins. ***NOTE:*** *This clock signal is always active.*
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin "High" (for one period of the RxOHClk signal) whenever the first overhead bit within a given E3 frame is being driven onto the RxOH output pin.

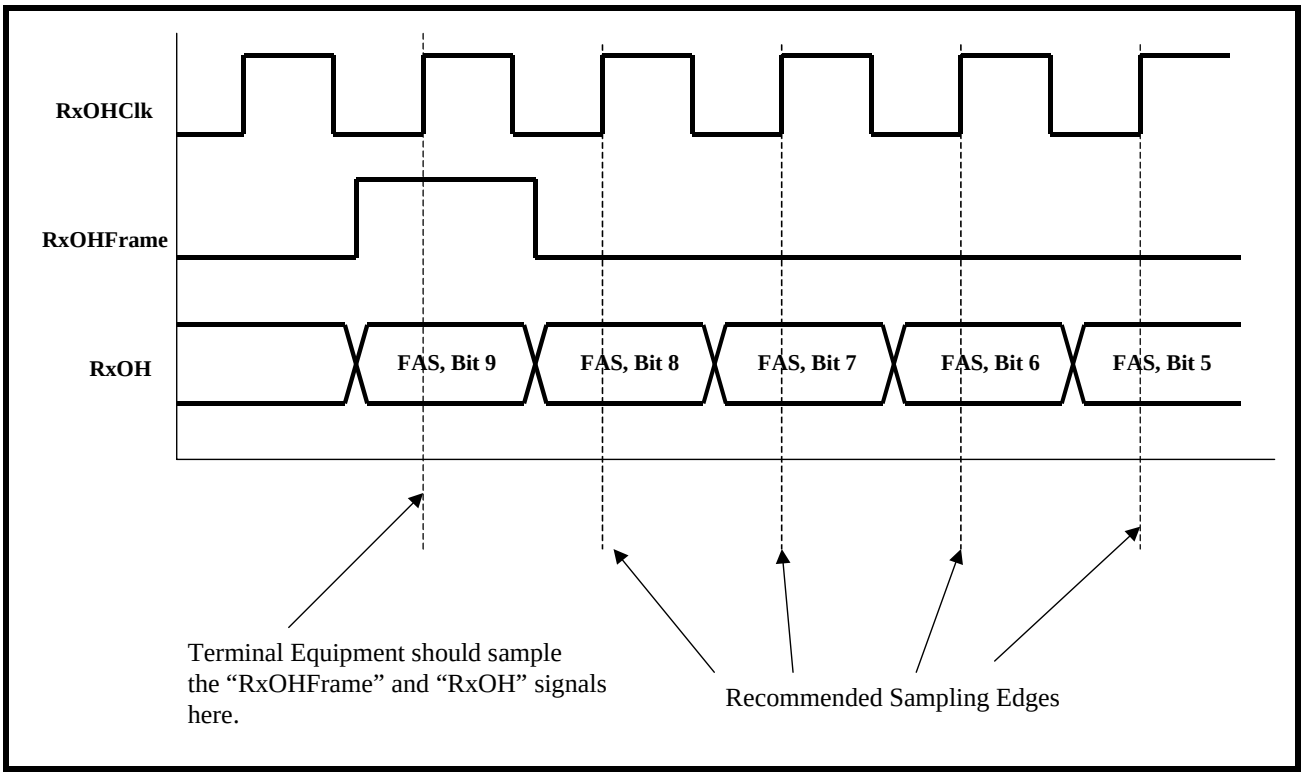
Table 80 relates the number of rising clock edges (in the RxOHClk signal, since the RxOHFrame signal was last sampled "High") to the E3 Overhead bit that is being output via the RxOH output pin.

TABLE 80: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (Clock edge is coincident with RxOHFrame being detected "High")	FAS Pattern - Bit 9
1	FAS Pattern - Bit 8
2	FAS Pattern - Bit 7
3	FAS Pattern - Bit 6
4	FAS Pattern - Bit 5
5	FAS Pattern - Bit 4
6	FAS Pattern - Bit 3
7	FAS Pattern - Bit 2
8	FAS Pattern - Bit 1
9	FAS Pattern - Bit 0
10	A Bit
11	N Bit

Figure 153 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 1 is being used to sample the incoming E3 overhead bits.

FIGURE 153. THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD OUTPUT INTERFACE FOR METHOD 1



Method 2 - Using RxOutClk and the RxOHEnable signals

Method 1 requires that the Terminal Equipment be able to handle an additional clock signal, RxOHClk. However, there may be a situation in which the Terminal Equipment circuitry does not have the means to deal with this extra clock signal, in order to use the Receive Overhead Data Output Interface. Method 2 involves the use of the following signals.

- RxOH
- RxOutClk
- RxOHEnable
- RxOHFrame

Each of these signals are listed and described below in Table 81 .

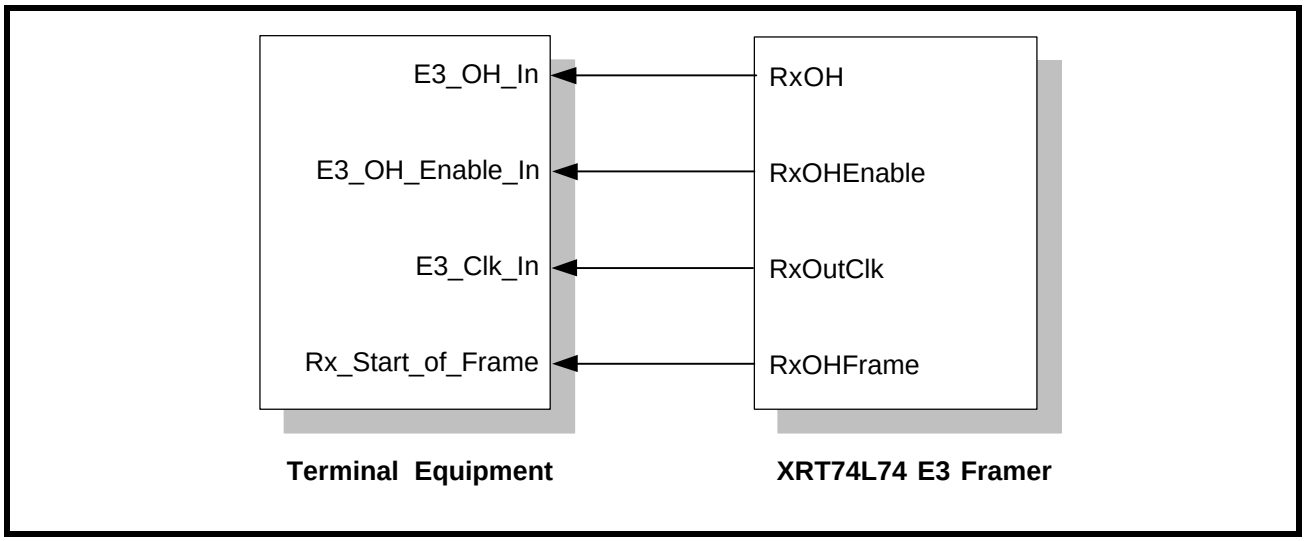
TABLE 81: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (METHOD 2)

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming E3 frames, via this pin. The Receive Overhead Output Interface will pulse the RxOHEnable output pin (for one RxOutClk period) at approximately the middle of the RxOH bit period. The user is advised to design the Terminal Equipment to latch the contents of the RxOH output pin, whenever the RxOHEnable output pin is sampled "High" on the falling edge of RxOutClk.
RxOHEnable	Output	**Receive Overhead Data Output Enable - Output pin:** The XRT74L74 will assert this output signal for one RxOutClk period when it is safe for the Terminal Equipment to sample the data on the RxOH output pin.
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin "High" (for one period of the RxOH signal), whenever the first overhead bit, within a given E3 frame is being driven onto the RxOH output pin.
RxOutClk	Output	**Receive Section Output Clock Signal:** This clock signal is derived from the RxLineClk signal (from the LIU) for loop-timing applications, and the TxInClk signal (from a local oscillator) for local-timing applications. For E3 applications, this clock signal will operate at 34.368MHz. The user is advised to design the Terminal Equipment to latch the contents of the RxOH pin, anytime the RxOHEnable output signal is sampled "High" on the falling edge of this clock signal.

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 2)

Figure 154 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment, when using Method 2 to sample and process the overhead bits from the inbound E3 data stream.

FIGURE 154. HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 2



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the inbound E3 data stream (via

the Receive Overhead Data Output Interface), then it is expected to do the following.

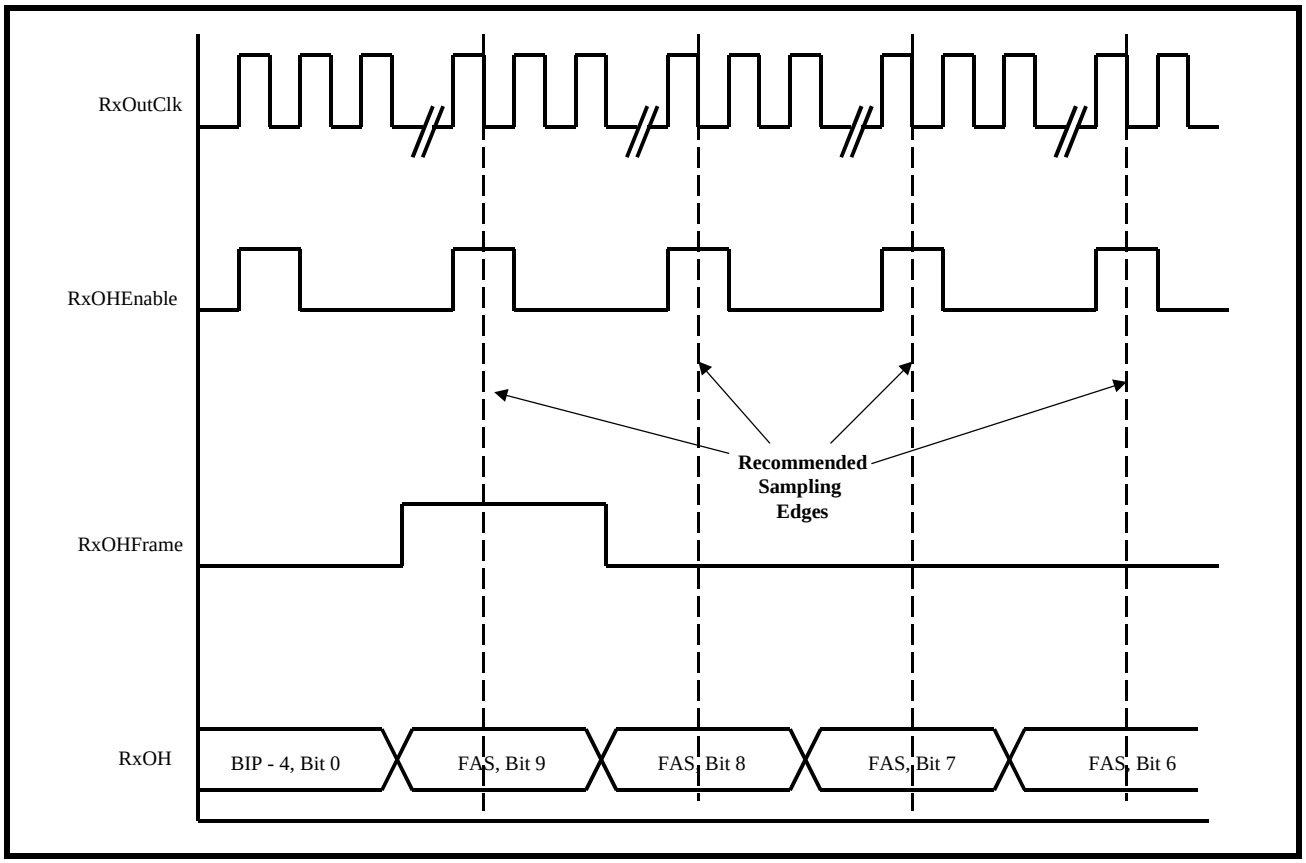
1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input) on the falling edge of the RxOutClk clock signal, whenever the RxOHEnable output signal is also sampled "High".
2. Keep track of the number of times that the RxOHEnable signal has been sampled "High" since the last time the RxOHFrame was also sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.
3. Table 82 relates the number of RxOHEnable output pulses (that have occurred since both the RxOHFrame and the RxOHEnable pins were both sampled "High") to the E3 overhead bit that is being output via the RxOH output pin.

TABLE 82: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (Clock edge is coincident with RxOHFrame being detected "High")	FAS Pattern - Bit 9
1	FAS Pattern - Bit 8
2	FAS Pattern - Bit 7
3	FAS Pattern - Bit 6
4	FAS Pattern - Bit 5
5	FAS Pattern - Bit 4
6	FAS Pattern - Bit 3
7	FAS Pattern - Bit 2
8	FAS Pattern - Bit 1
9	FAS Pattern - Bit 0
10	A Bit
11	N Bit

Figure 155 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 2 is being used to sample the incoming E3 overhead bits.

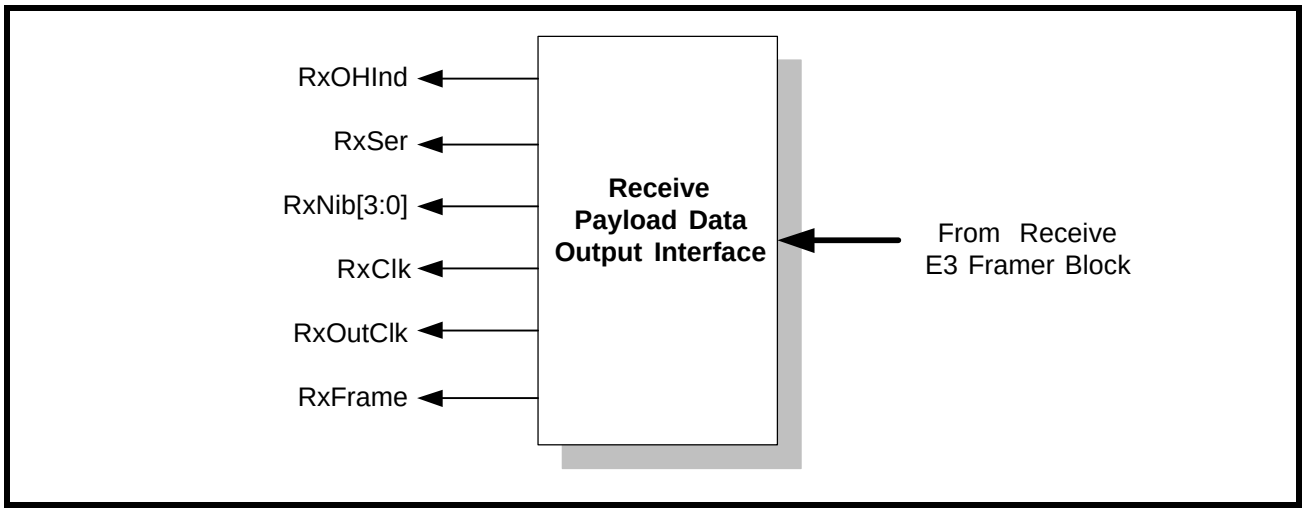
FIGURE 155. ILLUSTRATION OF THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (FOR METHOD 2).



6.3.5 The Receive Payload Data Output Interface

Figure 156 presents a simple illustration of the Receive Payload Data Output Interface block.

FIGURE 156. THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK



Each of the output pins of the Receive Payload Data Output Interface block are listed in Table 83 and described below. The exact role that each of these output pins assume, for a variety of operating scenarios are described throughout this section.

TABLE 83: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	TYPE	DESCRIPTION
RxSer	Output	**Receive Serial Payload Data Output pin:** If the user opts to operate the XRT74L74 in the serial mode, then the chip will output the payload data, of the incoming E3 frames, via this pin. The XRT74L74 will output this data upon the rising edge of RxClk. The user is advised to design the Terminal Equipment such that it will sample this data on the falling edge of RxClk. This signal is only active if the NibInt input pin is pulled "Low".
RxNib[3:0]	Output	**Receive Nibble-Parallel Payload Data Output pins:** If the user opts to operate the XRT74L74 in the nibble-parallel mode, then the chip will output the payload data, of the incoming E3 frames, via these pins. The XRT74L74 will output data via these pins, upon the falling edge of the RxClk output pin. The user is advised to design the Terminal Equipment such that it will sample this data upon the rising edge of RxClk. ***NOTE:*** *These pins are only active if the NibInt input pin is pulled "High".*
RxClk	Output	**Receive Payload Data Output Clock pin:** The exact behavior of this signal depends upon whether the XRT74L74 is operating in the Serial or in the Nibble-Parallel-Mode. **Serial Mode Operation** In the serial mode, this signal is a 34.368MHz clock output signal. The Receive Payload Data Output Interface will update the data via the RxSer output pin, upon the rising edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxSer pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation** In this Nibble-Parallel Mode, the XRT74L74 will derive this clock signal, from the RxLineClk signal. The XRT74L74 will pulse this clock 1060 times for each inbound E3 frame. The Receive Payload Data Output Interface will update the data, on the RxNib[3:0] output pins upon the falling edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxNib[3:0] output pins, upon the rising edge of this clock signal
RxOHInd	Output	**Receive Overhead Bit Indicator Output:** This output pin will pulse "High" whenever the Receive Payload Data Output Interface outputs an overhead bit via the RxSer output pin. The purpose of this output pin is to alert the Terminal Equipment that the current bit, (which is now residing on the RxSer output pin), is an overhead bit and should not be processed by the Terminal Equipment. The XRT74L74 will update this signal, upon the rising edge of RxOHInd. The user is advised to design (or configure) the Terminal Equipment to sample this signal (along with the data on the RxSer output pin) on the falling edge of the RxClk signal. ***NOTE:*** *For E3 applications, this output pin is only active if the XRT74L74 is operating in the Serial Mode. This output pin will be "Low" if the device is operating in the Nibble-Parallel Mode.*
RxFrame	Output	**Receive Start of Frame Output Indicator:** The exact behavior of this pin, depends upon whether the XRT74L74 has been configured to operate in the Serial Mode or the Nibble-Parallel Mode. **Serial Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" (for one bit period) when the Receive Payload Data Output Interface block is driving the very first bit (or Nibble) of a given E3 frame, onto the RxSer output pin. **Nibble-Parallel Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" for one nibble period, when the Receive Payload Data Output Interface is driving the very first nibble of a given E3 frame, onto the RxNib[3:0] output pins.

Operation of the Receive Payload Data Output Interface block

The Receive Payload Data Output Interface permits the user to read out the payload data of inbound E3 frames, via either of the following modes.

- Serial Mode
- Nibble-Parallel Mode

Each of these modes are described in detail, below.

6.3.5.1 Serial Mode Operation Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data, of the incoming E3 frames via the RxSer output pin, upon the rising edge of RxClk.

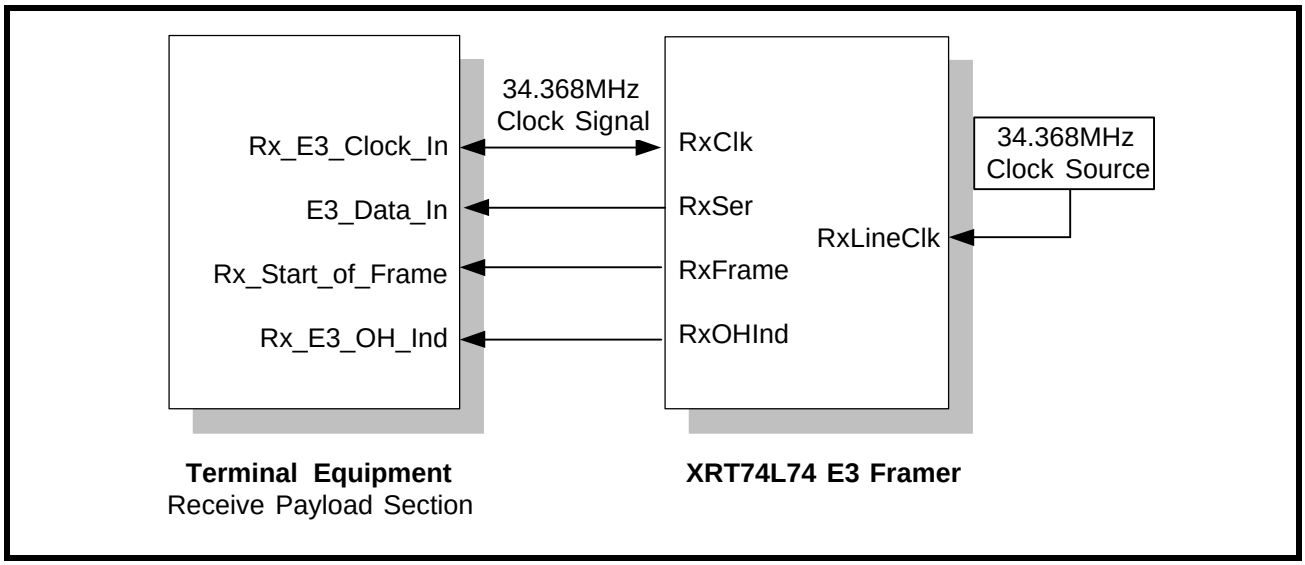
Delineation of inbound E3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one bit-period coincident with it driving the first bit within a given E3 frame, via the RxSer output pin.

Interfacing the XRT74L74 to the Receive Terminal Equipment

Figure 157 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 157. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE RECEIVE PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FRAMER IC (SERIAL MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxSer output pin, upon the rising edge of RxClk. Hence, the Terminal Equipment should sample the data on the RxSer output pin (or the E3_Data_In pin at the Terminal Equipment) upon the rising edge of RxClk. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the following signals.

- RxFrame
- RxOHInd

The Need for sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first bit of a given E3 frame onto the RxSer output pin. If knowledge of the E3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

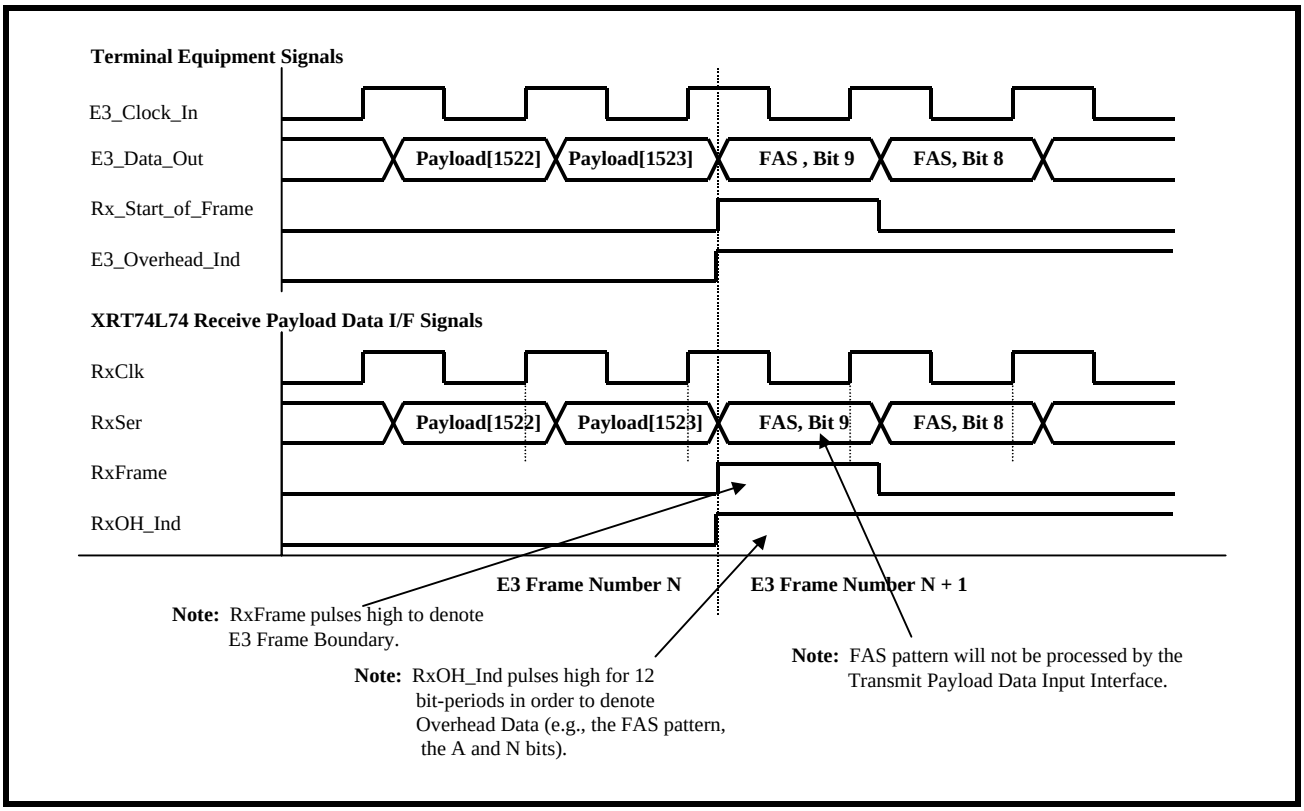
The Need for sampling RxOHInd

The XRT74L74 will indicate that it is currently driving an overhead bit onto the RxSer output pin, by pulsing the RxOHInd output pin "High". If the Terminal Equipment samples this signal "High", then it should know that the bit, that it is currently sampling via the RxSer pin is an overhead bit and should not be processed.

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Serial Mode Operation is illustrated in Figure 158 .

FIGURE 158. AN ILLUSTRATION OF THE BEHAVIOR OF THE SIGNALS BETWEEN THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT



6.3.5.2 Nibble-Parallel Mode Operation Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in the Nibble-Parallel Mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data of the incoming E3 frames, via the RxNib[3:0] output pins, upon the rising edge of RxClk.

NOTES:

1. *In this case, RxClk will function as the Nibble Clock signal between the XRT74L74 the Terminal Equipment. The XRT74L74 will pulse the RxClk output signal "High" 1060 times, for each inbound E3 frame.*
2. *Unlike Serial Mode operation, the duty cycle of RxClk, in Nibble-Parallel Mode operation is approximately 25%.*

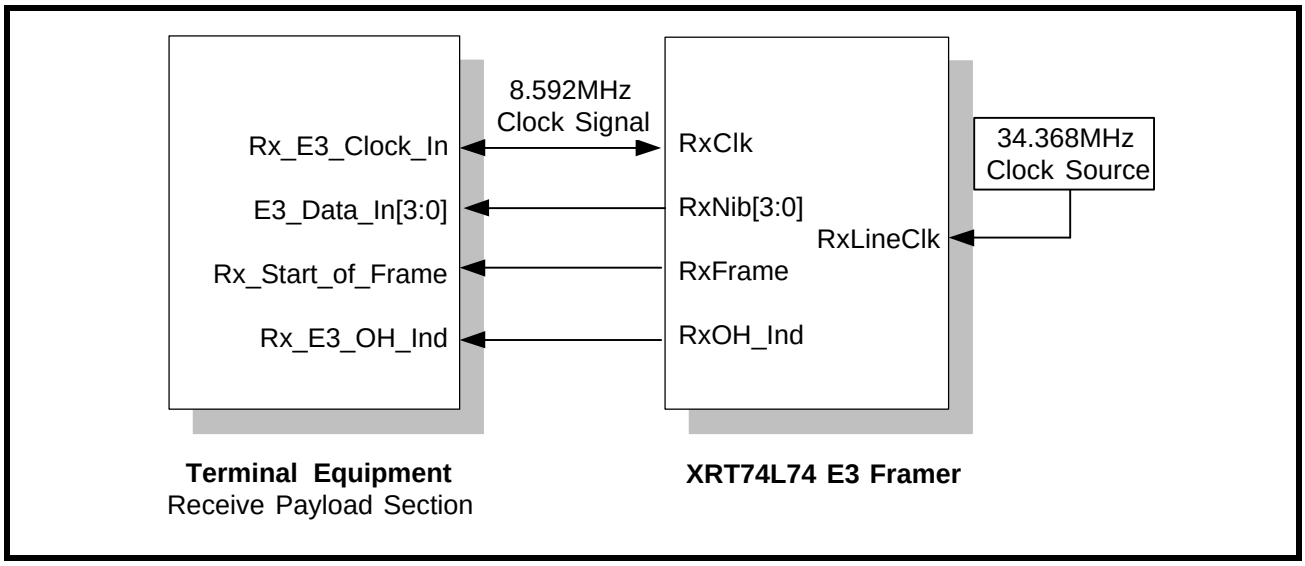
Delineation of Inbound E3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one nibble-period coincident with it driving the very first nibble, within a given inbound E3 frame, via the RxNib[3:0] output pins.

Interfacing the XRT74L74 the Terminal Equipment.

Figure 159 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 159. THE XRT74L74 DS3/E3 FRAMER IC BEING INTERFACED TO THE RECEIVE SECTION OF THE TERMINAL EQUIPMENT (NIBBLE-PARALLEL MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxNib[3:0] line, upon the rising edge of RxClk. Hence, the Terminal Equipment should sample the data on the RxNib[3:0] output pins (or the E3_Data_In[3:0] input pins at the Terminal Equipment) upon the rising edge of RxClk. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the RxFrame signal.

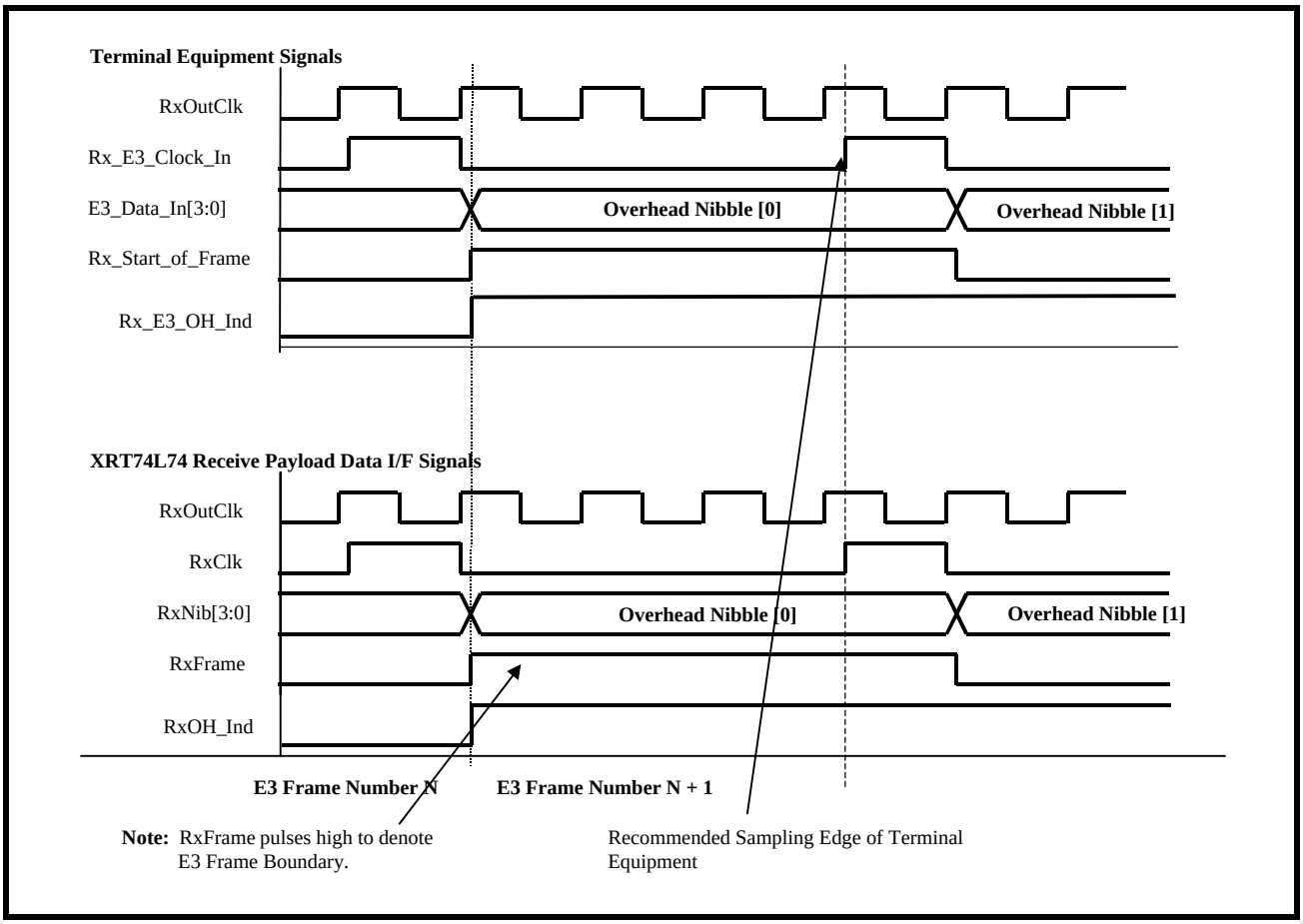
The Need for Sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first nibble of a given E3 frame, onto the RxNib[3:0] output pins. If knowledge of the E3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Nibble-Mode operation is illustrated in Figure 160 .

FIGURE 160. ILLUSTRATION OF THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK (FOR NIBBLE-PARALLEL MODE OPERATION).



6.3.6 Receive Section Interrupt Processing

The Receive Section of the XRT74L74 can generate an interrupt to the MIcrocontroller/Microprocessor for the following reasons.

- Change in Receive LOS Condition
- Change in Receive OOF Condition
- Change in Receive LOF Condition
- Change in Receive AIS Condition
- Change in Receive FERF Condition
- Change of Framing Alignment
- Detection of FEBE (Far-End Block Error) Event
- Detection of BIP-4 Error
- Detection of Framing Error
- Reception of a new LAPD Message

6.3.6.1 Enabling Receive Section Interrupts

As mentioned in Section 1.6, the Interrupt Structure within the XRT74L74 contains two hierarchical levels.

- Block Level
- Source Level

The Block Level

The Enable state of the Block level for the Receive Section Interrupts dictates whether or not interrupts (if enabled at the source level), are actually enabled.

The user can enable or disable these Receive Section interrupts, at the Block Level by writing the appropriate data into Bit 7 (Rx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
X	0	0	0	0	0	0	0

Setting this bit-field to “1” enables the Receive Section at the Block Level) for interrupt generation. Conversely, setting this bit-field to “0” disables the Receive Section for interrupt generation.

6.3.6.2 Enabling/Disabling and Servicing Interrupts

As mentioned previously, the Receive Section of the XRT74L74 Framer IC contains numerous interrupts. The Enabling/Disabling and Servicing of each of these interrupts is described below.

6.3.6.2.1 The Change in Receive LOS Condition Interrupt

If the Change in Receive LOS Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an LOS (Loss of Signal) Condition, and
2. When the XRT74L74 Framer IC clears the LOS condition.

Conditions causing the XRT74L74 Framer IC to declare an LOS Condition.

- If the XRT7300 LIU IC declares an LOS condition, and drives the RLOS input pin (of the XRT74L74 Framer IC) “High”.
- If the XRT74L74 Framer IC detects 32 consecutive “0”, via the RxPOS and RxNEG input pins.

Conditions causing the XRT74L74 Framer IC to clear the LOS Condition.

- If the XRT7300 LIU IC clears the LOS condition and drives the RLOS input pin (of the XRT74L74 Framer IC) “Low”.
- If the XRT74L74 Framer IC detects a string of 32 consecutive bits (via the RxPOS and RxNEG input pins) that does NOT contain a string of 4 consecutive “0’s”.

Enabling and Disabling the Change in Receive LOS Condition Interrupt

The user can enable or disable the Change in Receive LOS Condition Interrupt, by writing the appropriate value into Bit 1 (LOS Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive LOS Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 1 (LOS Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to “1”, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Whenever the user's system encounters the Change in Receive LOS Condition Interrupt, then it should do the following.

1. It should determine the current state of the LOS condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the LOS defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 4 (RxLOS) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

If the LOS state is TRUE

1. It should transmit a FERF (Far-End-Receive Failure) indicator to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 on how to configure the XRT74L74 to transmit a FERF indicator to the Remote Terminal Equipment.

If the LOS state is FALSE

1. It should cease transmitting the FERF indication to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 on how to control the state of the "A" bit, which is transmitted on each outbound E3 frame.

6.3.6.2.2 The Change in Receive OOF Condition Interrupt

If the Change in Receive OOF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an OOF (Out of Frame) Condition, and
2. When the XRT74L74 Framer IC clears the OOF condition.

Conditions causing the XRT74L74 Framer IC to declare an OOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) detects Framing bit errors, within four consecutive incoming E3 frames.

Conditions causing the XRT74L74 Framer IC to clear the OOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) transitions from the FAS Pattern Verification state to the In-Frame state (see Figure 115).
- If the Receive E3 Framer block transitions from the OOF Condition state to the In-Frame state (see Figure 115).

Enabling and Disabling the Change in Receive OOF Condition Interrupt

The user can enable or disable the Change in Receive OOF Condition Interrupt, by writing the appropriate value into Bit 3 (OOF Interrupt Enable), within

the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	X	0	X	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Change in Receive OOF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "Low".
- It will set Bit 3 (OOF Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to "1", as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0	0	0

Whenever the user's system encounters the Change in Receive OOF Condition Interrupt, then it should do the following.

1. It should determine the current state of the OOF condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the OOF defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 5 (RxOOF) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

If the OOF state is TRUE

1. It should transmit a FERF (Far-End-Receive Failure) indicator to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 on how to configure the XRT74L74 to transmit the FERF indicator to the Remote Terminal Equipment.

If the OOF state is FALSE

1. It should cease transmitting the FERF indication to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 on how to control the state of the "A" bit, which is transmitted via each outbound E3 frame.

6.3.6.2.3 The Change in Receive LOF Condition Interrupt

If the Change in Receive LOF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an LOF (Out of Frame) Condition, and
2. When the XRT74L74 Framer IC clears the LOF condition.

Conditions causing the XRT74L74 Framer IC to declare an LOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) detects Framing Bit errors, within four consecutive incoming E3 frames, and is not capable of transition back into the In-Frame state within a 1ms or 3ms period.

Conditions causing the XRT74L74 Framer IC to clear the LOF Condition.

- If the Receive E3 Framer block transitions from the OOF Condition state to the LOF Condition state (see Figure 115).
- If the Receive E3 Framer block transitions back into the In-Frame state.

Enabling and Disabling the Change in Receive LOF Condition Interrupt

The user can enable or disable the Change in Receive LOF Condition Interrupt, by writing the appropriate value into Bit 3 (LOF Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive LOF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 6 (LOF Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to “1”, as indicated below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

6.3.6.2.4 The Change in Receive AIS Condition Interrupt

If the Change in Receive AIS Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an AIS (Loss of Signal) Condition, and
2. When the XRT74L74 Framer IC clears the AIS condition.

Conditions causing the XRT74L74 Framer IC to declare an AIS Condition.

- If the XRT74L74 Framer IC detects 7 or less “0” within 2 consecutive E3 frames.

Conditions causing the XRT74L74 Framer IC to clear the AIS Condition.

- If the XRT74L74 Framer IC detects 2 consecutive E3 frames that each contain 8 or more “0’s”.

Enabling and Disabling the Change in Receive AIS Condition Interrupt

The user can enable or disable the Change in Receive LOS Condition Interrupt, by writing the appropriate value into Bit 0 (AIS Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive AIS Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 0 (AIS Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to “1”, as indicated below.

Whenever the user’s system encounters the Change in Receive AIS Condition Interrupt, then it should do the following.

1. It should determine the current state of the AIS condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the AIS defect. Hence, the user can determine the current state of the AIS defect by reading the state of Bit 3 (RxAIS) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

If the AIS Condition is TRUE

1. It should begin transmitting the FERF indication to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 for instructions on how to transmit a FERF condition.

If the AIS Condition is FALSE

2. It should cease transmitting the FERF indication to the Remote Terminal Equipment. Please see Section 4.2.4.2.1.3 for instructions on how to control the state of the “A” bit-field, within each outbound E3 frame.

6.3.6.2.5 The Change of Framing Alignment Interrupt

If the Change of Framing Alignment Interrupt is enabled then the XRT74L74 Framer IC will generate an interrupt any time the Receive E3 Framer block detects an abrupt change of framing alignment.

NOTE: *This interrupt is typically accompanied with the Change in Receive OOF Condition interrupt as well.*

Conditions causing the XRT74L74 Framer IC to generate this interrupt.

If the XRT74L74 Framer detects receives at least four consecutive E3 frames, within its Framing Alignment bytes in Error, then the XRT74L74 Framer IC will declare an OOF condition. However, while the XRT74L74 Framer IC is operating in the OOF condi-

tion, it will still rely on the old framing alignment for E3 payload data extraction, etc.

However, if the Receive E3 Framer had to change alignment, in order to re-acquire frame synchronization, then this interrupt will occur.

Enabling and Disabling the Change of Framing Alignment Interrupt

The user can enable or disable the Change of Framing Alignment Interrupt by writing the appropriate value into Bit 4 (COFA Interrupt Enable), within the Rx E3 Interrupt Enable Register - 1.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	X	0	0	0	0

Writing a “1” into this bit-field enables the Change of Framing Alignment Interrupt. Conversely, writing a “0” into this bit-field disables the Change of Framing Alignment Interrupt.

Servicing the Change of Framing Alignment Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it “Low”.
- It will set Bit 4 (COFA Interrupt Status), within the Rx E3 Interrupt Status Register -1, to “1”, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	0	0	0	0

6.3.6.2.6 The Change in Receive FERF Condition Interrupt

If the Change in Receive FERF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares a FERF (Far-End Receive Failure) Condition, and
2. When the XRT74L74 Framer IC clears the FERF condition.

Conditions causing the XRT74L74 Framer IC to declare an FERF Condition.

- If the XRT74L74 Framer IC begins receiving E3 frames which have the “A” bit set to “1”).

Conditions causing the XRT74L74 Framer IC to clear the AIS Condition.

- If the XRT74L74 Framer IC begins receiving E3 frames that do NOT have the “A” bit set to “1”.

Enabling and Disabling the Change in Receive AIS Condition Interrupt

The user can enable or disable the Change in Receive FERF Condition Interrupt, by writing the appropriate value into Bit 3 (FERF Interrupt Enable), within

the RxE3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Enable	BIP-4 Error Interrupt Enable	Framing Error Interrupt Enable	Not Used
R/W	RO	RO	RO	R/W	R/W	R/W	RO
0	0	0	0	0	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive FERF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 3 (FERF Interrupt Status), within the Rx E3 Interrupt Status Register - 2 to “1”, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Status	BIP-4 Error Interrupt Status	Framing Error Interrupt Status	Not Used
RO	RO	RO	RO	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Whenever the user’s system encounters the Change in Receive FERF Condition Interrupt, then it should do the following.

1. It should determine the current state of the FERF condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the FERF defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 0 (RxFERF) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER - 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	Not Used		RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	1	1	1

6.3.6.2.7 The Detection of BIP-4 Error Interrupt

If the Detection of BIP-4 Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block has detected an error in the BIP-4 Nibble, within an incoming E3 frame.

NOTE: *This interrupt is only active if the XRT74L74 Framer IC has been configured to process the BIP-4 nibble within each incoming and outbound E3 frame.*

Enabling and Disabling the Detection of FEBE Event Interrupt

The user can enable or disable the Detection of BIP-4 Error' interrupt by writing the appropriate value into Bit 2 (BIP-4 Interrupt Enable) within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Enable	BIP-4 Error Interrupt Enable	Framing Error Interrupt Enable	Not Used
R/W	RO	RO	RO	R/W	R/W	R/W	RO
0	0	0	0	0	X	0	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Detection of the BIP-4 Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "High".
- It will set the Bit 2 (BIP-4 Interrupt Status), within the RxE3 Interrupt Status Register - 2 as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Status	BIP-4 Error Interrupt Status	Framing Error Interrupt Status	Not Used
RO	RO	RO	RO	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

Whenever the Terminal Equipment encounters the Detection of BIP-4 Error Interrupt, it should do the following.

