GENERAL DESCRIPTION

The XRT79L71 is a single channel, integrated DS3/E3 framing controller and Line Interface Unit with Jitter Attenuator that is designed to support Frame processing. For Clear-Channel Framer applications, this device supports the transmission and reception of "user data" via the DS3/E3