- It should read the contents of the PMON Parity Error Event Count Registers (located at Addresses 0x54 and 0x55) in order to determine the number of BIP-4 Errors that have been received by the XRT74L74 Framer IC.

6.3.6.2.8 The Detection of Framing Error Interrupt

If the Detection of Framing Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block has received an E3 frame with an incorrect FAS pattern value.

Enabling and Disabling the Detection of FEBE Event Interrupt

The user can enable or disable the Detection of Framing Error' interrupt by writing the appropriate value into Bit 1 (Framing Error Interrupt Enable) within

the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Enable	BIP-4 Error Interrupt Enable	Framing Error Interrupt Enable	Not Used
R/W	RO	RO	RO	R/W	R/W	R/W	RO
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Detection of Framing Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “High”.
- It will set the Bit 1 (Framing Error Interrupt Status), within the RxE3 Interrupt Status Register - 2 as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				FERF Interrupt Status	BIP-4 Error Interrupt Status	Framing Error Interrupt Status	Not Used
RO	RO	RO	RO	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Whenever the Terminal Equipment encounters the Detection of Framing Error Interrupt, it should do the following.

- It should read the contents of the PMON Framing Bit/Byte Error Count Registers (located at Addresses 0x52 and 0x53) in order to determine the number of Framing errors that have been received by the XRT74L74 Framer IC.

6.3.6.2.9 The Receipt of New LAPD Message Interrupt

If the Receive LAPD Message Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt anytime the Receive HDLC Controller block has received a new LAPD Message frame from the Remote Terminal Equipment, and has stored the contents of this message into the Receive LAPD Message buffer.

Enabling/Disabling the Receive LAPD Message Interrupt

The user can enable or disable the Receive LAPD Message Interrupt by writing the appropriate data into Bit 1 (RxLAPD Interrupt Enable) within the Rx E3 LAPD Control Register, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Enable
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	0	0

Writing a “1” into this bit-field enables the Receive LAPD Message Interrupt. Conversely, writing a “0” into this bit-field disables the Receive LAPD Message Interrupt.

Servicing the Receive LAPD Message Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 0 (RxLAPD Interrupt Status), within the Rx E3 LAPD Control register to “1”, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used					RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Enable
RO	RO	RO	RO	RO	R/W	R/W	RUR
0	0	0	0	0	0	0	0

- It will write the contents of the newly Received LAPD Message into the Receive LAPD Message buffer (located at 0xDE through 0x135).

Whenever the Terminal Equipment encounters the Receive LAPD Message Interrupt, then it should read out the contents of the Receive LAPD Message buffer, and respond accordingly.

7.0 E3/ITU-T G.832 OPERATION OF THE XRT74L74

Configuring the XRT74L74 to Operate in the E3, ITU-T G.832 Mode

The XRT74L74 can be configured to operate in the E3/ITU-T G.832 Mode by writing a "0" into bit-field 6 and a "1" into bit-field 2, within the Framer Operating Mode register, as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	0	x	0	x	1	x	x

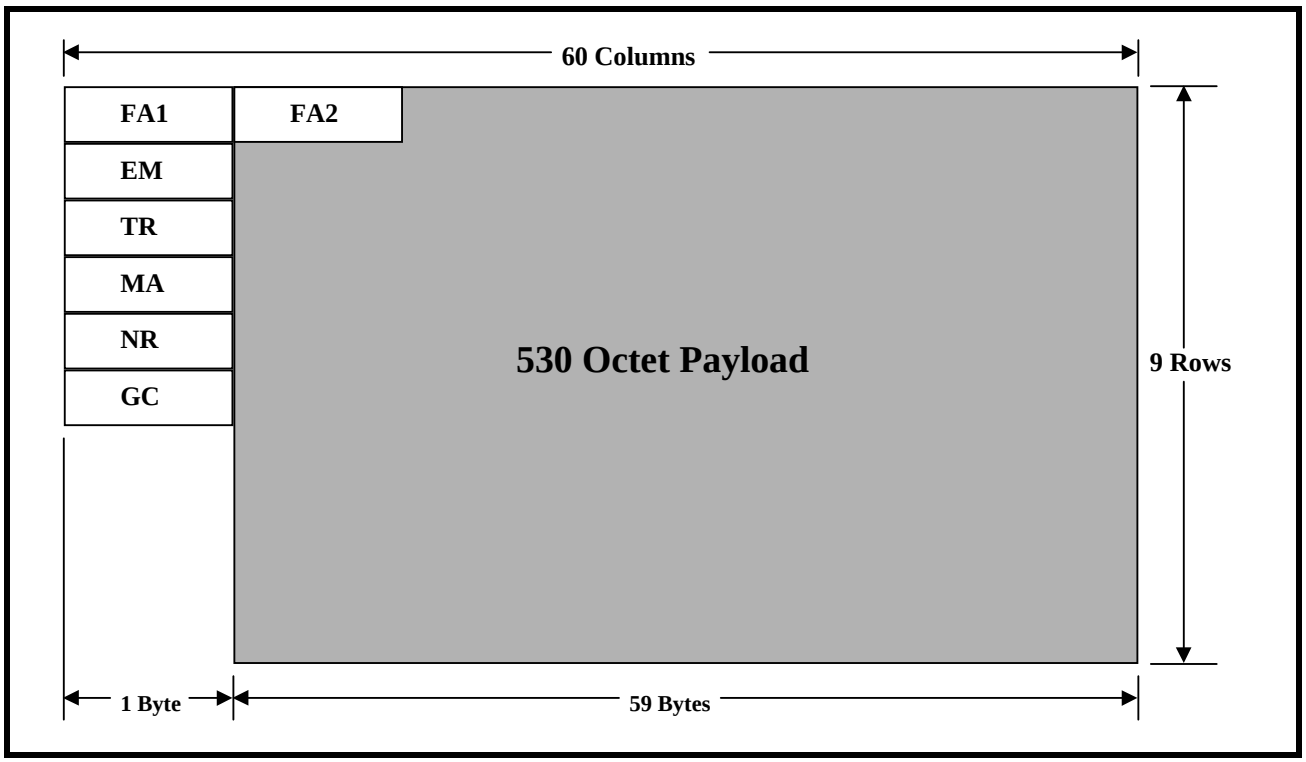
Prior to describing the functional blocks within the Transmit and Receive Sections of the XRT74L74, it is important to describe the E3, ITU-T G.832 framing format.

7.1 DESCRIPTION OF THE E3, ITU-T G.832 FRAMES AND ASSOCIATED OVERHEAD BYTES

The role of the various overhead bytes are best described by discussing the E3, ITU-T G.832 Frame Format as a whole. The E3, ITU-T G.832 Frame contains 537 bytes, of which 7 bytes are overhead and the remaining 530 bytes are payload bytes.

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 times per second, thereby resulting in the standard E3 bit rate of 34.368 Mbps. Figure 161 presents an illustration of the E3, ITU-T G.832 Frame Format.

FIGURE 161. E3, ITU-T G.832 FRAMING FORMAT.



7.1.1 Definition of the Overhead Bytes

The seven (7) overhead bytes are shown in Figure 161 , as FA1, FA2, EM, TR, MA, NR and GC. Each of these Overhead Bytes are further defined below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	1	x	0	x	x	x	x

7.1.1.1 Frame Alignment (FA1 and FA2) Bytes

FA1 and FA2 are known as the frame alignment bytes. The Receive E3 Framer, while trying to acquire or maintain framing synchronization with its incoming E3 frames, will attempt to locate these two bytes. FA1 is assigned the value “0xF6” and FA2 is assigned the value “0x28”.

7.1.1.2 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 Local Terminal Equipment and the Remote Terminal Equipment.

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 bytes 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 manner. Further, the Receive E3 Framer will block will inform the Remote Terminal Equipment of this fact by having the Local Terminal Equipment set the FEBE (Far-End-Block Error) bit, within the MA Byte of an Outbound E3 frame (to the Remote Terminal Equipment) to “0”. Please see Section 5.1.1 for a discussion of the MA Byte.

However, if the Receive E3 Framer block detects an error in the incoming EM byte, then it will conclude that the corresponding E3 frame is errored. Further, the Receive E3 Framer block will inform the Remote Terminal (e.g., the source of this erred E3 frame) of this fact by having the Local Terminal Equipment (e.g., the Transmit E3 Framer block) set the FEBE bit, within an Outbound E3 frame (destined to the Remote Terminal) to “1”.

NOTE: *A detailed discussion on the practical use of the EM byte is presented in Section 5.2.2.*

7.1.1.3 The Trail-Trace Buffer (TTB) 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 in Table 84 .

TABLE 84: DEFINITION OF THE TRAIL TRACE BUFFER BYTES, WITHIN THE E3, ITU-T G.832 FRAMING FORMAT

	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	X	X	X	X	X	X	X	X
*	X	X	X	X	X	X	X	X
*	X	X	X	X	X	X	X	X
16	X	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., the 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.

7.1.1.4 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.

THE MAINTENANCE AND ADAPTATION (MA) BYTE FORMAT

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 block (at a Local Terminal) is experiencing problems receiving E3 frame data from a Remote Terminal (e.g., an LOS, OOF or AIS condition), then it will inform the Remote Terminal Equipment of this fact by commanding the Local Transmit E3 Framer block to set the FERF bit-field (within the MA byte) of an Outbound E3 frame, to "1". The Local Transmit E3 Framer block will continue to set the FERF bit-field (within the subsequent Outbound E3 frames) to "1" until the Receive E3 Framer block no longer experiences problems in receiving the E3 frame data. If the Remote Terminal Equipment receives a certain number of consecutive E3 frames, with the FERF bit-field set to "1", then the Remote Terminal Equipment will interpret this signaling as an indication of a Far-End Receive Failure (e.g., a problem with the Local Terminal Equipment).

Conversely, if the Receive E3 Framer block (at a Local Terminal Equipment) is not experiencing any problems receiving E3 frame data from a Remote Terminal Equipment, then it will also inform the Remote Terminal Equipment of this fact by commanding the Local Transmit E3 Framer block to set the FERF bit-field (within the MA byte-field) of an Outbound E3 frame (which is destined for the Remote Terminal) to "0". The Remote Terminal Equipment will interpret this form of signaling as an indication of a normal operation.

NOTE: *A detailed discussion into the practical use of the FERF bit-field is presented in Section 5.2.4.2.*

Bit 6 - FEBE (Far-End Block Error)

If a Local Receive E3 Framer block detects an error in the EM byte, within an incoming E3 frame that it has received from the Remote Terminal Equipment, then it will inform the Remote Terminal Equipment of this error by commanding the Local Transmit E3 Framer block to set the FEBE bit-field (within the MA byte-field) of an Outbound E3 frame (which is destined for the Remote Terminal Equipment) to "1". The Remote Terminal Equipment will interpret this signaling as an indication that the E3 frames that it is transmitting back out to the Local Receive E3 Framer block are erred.

Conversely, if the Local Receive E3 Framer block does not detect any errors in the EM byte, within the incoming E3 frame, then it will also inform the Remote Terminal Equipment of this fact by commanding the Local Transmit E3 Framer block to set the FEBE bit-field of an Outbound E3 frame (which is destined for the Remote Terminal Equipment) to "0".

NOTE: *A detailed discussion into the practical use of the FEBE bit-field is presented in Section 5.2.4.2.*

Bits 5 - 3 Payload Type

These bit-fields indicates to the Remote Terminal Equipment, 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 in Table 85 .

TABLE 85: VARIOUS PAYLOAD TYPE VALUES AND THEIR CORRESPONDING MEANING

PAYLOAD TYPE VALUE	MEANING
000	Unequipped
001	Equipped
010	ATM Cells
011	SDH TU-12s

Bits 2 - 1 Payload Dependent

To be provided later.

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”.

7.1.1.5 The Network Operator (NR) Byte

The NR byte or the GC byte can be configured to transport LAP-D Message frame octets from the LAPD Transmitter to the LAPD Receiver (of the Remote Terminal Equipment) at a data rate of 64kbps (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 block will read in the contents of the TxNR Byte Register (Address = 0x37), and insert this value into the NR byte-field of each Outbound E3 frame. The Receive E3 Framer block will read in the contents of the NR byte-field within each incoming E3 frame and will write it into the RxNR 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 = 0x1A).

7.1.1.6 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 the Remote Terminal Equipment) at a data rate of 64kbps (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 block will read in the contents of the Tx GC Byte Register (Address = 0x35), and insert this value into the GC byte-field of each Outbound E3 frame.

The Receive E3 Framer block will read in the contents of the GC byte-field, within each incoming E3 frame, and will write it into the RxGC 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 = 0x1B).

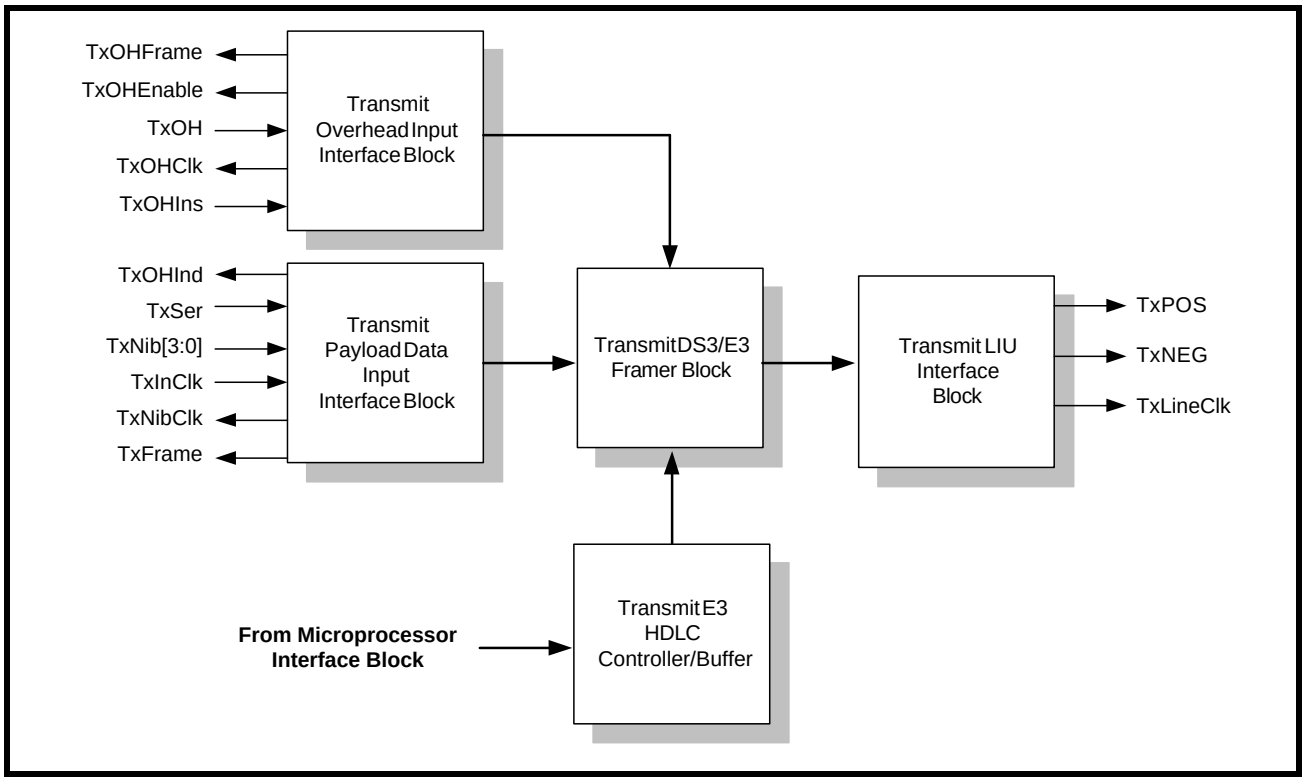
7.2 THE TRANSMIT SECTION OF THE XRT74L74 (E3 MODE OPERATION)

When the XRT74L74 has been configured to operate in the E3, ITU-T G.832 Mode, the Transmit Section of the XRT74L74 consists of the following functional blocks.

- Transmit Payload Data Input Interface block
- Transmit Overhead Data Input Interface block
- Transmit E3 Framer block
- Transmit HDLC Controller block
- Transmit LIU Interface block

Figure 162 presents a simple illustration of the Transmit Section of the XRT74L74 Framer IC.

FIGURE 162. THE TRANSMIT SECTION WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE E3 MODE

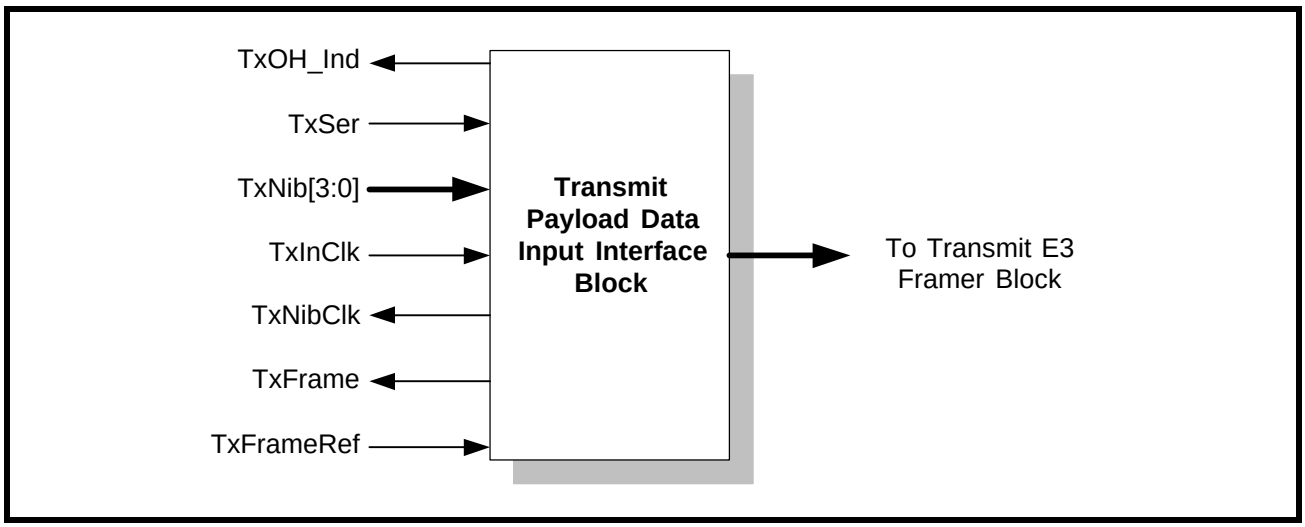


Each of these functional blocks will be discussed in detail in this document.

7.2.1 The Transmit Payload Data Input Interface Block

Figure 163 presents a simple illustration of the Transmit Payload Data Input Interface block.

FIGURE 163. THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK



Each of the input and output pins of the Transmit Payload Data Input Interface are listed in Table 86 and described below. The exact role that each of these inputs and output pins assume, for a variety of operating scenarios are described throughout this section.

TABLE 86: PIN LIST AND DESCRIPTIONS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE

SIGNAL NAME	TYPE	DESCRIPTION
TxSer	Input	**Transmit Serial Payload Data Input Pin:** If the user opts to operate the XRT74L74 in the serial mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the Outbound E3 data stream) to this input pin. The XRT74L74 will sample the data that is at this input pin upon the rising edge either the RxOutClk or the TxInClk signal (whichever is appropriate). ***NOTE:*** *This signal is only active if the NibInt input pin is pulled "Low".*
TxNib[3:0]	Input	**Transmit Nibble-Parallel Payload Data Input pins:** If the user opts to operate the XRT74L74 in the Nibble-Parallel mode, then the Terminal Equipment is expected to apply the payload data (that is to be transported via the Outbound E3 data stream) to these input pins. The XRT74L74 will sample the data that is at these input pins upon the rising edge of the TxNibClk signal. ***NOTE:*** *These pins are only active if the NibInt input pin is pulled "High".*
TxInClk	Input	**Transmit Section Timing Reference Clock Input pin:** The Transmit Section of the XRT74L74 can be configured to use this clock signal as the Timing Reference. If the user has made this configuration selection, then the XRT74L74 will use this clock signal to sample the data on the TxSer input pin. ***NOTE:*** *If this configuration is selected, then a 34.368 MHz clock signal must be applied to this input pin.*
TxNibClk	Output	**Transmit Nibble Mode Output** If the user opts to operate the XRT74L74 in the Nibble-Parallel mode, then the XRT74L74 will derive this clock signal from the selected Timing Reference for the Transmit Section of the chip (e.g., either the TxInClk or the RxLineClk signals). The XRT74L74 will use this signal to sample the data on the TxNib[3:0] input pins.
TxOHInd	Output	**Transmit Overhead Bit Indicator Output:** This output pin will pulse "High" one-bit period prior to the time that the Transmit Section of the XRT74L74 will be processing an Overhead bit. The purpose of this output pin is to warn the Terminal Equipment that, during the very next bit-period, the XRT74L74 is going to be processing an Overhead bit and will be ignoring any data that is applied to the TxSer input pin.
TxFrame	Output	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT74L74 will pulse this output pin "High" (for one bit-period), when the Transmit Payload Data Input Interface is processing the last bit of a given E3 frame. The purpose of this output pin is to alert the Terminal Equipment that it needs to begin transmission of a new E3 frame to the XRT74L74 (e.g., to permit the XRT74L74 to maintain Transmit E3 framing alignment control over the Terminal Equipment).
TxFrameRef	Input	**Transmit Frame Reference Input:** The XRT74L74 permits the user to configure the Transmit Section to use this input pin as a frame reference. If the user makes this configuration selection, then the Transmit Section will initiate its transmission of a new E3 frame, upon the rising edge of this signal. The purpose of this input pin is to permit the Terminal Equipment to maintain Transmit E3 Framing alignment control over the XRT74L74.
RxOutClk	Output	**Loop-Timed Timing Reference Clock Output pin:** The Transmit Section of the XRT74L74 can be configured to use the RxLineClk signal as the Timing Reference (e.g., loop-timing). If the user has made this configuration selection, then the XRT74L74 will: • Output a 34.368 MHz clock signal via this pin, to the Terminal Equipment. • Sample the data on the TxSer input pin, upon the rising edge of this clock signal.

Operation of the Transmit Payload Data Input Interface

The Transmit Terminal Input Interface is extremely flexible, in that it permits the user to make the following configuration options.

- The Serial or the Nibble-Parallel Interface Mode
- The Loop-Timing or the TxInClk (Local Timing) Mode

Further, if the XRT74L74 has been configured to operate in the TxInClk mode, then the user has two additional options.

- The XRT74L74 is the Frame Master (e.g., it dictates when the Terminal Equipment will initiate the transmission of data within a new E3 frame).
- The XRT74L74 is the Frame Slave (e.g., the Terminal Equipment will dictate when the XRT74L74 initiates the transmission of a new E3 frame).

Given these three set of options, the Transmit Terminal Input Interface can be configured to operate in one of the six (6) following modes.

- Mode 1 - Serial/Loop-Timed Mode
- Mode 2 - Serial/Local-Timed/Frame Slave Mode
- Mode 3 - Serial/Local-Timed/Frame Master Mode
- Mode 4 - Nibble/Loop-Timed Mode
- Mode 5 - Nibble/Local-Timed/Frame Slave Mode
- Mode 6 - Nibble/Local-Timed/Frame Master Mode

Each of these modes are described, in detail, below.

7.2.1.1 Mode 1 - The Serial/Loop-Timing Mode The Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Loop-Timing (Uses the RxLineClk signal as the Timing Reference)

Since the XRT74L74 is configured to operate in the loop-timed mode, the Transmit Section (of the XRT74L74) will use the RxLineClk input clock signal (e.g., the Recovered Clock signal, from the LIU) as its timing source. When the XRT74L74 is operating in this mode it will do the following.

1. It will ignore any signal at the TxInClk input pin.
2. The XRT74L74 will output a 34.368MHz clock signal via the RxOutClk output pin. This clock signal functions as the Transmit Payload Data Input Interface block clock signal.
3. The XRT74L74 will use the rising edge of the RxOutClk signal to latch in the data residing on the TxSer input pin.

B. Serial Mode

The XRT74L74 will accept the E3 payload data from the Terminal Equipment, in a serial-manner, via the TxSer input pin The Transmit Payload Data Input Interface will latch this data into its circuitry, on the rising edge of the RxOutClk output clock signal.

C. Delineation of Outbound E3 frames

The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period, coincident with the XRT74L74 processing the last bit of a given E3 frame.

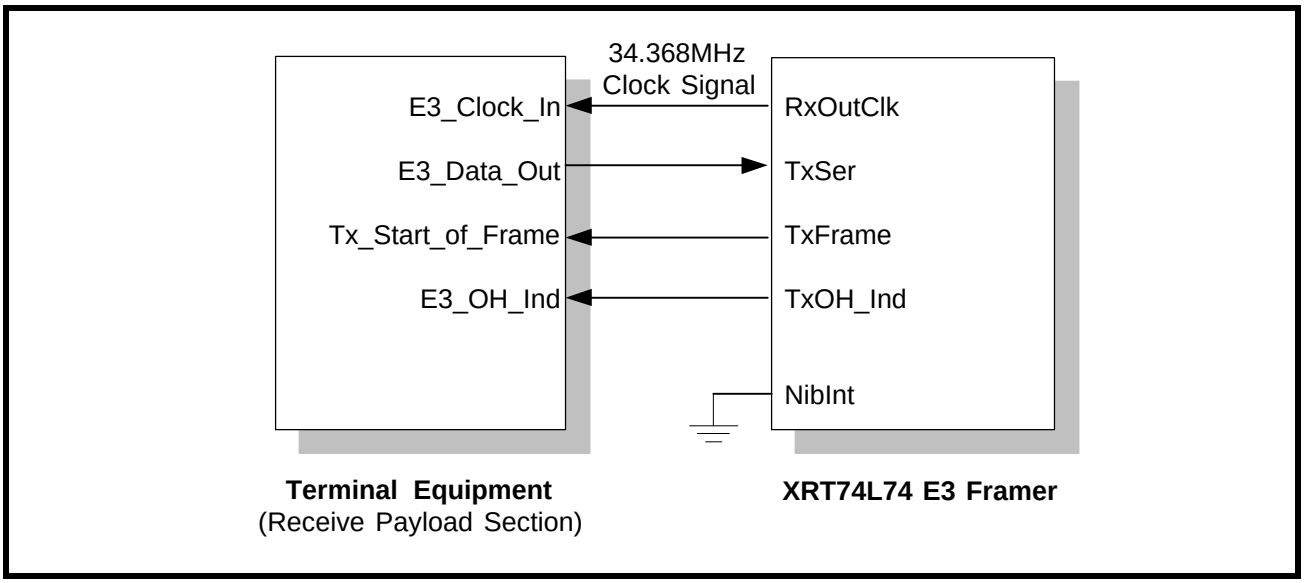
D. Sampling of Payload Data, from the Terminal Equipment

In Mode 1, the XRT74L74 will sample the data at the TxSer input, on the rising edge of RxOutClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 1 Operation

Figure 164 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 1 operation.

FIGURE 164. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 1 (SERIAL/LOOP-TIMED) OPERATION



Mode 1 Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode it will function as the source of the 34.368MHz clock signal. This clock signal will be used as the Terminal Equipment Interface clock by both the XRT74L74 IC and the Terminal Equipment.

The Terminal Equipment will serially output the payload data of the Outbound E3 data stream via its E3_Data_Out pin. The Terminal Equipment will update the data on the E3_Data_Out pin upon the rising edge of the 34.368 MHz clock signal, at its E3_Clock_In input pin (as depicted in Figures 19 and 20).

The XRT74L74 will latch the Outbound E3 data stream (from the Terminal Equipment) on the rising edge of the RxOutClk signal.

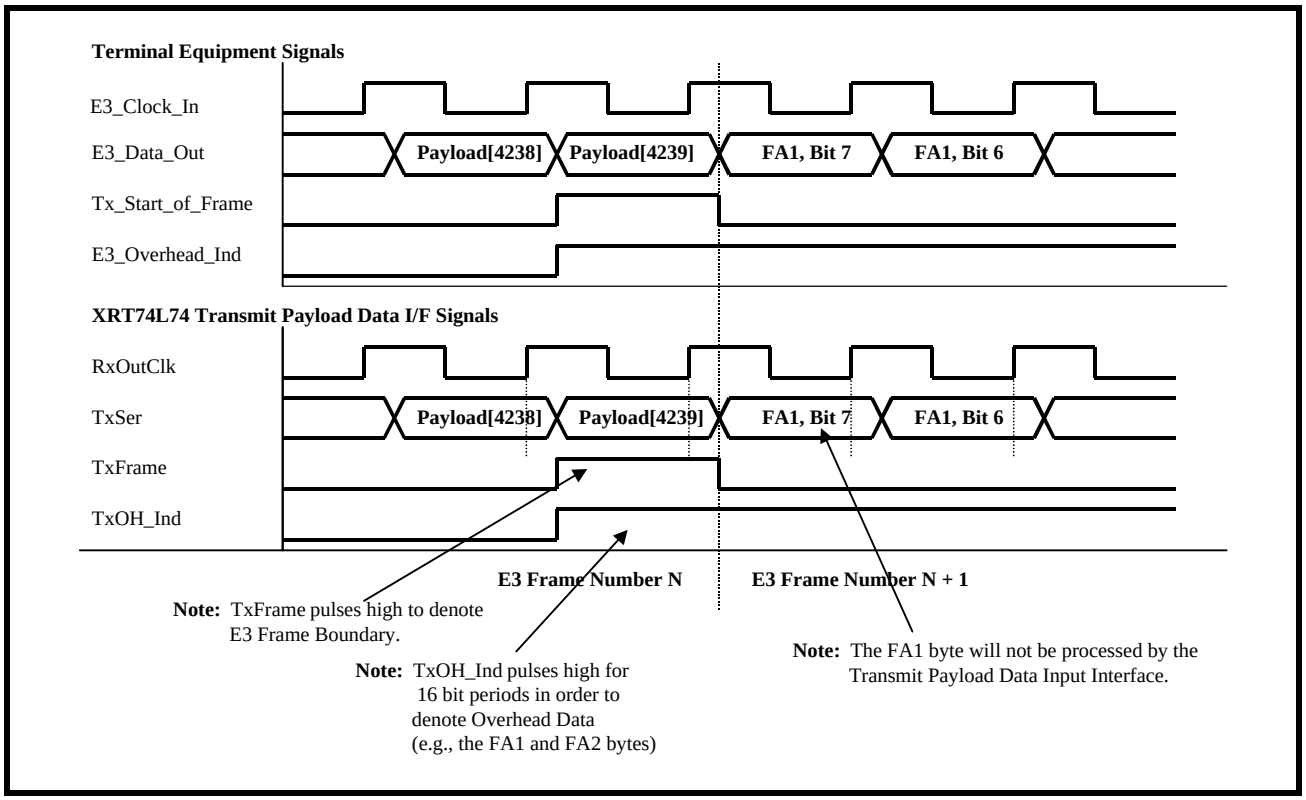
The XRT74L74 will indicate that it is processing the last bit, within a given Outbound E3 frame, by pulsing its TxFrame output pin “High” for one bit-period.

When the Terminal Equipment detects this pulse at its Tx_Start_of_Frame input, it is expected to begin transmission of the very next Outbound E3 frame to the XRT74L74 via the E3_Data_Out (or TxSer pin).

Finally, the XRT74L74 will indicate that it is about to process an overhead bit by pulsing the TxOH_Ind output pin "High" one bit period prior to its processing of an OH (Overhead) bit. In Figure 164 , the TxOH_Ind output pin is connected to the E3_Overhead_Ind input pin, of the Terminal Equipment. Whenever the E3_Overhead_Ind pin is pulsed "High" the Terminal Equipment is expected to not transmit a E3 payload bit upon the very next clock edge. Instead, the Terminal Equipment is expected to delay its transmission of the very next payload bit, by one clock cycle.

The behavior of the signals, between the XRT74L74 and the Terminal Equipment, for E3 Mode 1 operation is illustrated in Figure 165 .

FIGURE 165. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT (FOR MODE 1 OPERATION)



How to configure the XRT74L74 into the Serial/ Loop-Timed/Non-Overhead Interface Mode

1. Set the NibIntf input pin "Low".
2. Set the TimRefSel[1:0] bit fields (within the Framer Operating Mode Register) to "00" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 164 .

7.2.1.2 Mode 2 - The Serial/Local-Timed/ Frame-Slave Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local Timing - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the E3 payload data, in a serial manner, via the TxSer input pin. The Transmit Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of Outbound E3 frames (Frame Slave Mode)

The Transmit Section of the XRT74L74 will use the TxInClk input as its timing reference, and will use the TxFrameRef input signal as its framing reference. In other words, the Transmit Section of the XRT74L74

will initiate frame generation upon the rising edge of the TxFrameRef input signal).

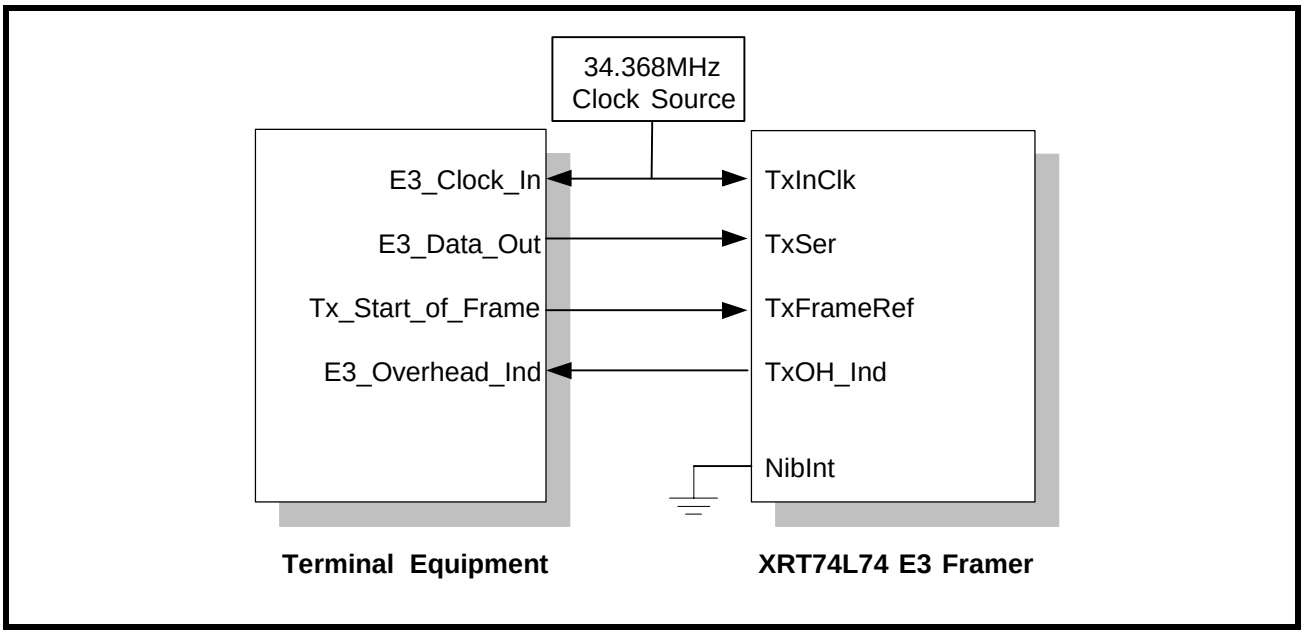
D. Sampling of payload data, from the Terminal Equipment

In Mode 2, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 2 Operation

Figure 166 presents an illustration of the Transmit Payload Data Input Interface block being interfaced to the Terminal Equipment, for Mode 2 operation.

FIGURE 166. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 2 (SERIAL/LOCAL-TIMED/FRAME-SLAVE) OPERATION



Mode 2 Operation of the Terminal Equipment

As shown in Figure 166 , both the Terminal Equipment and the XRT74L74 will be driven by an external 34.368MHz clock signal. The Terminal Equipment will receive the 34.368MHz clock signal via its E3_Clock_In input pin, and the XRT74L74 Framer IC will receive the 34.368MHz clock signal via the TxInClk input pin.

The Terminal Equipment will serially output the payload data of the Outbound E3 data stream, via the E3_Data_Out output pin, upon the rising edge of the signal at the E3_Clock_In input pin. (Note: The E3_Data_Out output pin of the Terminal Equipment is electrically connected to the TxSer input pin). The XRT74L74 Framer IC will latch the data, residing on the TxSer input line, on the rising edge of the TxInClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing its Tx_Start_of_Frame output signal (and in turn, the TxFrameRef input pin of the XRT74L74), “High” for one-bit period, coincident with the first bit of a new E3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new E3 frame.

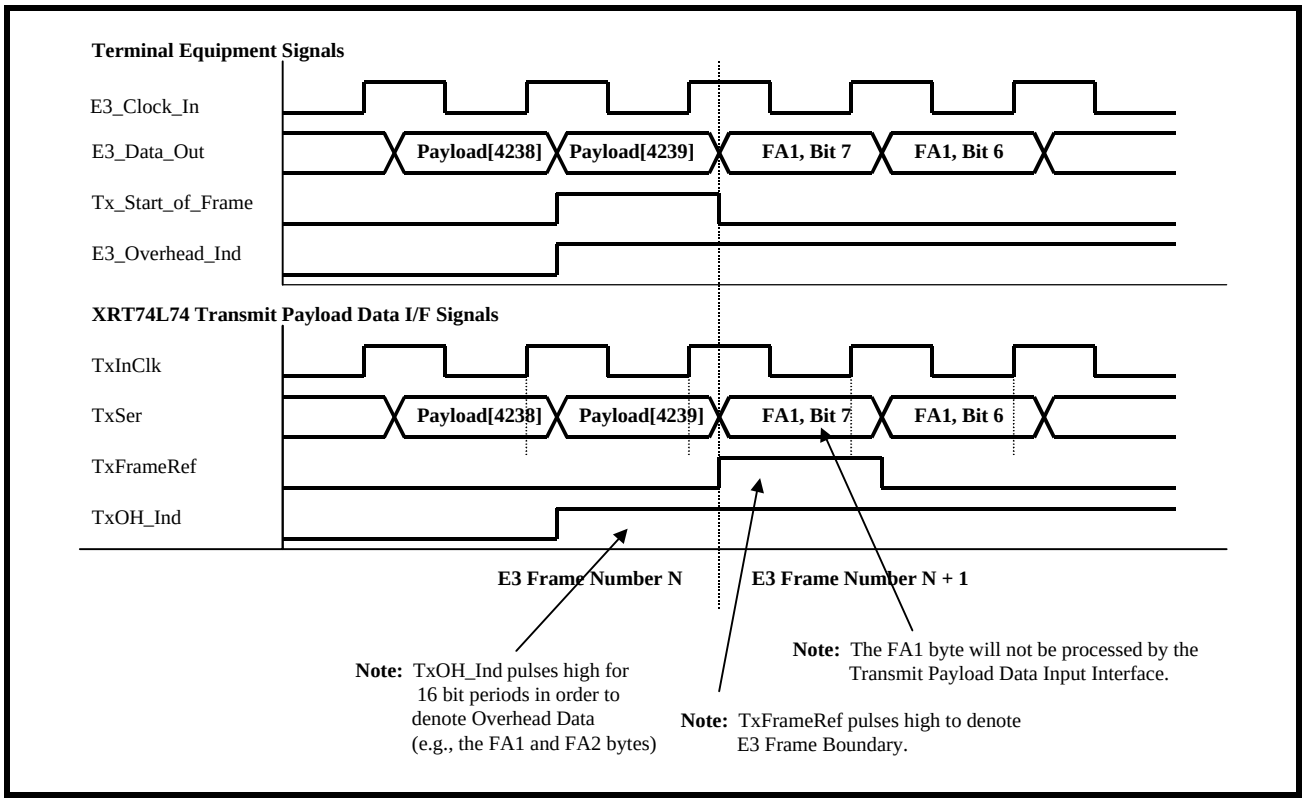
NOTES:

1. *In this case, the Terminal Equipment is controlling the start of Frame Generation, and is therefore referred to as the Frame Master. Conversely, since the XRT74L74 does not control the generationi of a new E3 frame, but is rather driven by the Terminal Equipment, the XRT74L74 is referred to as the Frame Slave.*
2. *If the user opts to configure the XRT74L74 to operate in Mode 2, it is imperative that the Tx_Start_of_Frame (or TxFrameRef) signal is synchronized to the TxInClk input clock signal.*

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the Outbound E3 frame. Since the TxOH_Ind output pin (of the XRT74L74) is electrically connected to the E3_Overhead_Ind, whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the E3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it should delay transmission of the very next E3 frame payload bit by one clock cycle.

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 2 Operation is illustrated in Figure 167 .

FIGURE 167. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 2 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 166 .

7.2.1.3 Mode 3 - The Serial/Local-Timed/ Frame-Master ModeBehavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows.

A. Local Timed - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference.

B. Serial Mode

The XRT74L74 will receive the E3 payload data, in a serial manner, via the TxSer input pin. The Transmit Payload Data Input Interface (within the XRT74L74) will latch this data into its circuitry, on the rising edge of the TxInClk input clock signal.

C. Delineation of Outbound DS3 frames (Frame Master Mode)

The Transmit Section of the XRT74L74 will use the TxInClk signal as its timing reference, and will initiate E3 frame generation, asynchronously with respect to any externally applied signal. The XRT74L74 will pulse its TxFrame output pin "High" whenever its it processing the very last bit-field within a given E3 frame.

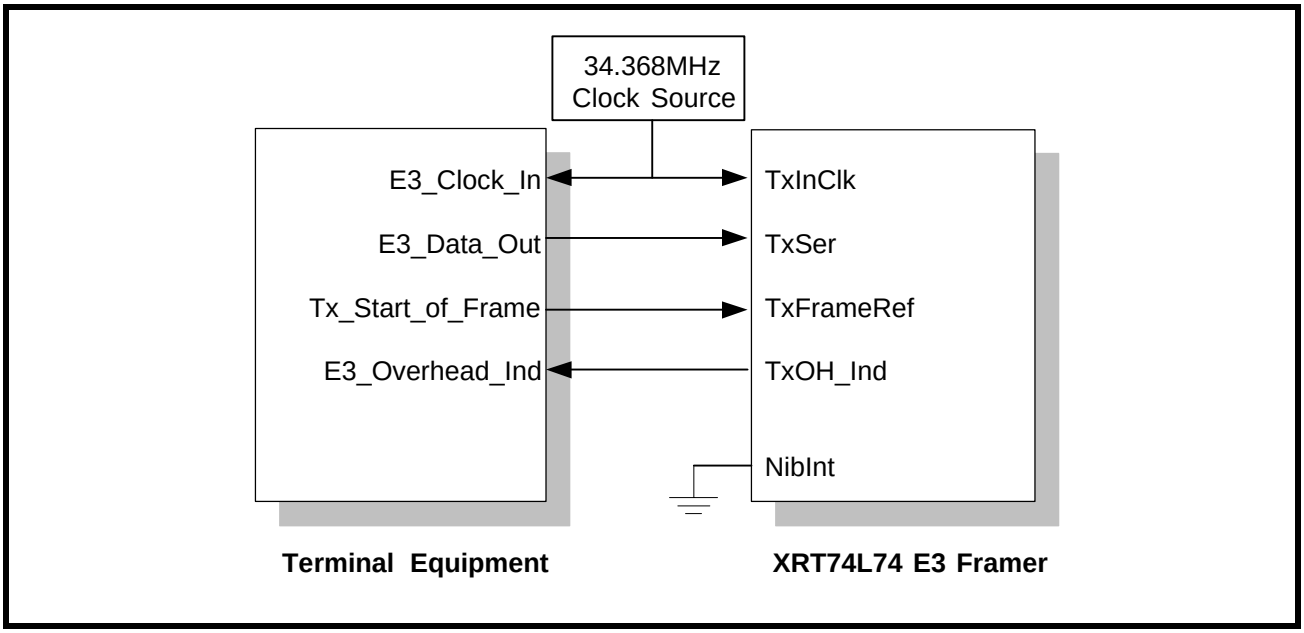
D. Sampling of payload data, from the Terminal Equipment

In Mode 3, the XRT74L74 will sample the data, at the TxSer input pin, on the rising edge of TxInClk.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 3 Operation

Figure 168 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 3 operation.

FIGURE 168. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 3 (SERIAL/LOCAL-TIMED/FRAME-MASTER) OPERATION



Mode 3 Operation of the Terminal Equipment

In Figure 168 , both the Terminal Equipment and the XRT74L74 are driven by an external 34.368 MHz clock signal. This clock signal is connected to the E3_Clock_In input of the Terminal Equipment and the TxInClk input pin of the XRT74L74.

The Terminal Equipment will serially output the payload data on its E3_Data_Out output pin, upon the rising edge of the signal at the E3_Clock_In input pin. Similarly, the XRT74L74 will latch the data, residing on the TxSer input pin, on the rising edge of TxInClk.

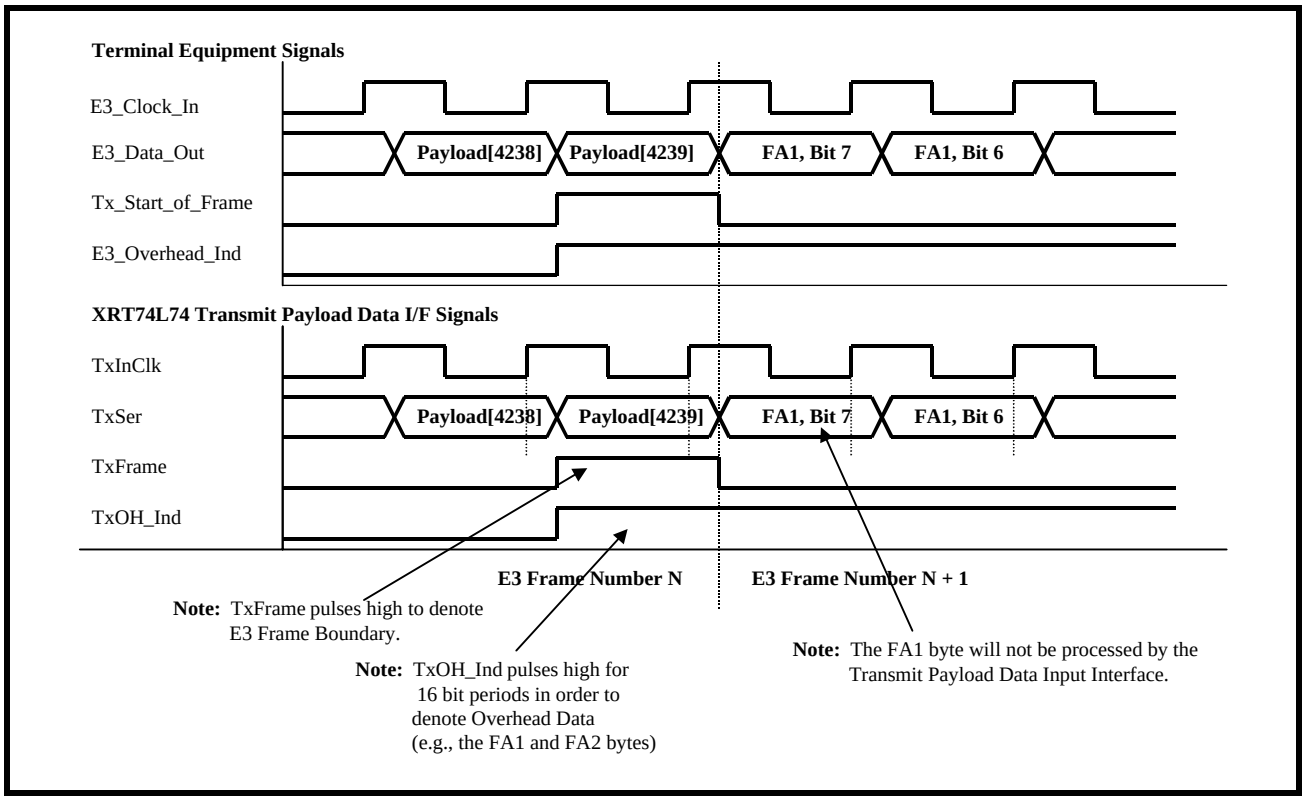
The XRT74L74 will pulse the TxFrame output pin "High" for one bit-period, coincident while it is processing the last bit-field within a given Outbound E3 frame. The Terminal Equipment is expected to monitor the TxFrame signal (from the XRT74L74) and to place the first bit, within the very next Outbound E3 frame on the TxSer input pin.

NOTE: *In this case, the XRT74L74 dictates exactly when the very next E3 frame will be generated. The Terminal Equipment is expected to respond appropriately by providing the XRT74L74 with the first bit of the new E3 frame, upon demand. Hence, in this mode, the XRT74L74 is referred to as the Frame Master and the Terminal Equipment is referred to as the Frame Slave.*

Finally, the XRT74L74 will pulse its TxOH_Ind output pin, one bit-period prior to it processing a given overhead bit, within the Outbound E3 frame. Since the TxOH_Ind output pin of the XRT74L74 is electrically connected to the E3_Overhead_Ind whenever the XRT74L74 pulses the TxOH_Ind output pin "High", it will also be driving the E3_Overhead_Ind input pin (of the Terminal Equipment) "High". Whenever the Terminal Equipment detects this pin toggling "High", it should delay transmission of the very next DS3 frame payload bit by one clock cycle.

The behavior of the signal between the XRT74L74 and the Terminal Equipment for E3 Mode 3 Operation is illustrated in Figure 169 .

FIGURE 169. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3 MODE 3 OPERATION)



How to configure the XRT74L74 to operate in this mode.

1. Set the NibIntf input pin "Low".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01".

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 169 .

7.2.1.4 Mode 4 - The Nibble-Parallel/Loop-Timed Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

A. Looped Timing (Uses the RxLineClk as the Timing Reference)

In this mode, the Transmit Section of the XRT74L74 will use the RxLineClk signal as its timing reference. When the XRT74L74 is operating in the Nibble-Mode, it will internally divide the RxLineClk signal, by a factor of four (4) and will output this signal via the TxNibClk output pin.

B. Nibble-Parallel Mode

The XRT74L74 will accept the E3 payload data, from the Terminal Equipment in a nibble-parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal

Equipment Input Interface block will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of the Outbound E3 frames

The XRT74L74 will pulse the TxNibFrame output pin "High" for one bit-period, coincident with the XRT74L74 processing the last nibble of a given E3 frame.

D. Sampling of payload data, from the Terminal Equipment

In Mode 4, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the RxOutClk clock signal, following a pulse in the TxNibClk signal (see Figure 171).

NOTE: *The TxNibClk signal, from the XRT74L74, operates nominally at 11.184 MHz (e.g., 44.736 MHz divided by 4). However, for reasons described below, TxNibClk effectively operates at a lower clock frequency. The Transmit Payload Data Input Interface is only used to accept the payload data, which is intended to be carried by Outbound DS3 frames. The Transmit Payload Data Input Interface is not designed to accommodate the entire DS3 data stream.*

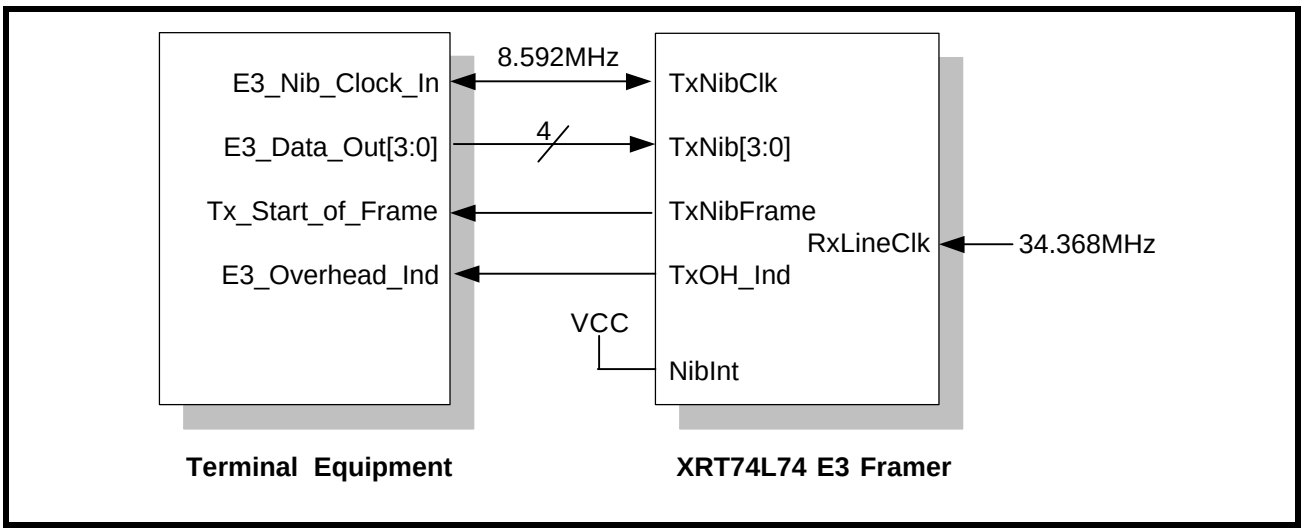
The E3 Frame consists of 537 bytes or 1074 nibbles. Therefore, the XRT74L74 will supply 1074 TxNibClk pulses between the rising edges of two consecutive TxNibFrame pulses. The E3 Frame repetition rate is 8.0kHz. Hence, 1074 TxNibClk pulses for each E3 frame period amounts to TxNibClk running at approximately 8.592 MHz. The method by which the 1074 TxNibClk pulses are distributed throughout the E3 frame period is presented below.

Nominally, the Transmit Section within the XRT74L74 will generate a TxNibClk pulse for every 4 RxOutClk (or TxInClk) periods.

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 4 Operation

Figure 170 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 4 Operation.

FIGURE 170. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMED) OPERATION



Mode 4 Operation of the Terminal Equipment

When the XRT74L74 is operating in this mode, it will function as the source of the 8.592MHz (e.g., the 34.368MHz clock signal divided by 4) clock signal that will be used as the Terminal Equipment Interface clock by both the XRT74L74 and the Terminal Equipment.

The Terminal Equipment will output the payload data of the Outbound E3 data stream via its E3_Data_Out[3:0] pins on the rising edge of the 8.592MHz clock signal at the E3_Nib_Clock_In input pin.

The XRT74L74 will latch the Outbound E3 data stream (from the Terminal Equipment) on the rising edge of the TxNibClk output clock signal. The XRT74L74 will indicate that it is processing the last nibble, within a given E3 frame, by pulsing its TxNibFrame output pin "High" for one TxNibClk clock period. When the Terminal Equipment detects a pulse at its Tx_Start_of_Frame input pin, it is expected to transmit the first nibble, of the very next Outbound E3

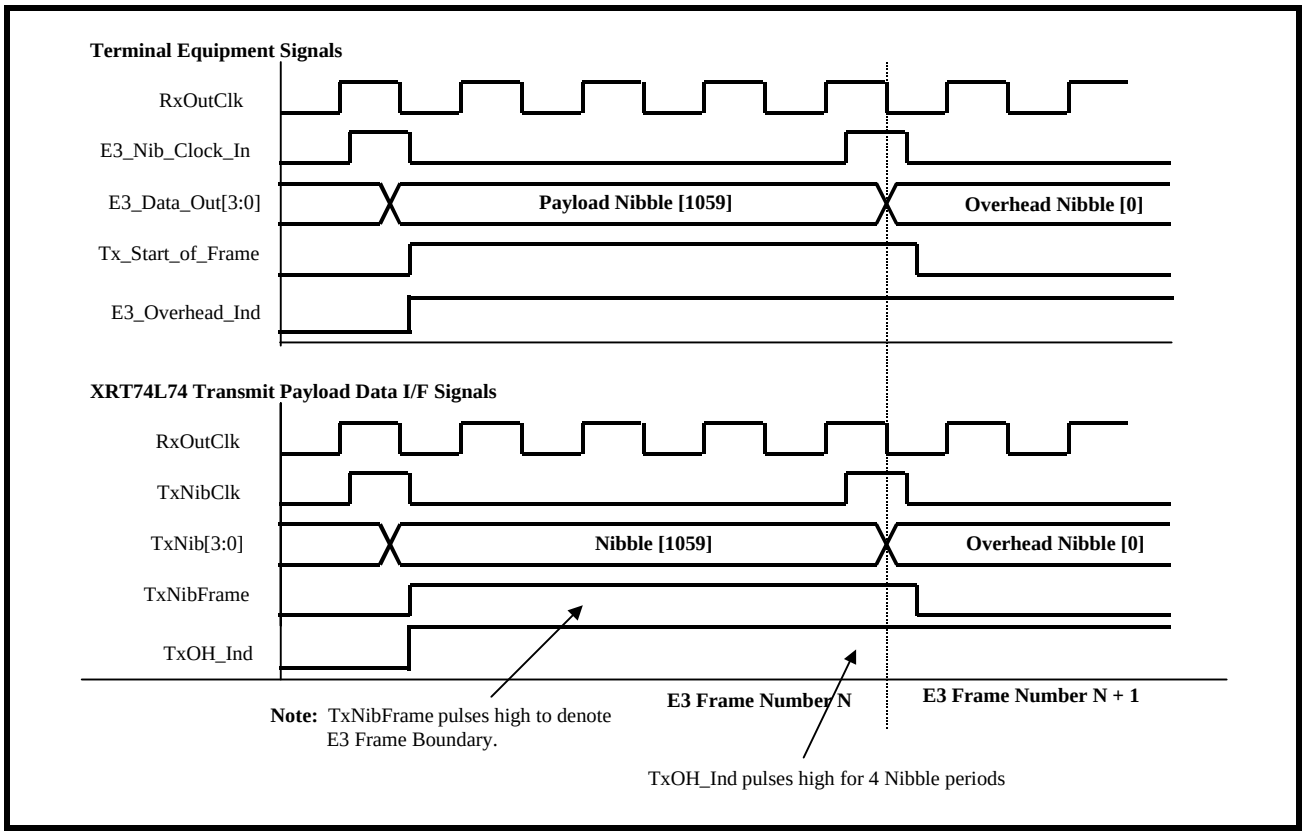
frame to the XRT74L74 via the E3_Data_Out[3:0] (or TxNib[3:0] pins).

Finally, for the Nibble-Parallel Mode operation, the XRT74L74 will pulse the TxOHInd output pin "High" for a total of 14 nibble periods (e.g., for the 7 overhead bytes, within each of the E3, ITU-T G.832 frames). At the beginning of an E3 frame, the XRT74L74 will pulse the TxOHInd output pin "High" for 4 nibble periods. These four nibbles represent the "FA1" and "FA2" bytes within each E3 frame.

Throughout the remainder of the E3 framing period, the XRT74L74 will pulse the TxOHInd output pin 5 times. The width (or duration) of each of these pulses will be two nibbles. Clearly, each of these 5 pulses corresponds to the five remaining overhead bytes, within the E3, ITU-T G.832 framing structure.

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 4 Operation is illustrated in Figure 171 .

FIGURE 171. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (MODE 4 OPERATION)



How to configure the XRT74L74 into Mode 4

1. Set the NibIntf input pin "High".

2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "00" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 170 .

7.2.1.5 Mode 5 - The Nibble-Parallel/Local-Time/Frame-Slave Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local Timing - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the DS3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of Outbound E3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will use the TxFrameRef input signal as its Framing Reference (e.g., the Transmit Section of the XRT74L74 initiates frame generation upon the rising edge of the TxFrameRef signal).

D. Sampling of payload data, from the Terminal Equipment

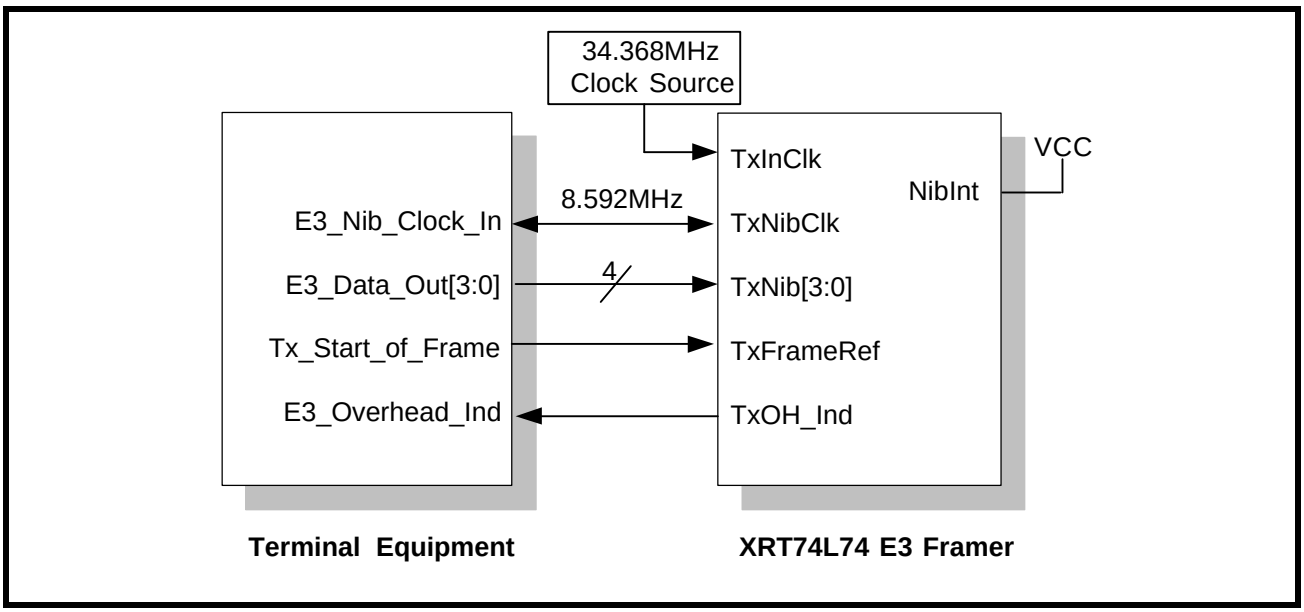
In Mode 5, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 173).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 8.592 MHz (e.g., 34.368 MHz divided by 4).*

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 5 Operation

Figure 172 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 5 Operation.

FIGURE 172. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIME/FRAME-SLAVE) OPERATION



Mode 5 Operation of the Terminal Equipment

In Figure 172 both the Terminal Equipment and the XRT74L74 will be driven by an external 8.592MHz clock signal. The Terminal Equipment will receive the 8.592MHz clock signal via the E3_Nib_Clock_In input pin. The XRT74L74 will output the 8.592MHz clock signal via the TxNibClk output pin.

The Terminal Equipment will serially output the data on the E3_Data_Out[3:0] pins, upon the rising edge of the signal at the E3_Clock_In input pin.

NOTE: *The E3_Data_Out[3:0] output pins of the Terminal Equipment is electrically connected to the TxNib[3:0] input pins.*

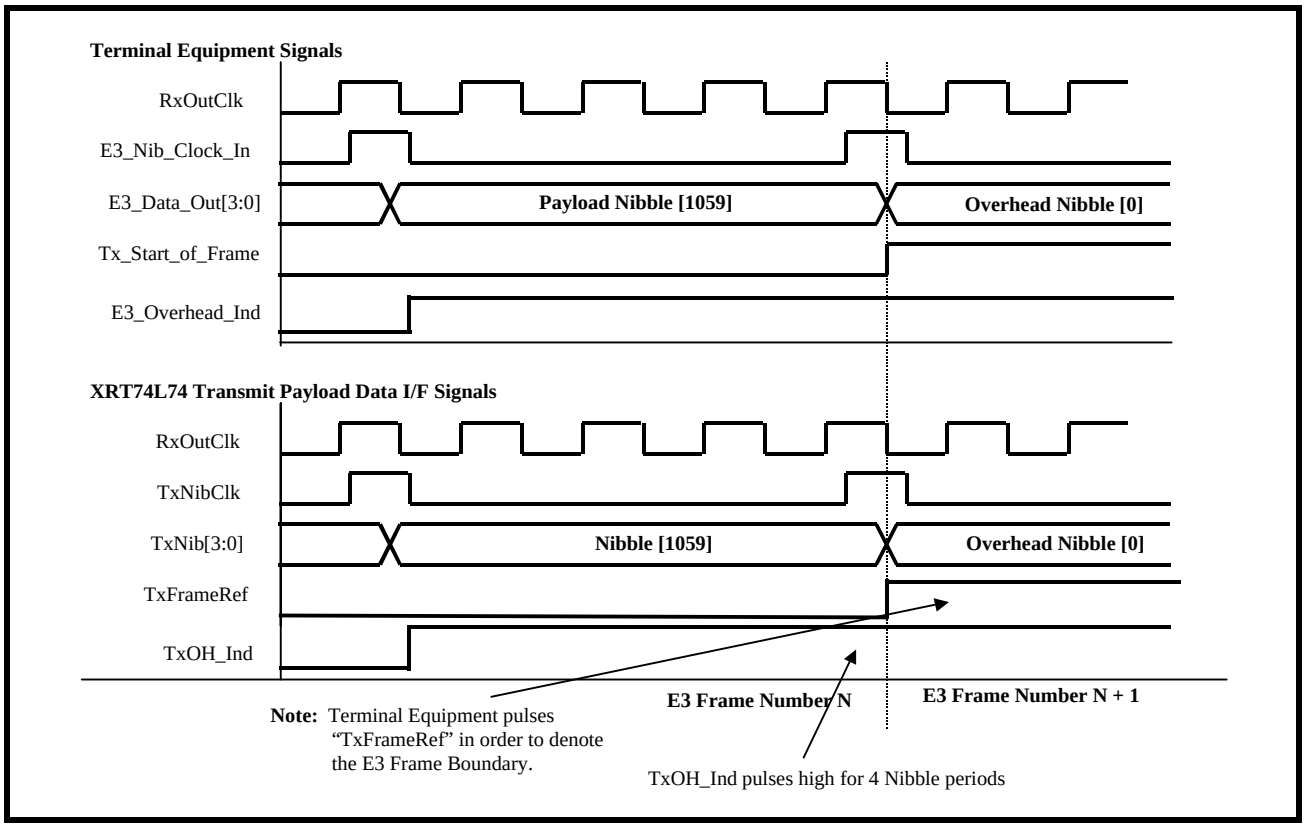
The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case, the Terminal Equipment has the responsibility of providing the framing reference signal by pulsing the Tx_Start_of_Frame output pin (and in turn, the TxFrameRef input pin of the XRT74L74) "High" for one bit-period, coincident with the first bit of a new E3 frame. Once the XRT74L74 detects the rising edge of the input at its TxFrameRef input pin, it will begin generation of a new E3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the E3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 5 Operation is illustrated in Figure 173 .

FIGURE 173. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3 MODE 5 OPERATION)



How to configure the XRT74L74 into Mode 5

1. Set the NibIntf input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "01" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 172 .

7.2.1.6 Mode 6 - The Nibble-Parallel/Local-Timed/Frame-Master Interface Mode Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will function as follows:

A. Local Timing - Uses the TxInClk signal as the Timing Reference

In this mode, the Transmit Section of the XRT74L74 will use the TxInClk signal at its timing reference. Further, the chip will internally divide the TxInClk clock signal by a factor of 4 and will output this divided clock signal via the TxNibClk output pin. The Transmit Terminal Equipment Input Interface block (within the XRT74L74) will use the rising edge of the TxNibClk signal, to latch the data, residing on the TxNib[3:0] into its circuitry.

B. Nibble-Parallel Mode

The XRT74L74 will accept the E3 payload data, from the Terminal Equipment, in a parallel manner, via the TxNib[3:0] input pins. The Transmit Terminal Equipment Input Interface will latch this data into its circuitry, on the rising edge of the TxNibClk output signal.

C. Delineation of Outbound E3 Frames

The Transmit Section will use the TxInClk input signal as its timing reference and will initiate the generation of E3 frames, asynchronous with respect to any external signal. The XRT74L74 will pulse the TxFrame output pin "High" whenever it is processing the last bit, within a given Outbound E3 frame.

D. Sampling of payload data, from the Terminal Equipment

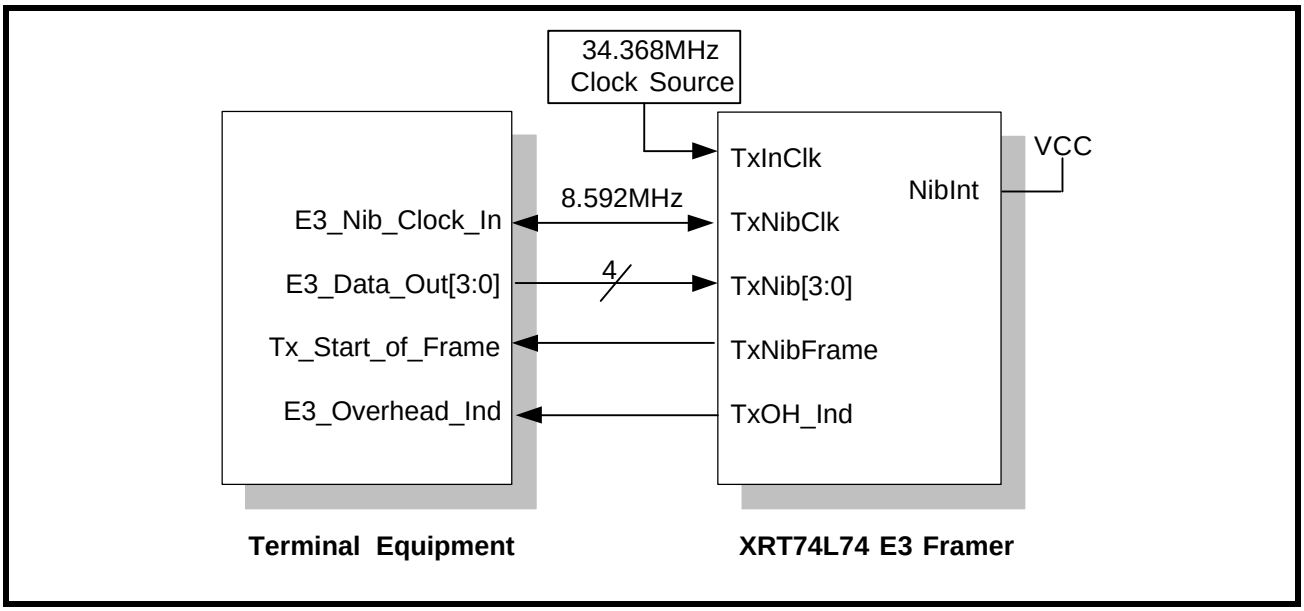
In Mode 6, the XRT74L74 will sample the data, at the TxNib[3:0] input pins, on the third rising edge of the TxInClk clock signal, following a pulse in the TxNibClk signal (see Figure 175).

NOTE: *The TxNibClk signal, from the XRT74L74 operates nominally at 8.592 MHz (e.g., 34.368 MHz divided by 4).*

Interfacing the Transmit Payload Data Input Interface block of the XRT74L74 to the Terminal Equipment for Mode 6 Operation

Figure 174 presents an illustration of the Transmit Payload Data Input Interface block (within the XRT74L74) being interfaced to the Terminal Equipment, for Mode 6 Operation.

FIGURE 174. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT74L74 FOR MODE 6 OPERATION



Mode 6 Operation of the Terminal Equipment

In Figure 174 both the Terminal Equipment and the XRT74L74 will be driven by an external 8.592MHz clock signal. The Teriminal Equipment will receive the 8.592MHz clock signal via the E3_Nib_Clock_In input pin. The XRT74L74 will output the 8.592MHz clock signal via the TxNibClk output pin.

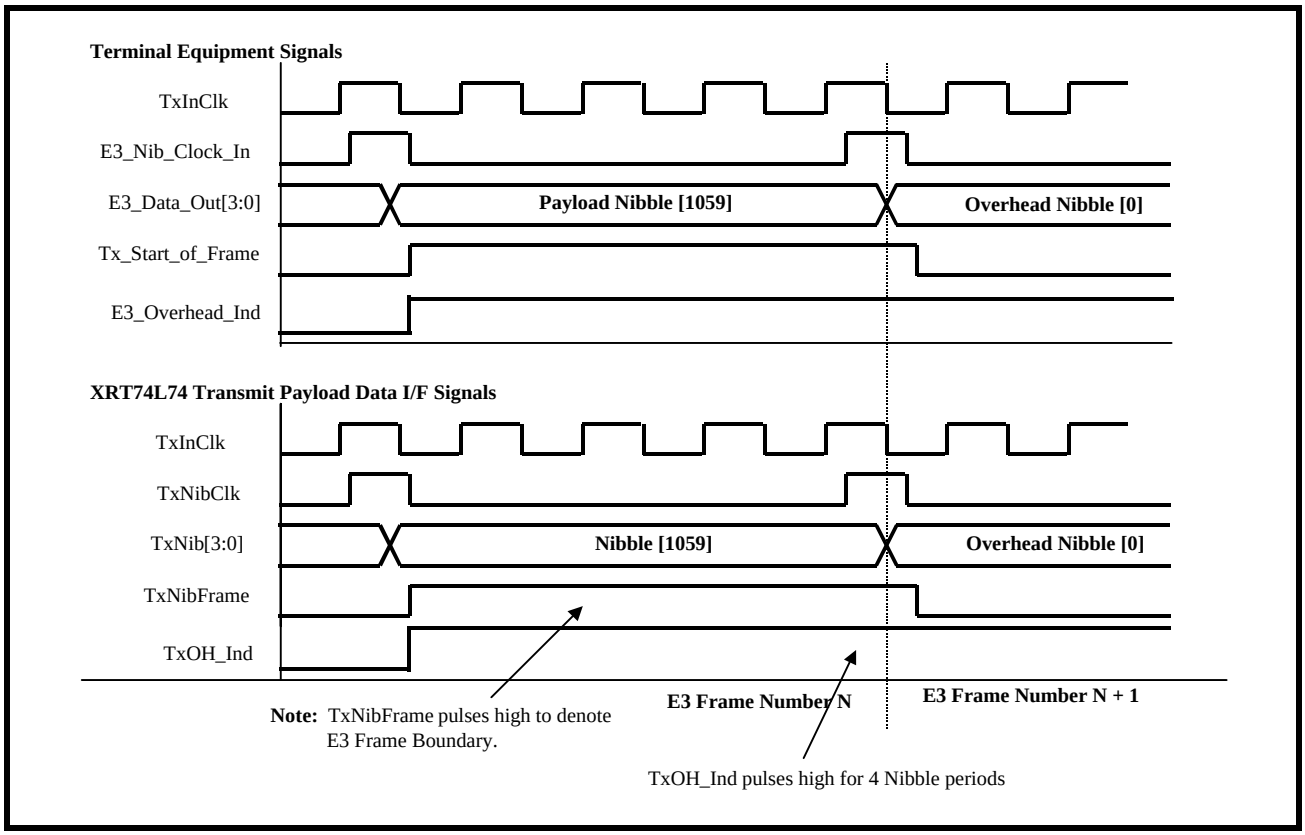
The Terminal Equipment will serially output the data on the E3_Data_Out[3:0] pins upon the rising edge of the signal at the E3_Clock_In input pin. The XRT74L74 will latch the data, residing on the TxNib[3:0] input pins, on the rising edge of the TxNibClk signal.

In this case the XRT74L74 has the responsibility of providing the framing reference signal by pulsing the TxFrame output pin (and in turn the Tx_Start_of_Frame input pin of the Terminal Equipment) "High" for one bit-period, coincident with the last bit within a given E3 frame.

Finally, the XRT74L74 will always internally generate the Overhead bits, when it is operating in both the E3 and Nibble-parallel modes. The XRT74L74 will pull the TxOHInd input pin "Low".

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Mode 6 Operation is illustrated in Figure 175 .

FIGURE 175. BEHAVIOR OF THE TERMINAL INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT (E3 MODE 6 OPERATION)



How to configure the XRT74L74 into Mode 6

1. Set the NibInt input pin "High".
2. Set the TimRefSel[1:0] bit-fields (within the Framer Operating Mode Register) to "1X" as illustrated below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

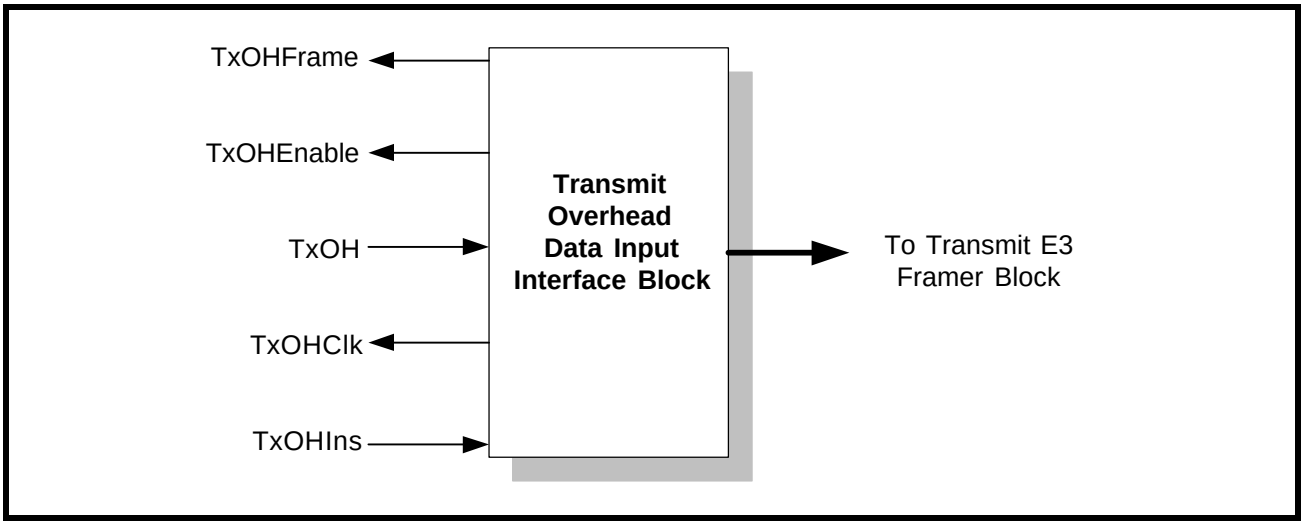
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	x

3. Interface the XRT74L74, to the Terminal Equipment, as illustrated in Figure 174 .

7.2.2 The Transmit Overhead Data Input Interface

Figure 176 presents a simple illustration of the Transmit Overhead Data Input Interface block within the XRT74L74.

FIGURE 176. THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK



The E3, ITU-T G.832 Frame consists of 537 bytes. Of these bytes, 530 bytes are payload bytes and the remaining 7 are overhead bytes. The XRT74L74 has been designed to handle and process both the payload type and overhead type bits for each E3 frame. Within the Transmit Section within the XRT74L74, the Transmit Payload Data Input Interface has been designed to handle the payload data. Likewise, the Transmit Overhead Input Interface has been designed to handle and process the overhead bits.

The Transmit Section of the XRT74L74 generates or processes the various overhead bits within the E3 frame, in the following manner.

The Frame Alignment Overhead Bytes (e.g., the "FA1" and "FA2" bytes)

The "FA1" and "FA2" bytes are always internally generated by the Transmit Section of the XRT74L74. Hence, the user cannot insert his/her value for the "FA1" and "FA2" bytes into the Outbound DS3 data stream, via the Transmit Overhead Data Input Interface.

The Error Monitoring (EM) Overhead Byte

The EM byte is always internally generated by the Transmit Section of the XRT74L74. Hence, the user cannot insert his/her value for the EM byte into the Outbound E3 data stream, via the Transmit Overhead Data Input Interface.

The Alarm and signaling related Overhead bytes

Bytes that are used to transport the alarm conditions can be either internally generated by the Transmit Section within the XRT74L74, or can be externally generated and inserted into the Outbound E3 data stream, via the Transmit Overhead Data Input Interface. The E3 frame overhead bits that fall into this category are:

- The "MA" byte
- The "TR" byte

The Data Link Related Overhead Bits

The E3 frame structure also contains bits which can be used to transport User Data Link information and Path Maintenance Data Link information. The UDL (User Data Link) bits are only accessible via the Transmit Overhead Data Input Interface. The Path Maintenance Data Link (PMDL) bits can either be sourced from the Transmit LAPD Controller/Buffer or via the Transmit Overhead Data Input Interface.

Table 87 lists the Overhead Bits within the DS3 frame. Additionally, this table also indicates whether or not these overhead bits can be sourced by the Transmit Overhead Data Input Interface or not.

TABLE 87: THE OVERHEAD BITS WITHIN THE E3 FRAME AND THEIR POTENTIAL SOURCES

OVERHEAD BIT	INTERNALLY GENERATED	ACCESSIBLE VIA THE TRANSMIT OVERHEAD DATA INPUT INTERFACE	BUFFER/REGISTER ACCESSIBLE
FA1 - Bit 7	Yes	No	Yes*
FA1 - Bit 6	Yes	No	Yes
FA1 - Bit 5	Yes	No	Yes*
FA1 - Bit 4	Yes	No	Yes*
FA1 - Bit 3	Yes	No	Yes
FA1 - Bit 2	Yes	No	Yes
FA1 - Bit 1	Yes	No	Yes+
FA1 - Bit 0	Yes	No	Yes
FA2 - Bit 7	Yes	No	Yes
FA2 - Bit 6	Yes	No	Yes
FA2 - Bit 5	Yes	No	Yes
FA2 - Bit 4	Yes	No	Yes
FA2 - Bit 3	Yes	No	Yes
FA2 - Bit 2	Yes	No	Yes
FA2 - Bit 1	Yes	No	Yes
FA2 - Bit 0	Yes	No	Yes
EM - Bit 7	Yes	Yes	Yes
EM - Bit 6	Yes	Yes	Yes
EM - Bit 5	Yes	Yes	Yes
EM - Bit 4	Yes	Yes	Yes
EM - Bit 3	Yes	Yes	Yes
EM - Bit 2	Yes	Yes	Yes
EM - Bit 1	Yes	Yes	Yes
EM - Bit 0	Yes	Yes	Yes
TR - Bit 7	No	Yes	Yes
TR - Bit 6	No	Yes	Yes
TR - Bit 5	No	Yes	Yes
TR - Bit 4	No	Yes	Yes
TR - Bit 3	No	Yes	Yes
TR - Bit 2	No	Yes	Yes
TR - Bit 1	No	Yes	Yes

TABLE 87: THE OVERHEAD BITS WITHIN THE E3 FRAME AND THEIR POTENTIAL SOURCES

OVERHEAD BIT	INTERNALLY GENERATED	ACCESSIBLE VIA THE TRANSMIT OVERHEAD DATA INPUT INTERFACE	BUFFER/REGISTER ACCESSIBLE
TR - Bit 0	No	Yes	Yes
MA - Bit 7	Yes	Yes	Yes
MA - Bit 6	Yes	Yes	Yes
MA - Bit 5	Yes	Yes	Yes
MA - Bit 4	Yes	Yes	Yes
MA - Bit 3	Yes	Yes	Yes
MA - Bit 2	Yes	Yes	Yes
MA - Bit 1	Yes	Yes	Yes
MA - Bit 0	Yes	Yes	Yes
NR - Bit 7	No	Yes	Yes
NR - Bit 6	No	Yes	Yes
NR - Bit 5	No	Yes	Yes
NR - Bit 4	No	Yes	Yes
NR - Bit 3	No	Yes	Yes
NR - Bit 2	No	Yes	Yes
NR - Bit 1	No	Yes	Yes
NR - Bit 0	No	Yes	Yes
GC - Bit 7	No	Yes	Yes
GC - Bit 6	No	Yes	Yes
GC - Bit 5	No	Yes	Yes
GC - Bit 4	No	Yes	Yes
GC - Bit 3	No	Yes	Yes
GC - Bit 2	No	Yes	Yes
GC - Bit 1	No	Yes	Yes
GC - Bit 0	No	Yes	Yes

NOTES:

1. *The XRT74L74 contains mask register bits that permit the user to alter the state of the internally generated value for these bits.*
2. *The Transmit LAPD Controller/Buffer can be configured to be the source of the DL bits, within the Outbound E3 data stream.*

In all, the Transmit Overhead Data Input Interface permits the user to insert overhead data into the Outbound E3 frames via the following two different methods.

- Method 1 - Using the TxOHClk clock signal
- Method 2 - Using the TxInClk and the TxOHEnable signals.

Each of these methods are described below.

7.2.2.1 Method 1 - Using the TxOHClk Clock Signal

The Transmit Overhead Data Input Interface consists of the five signals. Of these five (5) signals, the following four (4) signals are to be used when implementing Method 1.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

Each of these signals are listed and described below. Table 88 .

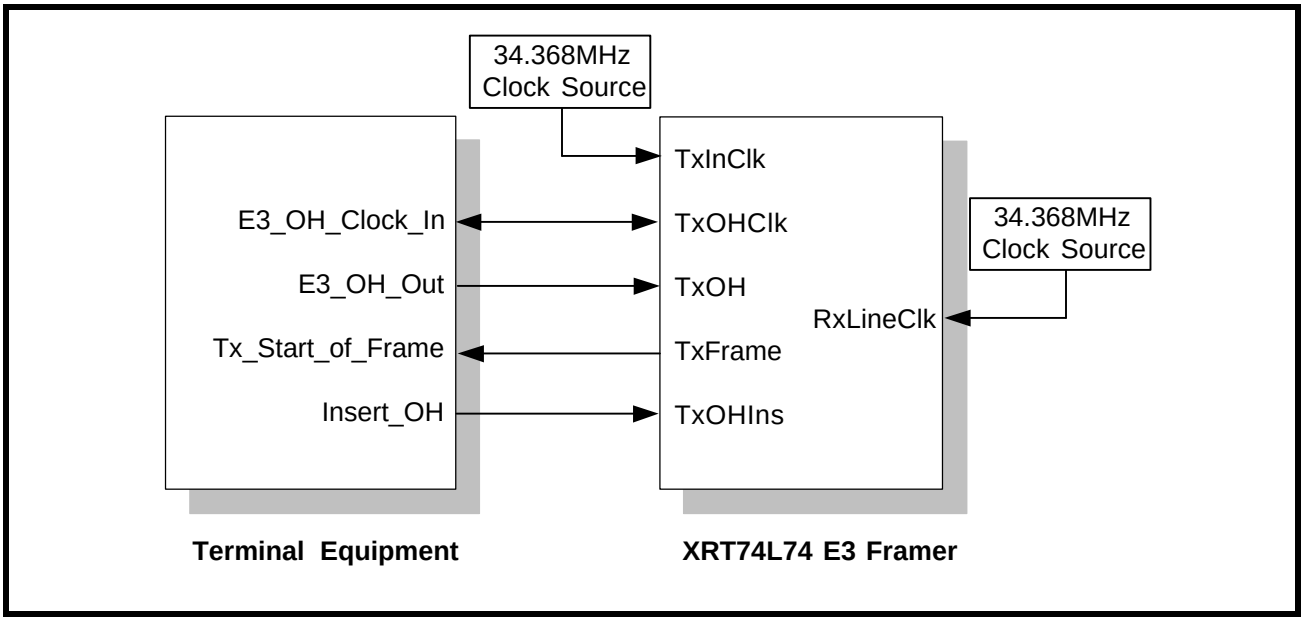
TABLE 88: DESCRIPTION OF METHOD 1 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next Outbound E3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.
TxOHClk	Output	**Transmit Overhead Input Interface Clock Output signal:** This output signal serves two purposes: 1. The Transmit Overhead Data Input Interface will provide a rising clock edge on this signal, one bit-period prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit. 2. The Transmit Overhead Data Input Interface will sample the data at the TxOH input, on the falling edge of this clock signal (provided that the TxOHIns input pin is "High"). ***NOTE:*** *The Transmit Overhead Data Input Interface will supply a clock edge for all overhead bits within the DS3 frame (via the TxOHClk output signal). This includes those overhead bits that the Transmit Overhead Data Input Interface will not accept from the Terminal Equipment.*
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given E3 frame. The purpose of this output signal is to alert the Terminal Equipment that the Transmit Overhead Data Input Interface block is about to begin processing the overhead bits for a new E3 frame.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment.

Figure 177 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment, when using Method 1.

FIGURE 177. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 1)



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the Outbound E3 data stream, (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of the TxOHFrame signal (e.g., the Tx_Start_of_Frame input signal) on the rising edge of the TxOHClk (e.g., the E3_OH_Clock_In signal).
2. To keep track of the number of rising clock edges that have occurred, via the TxOHClk (e.g., the E3_OH_Clock_In signal) since the last time the TxOHFrame signal was sampled "High". By doing this the Terminal Equipment will be able to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface block at any given time. When the Terminal Equipment knows which overhead bit is being processed, at a given TxOHClk period, it will know when to insert a desired overhead bit value into the Outbound E3 data stream. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin (of the XRT74L74).

Table 89 relates the number of rising clock edges (in the TxOHClk signal, since TxOHFrame was sampled "High") to the E3 Overhead Bit, that is being processed.

TABLE 89: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN TXOHCLK, (SINCE "TXOHFRAME" WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED

NUMBER OF RISING CLOCK EDGES IN TXOHCLK	THE OVERHEAD BIT EXPECTED BY THE "XRT74L74"	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (Clock edge is coincident with TxOHFrame being detected “High”)	FA1 Byte - Bit 7	No
1	FA1 Byte - Bit 6	No
2	FA1 Byte - Bit 5	No
3	FA1 Byte - Bit 4	No
4	FA1 Byte - Bit 3	No
5	FA1 Byte - Bit 2	No
6	FA1 Byte - Bit 1	No
7	FA1 Byte - Bit 0	No
8	FA2 Byte - Bit 7	No
9	FA2 Byte - Bit 6	No
10	FA2 Byte - Bit 5	No
11	FA2 Byte - Bit 4	No
12	FA2 Byte - Bit 3	No
13	FA2 Byte - Bit 2	No
14	FA2 Byte - Bit 1	No
15	FA2 Byte - Bit 0	No
16	EM Byte - Bit 7	No
17	EM Byte - Bit 6	No
18	EM Byte - Bit 5	No
19	EM Byte - Bit 4	No
20	EM Byte - Bit 3	No
21	EM Byte - Bit 2	No
22	EM Byte - Bit 1	No
23	EM Byte - Bit 0	No
24	TR Byte - Bit 7	Yes
25	TR Byte - Bit 6	Yes
26	TR Byte - Bit 5	Yes
27	TR Byte - Bit 4	Yes
28	TR Byte - Bit 3	Yes
29	TR Byte - Bit 2	Yes

TABLE 89: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN TXOHCLK, (SINCE "TXOHFRAME" WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED

NUMBER OF RISING CLOCK EDGES IN TXOHCLK	THE OVERHEAD BIT EXPECTED BY THE "XRT74L74"	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
30	TR Byte - Bit 1	Yes
31	TR Byte - Bit 0	Yes
32	MA Byte - Bit 7	Yes (FERF Bit)
33	MA Byte - Bit 6	Yes (FEBE Bit)
34	MA Byte - Bit 5	Yes
35	MA Byte - Bit 4	Yes
36	MA Byte - Bit 3	Yes
37	MA Byte - Bit 2	Yes
38	MA Byte - Bit 1	Yes
39	MA Byte - Bit 0	Yes
40	NR Byte - Bit 7	Yes
41	NR Byte - Bit 6	Yes
42	NR Byte - Bit 5	Yes
43	NR Byte - Bit 4	Yes
44	NR Byte - Bit 3	Yes
45	NR Byte - Bit 2	Yes
46	NR Byte - Bit 1	Yes
47	NR Byte - Bit 0	Yes
48	GC Byte - Bit 7	Yes
49	GC Byte - Bit 6	Yes
50	GC Byte - Bit 5	Yes
51	GC Byte - Bit 4	Yes
52	GC Byte - Bit 3	Yes
53	GC Byte - Bit 2	Yes
54	GC Byte - Bit 1	Yes
55	GC Byte - Bit 0	Yes

3. After the Terminal Equipment has waited the appropriate number of clock edges (from the TxOHFrame signal being sampled "High"), it should assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal, stable until the next rising edge of TxOHClk is detected.

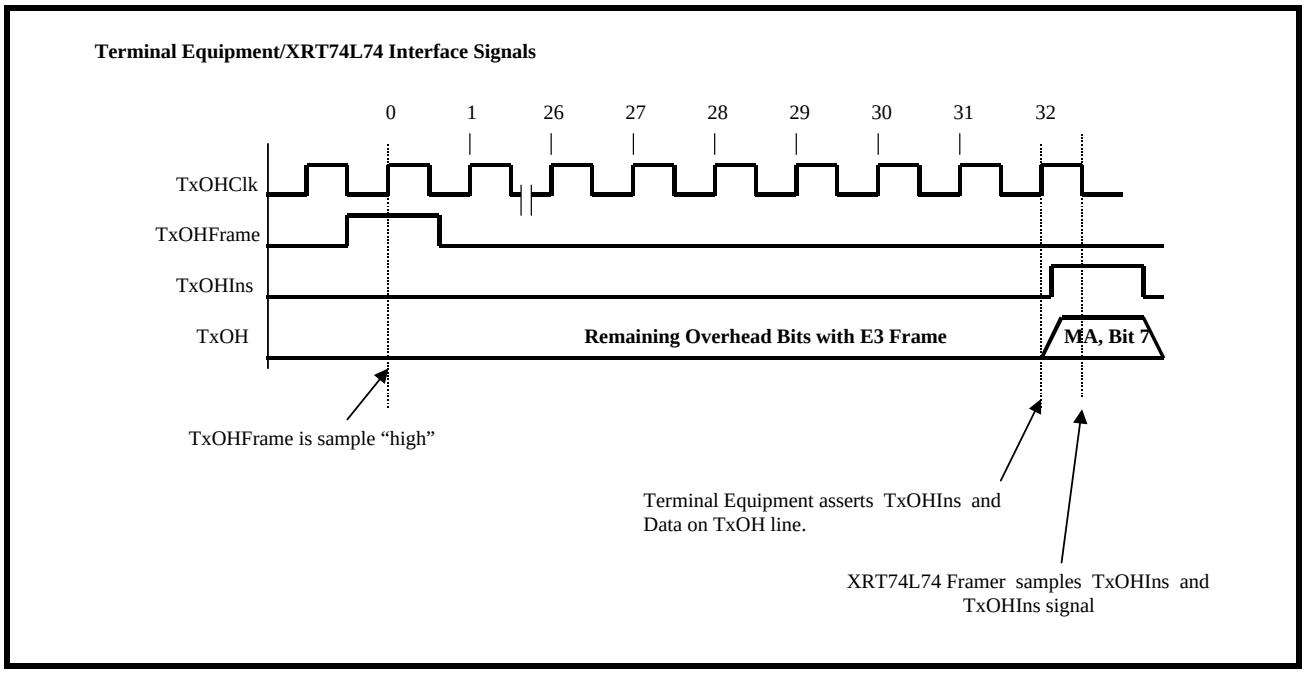
Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using

Method 1) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this example, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface, such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for E3 Applications, a Yellow Alarm is transmitted by setting the FERF bit (within the MA Byte) to "0".

If one assumes that the connection between the Terminal Equipment and the XRT74L74 are as illustrated in Figure 177 then Figure 178 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 178. ILLUSTRATION OF THE SIGNAL THAT MUST OCCUR BETWEEN THE TERMINAL EQUIPMENT AND THE XRT74L74, IN ORDER TO CONFIGURE THE XRT74L74 TO TRANSMIT A YELLOW ALARM TO THE REMOTE TERMINAL EQUIPMENT



In Figure 178 the Terminal Equipment samples the TxOHFrame signal being "High" at rising clock edge # “0". From this point, the Terminal Equipment waits until it has detected 32 rising edges in the TxOHClk signal. At this point, the Terminal Equipment knows that the XRT74L74 is just about to process the FERF bit within the MA byte (in a given Outbound E3 frame). Additionally, according to Table 89 , the 32nd overhead bit to be processed is the FERF bit. In order to facilitate the transmission of the Yellow Alarm, the Terminal Equipment must set this FERF bit to "1". Hence, the Terminal Equipment starts this process by implementing the following steps concurrently.

a. Assert the TxOHIns input pin by setting it "High".

b. Set the TxOH input pin to "0".

After the Terminal Equipment has applied these signals, the XRT74L74 will sample the data on both the TxOHIns and TxOH signals upon the very next falling edge of TxOHClk (designated at 32- in Figure 178). Once the XRT74L74 has sampled this data, it will then insert a "1" into the FERF bit position, in the Outbound E3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as clock edge 1 in Figure 178), the Terminal Equipment will negate the TxOHIns signal (e.g., toggles it "Low") and will cease inserting data into the Transmit Overhead Data Input Interface.

7.2.2.2 Method 2 - Using the TxInClk and TxOHEnable Signals

Method 1 requires the use of an additional clock signal, TxOHClk. However, there may be a situation in which the user does not wish to add this extra clock signal to their design, in order to use the Transmit Overhead Data Input Interface. Hence, Method 2 is available. When using Method 2, either the TxInClk or RxOutClk signal is used to sample the overhead bits and signals which are input to the Transmit Overhead Data Input Interface. Method 2 involves the use of the following signals:

- TxOH
- TxInClk
- TxOHFrame
- TxOHEnable

Each of these signals are listed and described in Table 90 .

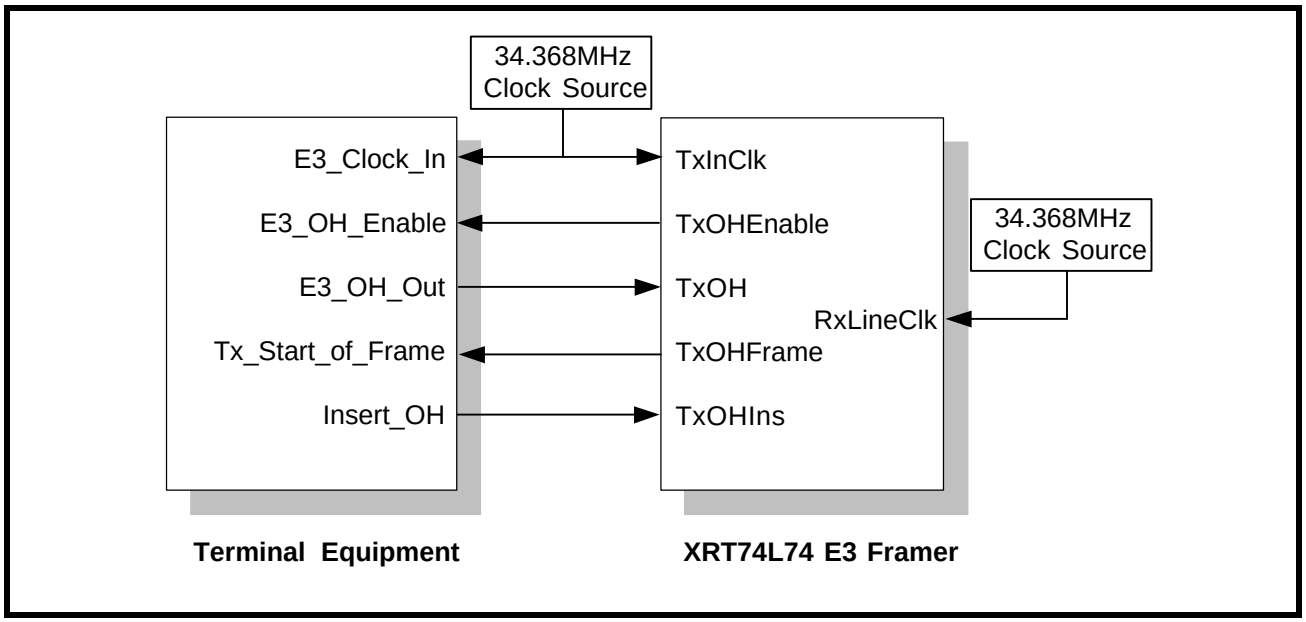
TABLE 90: DESCRIPTION OF METHOD 1 TRANSMIT OVERHEAD INPUT INTERFACE SIGNALS

NAME	TYPE	DESCRIPTION
TxOHEnable	Output	**Transmit Overhead Data Enable Output pin** The XRT74L74 will assert this signal, for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface is processing an overhead bit.
TxOHFrame	Output	**Transmit Overhead Input Interface Frame Boundary Indicator Output:** This output signal pulses "High" when the XRT74L74 is processing the last bit within a given DS3 frame.
TxOHIns	Input	**Transmit Overhead Data Insert Enable input pin.** Asserting this input signal (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data at the TxOH input pin, on the falling edge of the TxInClk output signal. Conversely, setting this pin "Low" configures the Transmit Overhead Data Input Interface to NOT sample (e.g., ignore) the data at the TxOH input pin, on the falling edge of the TxOHClk output signal. ***NOTE:*** *If the Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface (e.g., if the Terminal Equipment asserts the TxOHIns signal, at a time when one of these non-insertable overhead bits are being processed), that particular insertion effort will be ignored.*
TxOH	Input	**Transmit Overhead Data Input pin:** The Transmit Overhead Data Input Interface accepts the overhead data via this input pin, and inserts into the overhead bit position within the very next Outbound DS3 frame. If the TxOHIns pin is pulled "High", the Transmit Overhead Data Input Interface will sample the data at this input pin (TxOH), on the falling edge of the TxOHClk output pin. Conversely, if the TxOHIns pin is pulled "Low", then the Transmit Overhead Data Input Interface will NOT sample the data at this input pin (TxOH). Consequently, this data will be ignored.

Interfacing the Transmit Overhead Data Input Interface to the Terminal Equipment

Figure 179 illustrates how one should interface the Transmit Overhead Data Input Interface to the Terminal Equipment when using Method 2.

FIGURE 179. THE TERMINAL EQUIPMENT BEING INTERFACED TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE (METHOD 2)



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to insert any overhead data into the Outbound E3 data stream (via the Transmit Overhead Data Input Interface), then it is expected to do the following.

1. To sample the state of both the TxOHFrame and the TxOHEnable input signals, via the E3_Clock_In (e.g., either the TxInClk or the RxOutClk signal of the XRT74L74) signal. If the Terminal Equipment samples the TxOHEnable signal "High", then it knows that the XRT74L74 is about to process an overhead bit. Further, if the Terminal Equipment samples both the TxOHFrame and the TxOHEnable pins "High" (at the same time) then the Terminal Equipment knows that the XRT74L74 is about to process the first overhead bit, within a new E3 frame.
2. To keep track of the number of times that the TxOHEnable signal has been sampled "High" since the last time both the TxOHFrame and the TxOHEnable signals were sampled "High". By doing this, the Terminal Equipment will be able to keep track of which overhead bit the Transmit Overhead Data Input Interface is about ready to process. From this, the Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pins of the XRT74L74.

Table 91 also relates the number of TxOHEnable output pulses (that have occurred since both the TxOHFrame and TxOHEnable pins were sampled "High") to the E3 overhead bit, that is being processed.

TABLE 91: THE RELATIONSHIP BETWEEN THE NUMBER OF TXOHENABLE PULSES (SINCE THE LAST OCCURRENCE OF THE TXOHFRAME PULSE) TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED BY THE XRT74L74

NUMBER OF TXOHENABLE PULSES	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
0 (Clock edge is coincident with TxOHFrame being detected “High”)	FA1 Byte - Bit 7	Yes
1	FA1 Byte - Bit 6	No
2	FA1 Byte - Bit 5	No
3	FA1 Byte - Bit 4	No
4	FA1 Byte - Bit 3	No
5	FA1 Byte - Bit 2	No
6	FA1 Byte - Bit 1	No
7	FA1 Byte - Bit 0	No
8	FA2 Byte - Bit 7	No
9	FA2 Byte - Bit 6	No
10	FA2 Byte - Bit 5	No
11	FA2 Byte - Bit 4	No
12	FA2 Byte - Bit 3	No
13	FA2 Byte - Bit 2	No
14	FA2 Byte - Bit 1	No
15	FA2 Byte - Bit 0	No
16	EM Byte - Bit 7	No
17	EM Byte - Bit 6	No
18	EM Byte - Bit 5	No
19	EM Byte - Bit 4	No
20	EM Byte - Bit 3	No
21	EM Byte - Bit 2	No
22	EM Byte - Bit 1	No
23	EM Byte - Bit 0	No
24	TR Byte - Bit 7	Yes
25	TR Byte - Bit 6	Yes
26	TR Byte - Bit 5	Yes
27	TR Byte - Bit 4	Yes
28	TR Byte - Bit 3	Yes
29	TR Byte - Bit 2	Yes

TABLE 91: THE RELATIONSHIP BETWEEN THE NUMBER OF TXOHENABLE PULSES (SINCE THE LAST OCCURRENCE OF THE TXOHFRAME PULSE) TO THE E3 OVERHEAD BIT, THAT IS BEING PROCESSED BY THE XRT74L74

NUMBER OF TXOHENABLE PULSES	THE OVERHEAD BIT EXPECTED BY THE XRT74L74	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT74L74?
30	TR Byte - Bit 1	Yes
31	TR Byte - Bit 0	Yes
32	MA Byte - Bit 7 (FERF)	Yes
33	MA Byte - Bit 6 (FEBE)	Yes
34	MA Byte - Bit 5	Yes
35	MA Byte - Bit 4	Yes
36	MA Byte - Bit 3	Yes
37	MA Byte - Bit 2	Yes
38	MA Byte - Bit 1	Yes
39	MA Byte - Bit 0	Yes
40	NR Byte - Bit 7	Yes
41	NR Byte - Bit 6	Yes
42	NR Byte - Bit 5	Yes
43	NR Byte - Bit 4	Yes
44	NR Byte - Bit 3	Yes
45	NR Byte - Bit 2	Yes
46	NR Byte - Bit 1	Yes
47	NR Byte - Bit 0	Yes
48	GC Byte - Bit 7	Yes
49	GC Byte - Bit 6	Yes
50	GC Byte - Bit 5	Yes
51	GC Byte - Bit 4	Yes
52	GC Byte - Bit 3	Yes
53	GC Byte - Bit 2	Yes
54	GC Byte - Bit 1	Yes
55	GC Byte - Bit 0	Yes

3. After the Terminal Equipment has waited through the appropriate number of pulses via the TxOHEnable pin, it should then assert the TxOHIns input signal. Concurrently, the Terminal Equipment should also place the appropriate value (of the inserted overhead bit) onto the TxOH signal.
4. The Terminal Equipment should hold both the TxOHIns input pin "High" and the value of the TxOH signal stable, until the next TxOHEnable pulse is detected.

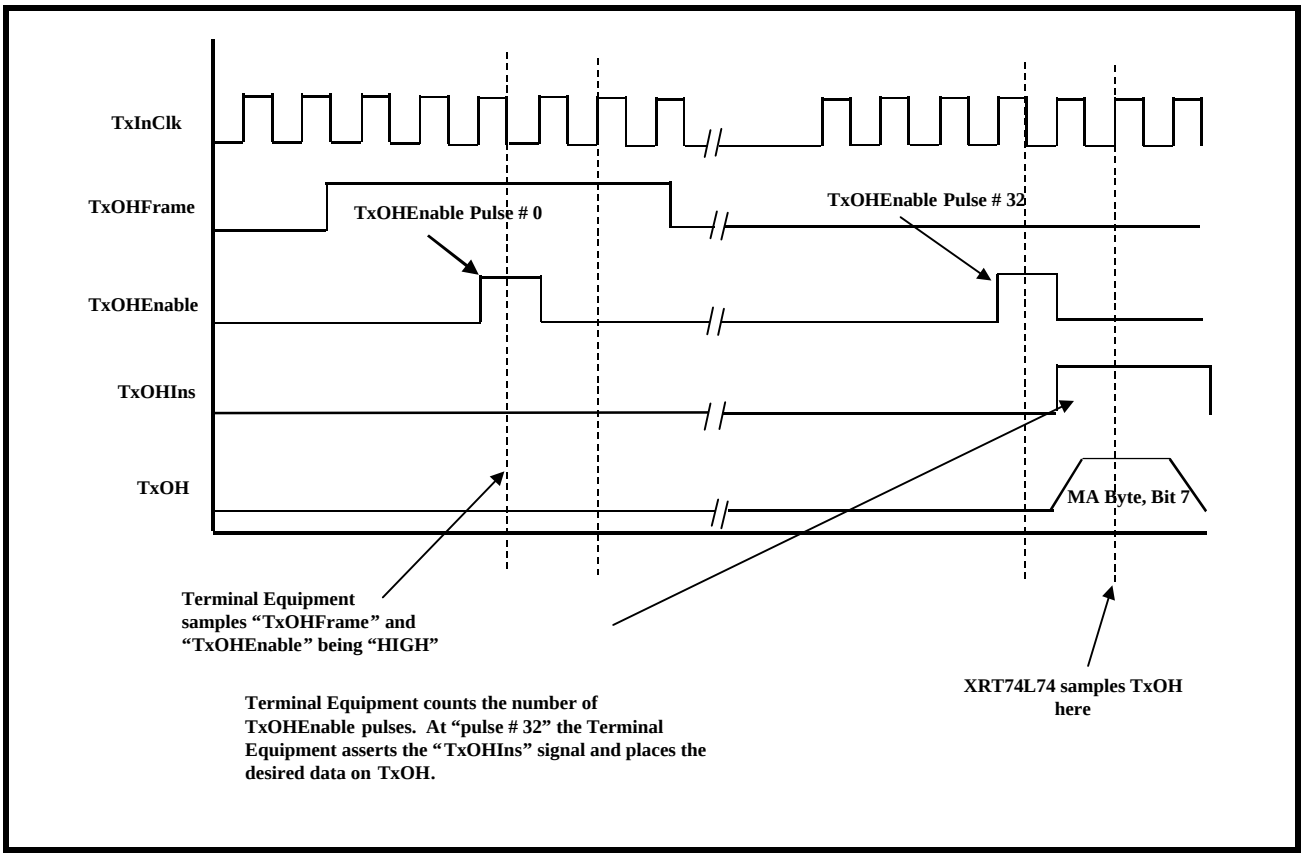
Case Study: The Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface (using

Method 2) in order to transmit a Yellow Alarm to the remote terminal equipment.

In this case, the Terminal Equipment intends to insert the appropriate overhead bits, into the Transmit Overhead Data Input Interface such that the XRT74L74 will transmit a Yellow Alarm to the remote terminal equipment. Recall that, for E3, ITU-T G.832 applications, a Yellow Alarm is transmitted by setting the FERF bit (within the MA byte) to "1".

If one assumes that the connection between the Terminal Equipment and the XRT74L74 is as illustrated in Figure 179 then, Figure 180 presents an illustration of the signaling that must go on between the Terminal Equipment and the XRT74L74.

FIGURE 180. BEHAVIOR OF TRANSMIT OVERHEAD DATA INPUT INTERFACE SIGNALS BETWEEN THE XRT74L74 AND THE TERMINAL EQUIPMENT FOR METHOD 2



7.2.3 The Transmit E3 HDLC Controller

The Transmit E3 HDLC Controller block can be used to transport Message-Oriented Signaling (MOS) type messages to the remote terminal equipment as discussed in detail below.

NOTE: *While executing this particular write operation, the user should write the binary value "000xx110b" into the Tx Controller block), please see Section 5.3.3.1.*

7.2.3.1 Message-Oriented Signaling (e.g., LAP-D) processing via the Transmit DS3 HDLC Controller

The LAPD Transmitter (within the Transmit E3 HDLC Controller Block) allows the user to transmit path maintenance data link (PMDL) messages to the remote terminal via the Outbound E3 Frames. In this case the message bits are either inserted into and carried by the "NR" or the "GC" bytes, within the Outbound E3 frames. The on-chip LAPD transmitter supports both the 76 byte and 82 byte length message formats, and the Framer IC allocates 88 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer) to store the message to be transmitted. The message format complies with ITU-T Q.921 (LAP-D) protocol with different addresses and is presented below in Figure 181 .

FIGURE 181. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following sections defines each of these bit/byte-fields within the LAPD Message Frame Format.

Flag Sequence Byte

The Flag Sequence byte is of the value 0x7E, 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 (decimal).

TEI - Terminal Endpoint Identifier

The TEI bit-fields are assigned the value of 0x00. The TEI field is used in N-ISDN systems to identify a terminal out of multiple possible terminal. However, since the Framer IC 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 Framer assigned the Control byte the value 03h. Hence, the Framer 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 0x86 through 0xDD).

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 = 0x86, within the Framer). The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. Table 92 presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT74L74 Framer device and the corresponding octet value that the user must write into to the first octet position within the Transmit LAPD Message buffer.

TABLE 92: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD

LAPD MESSAGE TYPE	VALUE OF FIRST BYTE, WITHIN INFORMATION PAYLOAD OF MESSAGE	MESSAGE SIZE
CL Path Identification	0x38	76 bytes
IDLE Signal Identification	0x34	76 bytes
Test Signal Identification	0x32	76 bytes
ITU-T Path Identification	0x3F	82 bytes

Frame Check Sequence Bytes

The 16 bit FCS (Frame Check Sequence) is calculated over the LAPD Message Header and Information Payload bytes, by using the CRC-16 polynomial, $x^{16} + x^{12} + x^5 + 1$.

Operation of the LAPD Transmitter

If a message is to be transmitted via the LAPD Transmitter then, the information portion (or the body) of the message must be written into the Transmit LAPD Message Buffer, which is located at 0x86 through 0xDD in on-chip RAM via the Microprocessor Interface. Afterwards, the user must do three things:

1. Specify the length of LAPD message to be transmitted.
2. Specify which bit-field (within the E3 frame) that the LAPD Message frame is to be transported on (e.g., either the “GC” or the “NR” byte).
3. Specify whether the LAPD Transmitter should transmit this LAPD Message frame only once, or an indefinite number of times at One-Second intervals.
4. Enable the LAPD Transmitter.
5. Initiate the Transmission of the PMDL Message.

Each of these steps will be discussed in detail.

STEP 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 Buffer (e.g., at Address Location 0x86 within the Framer IC). 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 that is to be transmitted. This octet will ultimately be used by the Remote Terminal Equipment in order to help it identify the type of LAPD message frame that it is receiving. Table 92 lists these octets and the corresponding LAPD Message types.

STEP 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 0x87 through 0x135 within the Framer IC).

STEP 3 - Specifying the Length of the LAPD Message

One of two different sizes of LAPD Messages can be transmitted, by writing the appropriate data to bit 1 within the Tx E3 LAPD Configuration Register. The bit-format of this register is presented below.

TRANSMIT E3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	X

The relationship between the contents of bit-fields 1 and the LAPD Message size is given in Table 93 .

TABLE 93: RELATIONSHIP BETWEEN TXLAPD MSG LENGTH AND THE LAPD MESSAGE SIZE

TXLAPD MESSAGE LENGTH	LAPD MESSAGE LENGTH
0	LAPD Message size is 76 bytes
1	LAPD Message size is 82 bytes

NOTE: *The Message Type selected must correspond with the contents of the first byte of the Information (Payload) portion, as presented in Table 92 .*

STEP 4 - Specifying which byte-field (within the E3 frame) that the LAPD Message frame octets are to be transported on.

The Transmit E3 Framer block allows the user to transport the LAPD Message frame octets via either the "NR" byte or 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.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			TxDL in NR	Not Used	TxAIS Enable	TxLOS Enable	TxMARx
RO	RO	RO	R/W	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

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. Additionally, the Transmit E3 Framer block will insert the contents of the TxNR Byte Register (Address = 0x37) 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 = 0x35) into the GC byte-field of each Outbound E3 frame.

STEP 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 HDLC Control block 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 3 (Auto Retransmit) within the Tx E3 LAPD Configuration Register (Address = 0x33), as depicted below.

TXE3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
RO	RO	RO	RO	R/W	RO	R/W	R/W
0	0	0	0	1	0	0	0

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.

STEP 6 - Enabling the LAPD Transmitter

Prior to the transmission of any data via the LAPD Transmitter, the LAPD Transmitter must be enabled by writing a "1" to bit 0 (TxLAPD Enable) of the Tx E3 LAPD Configuration Register, as depicted below.

TRANSMIT E3 LAPD CONFIGURATION REGISTER (ADDRESS = 0X33)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Not Used				Auto Retransmit	Not Used	TxLAPD Msg Length	TxLAPD Enable
R/O	R/O	R/O	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	X	X

If the user writes a “0” into this bit-field, then the LAPD Transmitter will be enabled, and the LAPD Transmitter will immediately begin to transmit a continuous stream of Flag Sequence octets (0x7E), via either the “GC” or the “NR” byte-field of each Outbound E3 frame (depending upon which byte has been selected to carry the PMDL channel).

Conversely, if the user writes a “1” into this bit-field, then the LAPD Transmitter will be disabled. The Transmit E3 Framer block will insert the contents of the Tx GC Byte Register into the “GC” byte-field for each Outbound E3 frame. Likewise, the Transmit E3 Framer block will also insert the contents of the Tx NR Byte Register into the NR” byte-field for each Outbound E3 frame. No transmission of PMDL data will occur.

STEP 7 - Initiate the Transmission

At this point, the user should have written the PMDL message into the on-chip Transmit LAPD Message buffer and should have specified the type of LAPD Message that is to be transmitted. The user should have also 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 LAPD Transmitter should have been enabled. Then initiate the transmission of this message by writing a “1” to Bit 3 (Tx DL Start) within the Tx E3 LAPD Status and Interrupt Register (Address = 0x34), as depicted below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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

A “0” to “1” transition in Bit 3 (TxDL Start) in this register, initiates the transmission of LAPD Message frames. At this point, the LAPD Transmitter will begin to search thorugh the PMDL message, which is residing within the Transmit LAPD Message buffer. If the LAPD Transmitter finds any string of five (5) consecutive “1’s” in the PMDL Message, then the LAPD Transmitter will insert a “0” immediately following these strings of consecutive “1’s”. 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 octet (e.g., six consecutive “1’s”) or the ABORT Sequence octet (e.g., seven consecutive “1’s”). 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-field (Bit 2) within the Tx E3 LAPD Status and Interrupt Register, will be set to “1”. This bit-field allows the user to poll the sta-

tus of the LAPD Transmitter. Once the LAPD Transmitter has completed the transmission of the LAPD Message, then this bit-field will toggle back to "0".

The user can configure the LAPD Transmitter to interrupt the local Microprocessor/Microcontroller upon completion of transmission of the LAPD Message frame, by setting bit-field 1 (TxLAPD Interrupt Enable) within the Tx E3 LAPD Status and Interrupt register (Address = 0x34). to "1" as depicted below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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

'The purpose of t his interrupt is to let the Microprocessor/Microcontroller know that the LAPD Transmitter is available and ready to transmit a LAPD Message frame (which contains a new PMDL Message) to the remote terminal equipment. Bit 0 (Tx LAPD Interrupt Status) within the Tx E3 LAPD Status and Interrupt Register will reflect the status for the Transmit LAPD 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., the Flag Sequence, SAPI, TEI, Control, etc.,) and insert them into the header byte 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 Sequence (FCS) value of the LAPD Message frame (e.g, of the LAPD Message header and Payload bytes) 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 octets).
- Fragment the resulting LAPD Message frame into octets and begin inserting these octets into either the GC or NR byte-fields within the Outbound E3 frames (depending upon the user's selection).
- Complete the transmission of the overhead bytes, information payload byte, FCS value, and the trailing Flag Sequence octets via the Transmit E3 Framer block.

Once the LAPD Transmitter has completed its transmission of the LAPD Message frame, the Framer will generate an Interrupt to the MIcroprocessor/Microcontroller (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 transmissions of the LAPD Message frames, the LAPD Transmitter will be sending a 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 Data Link) within the Tx E3 Configuration register.

NOTE: *In order to prevent the user's data (e.g., the PMDL Message within the LAPD Message frame) from mimicking the Flag Sequence byte or an ABORT Sequence, the LAPD Transmitter will parse through the PMDL Message data and insert a "0" into this data, immediately following the detection of five (5) consecutive "1's" (this stuffing occurs while the PMDL message data is being read in from the Transmit LAPD Message frame. The Remote LAPD Receive (See Section 5.3.5) will have the responsibility of checking the newly received PMDL messages for a string of five (5) consecutive "1's" and removing the subsequent "0" from the payload portion of the incoming LAPD Message.*

Figure 182 is a flow chart that 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 182. 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)

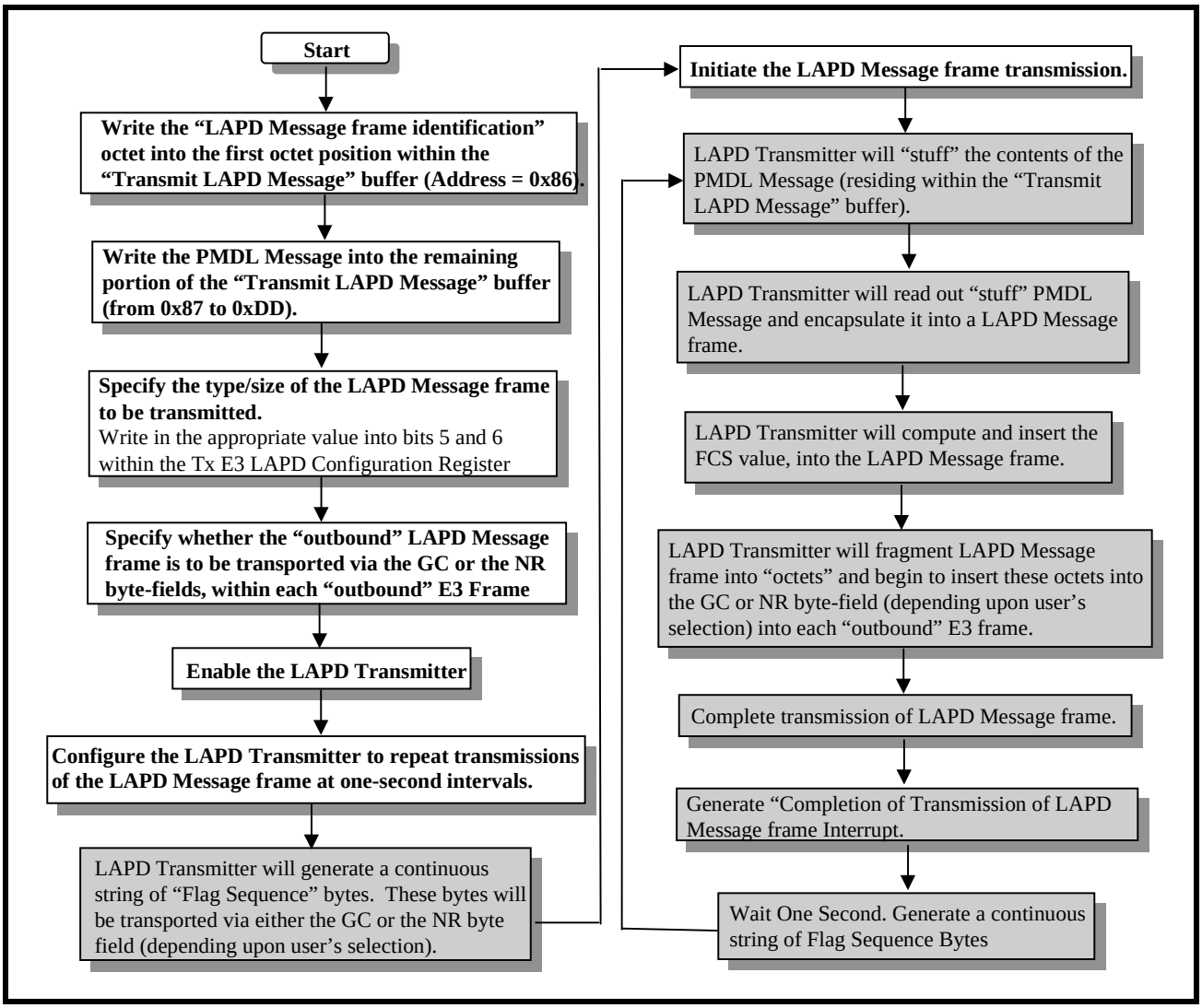
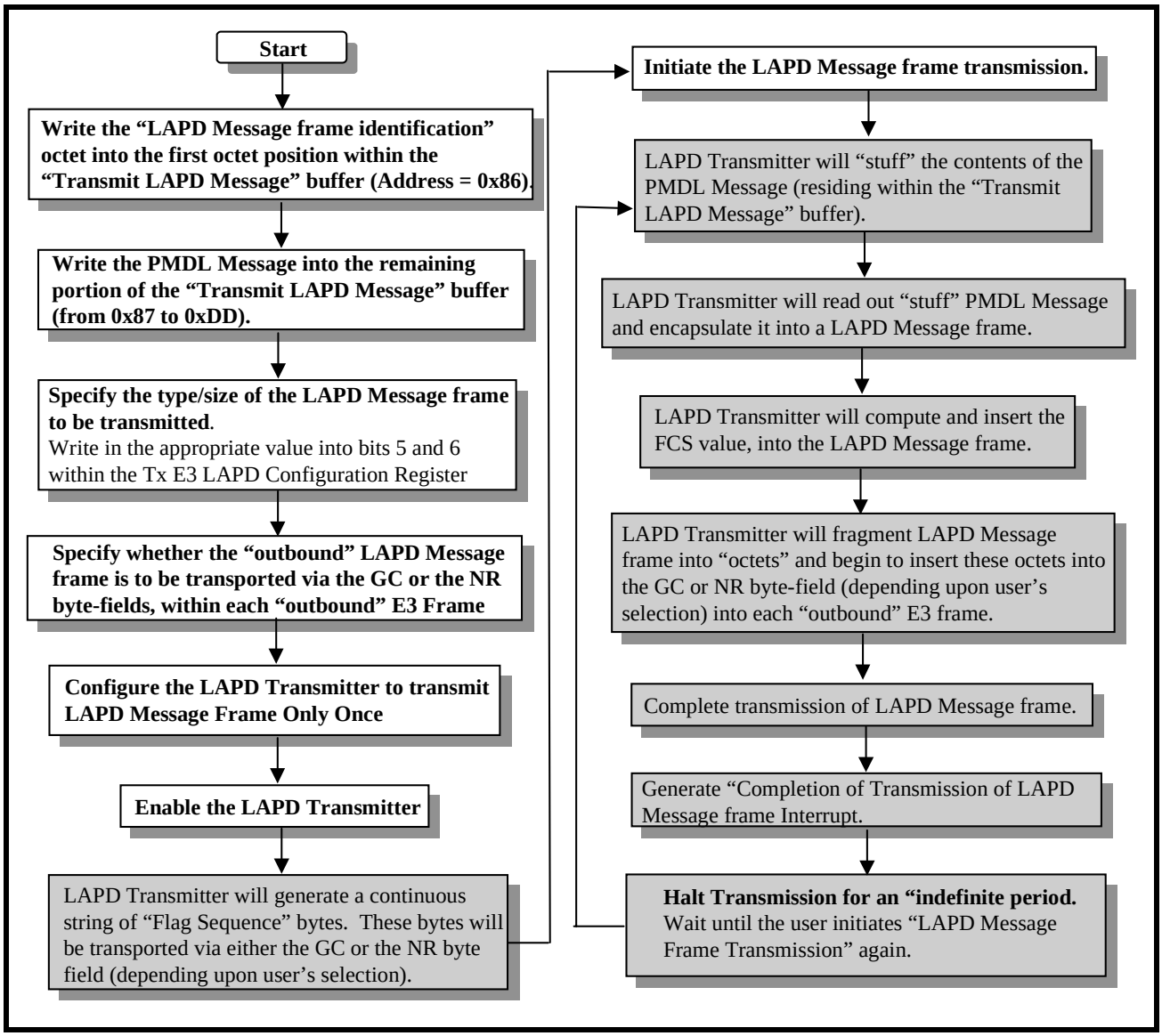


Figure 183 presents the procedure (in white boxes) which the user should use in order to transmit a PMDL Message via the LAPD Transmitter, when the LAPD Transmitter is configured to transmit a LAPD Message frame only once, and then halt transmission.

FIGURE 183. FLOW CHART DEPICTING HOW TO USE THE LAPD TRANSMITTER (LAPD TRANSMITTER IS CONFIGURED TO TRANSMIT A LAPD MESSAGE FRAME ONLY ONCE).



The Mechanics of Transmitting a New LAPD Message frame, if the LAPD Transmitter has been configured to re-transmit the LAPD Message frame, repeatedly, at One-Second intervals.

If the LAPD Transmitter has been configured to re-transmit the LAPD Message frame 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 Remote Terminal Equipment.

If another (e.g., a different) PMDL Message is to be transmitted to the Remote Terminal Equipment this new message will have to be written into the Transmit LAPD Message buffer, via the Microprocessor Interface block of the Framer IC. However, care must be taken when writing this new PMDL message. If this message is written 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 Remote Terminal Equip-

ment. 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 Framer to automatically reset activated interrupts.

The user can do this by writing a "1" into Bit 3 within the Framer Operating Mode register (Address = 0x00), as depicted below.

FRAMER OPERATING MODE REGISTER (ADDRESS = 0X00)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loopback	DS3/E3*	Internal LOS Enable	RESET	Interrupt Enable Reset	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	1

This action will prevent the LAPD Transmitter from generating its own One-Second interrupt (following each transmission of the LAPD Message frame).

2. Enable the One-Second Interrupt

This can be done by writing a "1" into Bit 0 (One-Second Interrupt Enable) within the Block Interrupt Enable Register, as depicted below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	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 synchronizing 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 Remote LAPD Receiver.

7.2.4 The Transmit E3 Framer Block

7.2.4.1 Brief Description of the Transmit E3 Framer

The Transmit E3 Framer block accepts data from any of the following three sources, and uses it to form the E3 data stream.

- The Transmit Payload Data Input block
- The Transmit Overhead Data Input block
- The Transmit HDLC Controller block
- The Internal Overhead Data Generator

The manner in how the Transmit E3 Framer block handles data from each of these sources is described below.

Handling of data from the Transmit Payload Data Input Interface

For E3 applications, all data that is input to the Transmit Payload Data Input Interface will be inserted into the payload bit positions within the Outbound E3 frames.

Handling of data from the Internal Overhead Bit Generator

By default, the Transmit E3 Framer block will internally generate the overhead bytes. However, if the Terminal Equipment inserts its own values for the overhead bits or bytes (via the Transmit Overhead Data Input Interface) or, if the user enables and employs the Transmit E3 HDLC Controller block, then these internally generated overhead bytes will be overwritten.

Handling of data from the Transmit Overhead Data Input Interface

For E3 applications, the Transmit E3 Framer block automatically generates and inserts the framing alignment bytes (e.g., the "FA1" and “FA2” framing alignment bytes) into the Outbound E3 frames. Further, the Transmit E3 Framer block will automatically compute and insert the EM byte into the Outbound E3 frames. Hence, the Transmit E3 Framer block will not accept data from the Transmit OH Data Input Interface block for the "FA1", “FA2” and “EM” bytes.

However, the Transmit E3 Framer block will accept (and insert) data from the Transmit Overhead Data Input Interface for the following byte-fields.

- MA byte
- TR byte
- NR byte
- GC byte

If the user’s local Data Link Equipment activates the Transmit Overhead Data Input Interface block and writes data into this interface for these bits or bytes, then the Transmit E3 Framer block will insert this data into the appropriate overhead bit/byte-fields, within the Outbound E3 frames.

7.2.4.2 Detailed Functional Description of the Transmit E3 Framer Block

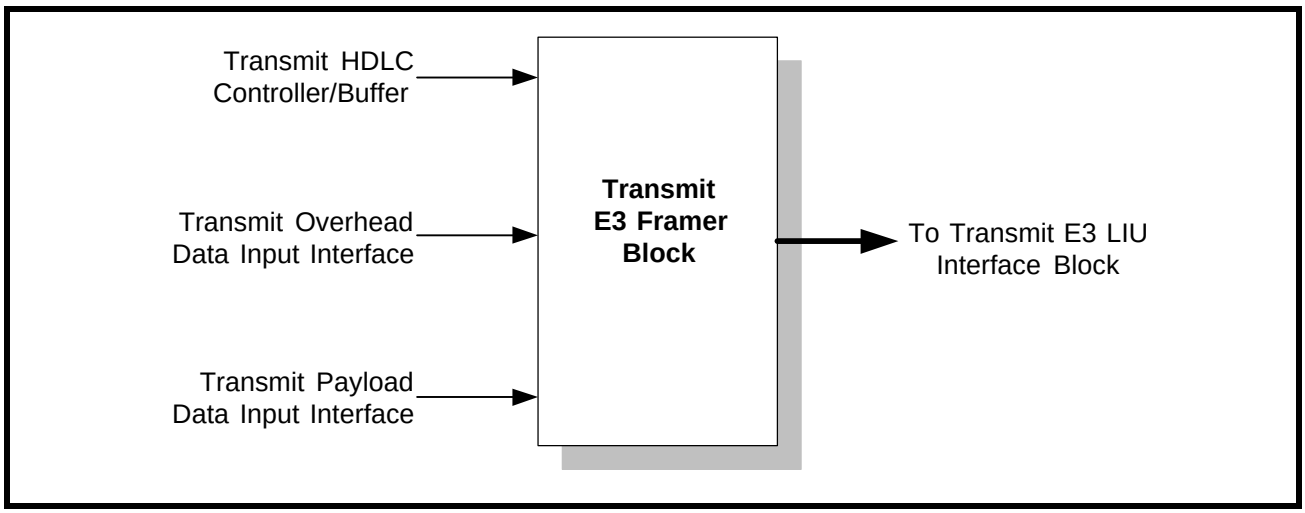
The Transmit E3 Framer receives data from the following three sources and combines them together to form a E3 data stream.

- The Transmit Payload Data Input Interface block.
- The Transmit Overhead Data Input Interface block
- The Transmit HDLC Controller block.

Afterwards, this E3 data stream will be routed to the Transmit E3 LIU Interface block, for further processing.

Figure 184 presents a simple illustration of the Transmit E3 Framer block, along with the associated paths to the other functional blocks within the chip.

FIGURE 184. THE TRANSMIT E3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO OTHER FUNCTIONAL BLOCKS



In addition to taking data from multiple sources and multiplexing them, in appropriate manner, to create the Outbound E3 frames, the Transmit E3 Framer block has the following roles.

- Generating Alarm Conditions
- Generating Errored Frames (for testing purposes)
- Routing Outbound E3 frames to the Transmit E3 LIU Interface block

Each of these additional roles are discussed below.

7.2.4.2.1 Generating Alarm Conditions

The Transmit E3 Framer block permits the user to, by writing the appropriate data into the on-chip registers, to override the data that is being written into the Transmit Payload Data and Overhead Data Input Interfaces and transmit the following alarm conditions.

- Generate the Yellow Alarms (or FERF indicators)
- Manipulate the FERF-bit, within the MA byte (set them to "0")
- Generate the AIS Pattern
- Generate the LOS pattern
- Generate FERF (Yellow) Alarms, in response to detection of a Red Alarm condition (via the Receive Section of the XRT74L74).
- Generate and transmit a desired value for the FEBE (Far-End-Block Error) bit, within the MA byte.

The procedure and results of generating any of these alarm conditions is presented below.

The user can exercise each of these options by writing the appropriate data to the Tx E3 Configuration Register (Address = 0x30). The bit format of this register is presented below.

TXE3 CONFIGURATION REGISTER (ADDRESS = 0X30)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			TxDL in NR	Not Used	TxAIS Enable	TxLOS Enable	TxMARx
RO	RO	RO	R/W	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Bit-field 2 through 0 permit the user to transmit various alarm conditions to the remote terminal equipment. The role/function of each of these three bit-fields within the register, are discussed below.

7.2.4.2.1.1 Tx AIS Enable - Bit 2

This read/write bit field permits the user to force the transmission of an AIS (Alarm Indication Signal) pattern to the remote terminal equipment via software control. If the user opts to transmit an AIS pattern, then the Transmit Section of the Framer IC will begin to transmit an unframed all ones pattern to the remote terminal equipment. Table 94 presents the relationship between the contents of this bit-field, and the resulting Framer action.

TABLE 94: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TX AIS ENABLE) WITHIN THE TX E3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT E3 FRAMER BLOCK'S ACTION

BIT 2	TRANSMIT E3 FRAMER'S ACTION
0	**Normal Operation:** The Transmit Section of the XRT74L74 Framer IC will transmit E3 traffic based upon data that it accepts via the Transmit Payload Data Input Interface block, the Transmit Overhead Data Input Interface block, the Transmit HDLC Controller block and internally generated overhead bytes.
1	**Transmit AIS Pattern:** The Transmit E3 Framer block will overwrite the E3 traffic, within an Unframed All Ones pattern.

NOTE: *This bit is ignored whenever the TxLOS bit-field is set.*

7.2.4.2.1.2 Transmit LOS Enable - Bit 1

This read/write bit field allows the user to transmit an LOS (Loss of Signal) pattern to the remote terminal, upon software control. Table 95 relates the contents of this bit field to the Transmit E3 Framer block's action.

TABLE 95: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (TX LOS) WITHIN THE TX E3 CONFIGURATION REGISTER, AND THE RESULTING TRANSMIT E3 FRAMER BLOCK'S ACTION

BIT 1	TRANSMIT E3 FRAMER'S ACTION
0	**Normal Operation:** The Overhead bits are either internally generated, or they are inserted via the Transmit Overhead Data Input Interface or the Transmit HDLC Controller blocks. The Payload bits are received from the Transmit Payload Data Input Interface.
1	**Transmit LOS Pattern:** When this command is invoked the Transmit E3 Framer will do the following. • Set all of the overhead bytes to "0" (including the FA1 and FA2 bytes) Overwrite the E3 payload bits with an "all zeros" pattern.

NOTE: *When this bit is set, it overrides all of the other bits in this register.*

7.2.4.2.1.3 TxMARx - Bit 0

This read/write bit-field permits the user to force the XRT74L74 Framer IC to transmit either a FERF (Far-End Receive Failure) or a FEBE (Far-End Block Error) indication to the remote terminal equipment.

7.2.4.2.2 Configuring the Transmit Trail Trace Buffer Message

The XRT74L74 Framer IC contains 16 bytes worth of Transmit Trail Trace Buffer registers and 16 bytes worth of Receive Trail Trace Buffer registers. The role of the Receive Trail Trace Buffer registers are described in Section 5.3.7.

The XRT74L74 Framer IC contains 16 Transmit Trail Trace Buffer registers (e.g., Tx TTB-0 through TxTTB-15). The purpose of these registers are to provide a 16-byte Trail Access Point Identifier to the Remote Terminal Equipment. The Remote Terminal Equipment will use this information in order 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 block will group 16 consecutive E3 frames, into a Trail Trace Buffer super-frame. When the Transmit E3 Framer block 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 = 0x38) 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 the Trail Trace Buffer super-frame), it will read in the contents of the Tx TTB-1 register (Address = 0x39) and insert this value into the TR byte-field of this Outbound E3 frame. As the Transmit E3 Framer block 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 block will then read in the contents of this particular Transmit Trail Trace Buffer register (Tx TTB-n) and insert this value into the TR byte-field of the very next Outbound E3 frame. After the Transmit E3 Framer block has created the 16th E3 frame, within a given Trail Trace Buffer super-frame (e.g., it has read in the contents of 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 (TxTTB-1 through TxTTB-15) will typically contain a "0" in their MSB 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 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.

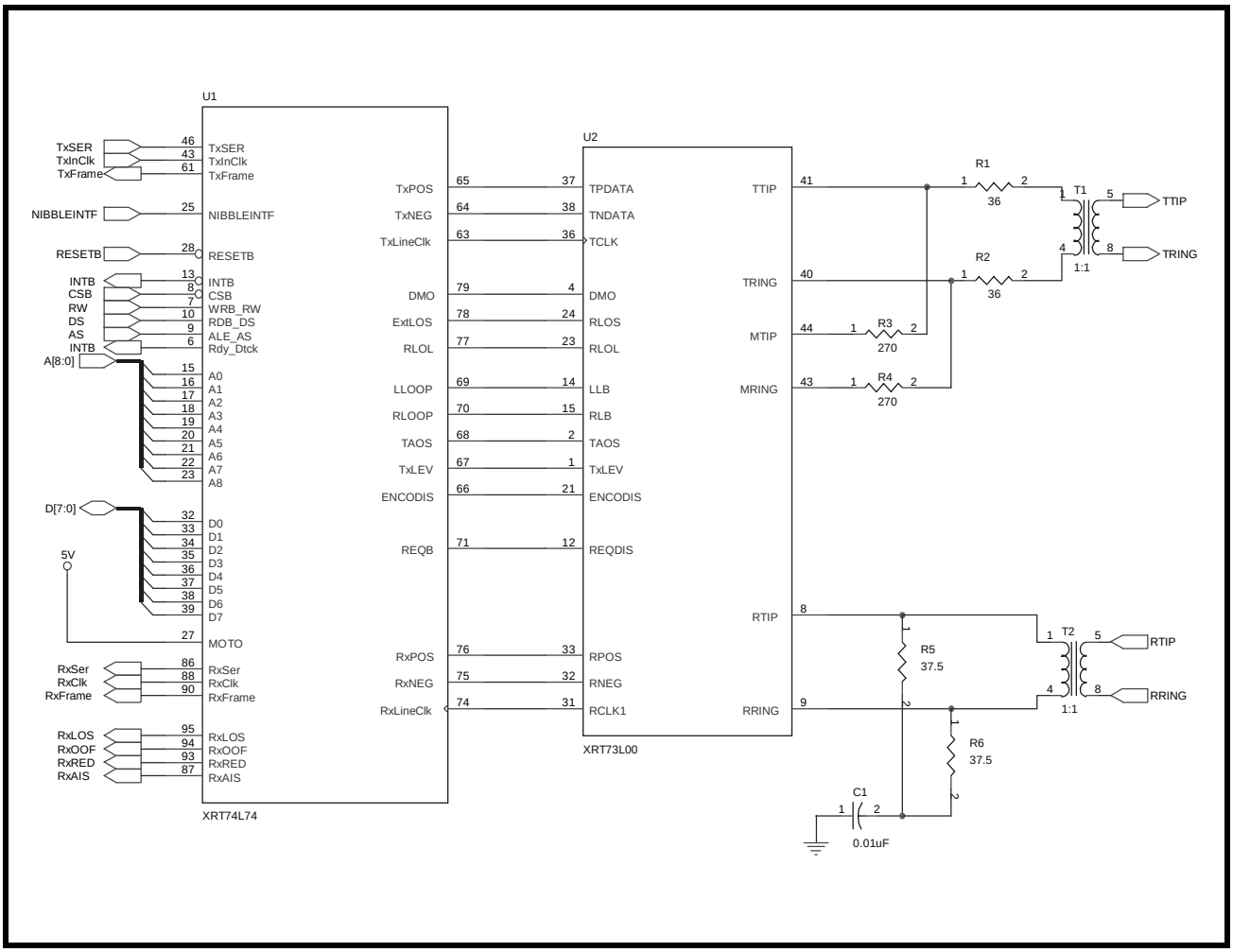
NOTES:

1. *The XRT74L74 Framer IC 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 5.3.2.9.*

7.2.5 The Transmit E3 Line Interface Block

The XRT74L74 Framer IC is a digital device that takes E3 payload and overhead bit information from some terminal equipment, processes this data and ultimately, multiplexes this information into a series of Outbound E3 frames. However, the XRT74L74 Framer IC lacks the current drive capability to be able to directly transmit this E3 data stream through some transformer-coupled coax cable with enough signal strength for it to be received by the remote receiver. Therefore, in order to get around this problem, the Framer IC requires the use of an LIU (Line Interface Unit) 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 185 presents a circuit drawing depicting the Framer IC interfacing to an LIU (XRT7300 DS3/E3/STS-1 Transmit LIU).

FIGURE 185. APPROACH TO INTERFACING THE XRT74L74 FRAMER IC TO THE XRT73L00 DS3/E3/STS-1 LIU

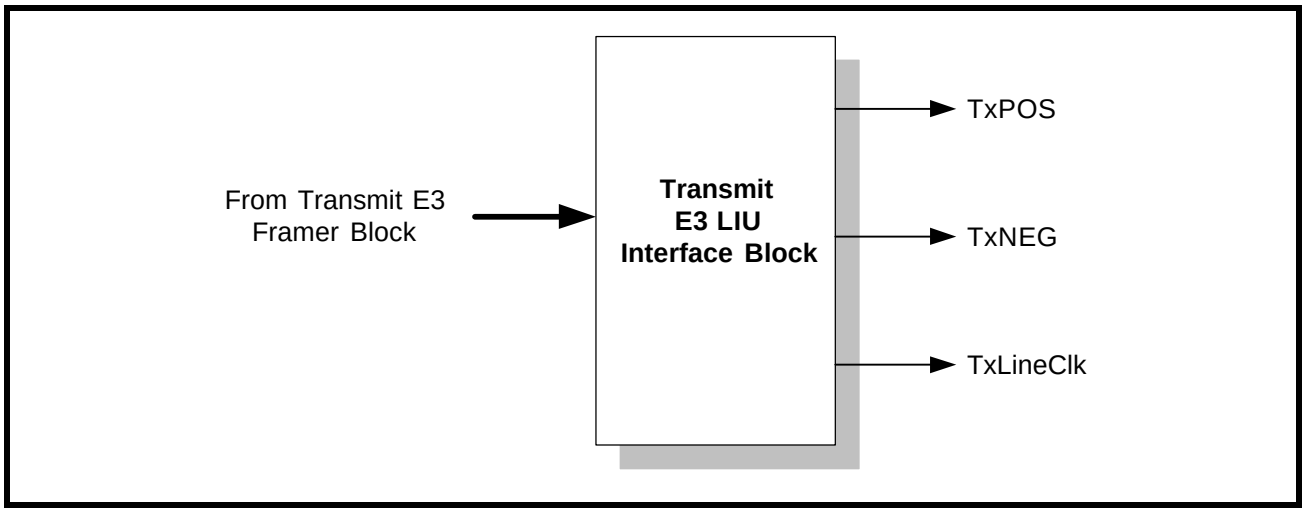


The Transmit Section of the XRT74L74 contains a block which is known as the Transmit E3 LIU Interface block. The purpose of the Transmit E3 LIU Interface block is to take the Outbound E3 data stream, from the Transmit E3 Framer block, and to do the following:

1. Encode this data into one of the following line codes
   - **a.** Unipolar (e.g., Single-Rail)
   - **b.** AMI (Alternate Mark Inversion)
   - **c.** HDB3 (High Density Bipolar - 3)
2. And to transmit this data to the LIU IC.

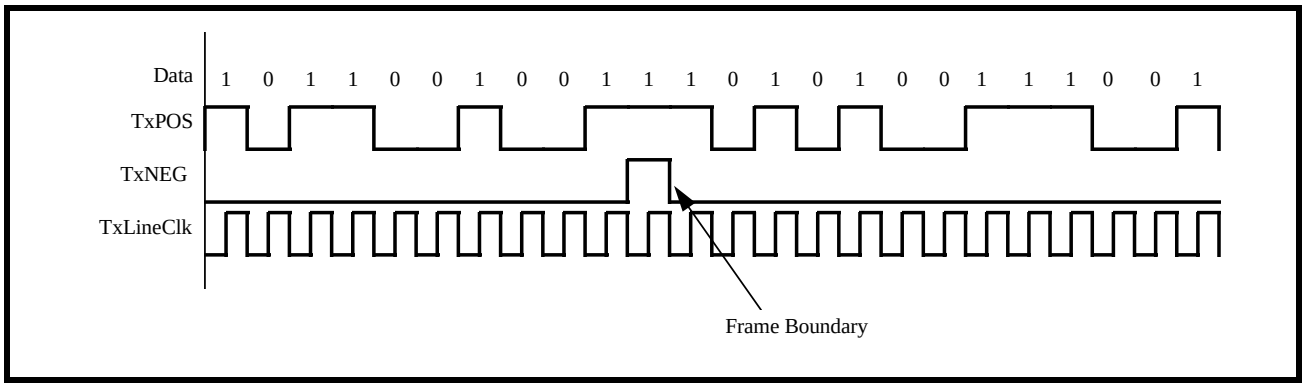
Figure 186 presents a simple illustration of the Transmit E3 LIU Interface block.

FIGURE 186. THE TRANSMIT E3 LIU INTERFACE BLOCK



The Transmit E3 LIU Interface block 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, in a binary (NRZ manner) data stream via the TxPOS pin to the LIU IC. 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. Figure 187 presents an illustration of the TxPOS and TxNEG signals during data transmission while the Transmit E3 LIU Interface block is operating in the Unipolar mode. This mode is sometimes referred to as Single Rail mode because the data pulses only exist in one polarity: positive.

FIGURE 187. THE BEHAVIOR OF TXPOS AND TXNEG SIGNALS DURING DATA TRANSMISSION WHILE THE TRANSMIT DS3 LIU INTERFACE IS OPERATING IN THE UNIPOLAR MODE



When the Transmit E3 LIU Interface block is operating in the Bipolar (or Dual Rail) mode, then the contents of the E3 Frame is output via both the TxPOS and TxNEG pins. If the Bipolar mode is chosen, then the E3 data can be transmitted to the LIU 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. Bipolar mode is sometimes referred to as Dual Rail because the data pulses occur in two polarities: positive and negative. The role of the TxPOS, TxNEG and TxLineClk output pins, for this mode are discussed below.

TxPOS - Transmit Positive Polarity Pulse: The Transmit E3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a positive polarity pulse to the remote terminal equipment.

TxNEG - Transmit Negative Polarity Pulse: The Transmit E3 LIU Interface block will assert this output to the LIU IC when it desires for the LIU to generate and transmit a negative polarity pulse to the remote terminal equipment.

TxLineClk - Transmit Line Clock: The LIU IC uses this signal from the Transmit E3 LIU Interface block 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 remote Receive E3 Framer.

7.2.5.1 Selecting the various Line Codes

The user can select either the Unipolar Mode or Bipolar Mode by writing the appropriate value to Bit 3 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 96 relates the value of this bit field to the Transmit E3 LIU Interface Output Mode.

TABLE 96: THE RELATIONSHIP BETWEEN THE CONTENT OF BIT 3 (UNIPOLAR/BIPOLAR*) WITHIN THE UNI I/O CONTROL REGISTER AND THE TRANSMIT E3 FRAMER LINE INTERFACE OUTPUT MODE

BIT 3	TRANSMIT E3 FRAMER LIU INTERFACE OUTPUT MODE
0	**Bipolar Mode:** AMI or HDB3 Line Codes are Transmitted and Received
1	**Unipolar (Single Rail) Mode** of transmission and reception of E3 data is selected.

NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the operation of the Receive E3 LIU Interface block*

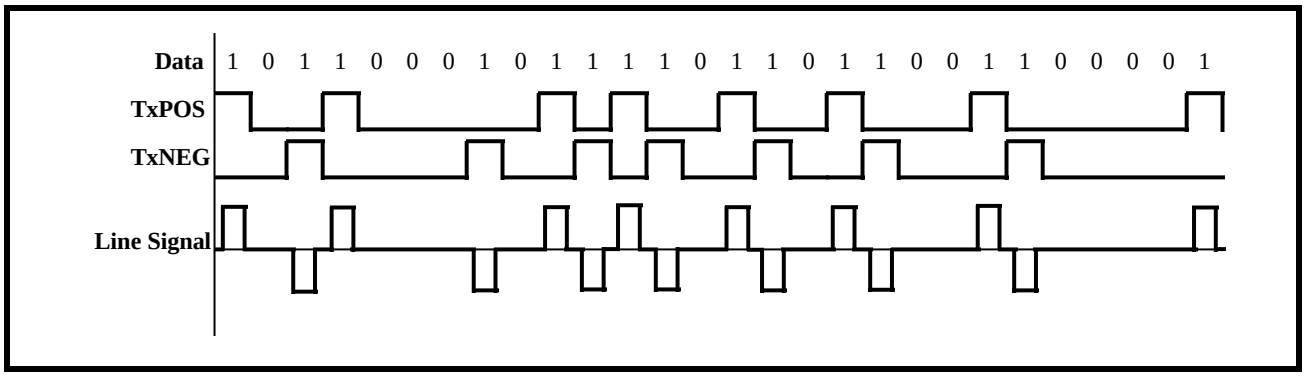
7.2.5.1.1 The Bipolar Mode Line Codes

If the Framer is selected to operate in the Bipolar Mode, then the DS3 data-stream can be transmitted via the AMI (Alternate Mark Inversion) or the HDB3 Line Codes. The definition of AMI and HDB3 line codes follow.

7.2.5.1.1.1 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, and -1. +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. Figure 188 presents an illustration of the AMI Line Code as would appear at the TxPOS and TxNEG pins of the Framer, as well as the output signal on the line.

FIGURE 188. AMI LINE CODE



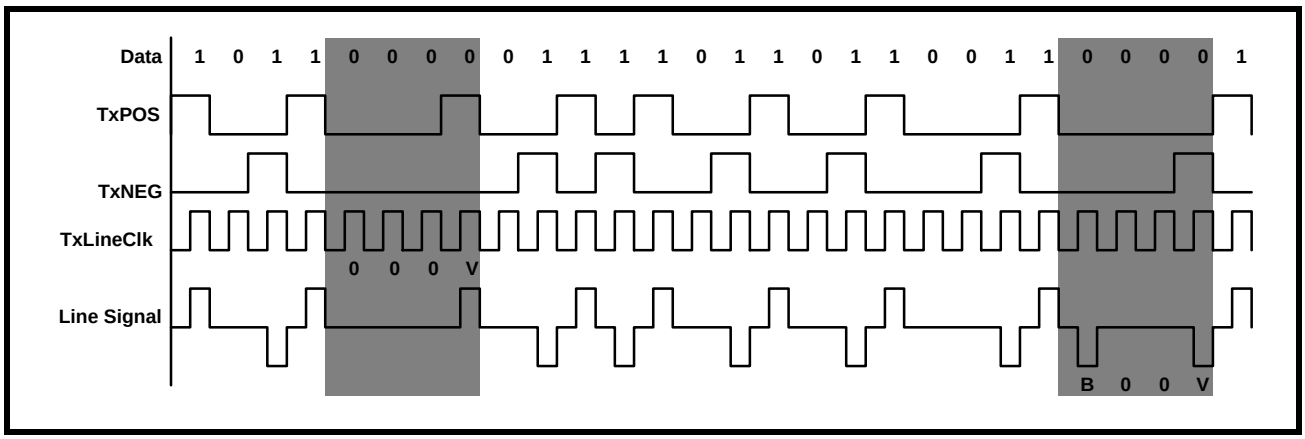
NOTE: *One of the main reasons that the AMI Line Code has been chosen for driving transformer-coupled media is that this line code introduces no dc component, thereby minimizing dc distortion in the line.*

7.2.5.1.1.2 The HDB3 Line Code

The Transmit E3 Framer and the associated LIU IC combine the data and timing information (originating from the TxLineClk signal) into the line signal that is transmitted to the remote receiver. The remote receiver has the task of recovering this data and timing information from the incoming E3 data stream. Many clock and data recovery schemes rely on the use of Phase Locked Loop technology. Phase-Locked-Loop (PLL) technology for clock recovery relies on transitions in the line signal, in order to maintain lock with the incoming E3 data stream. However, PLL-based clock recovery scheme, are vulnerable to the occurrence of a long stream of consecutive zeros (e.g., the absence of transitions). This scenario can cause the PLL to lose lock with the incoming E3 data, thereby causing the clock and data recovery process of the receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is HDB3 encoding. HDB3 (or High Density Bipolar - 3) is a form of AMI line coding that implements the following rule.

In general the HDB3 line code behaves just like AMI with the exception of the case when a long string of consecutive zeros occur on the line. 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. Figure 189 presents a timing diagram that illustrates examples of HDB3 encoding.

FIGURE 189. TWO EXAMPLES OF HDB3 ENCODING



The user chooses between AMI or HDB3 line coding by writing to bit 4 of the I/O Control Register (Address = 0x01), as shown below.

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 97 relates the content of this bit-field to the Bipolar Line Code which E3 Data will be transmitted and received at.

TABLE 97: THE RELATIONSHIP BETWEEN BIT 4 (AMI/ HDB3*) WITHIN THE I/O CONTROL REGISTER AND THE BIPOLAR LINE CODE THAT IS OUTPUT BY THE TRANSMIT E3 LIU INTERFACE BLOCK

BIT 4	BIPOLAR LINE CODE
0	HDB3
1	AMI

NOTES:

1. *This bit is ignored if the Unipolar mode is selected.*
2. *This selection also effects the operation of the Receive E3 LIU Interface block*

7.2.5.2 TxLineClk Clock Edge Selection

The Framer also allows the user to specify whether the E3 output data (via TxPOS and/or 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 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 98 relates the contents of this bit field to the clock edge of TxClk that E3 Data is output on the TxPOS and/or TxNEG output pins.

TABLE 98: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TXLINECLK INV) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG 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 190 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 191 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 190. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE RISING EDGE OF TXLINECLK

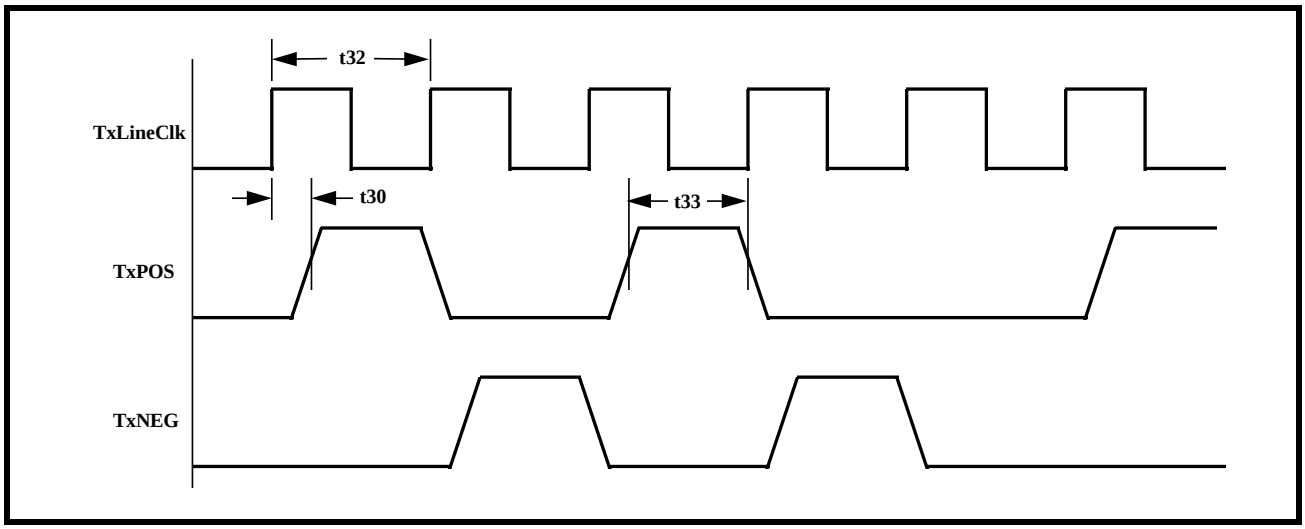
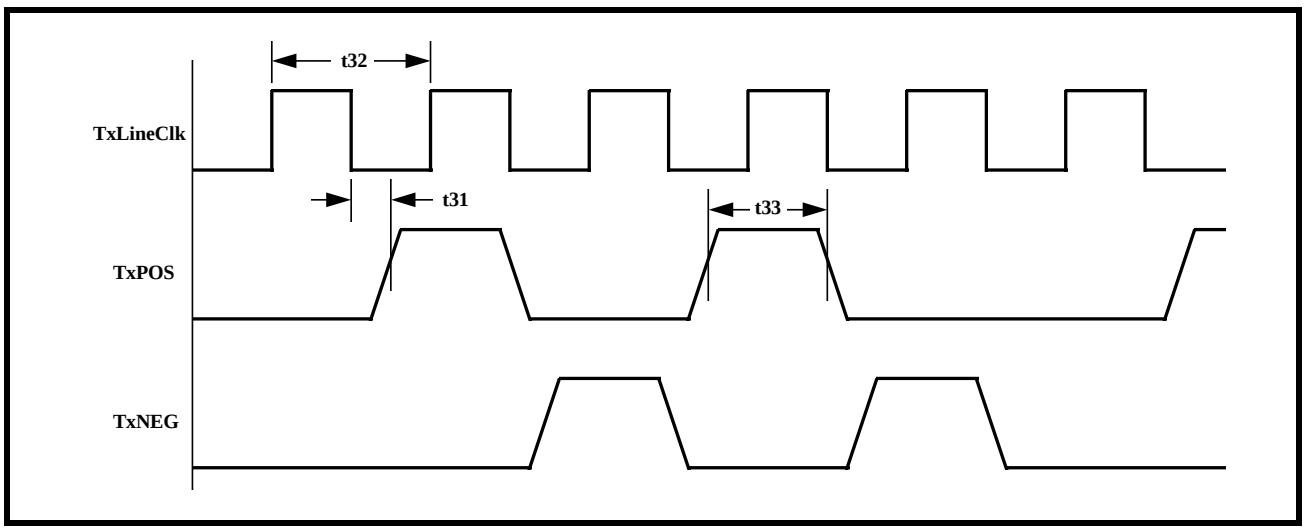


FIGURE 191. WAVEFORM/TIMING RELATIONSHIP BETWEEN TXLINECLK, TXPOS AND TXNEG - TXPOS AND TXNEG ARE CONFIGURED TO BE UPDATED ON THE FALLING EDGE OF TXLINECLK



7.2.6 Transmit Section Interrupt Processing

The Transmit Section of the XRT74L74 can generate an interrupt to the Microprocessor/Microcontroller for the following reasons.

- Completion of Transmission of LAPD Message

7.2.6.1 Enabling Transmit Section Interrupts

As mentioned in Section 36, the Interrupt Structure, within the XRT74L74 contains two hierarchical levels:

- Block Level
- Source Level

The Block Level

The Enable State of the Block Level for the Transmit Section Interrupts dictates whether or not interrupts (enabled) at the source level, are actually enabled.

The user can enable or disable these Transmit Section interrupts, at the Block Level by writing the appropriate data into Bit 1 (Tx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
0	0	0	0	0	0	0	0

Setting this bit-field to "1" enables the Transmit Section (at the Block Level) for Interrupt Generation. Conversely, setting this bit-field to "0" disables the Transmit Section for interrupt generation.

What does it mean for the Transmit Section Interrupts to be enabled or disabled at the Block Level?

If the Transmit Section is disabled (for interrupt generation) at the Block Level, then ALL Transmit Section interrupts are disabled, independent of the interrupt enable/disable state of the source level interrupts.

If the Transmit Section is enabled (for interrupt generation) at the block level, then a given interrupt will be enabled at the source level. Conversely, if the Transmit Section is enabled (for interrupt generation) at the Block level, then a given interrupt will still be disabled, if it is disabled at the source level.

As mentioned earlier, the Transmit Section of the XRT74L74 Framer IC contains the Completion of Transmission of LAPD Message Interrupt.

The Enabling/Disabling and Servicing of this interrupt is presented below.

7.2.6.1.1 The Completion of Transmission of the LAPD Message Interrupt

If the Transmit Section interrupts have been enabled at the Block level, then the user can enable or disable the Completion of Transmission of a LAPD Message Interrupt by writing the appropriate value into Bit 1 (TxLAPD Interrupt Enable) within the Tx E3 LAPD Status & Interrupt Register (Address = 0x34), as illustrated below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	X	0

Setting this bit-field to "1" enables the Completion of Transmission of a LAPD Message Interrupt. Conversely, setting this bit-field to "0" disables the Completion of Transmission of a LAPD Message interrupt.

7.2.6.1.2 Servicing the Completion of Transmission of a LAPD Message Interrupt

As mentioned previously, once the user commands the LAPD Transmitter to begin its transmission of a LAPD Message, it will do the following.

1. It will parse through the contents of the Transmit LAPD Message Buffer (located at address locations 0x86 through 0xDD) and search for a string of five (5) consecutive "1's". If the LAPD Transmitter finds a string of five consecutive "1's" (within the content of the LAPD Message Buffer, then it will insert a "0" immediately after this string.
2. It will compute the FCS (Frame Check Sequence) value and append this value to the back-end of the user-message.
3. It will read out of the content of the user (zero-stuffed) message and will encapsulate this data into a LAPD Message frame.
4. Finally, it will begin transmitting the contents of this LAPD Message frame via the "N" bits, within each Outbound E3 frame.
5. Once the LAPD Transmitter has completed its transmission of this LAPD Message frame (to the Remote Terminal Equipment), the XRT74L74 Framer IC will generate the Completion of Trans-

mission of a LAPD Message Interrupt to the Microcontroller/Microprocessor. Once the XRT74L74 Framer IC generates this interrupt, it will do the following.

- Assert the Interrupt Output pin (INT) by toggling it "Low".
- Set Bit 0 (TxLAPD Interrupt Status) within the TxE3 LAPD Status and Interrupt Register, to "1" as illustrated below.

TXE3 LAPD STATUS AND INTERRUPT REGISTER (ADDRESS = 0X34)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				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	1

The purpose of this interrupt is to alert the Microcontroller/MIcroprocessor that the LAPD Transmitter has completed its transmission of a given LAPD (or PMDL) Message, and is now ready to transmit the next PMDL Message, to the Remote Terminal Equipment.

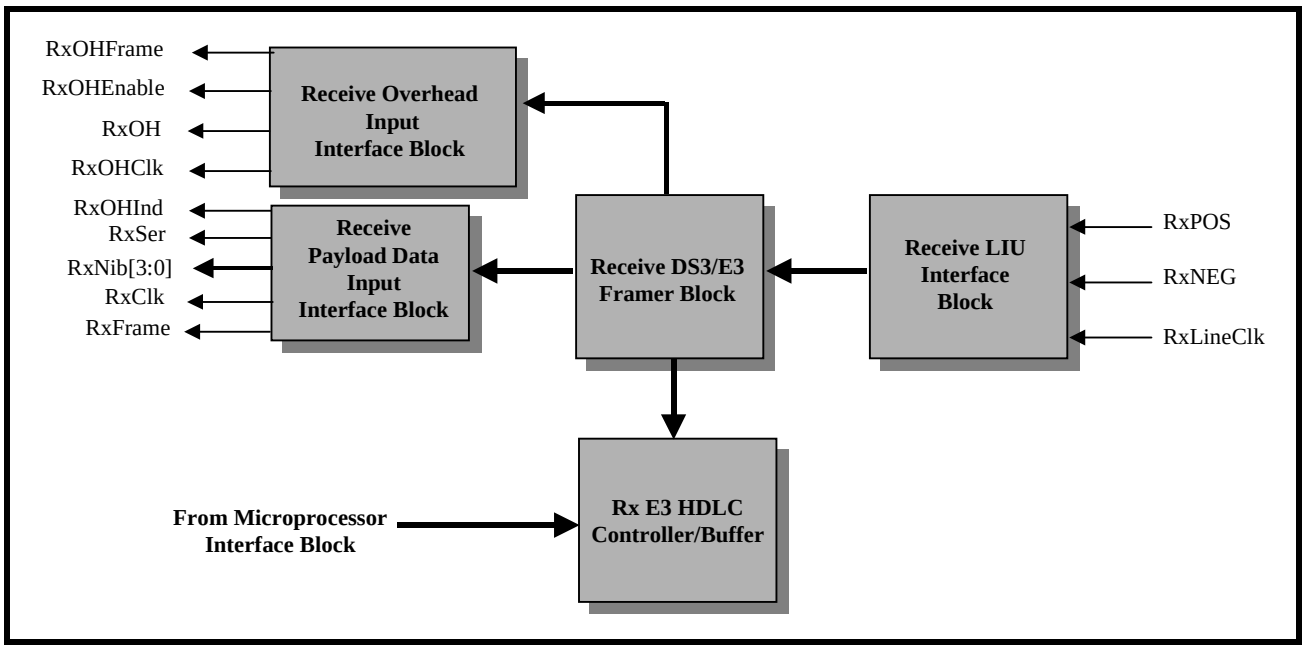
7.3 THE RECEIVE SECTION OF THE XRT74L74 (E3 MODE OPERATION)

When the XRT74L74 has been configured to operate in the E3 Mode, the Receive Section of the XRT74L74 consists of the following functional blocks.

- Receive LIU Interface block
- Receive HDLC Controller block
- Receive E3 Framer block
- Receive Overhead Data Output Interface block
- Receive Payload Data Output Interface block

Figure 192 presents a simple illustration of the Receive Section of the XRT74L74 Framer IC.

FIGURE 192. THE RECEIVE SECTION OF THE XRT74L74 WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE E3 MODE



Each of these functional blocks will be discussed in detail in this document.

7.3.1 The Receive E3 LIU Interface Block

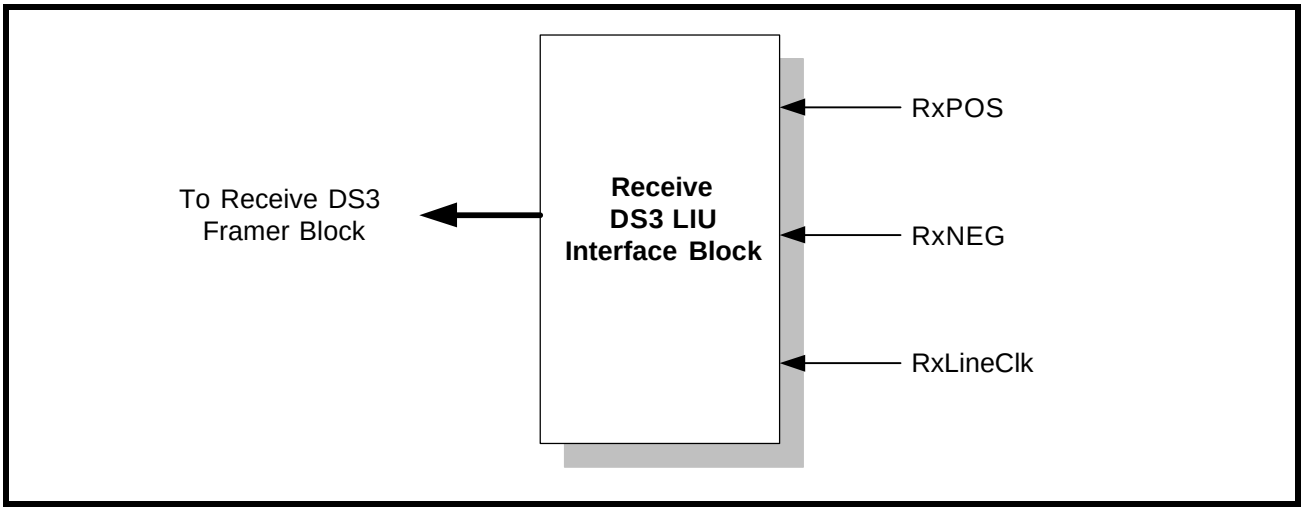
The purpose of the Receive E3 LIU Interface block is two-fold:

1. To receive encoded digital data from the E3 LIU IC.

2. To decode this data, convert it into a binary data stream and to route this data to the Receive E3 Framer block.

Figure 193 presents a simple illustration of the Receive E3 LIU Interface block.

FIGURE 193. THE RECEIVE E3 LIU INTERFACE BLOCK



The Receive Section of the XRT74L74 will via the Receive E3 LIU Interface Block receive 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 data information will be received via the RxPOS and RxNEG input pins. The Receive E3 LIU Interface block is capable of receiving E3 data pulses in unipolar or bipolar format. If the Receive E3 framer is operating in the bipolar format, 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.

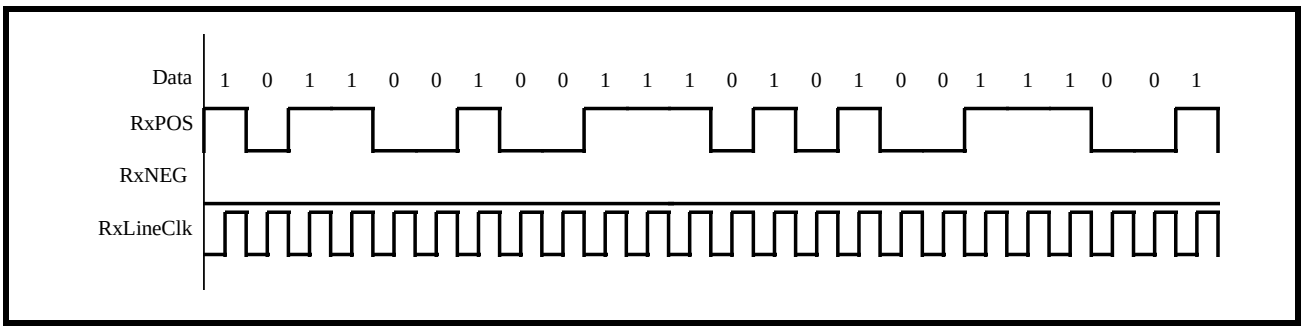
7.3.1.1 Unipolar Decoding

If the Receive E3 LIU Interface block is operating in the Unipolar (single-rail) mode, then it will receive the Single Rail NRZ DS3 data pulses via the RxPOS input pin. The Receive E3 LIU Interface block will also receive its timing signal via the RxLineClk signal.

NOTE: *The RxLineClk signal will function as the timing source for the entire Receive Section of the XRT74L74.*

No data pulses will be applied to the RxNEG input pin. The Receive E3 LIU Interface block receives a logic "1" when a logic "1" level signal is present at the RxPOS 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 194 presents an illustration of the behavior of the RxPOS, RxNEG and RxLineClk input pins when the Receive E3 LIU Interface block is operating in the Unipolar mode.

FIGURE 194. BEHAVIOR OF THE RXPOS, RXNEG AND RXLINECLK SIGNALS DURING DATA RECEPTION OF UNIPOLAR DATA



The user can configure the Receive E3 LIU Interface block to operate in either the Unipolar or the Bipolar Mode by writing the appropriate data to the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 99 relates the value of this bit-field to the Receive E3 LIU Interface Input Mode.

TABLE 99: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (TXLINECLK INV) WITHIN THE I/O CONTROL REGISTER AND THE TXLINECLK CLOCK EDGE THAT TXPOS AND TXNEG ARE UPDATED ON

BIT 3	RECEIVE E3 LIU INTERFACE INPUT MODE
0	.**Bipolar Mode (Dual Rail):** AMI or HDB3 Line Codes are Transmitted and Received.
1	**Unipolar Mode (Single Rail) Mode** of transmission and reception of E3 data is selected.

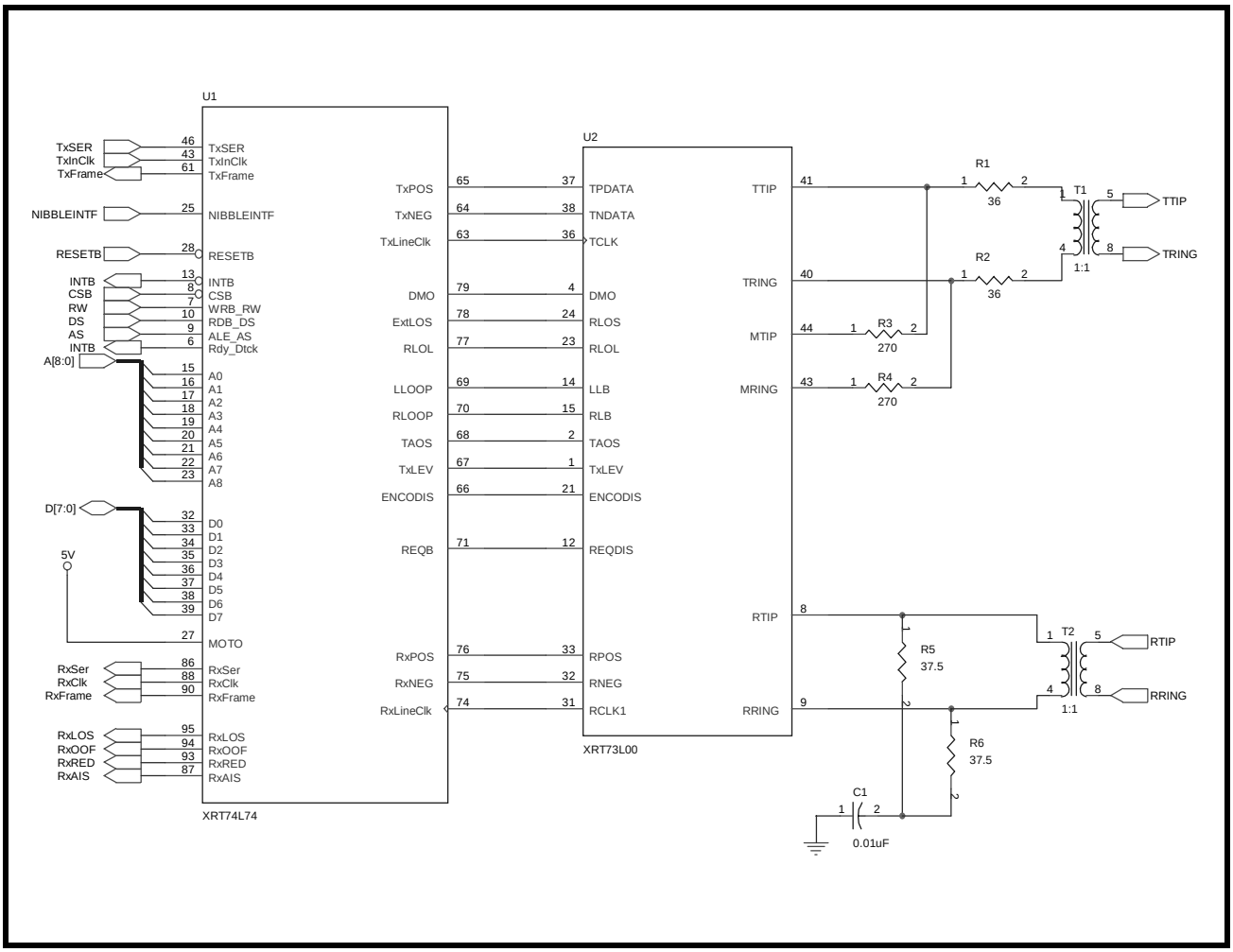
NOTES:

1. *The default condition is the Bipolar Mode.*
2. *This selection also effects the Transmit E3 Framer Line Interface Output Mode*

7.3.1.2 Bipolar Decoding

If the Receive E3 LIU Interface block is operating in the Bipolar Mode, then it will receive the E3 data pulses via both the RxPOS, RxNEG, and the RxLineClk input pins. Figure 195 presents a circuit diagram illustrating how the Receive E3 LIU Interface block interfaces to the Line Interface Unit while the Framer is operating in Bipolar mode. The Receive E3 LIU Interface block can be configured to decode either the AMI or HDB3 line codes.

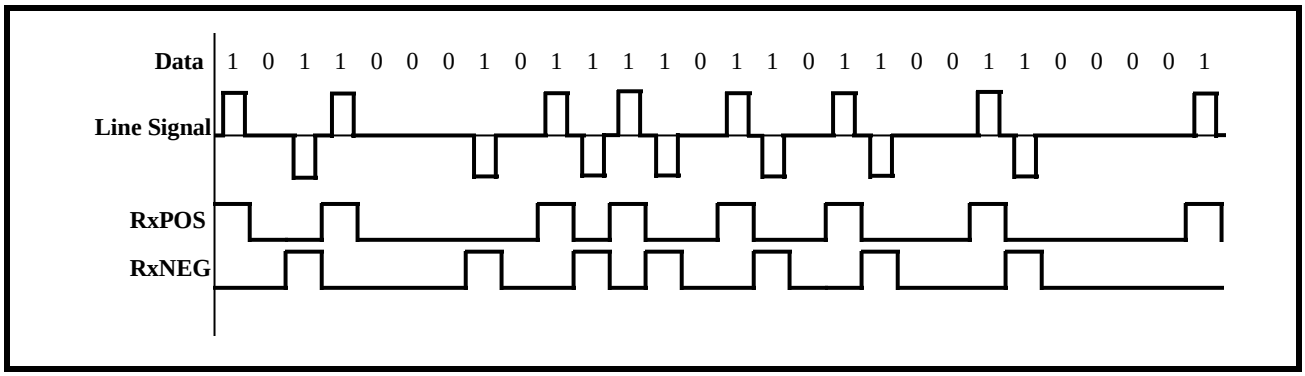
FIGURE 195. ILLUSTRATION ON HOW THE XRT74L74 RECEIVE E3 FRAMER IS INTERFACED TO THE XRT73L00 LINE INTERFACE UNIT WHILE OPERATING IN THE BIPOLAR MODE (ONE CHANNEL SHOWN)



7.3.1.2.1 AMI Decoding

AMI or Alternate Mark Inversion, means that consecutive "one's" pulses (or marks) 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 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 exist between these two pulses. Figure 196 presents an illustration of the AMI Line Code as would appear at the RxPOS and RxNEG pins of the Framer, as well as the output signal on the line.

FIGURE 196. AMI LINE CODE



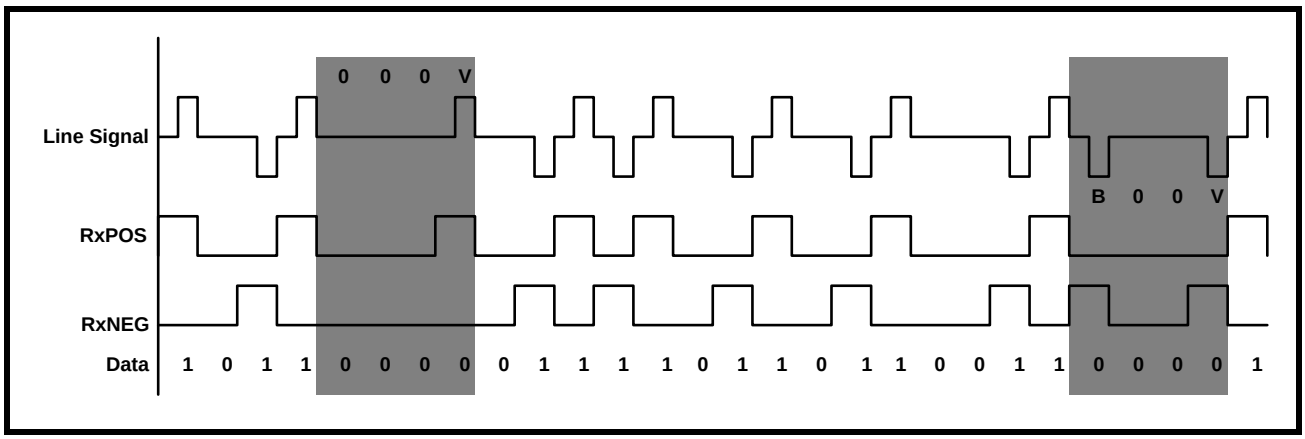
NOTE: *One of the reasons that the AMI Line Code has been chosen for driving copper medium, isolated via transformers, is that this line code has no dc component, thereby eliminating dc distortion in the line.*

7.3.1.2.2 HDB3 Decoding

The Transmit E3 LIU Interface block and the associated LIU embed and combine the data and clocking information into the line signal that is transmitted to the remote terminal equipment. The remote terminal equipment has the task of recovering this data and timing information from the incoming E3 data stream. Most clock and data recovery schemes rely on the use of Phase-Locked-Loop technology. One of the problems of using Phase-Locked-Loop (PLL) technology for clock recovery is that it relies on transitions in the line signal, in order to maintain lock with the incoming E3 data-stream. Therefore, these clock recovery scheme, 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 receiver to fail. Therefore, some approach is needed to insure that such a long string of consecutive zeros can never happen. One such technique is HDB3 (or High Density Bipolar -3) 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. Any 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 LIU Interface block, when operating with the HDB3 Line Code is responsible for decoding the HD-encoded data back into a unipolar (binary-format). For instance, if the Receive E3 LIU Interface block detects a "000V" or a "B00V" pattern in the incoming pattern, the Receive E3 LIU Interface block will replace it with four (4) consecutive zeros. Figure 197 presents a timing diagram that illustrates examples of HDB3 decoding.

FIGURE 197. TWO EXAMPLES OF HDB3 DECODING



7.3.1.2.3 Line Code Violations

The Receive E3 LIU Interface block will also check the incoming E3 data stream for line code violations. For example, when the Receive E3 LIU Interface block detects a valid bipolar violation (e.g., in HDB3 line code), it will substitute four zeros into the binary data stream. However, if the bipolar violation is invalid, then an LCV (Line Code Violation) is flagged and the PMON LCV Event Count Register (Address = 0x50 and 0x51) will also be incremented. Additionally, the LCV-One-Second Accumulation Registers (Address = 0x6E and 0x6F) will be incremented. For example: If the incoming E3 data is HDB3 encoded, the Receive E3 LIU Interface block will also increment the LCV One-Second Accumulation Register if three (or more) consecutive zeros are received.

7.3.1.2.4 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 LIU Interface block via the RxLineClk signal. The Framer IC allows the user to specify which edge (e.g, rising or falling) of the RxLineClk signal will sample and latch the signal at the RxPOS and RxNEG input signals into the Framer IC. The user can make this selection by writing the appropriate data to bit 1 of the I/O Control Register, as depicted below.

II/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ZeroSup*	Unipolar/ Bipolar*	TxLine CLK Invert	RxLine CLK Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	0

Table 100 depicts the relationship between the value of this bit-field to the sampling clock edge of RxLineClk.

TABLE 100: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (RXLINECLK INV) OF THE I/O CONTROL REGISTER, AND THE SAMPLING EDGE OF THE RXLINECLK SIGNAL

RXCLKINV (BIT 1)	RESULT
0	**.Rising Edge:** RxPOS and RxNEG are sampled at the rising edge of RxLineClk. See Figure 198 for timing relationship between RxLineClk, RxPOS, and RxNEG.
1	**Falling Edge:** RxPOS and RxNEG are sampled at the falling edge of RxLineClk. See Figure 199 for timing relationship between RxLineClk, RxPOS, and RxNEG.

Figure 198 and Figure 199 present the Waveform and Timing Relationships between RxLineClk, RxPOS and RxNEG for each of these configurations.

FIGURE 198. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE RISING EDGE OF RXLINECLK

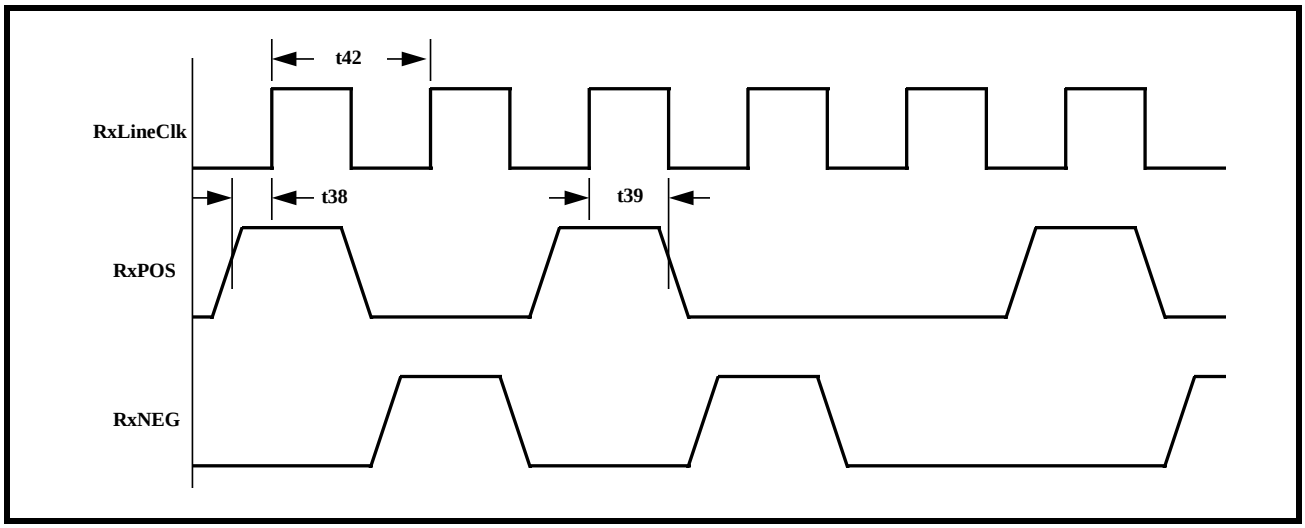
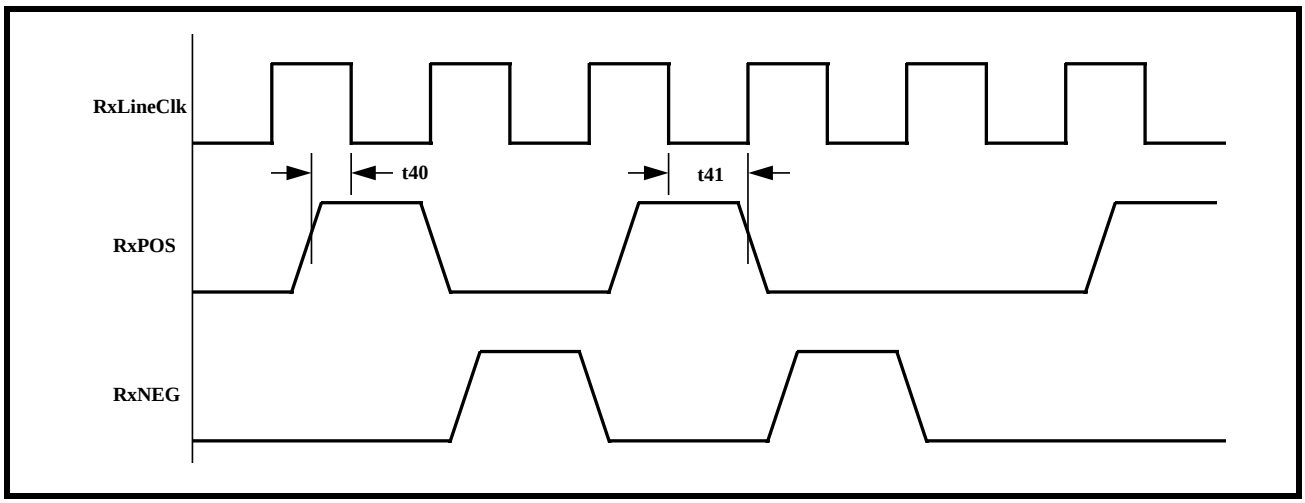


FIGURE 199. WAVEFORM/TIMING RELATIONSHIP BETWEEN RXLINECLK, RXPOS AND RXNEG - WHEN RXPOS AND RXNEG ARE TO BE SAMPLED ON THE FALLING EDGE OF RXLINECLK



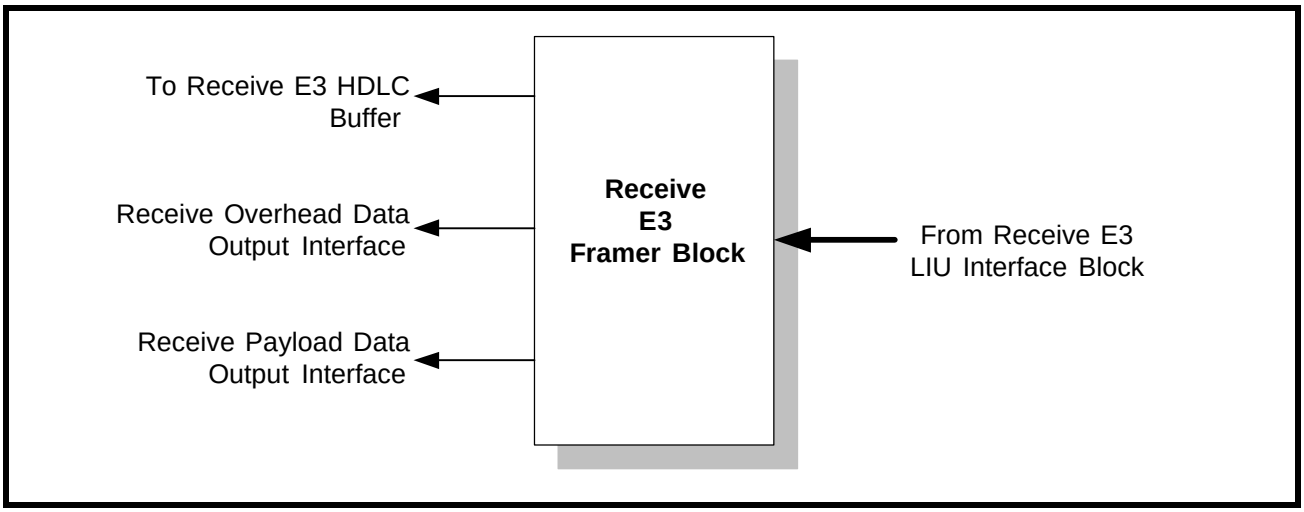
7.3.2 The Receive E3 Framer Block

The Receive E3 Framer block accepts decoded E3 data from the Receive E3 LIU Interface block, and routes data to the following destinations.

- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block.
- The Receive E3 HDLC Controller Block

Figure 200 presents a simple illustration of the Receive E3 Framer block, along with the associated paths to the other functional blocks within the Framer chip.

FIGURE 200. THE RECEIVE E3 FRAMER BLOCK AND THE ASSOCIATED PATHS TO THE OTHER FUNCTIONAL BLOCKS



Once the HDB3 (or AMI) encoded data has been decoded into a binary data-stream, the Receive E3 Framer block will use portions of this data-stream in order to synchronize itself to the remote terminal equipment. At any given time, the Receive E3 Framer block will be operating in one of two modes.

- **The Frame Acquisition Mode:** In this mode, the Receive E3 Framer block is trying to acquire synchronization with the incoming E3 frame, or
- **The Frame Maintenance Mode:** In this mode, the Receive E3 Framer block is trying to maintain frame synchronization with the incoming E3 Frames.

Figure 201 presents a State Machine diagram that depicts the Receive E3 Framer block's E3/ITU-T G.832 Frame Acquisition/Maintenance Algorithm.

7.3.2.1 The Framing Acquisition Mode

The Receive E3 Framer block 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 201 .

- 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 block 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 "0xF6", and FA2 is assigned the value of "0x28". Figure 202 , which presents an illustration of the E3, ITU-T G.832 Framing 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.

FIGURE 201. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER E3 FRAME ACQUISITION/MAINTENANCE ALGORITHM

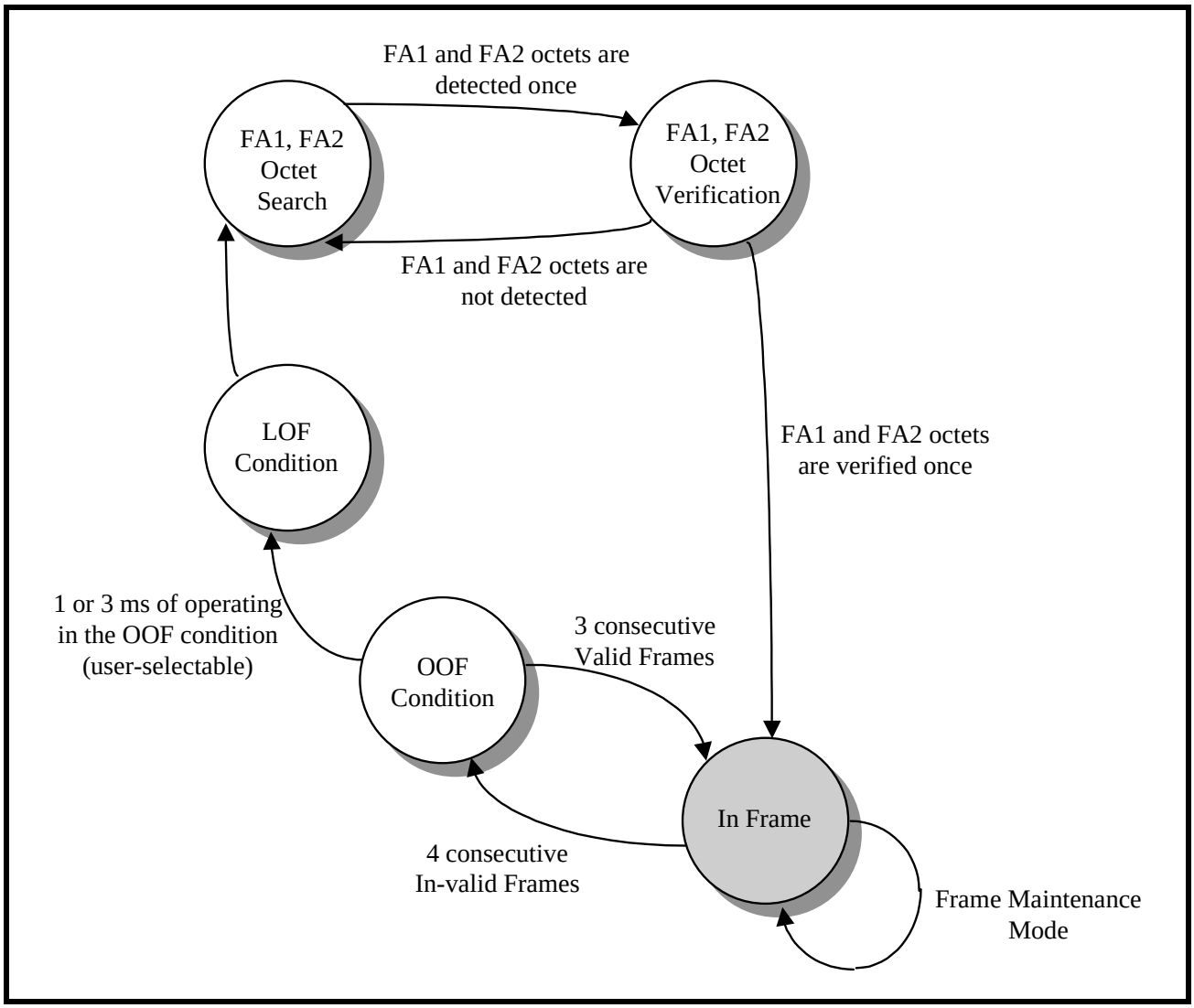
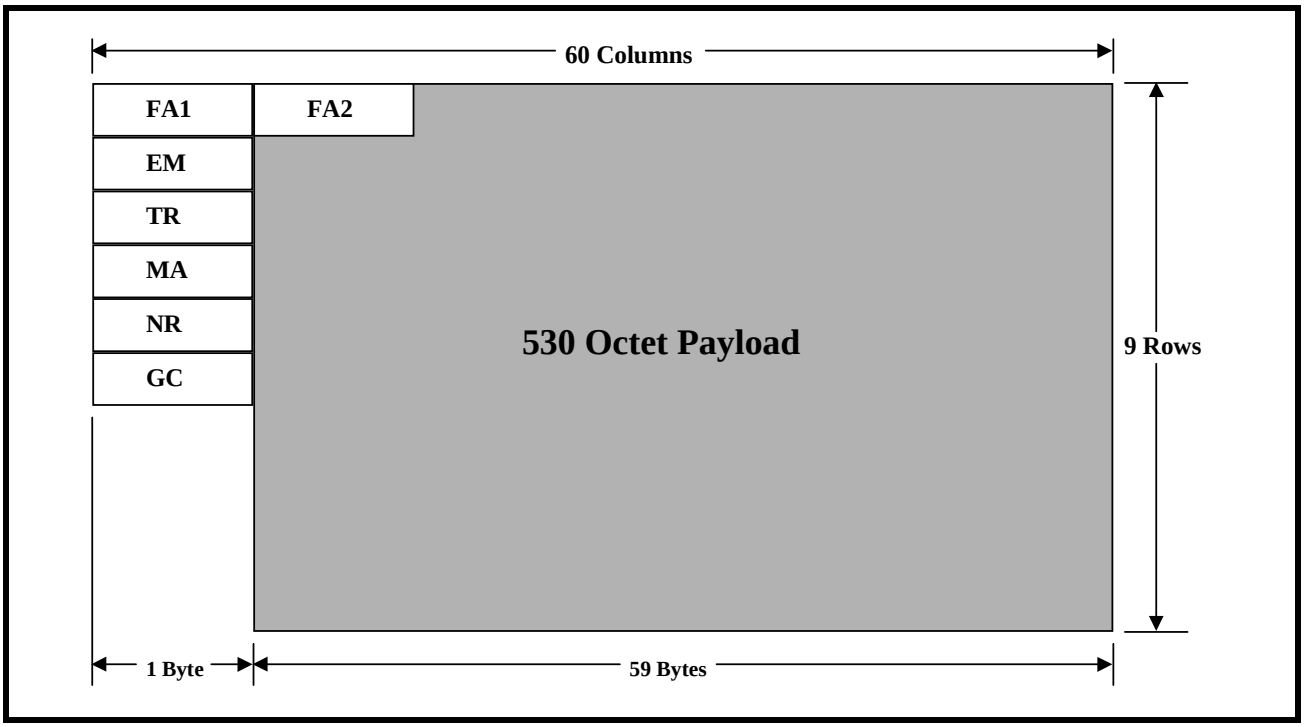


FIGURE 202. THE E3, ITU-T G.832 FRAMING FORMAT



When the Receive E3 Framer block detects the "FA1" octet, and determines that this octet is immediately followed by the "FA2" octet, then it will transition to the FA1, FA2 Octet Verification state, per Figure 202 .

The FA1, FA2 Octet Verification State

Once the Receive E3 Framer block has detected an "0xF628" pattern (e.g., the concatenation of the FA1 and 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 block enters this state, it will then quit performing its bit-by-bit search for the Frame Alignment bytes. Instead, the Receive E3 Framer block 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 block 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 block 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 block 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 block, while operating in the In-Frame state, can be found in Section 5.3.2.2 (The Framing Maintenance Mode).

OOF (Out of Frame) Condition State

If the Receive E3 Framer while operating in the In-Frame state detects four (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 block's operation, while in the OOF condition state is a unique mix of Framing Maintenance and Framing Acquisition operation. The Receive E3 Framer block will exhibit some Framing Acquisition characteristics by attempting to locate (once again) the Frame Alignment octets. However, the Receive E3 Framer block will also exhibit some Frame Maintenance behavior by still using the most recent frame synchronization for its overhead byte and payload byte processing.

The Receive E3 Framer block will inform the Microprocessor/Microcontroller 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.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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

The Receive E3 Framer block 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 block is capable of finding the Framing 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 block will then inform the Microprocessor/Microcontroller of its transition back into the In-Frame state by generating the Change in OOF Condition Interrupt.

However, if the Receive E3 Framer block 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 (Loss of Frame) Condition state.

The user can select this user-selectable number of E3 frame periods that the Receive E3 Framer block will remain in the OOF Condition state by writing the appropriate value into Bit 7 (RxLOF Algo) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	1	1	1	1	1

Writing a “0” into this bit-field causes the Receive E3 Framer block 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 block 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 block enters the LOF Condition state, then the following things will happen.

- The Receive E3 Framer block will discard the most recent frame synchronization and
- The Receive E3 Framer block will make an unconditional transition to the FA1, FA2 Octet Search state.
- The Receive E3 Framer block will notify the Microprocessor/Microcontroller 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.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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

Finally, the Receive E3 Framer block will also inform the external circuitry of this transition to the LOF Condition state by toggling the RxLOF output pin “High”.

7.3.2.2 The Framing Maintenance Mode

Once the Receive E3 Framer block enters the In-Frame state, then it will notify the Microprocessor/Microcontroller of this fact by generating both the Change in OOF Condition and Change in LOF Condition Interrupts. When this happens, bits 2 and 3 (LOF Interrupt Status and OOF Interrupt Status) will be set to “1”, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	1	0	0

Additionally, the Receive E3 Framer block will inform the external circuitry of its transition to the In-Frame state by toggling both the RxOOF and RxLOF output pins "Low”.

Finally, the Receive E3 Framer block will negate both the RxOOF and the RxLOF bit-fields within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	0	0	0	0	0

When the Receive E3 Framer block 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 block 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 block 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 Framing Alignment bytes, and still remain in the In-Frame state. However, each time the Receive E3 Framer block detects an error in the Frame Alignment bytes, it will increment the PMON Framing Error Event Count Registers (Address = 0x52 and 0x53). The bit-format for these two registers are depicted below.

PMON FRAMING BIT/BYTE ERROR COUNT REGISTER - MSB (ADDRESS = 0X52)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framing Bit/Byte Error Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FRAMING BIT/BYTE ERROR COUNT REGISTER - LSB (ADDRESS = 0X53)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framing Bit/Byte Error Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

7.3.2.3 Forcing a Reframe via Software Command

The XRT74L74 Framer IC permits the user to command a reframe procedure with the Receive E3 Framer block via software command. If the user writes a "1" into Bit 0 (Reframe) within the I/O Control Register (Address = 0x01), as depicted below, then the Receive E3 Framer block will be forced into the FA1, FA2 Octet Search state, per Figure 201 , and will begin its search for the "FA1" and "FA2" octets.)

I/O CONTROL REGISTER (ADDRESS = 0X01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	AMI/ ZeroSup*	Unipolar/Bipolar*	TxLine Clk Invert	RxLine Clk Invert	Reframe
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	0	0	0	1

The Framer IC will 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 Microprocessor.
3. Asserting both the RxLOF and RxOOF bit-fields within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	1	1	1	1	1	1	1

7.3.2.4 Performance Monitoring of the Frame Synchronization Section, within the Receive E3 Framer block

The user can monitor the number of framing bytes (FA1 and FA2 bytes) errors that have been detected by the Receive E3 Framer block. This is accom-

plished by periodically reading the PMON Framing Bit/Byte Error Event Count Registers (Address = 0x52 and 0x53). The byte format of these registers are presented below.

7.3.2.5 The RxOOF and RxLOF output pin.

The user can roughly determine the current framing state that the Receive E3 Framer block is operating in by reading the logic state of the RxOOF and the RxLOF output pins. Table 101 presents the relationship between the state of the RxOOF and RxLOF output pins, and the Framing State of the Receive E3 Framer block.

TABLE 101: THE RELATIONSHIP BETWEEN THE LOGIC STATE OF THE RXOOF AND RXLOF OUTPUT PINS, AND THE FRAMING STATE OF THE RECEIVE E3 FRAMER BLOCK

RXLOF	RXOOF	FRAMING STATE OF THE RECEIVE E3 FRAMER BLOCK
0	0	In Frame
0	1	OOF Condition (The Receive E3 Framer block is operating in the 3ms OOF period).
1	0	Invalid
1	1	LOF Condition

7.3.2.6 E3 Receive Alarms

7.3.2.6.1 The Loss of Signal (LOS) Alarm

Declaring an LOS Condition

The Receive E3 Framer block will declare a Loss of Signal (LOS) Condition, when it detects 32 consecutive incoming “0’s” via the RxPOS and RxNEG input pins or if the ExtLOS input pin (from the XRT7300 DS3/E3/STS-1 LIU IC) is asserted. The Receive E3 Framer block will indicate that it is declaring an LOS condition by.

- Asserting the RxLOS output pin (e.g., toggling it “High”).
- Setting Bit 4 (RxLOS) of the Rx E3 Configuration & Status Register to “1” as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	1	0	0	0	0

- The Receive E3 Framer block will generate a Change in LOS Condition interrupt request. Upon generating this interrupt request, the Receive E3 Framer block will assert Bit 1 (LOS Interrupt Status within the Rx E3 Framer Interrupt Status Register - 1, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Clearing the LOS Condition

The Receive E3 Framer block will clear 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 block clears the LOS condition, then it will notify the Microprocessor and the external circuitry of this occurrence by:

- Generating the Change in LOS Condition Interrupt to the Microprocessor.
- Clearing Bit 4 (RxLOS) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	1	0	0	0	0

- Clear the RxLOS output pin (e.g., toggle it "Low").

7.3.2.6.2 The AIS (Alarm Indication Status) Condition

Declaring the AIS Condition

The Receive E3 Framer block 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 block will declare an AIS Condition if 7 or less “0s” are detected in each of 2 consecutive E3 frames.

If the Receive E3 Framer block declares an AIS Condition, then it will do the following.

- Generate the Change in AIS Condition Interrupt to the Microprocessor. Hence, the Receive E3 Framer block will assert Bit 1 (AIS Interrupt Status) within the Rx E3 Framer Interrupt Status register - 1, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1

- Assert the RxAIS output pin.
- Set Bit 3 (Rx AIS) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	1	0	0	0

Clearing the AIS Condition

The Receive E3 Framer block will clear the AIS condition when it detects two consecutive E3 frames, with eight or more “zeros” in the incoming data stream. The Receive E3 Framer block will inform the Microprocessor that the AIS Condition has been cleared by:

- Generating the Change in AIS Condition Interrupt to the Microprocessor. Hence, the Receive E3 Framer block will assert Bit 1 (AIS Interrupt Status)

within the Rx E3 Framer Interrupt Status Register - 1.

- Clearing 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.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

7.3.2.6.3 The Far-End-Receive Failure (FERF) Condition

Declaring the FERF Condition

The Receive E3 Framer block 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”. Recall, the bit-format of the MA byte is presented below.

THE MAINTENANCE AND ADAPTATION (MA) BYTE FORMAT

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FERF	FEBE	Payload Type			Payload Dependent		Timing Marker

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) within the Rx E3 Configuration & Status Register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 1 - (E3, ITU-T G.832) (ADDRESS = 0X10)

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	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Writing a “0” into this bit-field causes the Receive E3 Framer block to declare a 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” into this bit-field causes the Receive E3 Framer block to declare a 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 block declares a FERF condition, then it will do the following.

- Generate a Change in FERF Condition interrupt to the MIcroprocessor. Hence, the Receive E3 Framer block will assert Bit 3 (FERF Interrupt Status) within the Rx E3 Framer Interrupt Status register - 2, as depicted below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	0	0

- Set the Rx FERF bit-field, within the Rx E3 Configuration/Status Register to “1”, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	1

Clearing the FERF Condition

The Receive E3 Framer block will clear the FERF condition once it has received a 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 clears the FERF status, then it will do the following:

1. Generate a Change in the FERF Status Interrupt to the Microprocessor.
2. Clear the Bit 0 (RxFERF) within the Rx E3 Configuration & Status register, as depicted below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	1

7.3.2.7 Error Checking of the Incoming E3 Frames

The Receive E3 Framer block performs error-checking on the incoming E3 frame data that it receives from the Remote Terminal Equipment. It performs this error-checking by computing the BIP-8 value of an incoming E3 frame. Once the Receive E3 Framer block 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 block will conclude that this particular frame has been properly received. The Receive E3 Framer block will then inform the Remote Terminal Equipment of this fact by having the Local Terminal Equipment Transmit E3 Framer block send the Remote Terminal an E3 frame, with the FEBE bit-field, within the MA byte, set to “0”.

This procedure is illustrated in Figure 203 and Figure 204 .

Figure 203 illustrates the Local Receive E3 Framer receiving an error-free E3 frame. In this figure, the locally computed BIP-8 value of “0x5A” matches that received from the Remote Terminal, within the EM byte-field. Figure 204 illustrates the subsequent ac-

tion of the Local Transmit E3 Framer block, which will transmit an E3 frame to the Remote Terminal, with the FEBE bit-field set to “0” . This signaling indicates that the Local Receive E3 Framer has received an error-free E3 frame.

FIGURE 203. THE LOCAL RECEIVE E3 FRAMER BLOCK, RECEIVING AN E3 FRAME FROM THE REMOTE TERMINAL WITH A CORRECT EM BYTE.

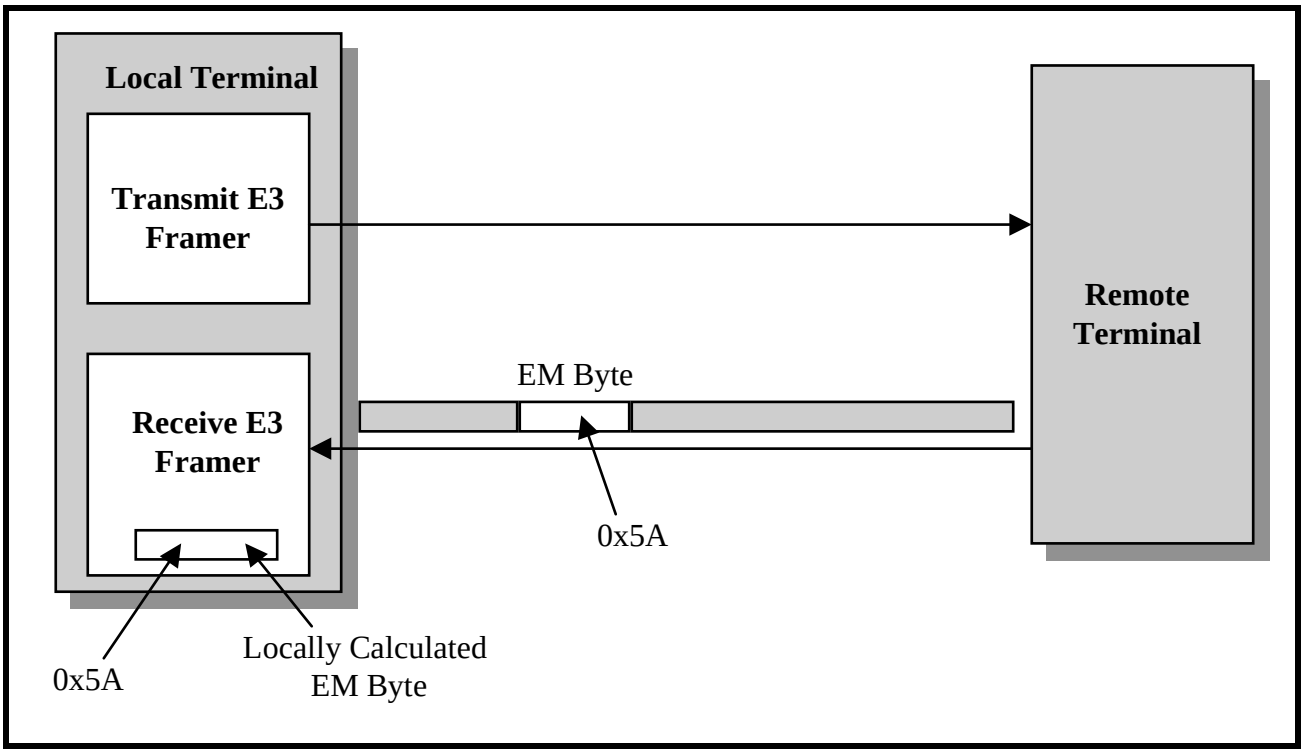
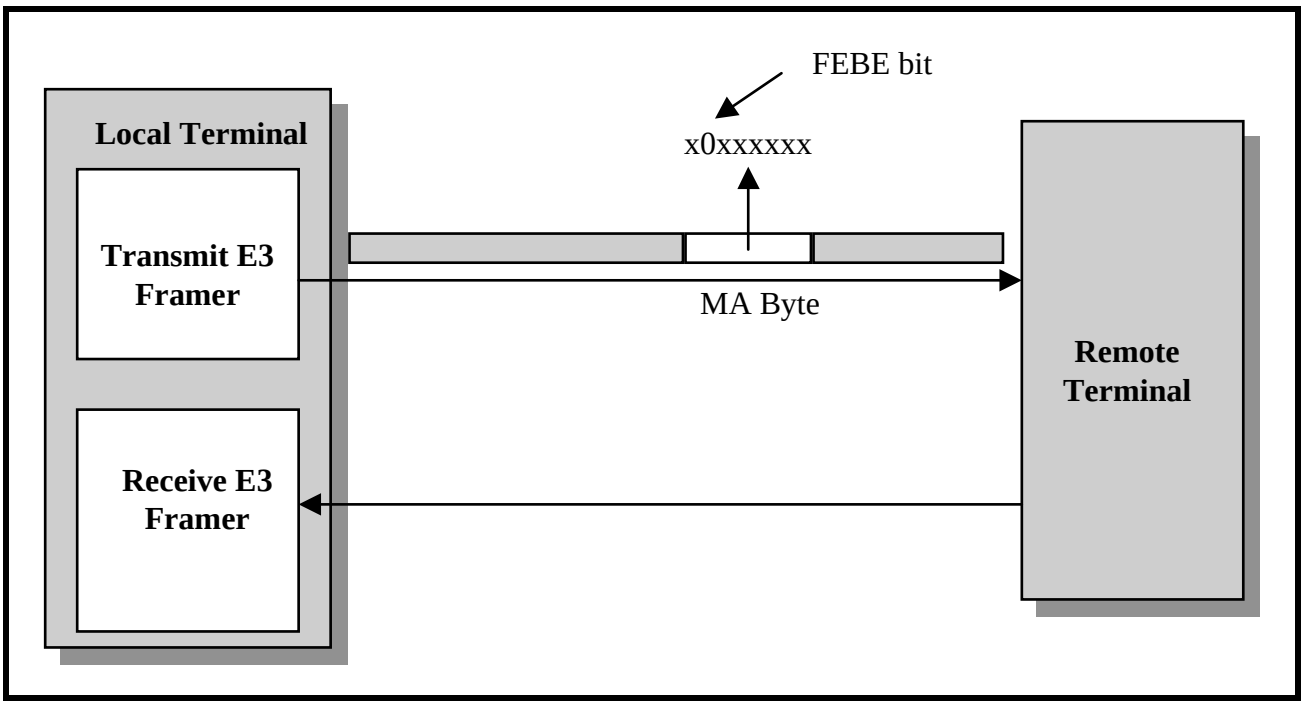


FIGURE 204. THE LOCAL RECEIVE E3 FRAMER BLOCK, TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL WITH THE FEBE BIT WITHIN THE MA BYTE-FIELD SET TO “0”



However, if the locally computed BIP-8 value does not match the EM byte of the corresponding E3 frame, then the Receive E3 Framer block will do the following.

- It will inform the Remote Terminal of this fact by having the Local Transmit E3 Framer block send the Remote Terminal an E3 frame, with the FEBE bit-field, within the MA byte, set to “1”. This phenomenon is illustrated below in Figure 205 and Figure 206 .

Figure 205 illustrates the Local Receive E3 Framer receiving an errored E3 frame. In this figure, the Local Receive E3 Frame block is receiving an E3 frame with an EM byte containing the value “0x5A”. This value does not match the locally computed EM byte value of “0x5B”. Consequently, there is an error in this E3 frame.

Figure 206 illustrates the subsequent action of the Local Transmit E3 Framer block, which will transmit an E3 frame, with the FEBE bit-field set to “1” to the Remote Terminal. This signaling indicates that the Local Receive E3 Framer block has received an errored E3 frame.

FIGURE 205. THE LOCAL RECEIVE E3 FRAMER BLOCK, RECEIVING AN E3 FRAME FROM THE REMOTE TERMINAL WITH AN INCORRECT EM BYTE.

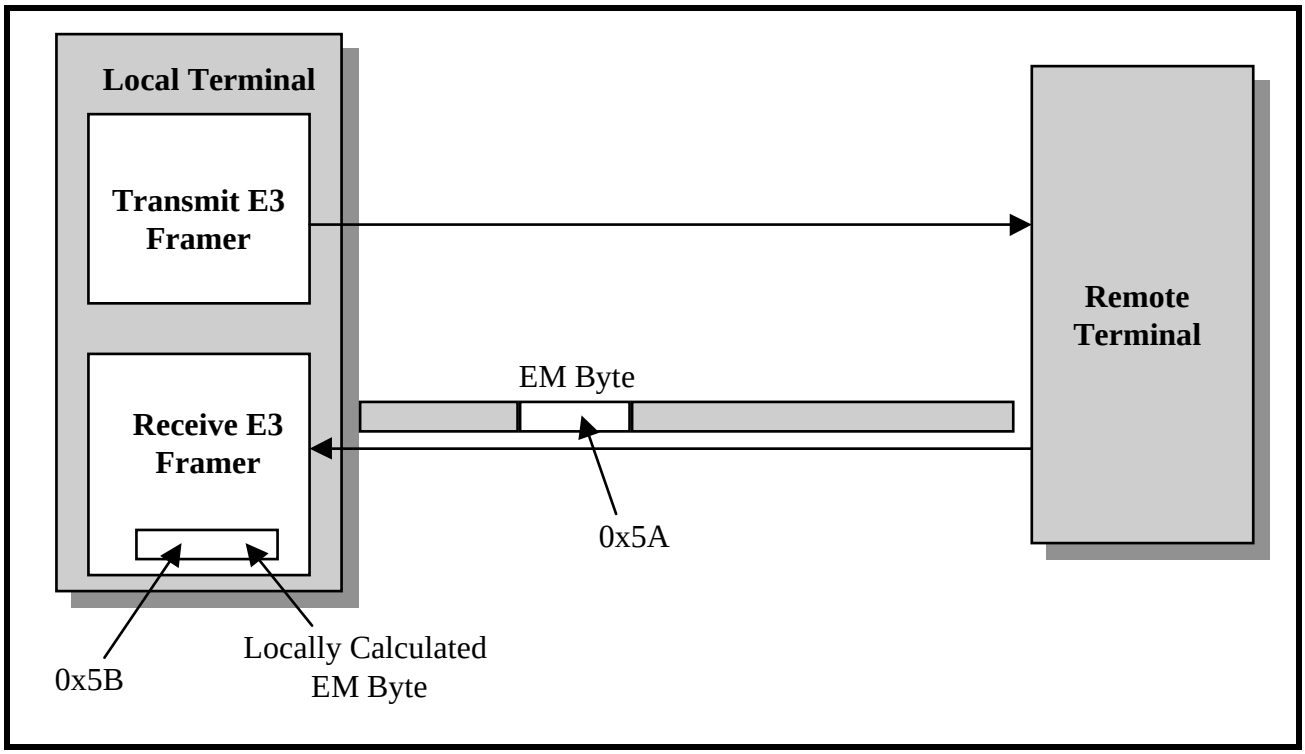
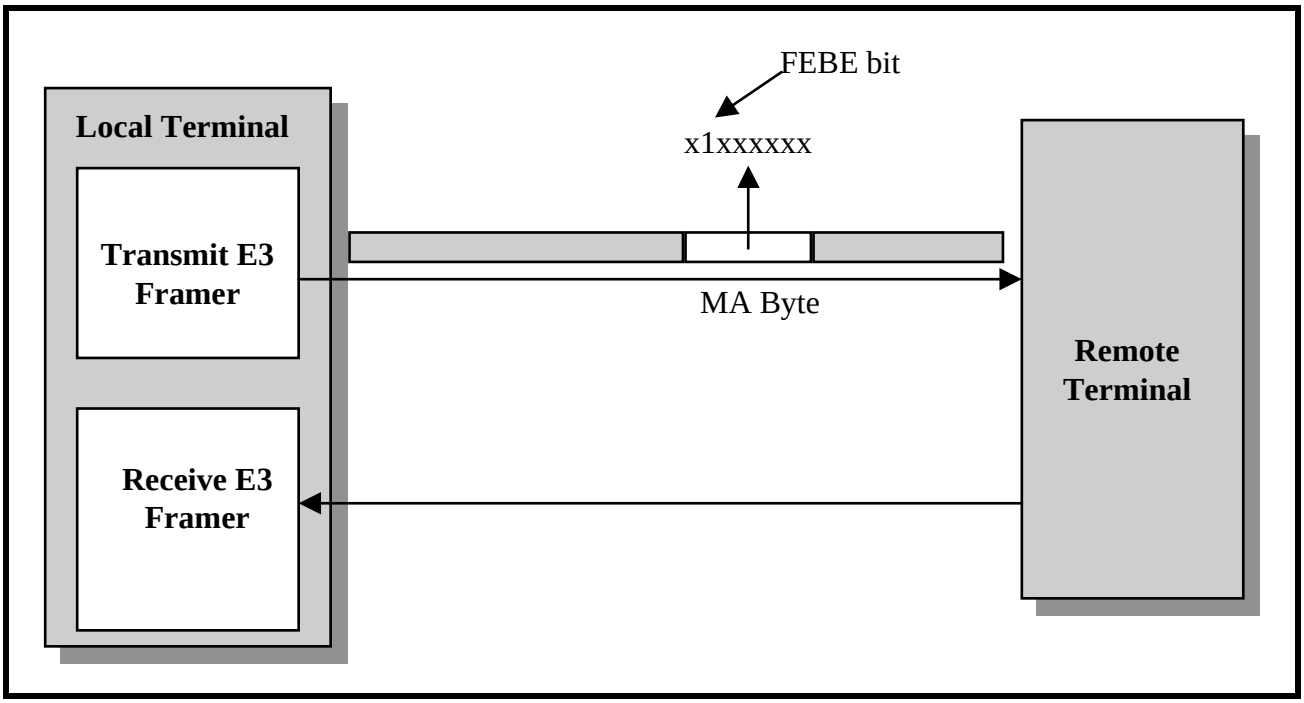


FIGURE 206. THE LOCAL RECEIVE E3 FRAMER BLOCK, TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL WITH THE FEBE BIT WITHIN THE MA BYTE-FIELD SET TO "1"



In additional to the FEBE bit-field signaling, the Receive E3 Framer block will generate the BIP-8 Error Interrupt to the Microprocessor. Hence, it will set bit 2 (BIP-8 Error Interrupt Status) to "1", as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

Finally, the Receive E3 Framer block will increment the PMON Parity Error Count registers. The byte format of these registers are presented below.

PMON PARITY ERROR COUNT REGISTER - MSB (ADDRESS = 0X54)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Parity Error Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON PARITY ERROR COUNT REGISTER - LSB (ADDRESS = 0X55)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Parity Error Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The user can determine the number of BIP-8 Errors that have been detected by the Receive E3 Framer block, since the last read of these registers. These registers are reset-upon-read.

7.3.2.8 Processing of the Far-End-Block Error (FEBE) Bit-fields

Whenever the Receive E3 Framer detects an error in the incoming E3 frame, via EM byte verification, it will inform the Local Transmit E3 Framer of this fact. The Local Transmit E3 Framer will, in turn, notify the Remote 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 Microprocessor/Microcontroller. Hence, the Receive E3 Framer block will set bit 4 (FEBE Interrupt Status) within the Rx E3 Framer Interrupt Status Register - 2, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	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 0x56 and 0x57 in the Framer Address space. The byte-format of these registers are presented below.

PMON FEBE EVENT COUNT REGISTER - MSB (ADDRESS = 0X56)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FEBE Event Count - High Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FEBE EVENT COUNT REGISTER - LSB (ADDRESS = 0X57)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FEBE Event Count - Low Byte							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

The user can determine the total number of FEBE Events (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.

7.3.2.9 Receiving the Trail Trace Buffer Messages

The XRT74L74 Framer IC device contains 16 bytes worth of Transmit Trail Trace Buffers, and 16 bytes worth of Receive Trail Trace Buffers, as described below. The role of the Transmit Trail Trace Buffers are described in Section 5.2.4.2..

The XRT74L74 DS3/E3 Framer IC contains 16 Receive Trail Trace Buffer registers (e.g., RxTTB-0 through RxTTB-15). The purpose of these registers are to receive and store the incoming Trail Access Point Identifier from the Remote Transmitting Terminal.

The Local 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 Remote Transmit E3 Framer block will group 16 consecutive E3 frames into a Trail Trace Buffer super-frame. When the Remote 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 Local Receive E3 Framer block receives an E3 frame, containing a value in the TR byte that has a “1” in the MSB position, then it (the Receive E3 Framer block) will write this value into the RxTTB-0 Register (Address = 0x1C). Once this occurs, the Receive E3 Framer block will notify the Microprocessor of this new incoming Trail Trace Buffer message by generating the Change in Trail Trace Buffer Message interrupt. The Receive E3 Framer block will also set bit 6 (TTB Change Interrupt Status) within the Rx E3 Framer Interrupt Status Register - 2, as depicted below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	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 = 0x1D), and so on until all 16 bytes have been received.

NOTES:

1. *Anytime the Receive E3 Framer block 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 block will (1) write the contents of the TR byte-field (in this E3 frame) into the RxTTB-0 Register,*
2. *It will generate the Change in Trail Trace Buffer Interrupt. The Receive E3 Framer will do these things independent of the number of E3 frames that have been received since the last occurrence of the Change in Trail Trace Buffer Interrupt. Hence, 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 position. All remaining Tx TTB registers (e.g., TxTTB-1 through TxTTB-15) must contain octets with a “0” in the MSB position.*
3. *The Framer IC 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.*

7.3.3 The Receive HDLC Controller Block

The Receive E3 HDLC Controller block can be used to receive message-oriented signaling (MOS) type data link messages from the remote terminal equipment.

The MOS types of HDLC message processing is discussed in detail below.

The Message Oriented Signaling (e.g., LAP-D) Processing via the Receive DS3 HDLC Controller block

The LAPD Receiver (within the Receive E3 HDLC Controller block) allows the user to receive PMDL messages from the remote terminal equipment, via

the Inbound E3 frames. In this case, the Inbound message bits will be carried by either the “GC” or the “NR” byte-fields within each E3 Frame. The remote LAPD Transmitter will transmit a LAPD Message to the Near-End Receiver via either one of these bytes within each E3 Frame. The LAPD Receiver will receive and store the information portion of the received LAPD frame into the Receive LAPD Message Buffer, which is located at addresses: 0xDE through 0x135 within the on-chip RAM. The LAPD Receiver has the following responsibilities.

- Framing to the incoming LAPD Messages
- Filtering out stuffed "0's" (within the information payload)
- Storing the Frame Message into the Receive LAPD Message Buffer
- Perform Frame Check Sequence (FCS) Verification
- Provide status indicators for
  - End of Message (EOM)
  - Flag Sequence Byte detected
  - Abort Sequence detected
  - Message Type
  - C/R Type
  - The occurrence of FCS Errors

The LAPD receiver's actions are facilitated 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 (0x7E), as depicted in Figure 207 .

FIGURE 207. LAPD MESSAGE FRAME FORMAT

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)		

Where: Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The 16 bit FCS is calculated using CRC-16, x16 + x12 + x5 + 1

The local µP (at the remote terminal), while assembling the LAPD Message frame, will insert an additional byte at the beginning of the information (payload) field. This first byte of the information field indicates the type and size of the message being transferred. The value of this information field and the corresponding message type/size follow:

CL Path Identification = 0x38 (76 bytes)

IDLE Signal Identification = 0x34 (76 bytes)

Test Signal Identification = 0x32 (76 bytes)

ITU-T Path Identification = 0x3F (82 bytes)

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. The byte-field within each E3 frame which will be carrying the comprising octets of the LAPD Message frame must be specified and
2. The LAPD Receiver must be enabled.

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.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				DL from NR	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	0	0	1	0

Writing a “0” into 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” into this bit-field causes the LAPD Receiver 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 2 (RxLAPD Enable) of the Rx E3 LAPD Control Register, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				DL from NR	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	0	1	1	0

Once the LAPD Receiver has been enabled, it will begin searching for the Flag Sequence octet (0x7E), 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.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	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 or end 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 “1’s”). If this next set of data is an

ABORT Sequence, then the LAPD Receiver will assert the RxABORT bit-field (Bit 6) within the Rx E3 LAPD Status Register, as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	1	0	0	0	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 Framer IC.

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 Remote 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 "1's".*

For more information on how the LAPD Transmitter inserted these stuff bits, please see Section 5.2.3.1.

2. 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 locally-computed CRC-16 value, then the LAPD Receiver will conclude that it has received this LAPD Message frame in an error-free manner.

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 erred.

The LAPD Receiver will indicate the results of this FCS Verification process by setting Bit 2 (RxFCS Error) within the Rx E3 LAPD Status Register, to the appropriate value as tabulated below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	1	0	0

If the LAPD Receiver detects an error in the FCS value, then it will set the RxFCS Error bit-field to "1". Conversely, if the LAPD Receiver does not detect an error in the FCS value, the it will clear the RxFCS Error bit-field to "0".

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 is read in from the Receive LAPD Message buffer, by first checking the state of this bit-field.*

3. Check and Report the State of the "C/R" Bit-field

After 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) within the Rx E3 LAPD Status Register, as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	1	0	0	0

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.

4. 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 that from Section _, 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 Remote LAPD Receiver. This Message-Type Identification octet is transported to the Remote LAPD Receiver, along with the rest of the LAPD frame. From this Message Type Identification octet, the LAPD Receiver will know the type of 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.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0

Table 102 presents the relationship between the contents of RxLAPDType[1:0] and the type of message received by the LAPD Receiver.

TABLE 102: 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.*

5. Inform the Local Microprocessor/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 Microprocessor that a LAPD Message frame has been received and is ready for user-system handling. The LAPD Receiver will inform the Microprocessor/Microcontroller and the external circuitry by:

- Generating a LAPD Message Frame Received interrupt to the Microprocessor. The purpose of this interrupt is to let the Microprocessor 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 0 (RxLAPD Interrupt Status) within the Rx E3 LAPD Control Register to "1" as depicted below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				DL from NR	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	0	0	0	1

- Setting Bit 1 (End of Message) within the Rx E3 LAPD Status Register, to "1" as depicted below.

RXE3 LAPD STATUS REGISTER (ADDRESS = 0X19)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	Rx ABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	1	0

In summary, Figure 208 presents a flow chart depicting how the LAPD Receiver functions.

FIGURE 208. FLOW CHART DEPICTING THE FUNCTIONALITY OF THE LAPD RECEIVER

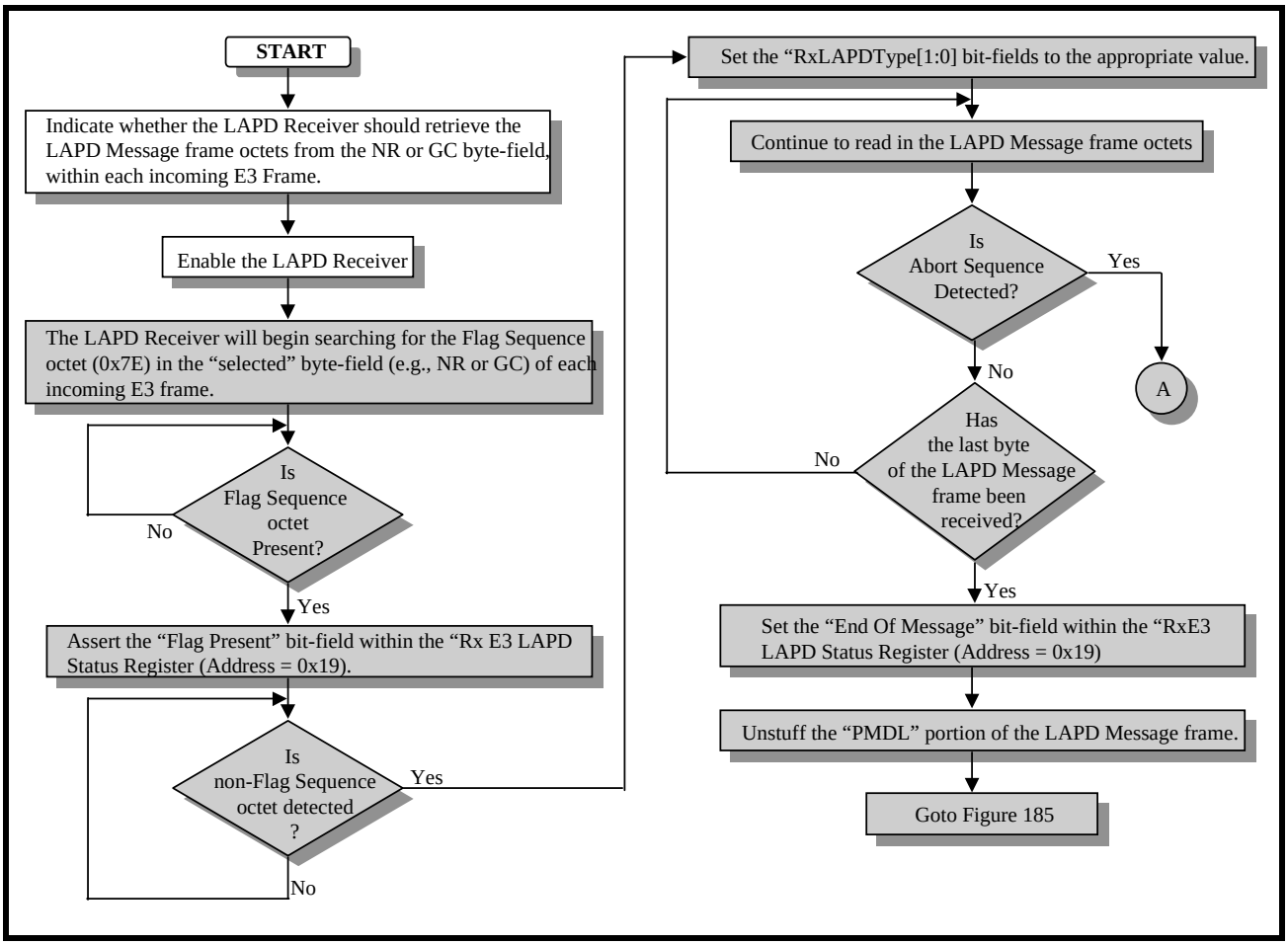
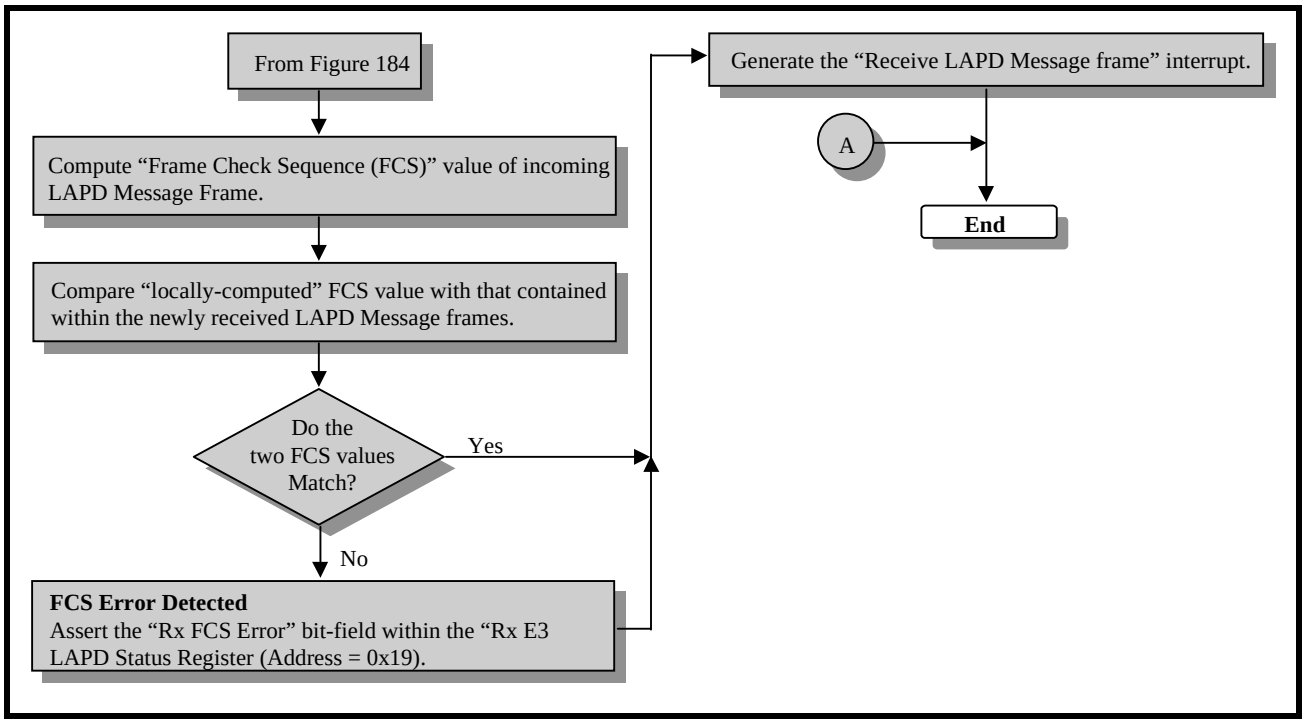


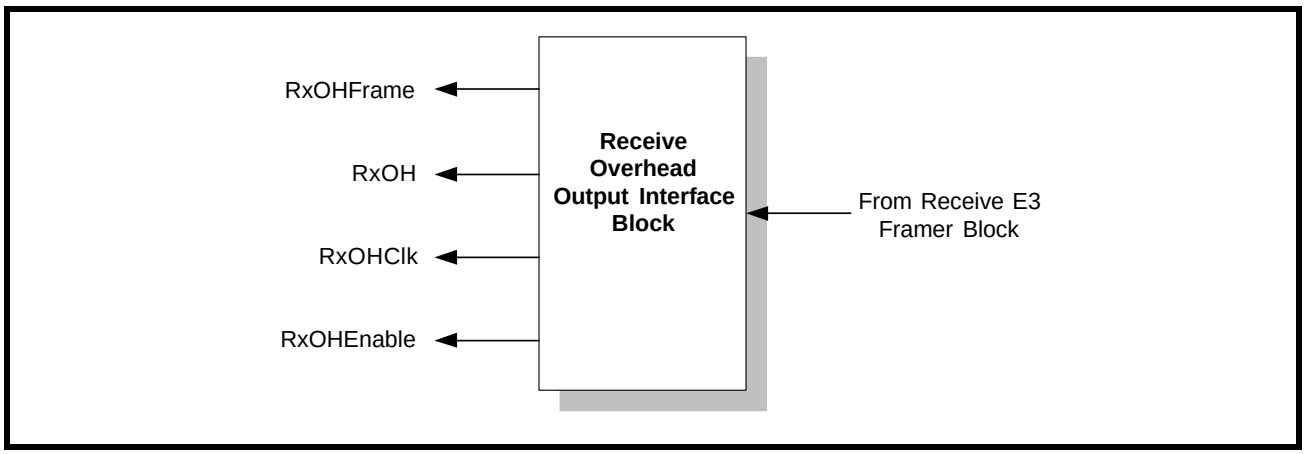
FIGURE 209. FLOW CHART DEPICTING THE FUNCTIONALITY OF THE LAPD RECEIVER (CONTINUED)



7.3.4 The Receive Overhead Data Output Interface

Figure 210 presents a simple illustration of the Receive Overhead Data Output Interface block within the XRT74L74.

FIGURE 210. THE RECEIVE OVERHEAD OUTPUT INTERFACE BLOCK



The E3, ITU-T G.832 frame consists of 537 bytes. Of these bytes, 530 bytes are payload bits and the remaining 7 bytes are overhead bytes. The XRT74L74 has been designed to handle and process both the payload type and overhead type bytes for each E3 frame.

Within the Receive Section of the XRT74L74, the Receive Payload Data Output Interface block has been designed to handle the payload bits. Likewise, the Receive Overhead Data Output Interface block has been designed to handle and process the overhead bits.

The Receive Overhead Data Output Interface block unconditionally outputs the contents of all overhead bits. The XRT74L74 does not offer the user a means to shut off this transmission of data. However, the Receive Overhead Output Interface block does provide the user with the appropriate output signals for

external Data Link Layer equipment to sample and process these overhead bits, via the following two methods.

- Method 1- Using the RxOHClk clock signal.
- Method 2 - Using the RxClk and RxOHEnable output signals.

Each of these methods are described below.

7.3.4.1 Method 1 - Using the RxOHClk Clock signal

The Receive Overhead Data Output Interface block consists of four (4) signals. Of these four signals, the following three signals are to be used when sampling the E3 overhead bits via Method 1.

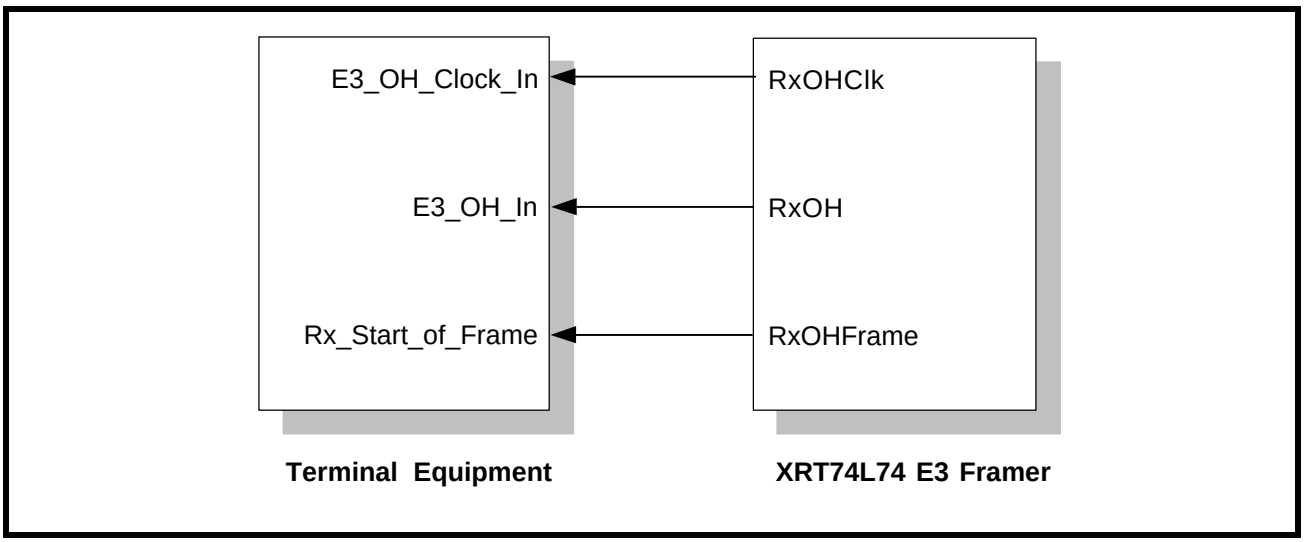
- RxOH
- RxOHClk
- RxOHFrame

Each of these signals are listed and described below in Table 103 .

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 1)

Figure 211 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment when using Method 1, to sample and process the overhead bits from the Inbound E3 data stream.

FIGURE 211. HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 1.



Method 1 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the Inbound E3 data stream (via the Receive Overhead Data Output Interface block) then it is expected to do the following:

1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input signal) on the rising edge of the RxOHClk (e.g., the E3_OH_Clock_In) signal.
2. Keep track of the number of rising clock edges that have occurred in the RxOHClk (e.g., the E3_OH_Clock_In) signal, since the last time the RxOHFrame signal was sampled “High”. By doing this, the Terminal Equipment will be able to keep track of which overhead byte is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.

TABLE 103: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming E3 frames, via this pin. The Receive Overhead Data Output Interface block will output a given overhead bit, upon the falling edge of RxOHClk. Hence, the external data link equipment should sample the data, at this pin, upon the rising edge of RxOHClk. The XRT74L74 will always output the E3 Overhead bits via this output pin. There are no external input pins or register bit settings available that will disable this output pin.
RxOHClk	Output	**Receive Overhead Data Output Interface Clock Signal:** The XRT74L74 will output the Overhead bits (within the incoming E3 frames), via the RxOH output pin, upon the falling edge of this clock signal. As a consequence, the user's data link equipment should use the rising edge of this clock signal to sample the data on both the RxOH and RxOHFrame output pins. This clock signal is always active.
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin "High" (for one period of the RxOHClk signal), whenever the first overhead bit within a given E3 frame is being driven onto the RxOH output pin.

Table 104 relates the number of rising clock edges (in the RxOHClk signal, since the RxOHFrame signal was last sampled "High") to the E3 Overhead bit that is being output via the RxOH output pin.

TABLE 104: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (Clock edge is coincident with RxOHFrame being detected "High")	FA1 Byte - Bit 7
1	FA1 Byte - Bit 6
2	FA1 Byte - Bit 5
3	FA1 Byte - Bit 4
4	FA1 Byte - Bit 3
5	FA1 Byte - Bit 2
6	FA1 Byte - Bit 1
7	FA1 Byte - Bit 0
8	FA2 Byte - Bit 7
9	FA2 Byte - Bit 6
10	FA2 Byte - Bit 5
11	FA2 Byte - Bit 4
12	FA2 Byte - Bit 3
13	FA2 Byte - Bit 2
14	FA2 Byte - Bit 1

TABLE 104: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

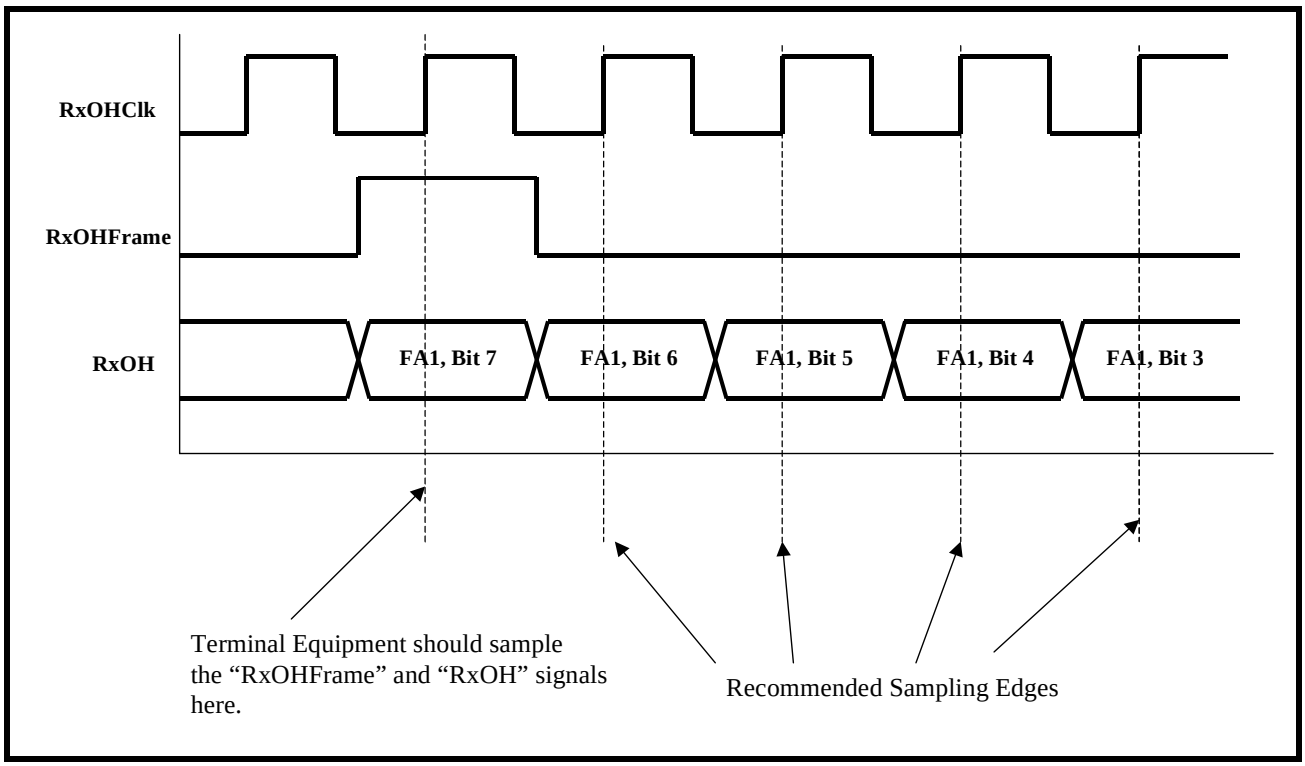
NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
15	FA2 Byte - Bit 0
16	EM Byte - Bit 7
17	EM Byte - Bit 6
18	EM Byte - Bit 5
19	EM Byte - Bit 4
20	EM Byte - Bit 3
21	EM Byte - Bit 2
22	EM Byte - Bit 1
23	EM Byte - Bit 0
24	TR Byte - Bit 7
25	TR Byte - Bit 6
26	TR Byte - Bit 5
27	TR Byte - Bit 4
28	TR Byte - Bit 3
29	TR Byte - Bit 2
30	TR Byte - Bit 1
31	TR Byte - Bit 0
32	MA Byte - Bit 7
33	MA Byte - Bit 6
34	MA Byte - Bit 5
35	MA Byte - Bit 4
36	MA Byte - Bit 3
37	MA Byte - Bit 2
38	MA Byte - Bit 1
39	MA Byte - Bit 0
40	NR Byte - Bit 7
41	NR Byte - Bit 6
42	NR Byte - Bit 5
43	NR Byte - Bit 4
44	NR Byte - Bit 3

TABLE 104: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN RXOHCLK, (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RISING CLOCK EDGES IN RXOHCLK	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
45	NR Byte - Bit 2
46	NR Byte - Bit 1
47	NR Byte - Bit 0
48	GC Byte - Bit 7
49	GC Byte - Bit 6
50	GC Byte - Bit 5
51	GC Byte - Bit 4
52	GC Byte - Bit 3
53	GC Byte - Bit 2
54	GC Byte - Bit 1
55	GC Byte - Bit 0

Figure 212 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 1 is being used to sample the incoming E3 overhead bits.

FIGURE 212. THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD OUTPUT INTERFACE FOR METHOD 1.



7.3.4.2 Method 2 - Using RxOutClk and the RxOHEnable signals

Method 1 requires that the Terminal Equipment be able to handle an additional clock signal, RxOHClk.

However, there may be a situation in which the Terminal Equipment circuitry does not have the means to deal with this extra clock signal, in order to use the Receive Overhead Data Output Interface. Method 2 involves the use of the following signals.

- RxOH
- RxOutClk
- RxOHEnable
- RxOHFrame

Each of these signals are listed and described in Table 105 .

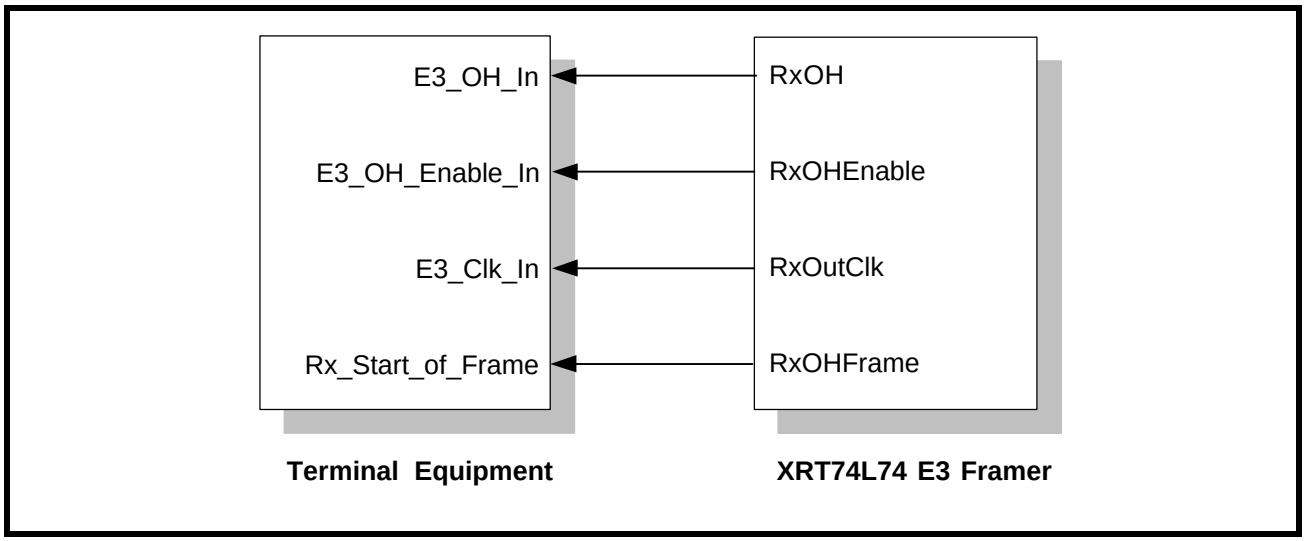
TABLE 105: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK (METHOD 2)

SIGNAL NAME	TYPE	DESCRIPTION
RxOH	Output	**Receive Overhead Data Output pin:** The XRT74L74 will output the overhead bits, within the incoming E3 frames, via this pin. The Receive Overhead Output Interface will pulse the RxOHEnable output pin (for one RxOutClk period) at approximately the middle of the RxOH bit period. The user is advised to design the Terminal Equipment to latch the contents of the RxOH output pin, whenever the RxOHEnable output pin is sampled “High” on the falling edge of RxOutClk.
RxOHEnable	Output	**Receive Overhead Data Output Enable - Output pin:** The XRT74L74 will assert this output signal for one RxOutClk period when it is safe for the Terminal Equipment to sample the data on the RxOH output pin.
RxOHFrame	Output	**Receive Overhead Data Output Interface - Start of Frame Indicator:** The XRT74L74 will drive this output pin “High” (for one period of the RxOH signal), whenever the first overhead bit, within a given E3 frame is being driven onto the RxOH output pin.
RxOutClk	Output	**Receive Section Output Clock Signal:** This clock signal is derived from the RxLineClk signal (from the LIU) for loop-timing applications, and the TxInClk signal (from a local oscillator) for local-timing applications. For E3 applications, this clock signal will operate at 34.368MHz. The user is advised to design the Terminal Equipment to latch the contents of the RxOH pin, anytime the RxOHEnable output signal is sampled “High” on the falling edge of this clock signal.

Interfacing the Receive Overhead Data Output Interface block to the Terminal Equipment (Method 2)

Figure 213 illustrates how one should interface the Receive Overhead Data Output Interface block to the Terminal Equipment, when using Method 2 to sample and process the overhead bits from the Inbound E3 data stream.

FIGURE 213. HOW TO INTERFACE THE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 2



Method 2 Operation of the Terminal Equipment

If the Terminal Equipment intends to sample any overhead data from the Inbound E3 data stream (via the Receive Overhead Data Output Interface), then it is expected to do the following.

1. Sample the state of the RxOHFrame signal (e.g., the Rx_Start_of_Frame input) on the falling edge of the RxOutClk clock signal, whenever the RxOHEnable output signal is also sampled “High”.
2. Keep track of the number of times that the RxOHEnable signal has been sampled “High” since the last time the RxOHFrame was also sampled “High”. By doing this, the Terminal Equipment will be able to keep track of which overhead bit is being output via the RxOH output pin. Based upon this information, the Terminal Equipment will be able to derive some meaning from these overhead bits.
3. Table 106 relates the number of RxOHEnable output pulses (that have occurred since both the RxOHFrame and the RxOHEnable pins were both sampled “High”) to the E3 overhead bit that is being output via the RxOH output pin.

TABLE 106: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
0 (Clock edge is coincident with RxOHFrame being detected “High”)	FA1 Byte - Bit 7
1	FA1 Byte - Bit 6
2	FA1 Byte - Bit 5
3	FA1 Byte - Bit 4
4	FA1 Byte - Bit 3
5	FA1 Byte - Bit 2
6	FA1 Byte - Bit 1
7	FA1 Byte - Bit 0
8	FA2 Byte - Bit 7
9	FA2 Byte - Bit 6

TABLE 106: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

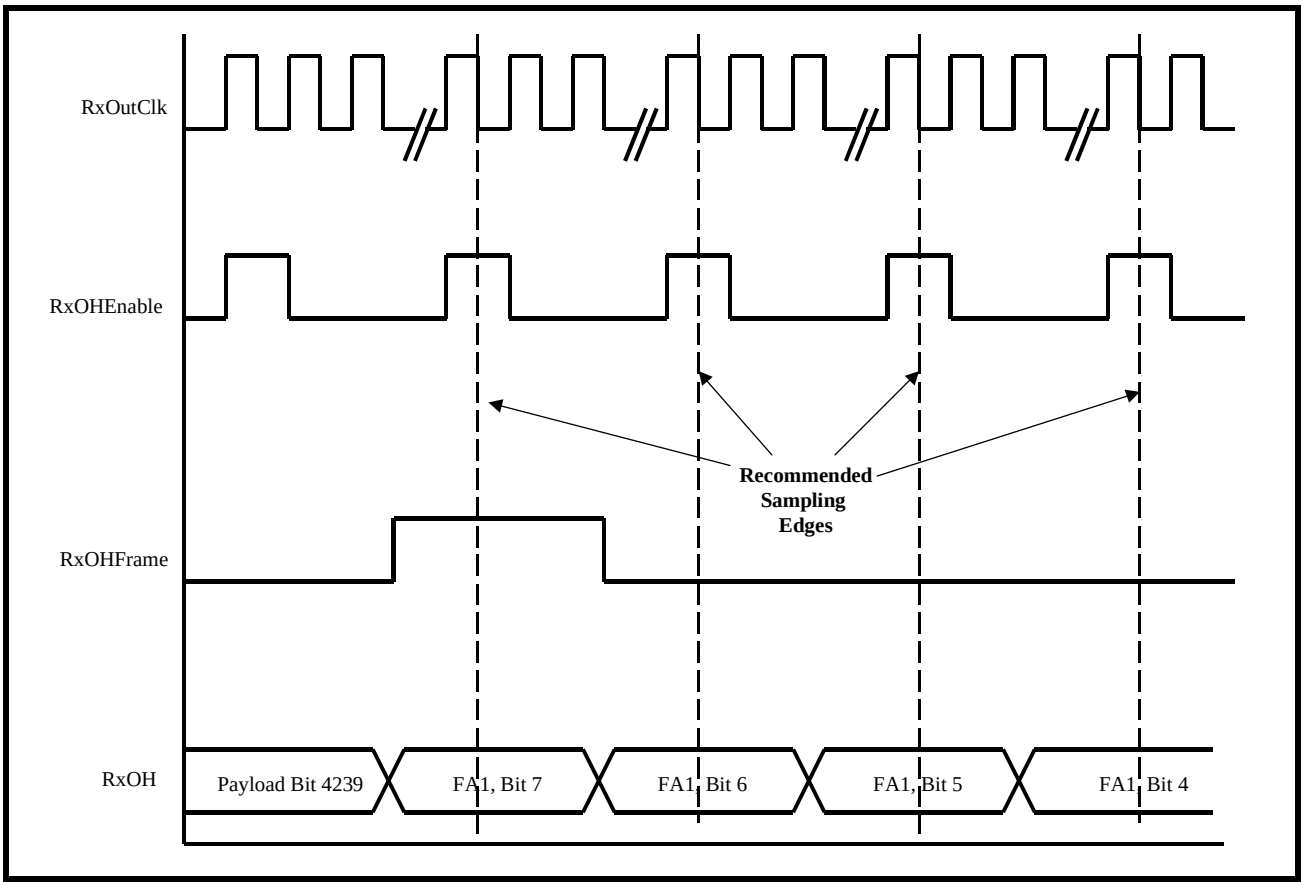
NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
10	FA2 Byte - Bit 5
11	FA2 Byte - Bit 4
12	FA2 Byte - Bit 3
13	FA2 Byte - Bit 2
14	FA2 Byte - Bit 1
15	FA2 Byte - Bit 0
16	EM Byte - Bit 7
17	EM Byte - Bit 6
18	EM Byte - Bit 5
19	EM Byte - Bit 4
20	EM Byte - Bit 3
21	EM Byte - Bit 2
22	EM Byte - Bit 1
23	EM Byte - Bit 0
24	TR Byte - Bit 7
25	TR Byte - Bit 6
26	TR Byte - Bit 5
27	TR Byte - Bit 4
28	TR Byte - Bit 3
29	TR Byte - Bit 2
30	TR Byte - Bit 1
31	TR Byte - Bit 0
32	MA Byte - Bit 7
33	MA Byte - Bit 6
34	MA Byte - Bit 5
35	MA Byte - Bit 4
36	MA Byte - Bit 3
37	MA Byte - Bit 2
38	MA Byte - Bit 1
39	MA Byte - Bit 0
40	NR Byte - Bit 7

TABLE 106: THE RELATIONSHIP BETWEEN THE NUMBER OF RXOHENABLE OUTPUT PULSES (SINCE RXOHFRAME WAS LAST SAMPLED "HIGH") TO THE E3 OVERHEAD BIT, THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN

NUMBER OF RXOHENABLE OUTPUT PULSES	THE OVERHEAD BIT BEING OUTPUT BY THE XRT74L74
41	NR Byte - Bit 6
42	NR Byte - Bit 5
43	NR Byte - Bit 4
44	NR Byte - Bit 3
45	NR Byte - Bit 2
46	NR Byte - Bit 1
47	NR Byte - Bit 0
48	GC Byte - Bit 7
49	GC Byte - Bit 6
50	GC Byte - Bit 5
51	GC Byte - Bit 4
52	GC Byte - Bit 3
53	GC Byte - Bit 2
54	GC Byte - Bit 1
55	GC Byte - Bit 0

Figure 214 presents the typical behavior of the Receive Overhead Data Output Interface block, when Method 2 is being used to sample the incoming E3 overhead bits.

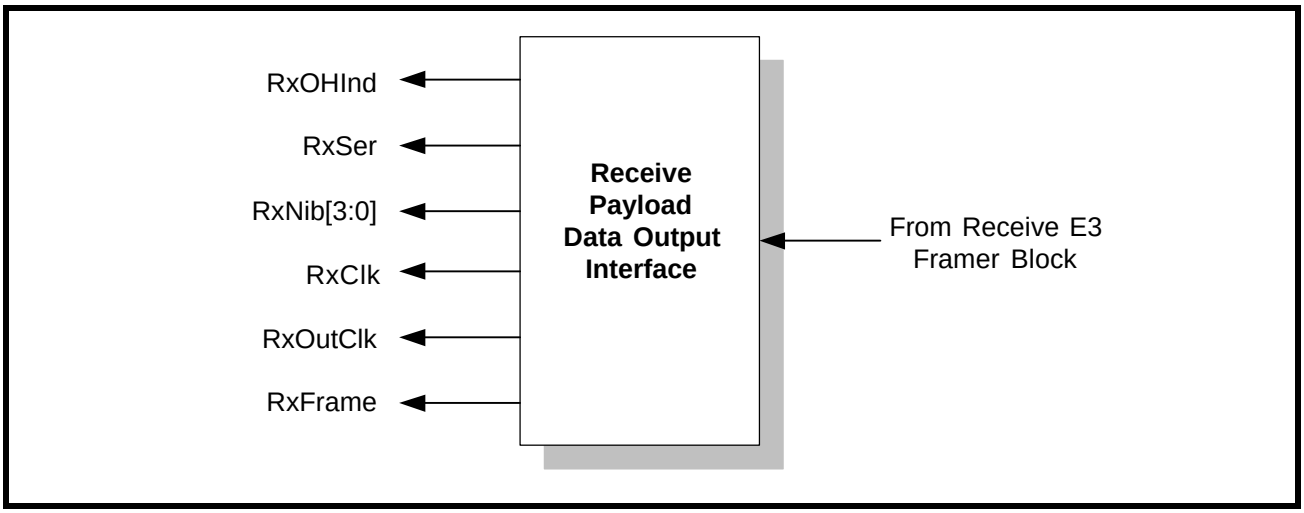
FIGURE 214. THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 2



7.3.5 The Receive Payload Data Output Interface

Figure 215 presents a simple illustration of the Receive Payload Data Output Interface block.

FIGURE 215. THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK



Each of the output pins of the Receive Payload Data Output Interface block are listed in Table 107 and

described below. The exact role that each of these output pins assume, for a variety of operating scenarios are described throughout this section.

TABLE 107: LISTING AND DESCRIPTION OF THE PIN ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	TYPE	DESCRIPTION
RxSer	Output	**Receive Serial Payload Data Output pin:** If the user opts to operate the XRT74L74 in the serial mode, then the chip will output the payload data, of the incoming E3 frames, via this pin. The XRT74L74 will output this data upon the rising edge of RxClk. The user is advised to design the Terminal Equipment such that it will sample this data on the falling edge of RxClk. ***NOTE:*** *This signal is only active if the NibInt input pin is pulled "Low".*
RxNib[3:0]	Output	**Receive Nibble-Parallel Payload Data Output pins:** If the user opts to operate the XRT74L74 in the nibble-parallel mode, then the chip will output the payload data, of the incoming E3 frames, via these pins. The XRT74L74 will output data via these pins, upon the falling edge of the RxClk output pin. The user is advised to design the Terminal Equipment such that it will sample this data upon the rising edge of RxClk. ***NOTE:*** *These pins are only active if the NibInt input pin is pulled "High".*
RxClk	Output	**Receive Payload Data Output Clock pin:** The exact behavior of this signal depends upon whether the XRT74L74 is operating in the Serial or in the Nibble-Parallel-Mode. **Serial Mode Operation** In the serial mode, this signal is a 34.368MHz clock output signal. The Receive Payload Data Output Interface will update the data via the RxSer output pin, upon the rising edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxSer pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation** In this Nibble-Parallel Mode, the XRT74L74 will derive this clock signal, from the RxLineClk signal. The XRT74L74 will pulse this clock 1060 times for each Inbound E3 frame. The Receive Payload Data Output Interface will update the data, on the RxNib[3:0] output pins upon the falling edge of this clock signal. The user is advised to design (or configure) the Terminal Equipment to sample the data on the RxNib[3:0] output pins, upon the rising edge of this clock signal
RxOHInd	Output	**Receive Overhead Bit Indicator Output:** This output pin will pulse "High" whenever the Receive Payload Data Output Interface outputs an overhead bit via the RxSer output pin. The purpose of this output pin is to alert the Terminal Equipment that the current bit, (which is now residing on the RxSer output pin), is an overhead bit and should not be processed by the Terminal Equipment. The XRT74L74 will update this signal, upon the rising edge of RxOHInd. The user is advised to design (or configure) the Terminal Equipment to sample this signal (along with the data on the RxSer output pin) on the falling edge of the RxClk signal. ***NOTE:*** *For E3 applications, this output pin is only active if the XRT74L74 is operating in the Serial Mode. This output pin will be "Low" if the device is operating in the Nibble-Parallel Mode.*
RxFrame	Output	**Receive Start of Frame Output Indicator:** The exact behavior of this pin, depends upon whether the XRT74L74 has been configured to operate in the Serial Mode or the Nibble-Parallel Mode. **Serial Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" (for one bit period) when the Receive Payload Data Output Interface block is driving the very first bit (or Nibble) of a given E3 frame, onto the RxSer output pin. **Nibble-Parallel Mode Operation:** The Receive Section of the XRT74L74 will pulse this output pin "High" for one nibble period, when the Receive Payload Data Output Interface is driving the very first nibble of a given E3 frame, onto the RxNib[3:0] output pins.

Operation of the Receive Payload Data Output Interface block

The Receive Payload Data Output Interface permits the user to read out the payload data of Inbound E3 frames, via either of the following modes.

- Serial Mode
- Nibble-Parallel Mode

Each of these modes are described in detail, below.

7.3.5.1 Serial Mode Operation Behavior of the XRT74L74

If the XRT74L74 has been configured to operate in this mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data, of the incoming E3 frames, upon the rising edge of RxClk.

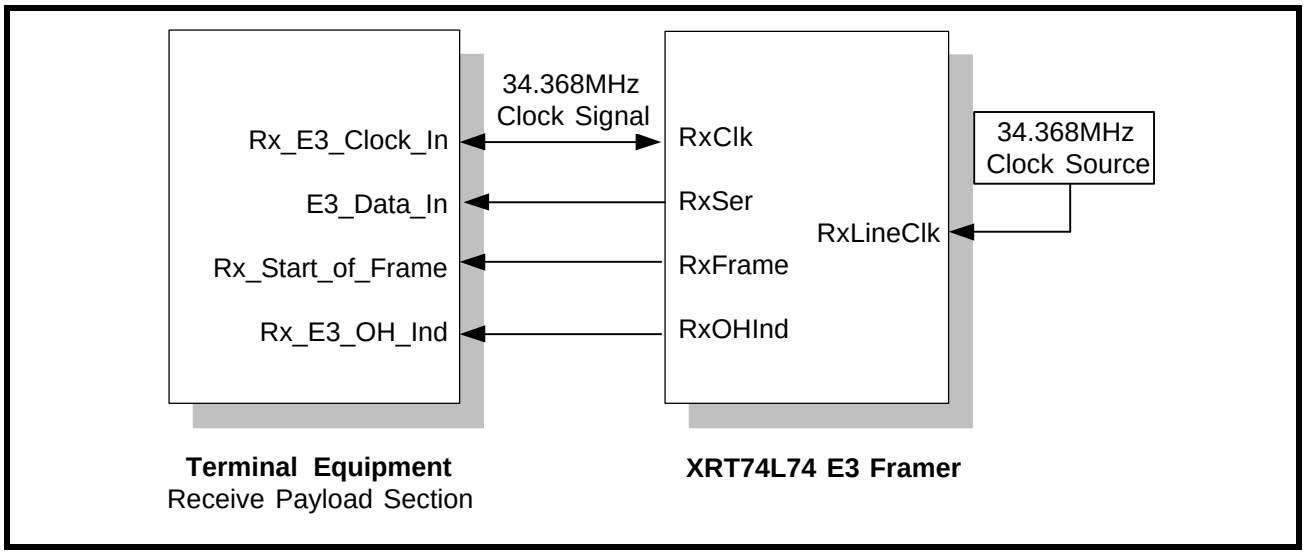
Delineation of Inbound DS3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one bit-period, coincident with it driving the first bit within a given E3 frame, via the RxSer output pin.

Interfacing the XRT74L74 to the Receive Terminal Equipment

Figure 216 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 216. THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK BEING INTERFACED TO THE RECEIVE TERMINAL EQUIPMENT (SERIAL MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxSer output pin, upon the rising edge of RxClk. Hence, the Terminal Equipment should sample the data on the RxSer output pin (or the E3_Data_In pin at the Terminal Equipment) upon the rising edge of RxClk. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the following signals.

- RxFrame
- RxOHInd

The Need for sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first bit of a given E3 frame onto the RxSer output pin. If knowledge of the E3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

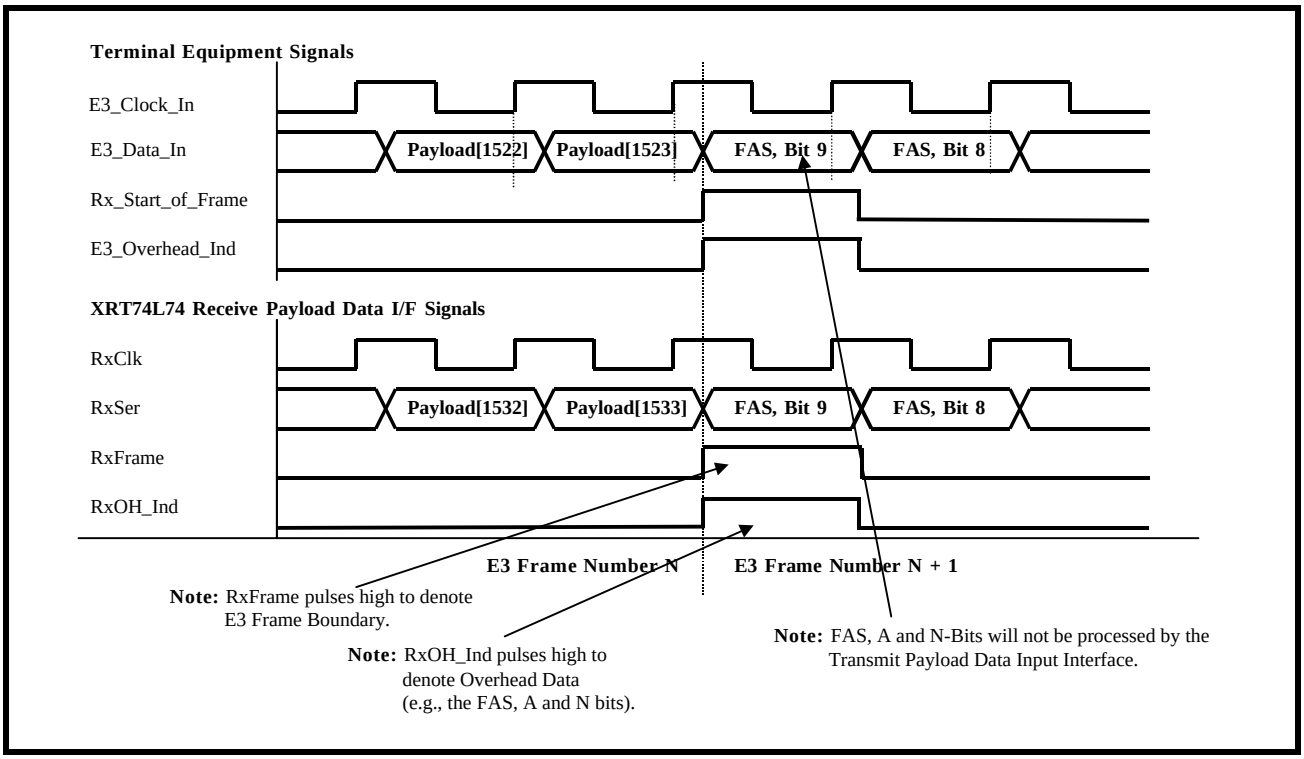
The Need for sampling RxOHInd

The XRT74L74 will indicate that it is currently driving an overhead bit onto the RxSer output pin, by pulsing the RxOHInd output pin "High". If the Terminal Equipment samples this signal "High", then it should know that the bit, that it is currently sampling via the RxSer pin is an overhead bit and should not be processed.

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Serial Mode Operation is illustrated in Figure 217 .

FIGURE 217. THE BEHAVIOR OF THE SIGNALS BETWEEN THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT74L74 AND THE TERMINAL EQUIPMENT



7.3.5.2 Nibble-Parallel Mode OperationBehavior of the XRT74L74

If the XRT74L74 has been configured to operate in the Nibble-Parallel Mode, then the XRT74L74 will behave as follows.

Payload Data Output

The XRT74L74 will output the payload data of the incoming E3 frames, via the RxNib[3:0] output pins, upon the rising edge of RxClk.

NOTES:

1. *In this case, RxClk will function as the Nibble Clock signal between the XRT74L74 the Terminal Equipment. The XRT74L74 will pulse the RxClk output signal "High" 1060 times, for each Inbound E3 frame.*
2. *Unlike Serial Mode operation, the duty cycle of RxClk, in Nibble-Parallel Mode operation is approximately 25%.*

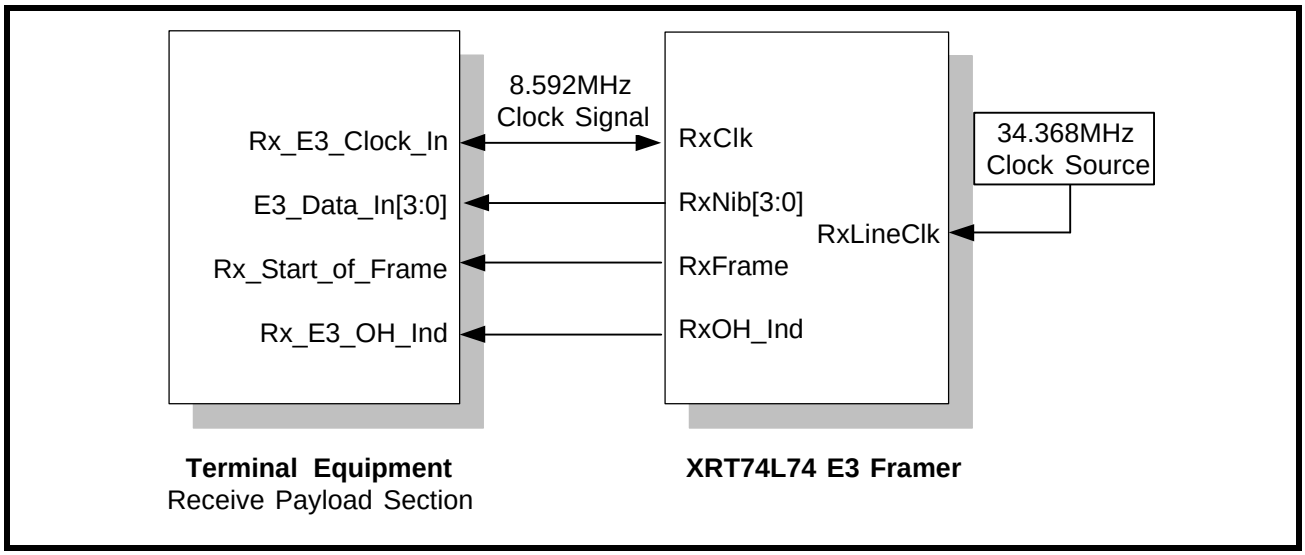
Delineation of Inbound DS3 Frames

The XRT74L74 will pulse the RxFrame output pin "High" for one nibble-period coincident with it driving the very first nibble, within a given Inbound E3 frame, via the RxNib[3:0] output pins.

Interfacing the XRT74L74 the Terminal Equipment.

Figure 218 presents a simple illustration as how the user should interface the XRT74L74 to that terminal equipment which processes Receive Direction payload data.

FIGURE 218. THE XRT74L74 DS3/E3 FRAMER IC BEING INTERFACED TO THE RECEIVE SECTION OF THE TERMINAL EQUIPMENT (NIBBLE-MODE OPERATION)



Required Operation of the Terminal Equipment

The XRT74L74 will update the data on the RxNib[3:0] line, upon the rising edge of RxClk. Hence, the Terminal Equipment should sample the data on the RxNib[3:0] output pins (or the E3_Data_In[3:0] input pins at the Terminal Equipment) upon the rising edge of RxClk. As the Terminal Equipment samples RxSer with each rising edge of RxClk it should also be sampling the RxFrame signal.

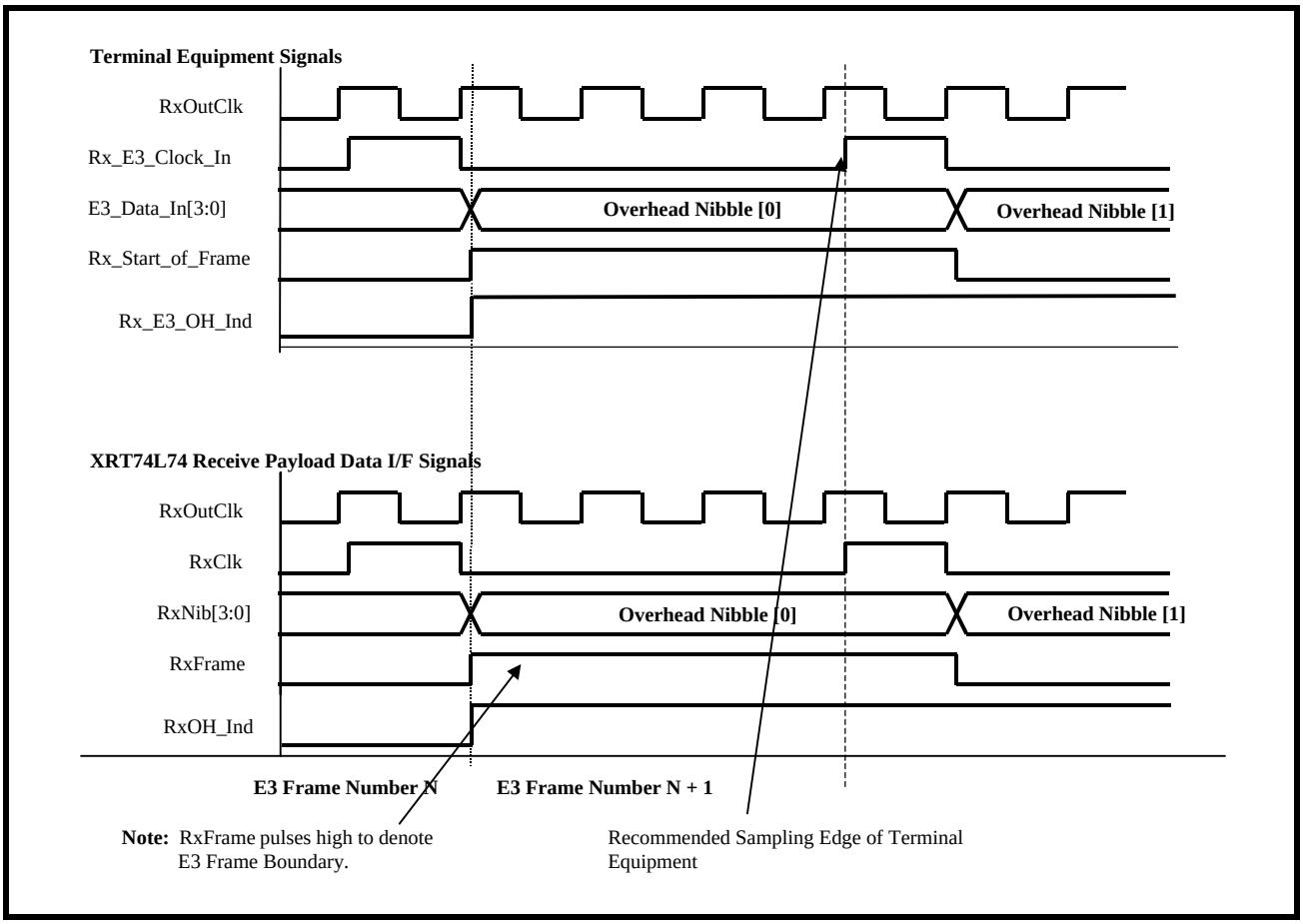
The Need for Sampling RxFrame

The XRT74L74 will pulse the RxFrame output pin "High" coincident with it driving the very first nibble of a given E3 frame, onto the RxNib[3:0] output pins. If knowledge of the E3 Frame Boundaries is important for the operation of the Terminal Equipment, then this is a very important signal for it to sample.

The Behavior of the Signals between the Receive Payload Data Output Interface block and the Terminal Equipment

The behavior of the signals between the XRT74L74 and the Terminal Equipment for E3 Nibble-Mode operation is illustrated in Figure 219 .

FIGURE 219. THE SIGNALS THAT ARE OUTPUT VIA THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK FOR METHOD 2.



7.3.6 Receive Section Interrupt Processing

The Receive Section of the XRT74L74 can generate an interrupt to the MIcrocontroller/Microprocessor for the following reasons.

- Change in Receive LOS Condition
- Change in Receive OOF Condition
- Change in Receive LOF Condition
- Change in Receive AIS Condition
- Change in Receive FERF Condition
- Change of Framing Alignment
- Change in Receive Trail Trace Buffer Message
- Detection of FEBE (Far-End Block Error) Event
- Detection of BIP-8 Error
- Detection of Framing Byte Error
- Detection of Payload Type Mismatch
- Reception of a new LAPD Message

7.3.6.1 Enabling Receive Section Interrupts

As mentioned in Section 1.6, the Interrupt Structure within the XRT74L74 contains two hierarchical levels.

- Block Level
- Source Level

The Block Level

The Enable state of the Block level for the Receive Section Interrupts dictates whether or not interrupts (if enabled at the source level), are actually enabled.

The user can enable or disable these Receive Section interrupts, at the Block Level by writing the appropriate data into Bit 7 (Rx DS3/E3 Interrupt Enable) within the Block Interrupt Enable register (Address = 0x04), as illustrated below.

BLOCK INTERRUPT ENABLE REGISTER (ADDRESS = 0X04)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxDS3/E3 Interrupt Enable	Not Used					TxDS3/E3 Interrupt Enable	One-Second Interrupt Enable
R/W	RO	RO	RO	RO	RO	R/W	R/W
X	0	0	0	0	0	0	0

Setting this bit-field to "1" enables the Receive Section at the Block Level) for interrupt generation. Conversely, setting this bit-field to "0" disables the Receive Section for interrupt generation.

7.3.6.2 Enabling/Disabling and Servicing Interrupts

As mentioned earlier, the Receive Section of the XRT74L74 Framer IC contains numerous interrupts. The Enabling/Disabling and Servicing of each of these interrupts is described below.

7.3.6.2.1 The Change in Receive LOS Condition Interrupt

If the Change in Receive LOS Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an LOS (Loss of Signal) Condition, and
2. When the XRT74L74 Framer IC clears the LOS condition.

Conditions causing the XRT74L74 Framer IC to declare an LOS Condition.

- If the XRT7300 LIU IC declares an LOS condition, and drives the RLOS input pin (of the XRT74L74 Framer IC) "High".
- If the XRT74L74 Framer IC detects 32 consecutive "0", via the RxPOS and RxNEG input pins.

Conditions causing the XRT74L74 Framer IC to clear the LOS Condition.

- If the XRT7300 LIU IC clears the LOS condition and drives the RLOS input pin (of the XRT74L74 Framer IC) "Low".
- If the XRT74L74 Framer IC detects a string of 32 consecutive bits (via the RxPOS and RxNEG input pins) that does NOT contain a string of 4 consecutive "0's".

Enabling and Disabling the Change in Receive LOS Condition Interrupt

The user can enable or disable the Change in Receive LOS Condition Interrupt, by writing the appropriate value into Bit 1 (LOS Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Change in Receive LOS Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "Low".
- It will set Bit 1 (LOS Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to "1", as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Whenever the user's system encounters the Change in Receive LOS Condition Interrupt, then it should do the following.

1. It should determine the current state of the LOS condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the LOS defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 4 (RxLOS) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

If the LOS state is TRUE

1. It should transmit a FERF (Far-End-Receive Failure) indicator to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-LOS feature.

If the LOS state is FALSE

1. It should cease transmitting the FERF indication to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-LOS feature.

7.3.6.2.2 The Change in Receive OOF Condition Interrupt

If the Change in Receive OOF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an OOF (Out of Frame) Condition, and
2. When the XRT74L74 Framer IC clears the OOF condition.

Conditions causing the XRT74L74 Framer IC to declare an OOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) detects Framing Byte errors, within four consecutive incoming E3 frames.

Conditions causing the XRT74L74 Framer IC to clear the OOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) transitions from the FA1, FA2 Octet Verification state to the In-Frame state (see Figure 175).
- If the Receive E3 Framer block transitions from the OOF Condition state to the In-Frame state (see Figure 175).

Enabling and Disabling the Change in Receive OOF Condition Interrupt

The user can enable or disable the Change in Receive OOF Condition Interrupt, by writing the appropriate value into Bit 3 (OOF Interrupt Enable), within

the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive OOF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 3 (OOF Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to “1”, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Whenever the user’s system encounters the Change in Receive OOF Condition Interrupt, then it should do the following.

1. It should determine the current state of the OOF condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the OOF defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 5 (RxOOF) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

If the OOF state is TRUE

1. It should transmit a FERF (Far-End-Receive Failure) indicator to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-OOF feature.

If the OOF state is FALSE

1. It should cease transmitting the FERF indication to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-OOF feature.

7.3.6.2.3 The Change in Receive LOF Condition Interrupt

If the Change in Receive LOF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an LOF (Out of Frame) Condition, and
2. When the XRT74L74 Framer IC clears the LOF condition.

Conditions causing the XRT74L74 Framer IC to declare an LOF Condition.

- If the Receive E3 Framer block (within the XRT74L74 Framer IC) detects Framing Byte errors, within four consecutive incoming E3 frames and is not able to transition back into the In-Frame state within 1 or 3ms.

Conditions causing the XRT74L74 Framer IC to clear the LOF Condition.

- If the Receive E3 Framer block transitions from the OOF Condition state to the LOF Condition state (see Figure 175).
- If the Receive E3 Framer block transitions back into the In-Frame state.

Enabling and Disabling the Change in Receive LOF Condition Interrupt

The user can enable or disable the Change in Receive LOF Condition Interrupt, by writing the appropriate value into Bit 3 (LOF Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive LOF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 6 (LOF Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to “1”, as indicated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

7.3.6.2.4 The Change of Framing Alignment (COFA) Interrupt

If the Change of Framing Alignment Interrupt is enabled then the XRT74L74 Framer IC will generate an interrupt any time the Receive E3 Framer block detects an abrupt change of framing alignment.

NOTE: *This interrupt is typically accompanied with the Change in Receive OOF Condition interrupt as well.*

Conditions causing the XRT74L74 Framer IC to generate this interrupt.

If the XRT74L74 Framer detects receives at least four consecutive E3 frames, within its Framing Alignment bytes in Error, then the XRT74L74 Framer IC will declare an OOF condition. However, while the XRT74L74 Framer IC is operating in the OOF condition, it will still rely on the old framing alignment for E3 payload data extraction, etc.

However, if the Receive E3 Framer had to change alignment, in order to re-acquire frame synchronization, then this interrupt will occur.

Enabling and Disabling the Change of Framing Alignment Interrupt

The user can enable or disable the Change of Framing Alignment Interrupt by writing the appropriate value into Bit 4 (COFA Interrupt

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	X	0	0	0	0

Writing a "1" into this bit-field enables the Change of Framing Alignment Interrupt. Conversely, writing a "0" into this bit-field disables the Change of Framing Alignment Interrupt.

Servicing the Change of Framing Alignment Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it "Low".
- It will set Bit 4 (COFA Interrupt Status), within the Rx E3 Interrupt Status Register -2, to "1", as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	0	0	0	0

7.3.6.2.5 The Change in Receive AIS Condition Interrupt

If the Change in Receive AIS Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares an AIS (Loss of Signal) Condition, and
2. When the XRT74L74 Framer IC clears the AIS condition.

Conditions causing the XRT74L74 Framer IC to declare an AIS Condition.

- If the XRT74L74 Framer IC detects 7 or less "0" within 2 consecutive E3 frames.

Conditions causing the XRT74L74 Framer IC to clear the AIS Condition.

- If the XRT74L74 Framer IC detects 2 consecutive E3 frames that each contain 8 or more "0's".

Enabling and Disabling the Change in Receive AIS Condition Interrupt

The user can enable or disable the Change in Receive LOS Condition Interrupt, by writing the appropriate value into Bit 0 (AIS Interrupt Enable), within the RxE3 Interrupt Enable Register - 1, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 1 (ADDRESS = 0X12)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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
0	0	0	0	0	0	X	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Change in Receive AIS Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "Low".
- It will set Bit 0 (AIS Interrupt Status), within the Rx E3 Interrupt Status Register - 1 to "1", as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 1 (ADDRESS = 0X14)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			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	1	0

Whenever the user's system encounters the Change in Receive AIS Condition Interrupt, then it should do the following.

1. It should determine the current state of the AIS condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the AIS defect. Hence, the user can determine the current state of the AIS defect by reading the state of Bit 4 (RxAIS) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

If the AIS Condition is TRUE

1. It should begin transmitting the FERF indication to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-AIS feature.

If the AIS Condition is FALSE

2. It should cease transmitting the FERF indication to the Remote Terminal Equipment. The XRT74L74 Framer IC automatically supports this action via the FERF-upon-AIS feature.

7.3.6.2.6 The Change in Trail Trace Buffer Message Interrupt

If the Change in Trail Trace Buffer Message Interrupt has been enabled, then the XRT74L74 Framer IC will generate an interrupt any time the Receive E3 Framer block receives a different Trail Trace Buffer message, then it has previously read in.

Enabling and Disabling the Change in Trail Trace Buffer Message Interrupt.

The user can enable or disable the Change in Trail Trace Buffer Message interrupt by writing the appropriate value into Bit 6 (TTB Change Interrupt Enable) within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	0	0	0

Writing a “1” into this bit-field enables the Change in Trail Trace Buffer Message Interrupt. Conversely, writing a “0” into this bit-field disables the Change in Trail Trace Buffer Message Interrupt.

Servicing the Change in Trail Trace Buffer Message Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it “Low”.
- It will set Bit 6 (TTB Change Interrupt Status), within the Rx E3 Interrupt Status Register - 2, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

- It will write the contents of this newly received Trail Trace Buffer Message, into the RxTTB-0 (located at 0x1C) through RxTTB-15 (located at 0x2B) registers.

Whenever the Terminal Equipment encounters the Change in Trail Trace Buffer Message Interrupt, then it should read out the contents of the 16 RxTTB registers.

7.3.6.2.7 The Change in Receive FERF Condition Interrupt

If the Change in Receive FERF Condition Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt in response to either of the following conditions.

1. When the XRT74L74 Framer IC declares a FERF (Far-End Receive Failure) Condition, and
2. When the XRT74L74 Framer IC clears the FERF condition.

Conditions causing the XRT74L74 Framer IC to declare an FERF Condition.

- If the XRT74L74 Framer IC begins receiving E3 frames which have the FERF bit (within the MA byte, set to “1”).

Conditions causing the XRT74L74 Framer IC to clear the AIS Condition.

- If the XRT74L74 Framer IC begins receiving E3 frames that do NOT have the FERF bit set to “1”.

Enabling and Disabling the Change in Receive AIS Condition Interrupt

The user can enable or disable the Change in Receive FERF Condition Interrupt, by writing the appropriate value into Bit 3 (FERF Interrupt Enable), within the RxE3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Change in Receive FERF Condition Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do all of the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.It will set Bit 3 (FERF Interrupt

Status), within the Rx E3 Interrupt Status Register - 2 to “1”, as indicated below

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Whenever the user’s system encounters the Change in Receive FERF Condition Interrupt, then it should do the following.

1. It should determine the current state of the FERF condition. Recall, that this interrupt can be generated, whenever the XRT74L74 Framer IC declares or clears the FERF defect. Hence, the user can determine the current state of the LOS defect by reading the state of Bit 0 (RxFERF) within the Rx E3 Configuration and Status Register - 2, as illustrated below.

RXE3 CONFIGURATION & STATUS REGISTER 2 (ADDRESS = 0X11)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Rx LOF Algo	RxLOF	RxOOF	RxLOS	RxAIS	RxPld Unstab	Rx TMark	RxFERF
R/W	RO	RO	RO	RO	RO	RO	RO
X	X	X	X	X	X	X	X

7.3.6.2.8 The Detection of FEBE (Far-End-Block Error) Event Interrupt

If the Detection of FEBE Event Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block has received an E3 frame with the FEBE bit-field (within the MA byte) set to “1”.

Enabling and Disabling the Detection of FEBE Event Interrupt

The user can enable or disable the Detection of FEBE Event’ interrupt by writing the appropriate value into Bit 4 (FEBE Interrupt Enable) within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	0	0	0

Setting this bit-field to “1” enables this interrupt. Conversely, setting this bit-field to “0” disables this interrupt.

Servicing the Detection of the FEBE Event Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “High”.It will set the Bit 4 (FEBE Interrupt Status), within the RxE3 Interrupt Status Register - 2 as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

Whenever the Terminal Equipment encounters the Detection of FEBE Event Interrupt, it should do the following.

- It should read the contents of the PMON FEBE Event Count Registers (located at Addresses 0x56 and 0x57) in order to determine the number of FEBE Events that have been received by the XRT74L74 Framer IC.

7.3.6.2.9 The Detection of BIP-8 Error Interrupt

If the Detection of BIP-8 Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block has detected an error in the EM (Error Monitoring) byte, within an incoming E3 frame.

Enabling and Disabling the Detection of FEBE Event Interrupt

The user can enable or disable the Detection of BIP-8 Error' interrupt by writing the appropriate value into Bit 2 (BIP-8 Interrupt Enable) within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	0	0	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Detection of the BIP-8 Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "High".
- It will set the Bit 2 (BIP-8 Interrupt Status), within the RxE3 Interrupt Status Register - 2 as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

Whenever the Terminal Equipment encounters the Detection of BIP-8 Error Interrupt, it should do the following.

- It should read the contents of the PMON Parity Error Event Count Registers (located at Addresses 0x54 and 0x55) in order to determine the number of BIP-8 Errors that have been received by the XRT74L74 Framer IC.

7.3.6.2.10 The Detection of Framing Byte Error Interrupt

If the Detection of Framing Byte Error Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block has received an E3 frame with an incorrect Framing Byte (e.g., FA1 or FA2) value.

Enabling and Disabling the Detection of FEBE Event Interrupt

The user can enable or disable the Detection of Framing Byte Error' interrupt by writing the appropriate value into Bit 1 (Framing Byte Error Interrupt Enable) within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	0	0	0

Setting this bit-field to "1" enables this interrupt. Conversely, setting this bit-field to "0" disables this interrupt.

Servicing the Detection of Framing Byte Error Interrupt

Whenever the XRT74L74 Framer IC detects this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it "High".
- It will set the Bit 4 (Framing Byte Error Interrupt Status), within the RxE3 Interrupt Status Register - 2 as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

Whenever the Terminal Equipment encounters the Detection of Framing Byte Error Interrupt, it should do the following.

- It should read the contents of the PMON Framing Bit/Byte Error Count Registers (located at Addresses 0x52 and 0x53) in order to determine the number of Framing Byte errors that have been received by the XRT74L74 Framer IC.

7.3.6.2.11 The Detection of Payload Type Mismatch Interrupt

If the Detection of Payload Type Mismatch Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt, anytime the Receive E3 Framer block receives a MA byte (within an incoming E3 frame) that contents a Payload Type value that is different from the expected Payload Type value.

Conditions causing this interrupt to be generated.

During system configuration, the user is expected to specify the Payload Type value that is expected of the Receive E3 Framer to receive (within each E3 frame), by writing this value into the RxPLDExp[2:0] bit-fields within the Rx E3 Configuration & Status Register - 1, as indicated below.

As long as the Receive E3 Framer block receives E3 frames that contains this Payload Type value, no interrupt will be generated. However, the instant that it receives an E3 frame, that contains a different Pay-

load Type value, then the XRT74L74 Framer IC will generate this interrupt.

RXE3 CONFIGURATION & STATUS REGISTER 1 (ADDRESS = 0X10)

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	RO	RO	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Enabling and Disabling the Detection of Payload Type Mismatch Interrupt.

The user can enable or disable the Detection of Payload Type Mismatch Interrupt by writing the appropriate data into Bit 0 (RxPld Mis Interrupt Enable), within the Rx E3 Interrupt Enable Register - 2, as indicated below.

RXE3 INTERRUPT ENABLE REGISTER - 2 (ADDRESS = 0X13)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Enable	Not Used	FEBE Interrupt Enable	FERF Interrupt Enable	BIP-8 Error Interrupt Enable	Framing Byte Error Interrupt Enable	RxPld Mis Interrupt Enable
RO	R/W	RO	R/W	R/W	R/W	R/W	R/W
0	0	0	X	X	0	0	X

Setting this bit-field to “1 enables the Detection of Payload Type Mismatch Interrupt. Conversely, setting this bit-field to “0” disables the Detection of Payload Type Mismatch Interrupt.

Servicing the Detection of Payload Type Mismatch Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$) by driving it “Low”.
- It will set Bit 0 (RxPld Mis Interrupt Status), within the Rx E3 Interrupt Enable Register -2 to “1”, as indicated below.

RXE3 INTERRUPT STATUS REGISTER - 2 (ADDRESS = 0X15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TTB Change Interrupt Status	Not Used	FEBE Interrupt Status	FERF Interrupt Status	BIP-8 Error Interrupt Status	Framing Byte Error Interrupt Status	RxPld Mis Interrupt Status
RO	RUR	RO	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

7.3.6.2.12 The Receive LAPD Message Interrupt

If the Receive LAPD Message Interrupt is enabled, then the XRT74L74 Framer IC will generate an interrupt anytime the Receive HDLC Controller block has received a new LAPD Message frame from the Remote Terminal Equipment, and has stored the contents of this message into the Receive LAPD Message buffer.

Enabling/Disabling the Receive LAPD Message Interrupt

The user can enable or disable the Receive LAPD Message Interrupt by writing the appropriate data into Bit 1 (RxLAPD Interrupt Enable) within the Rx E3 LAPD Control Register, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				DL from NR	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	0	0	0	X

Writing a “1” into this bit-field enables the Receive LAPD Message Interrupt. Conversely, writing a “0” into this bit-field disables the Receive LAPD Message Interrupt.

Servicing the Receive LAPD Message Interrupt

Whenever the XRT74L74 Framer IC generates this interrupt, it will do the following.

- It will assert the Interrupt Request output pin ($\overline{INT}$), by driving it “Low”.
- It will set Bit 0 (RxLAPD Interrupt Status), within the Rx E3 LAPD Control register to “1”, as indicated below.

RXE3 LAPD CONTROL REGISTER (ADDRESS = 0X18)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used				DL from NR	RxLAPD Enable	RxLAPD Interrupt Enable	RxLAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	0	0	1	1

- It will write the contents of the newly Received LAPD Message into the Receive LAPD Message buffer (located at 0xDE through 0x135).

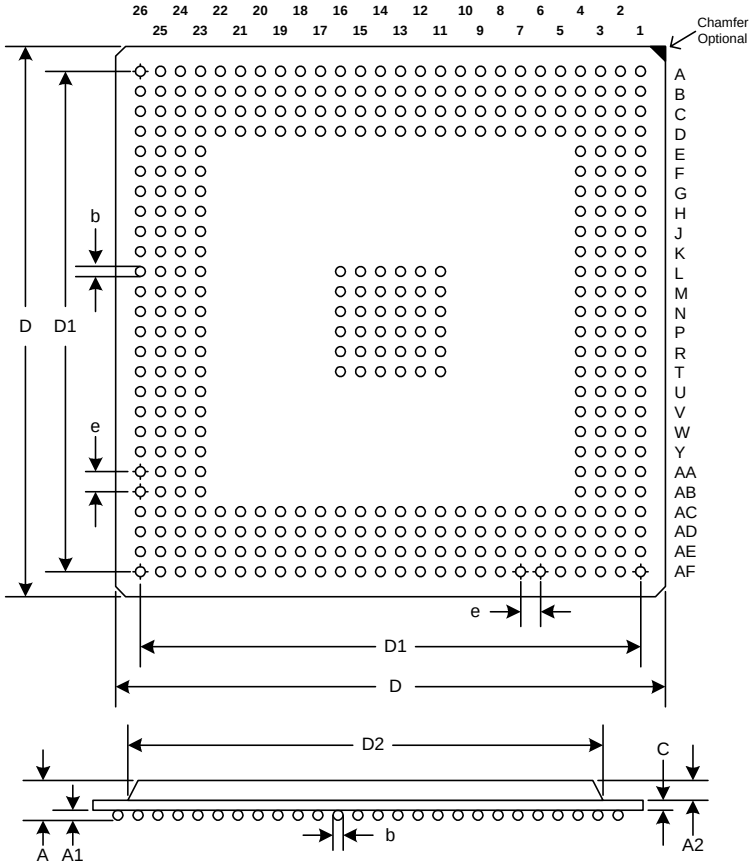
Whenever the Terminal Equipment encounters the Receive LAPD Message Interrupt, then it should read out the contents of the Receive LAPD Message buffer, and respond accordingly.

ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE
XRT74L74IB	35 x 35 mm, 388 Plastic Ball Grid Array	-40^{0}C to +85^{0}C

PACKAGE DIMENSIONS

388 Ball Plastic Ball Grid Array
(35 x 35 mm PBGA)
Rev. 1.0



Symbol	Inches MIN	Inches MAX	Millimeters MIN	Millimeters MAX
A	0.075	0.106	1.90	2.70
A1	0.020	0.028	0.50	0.70
A2	0.039	0.051	1.00	1.30
b	0.024	0.035	0.60	0.90
C	0.016	0.028	0.40	0.70
D	1.370	1.386	34.80	35.20
D1	1.250BSC		31.75BSC	
D2	1.177	1.185	29.90	30.10
e	0.050BSC		1.27BSC	

Note: The control dimension is the millimeter column

REVISION HISTORY

REVISION #	DATE	DESCRIPTION
P1.1.0		First release of the 4-Channel ATM UNI/PPP DS3/E3 Framing Controller Datasheet.
P1.1.1	10/03	Added the register map.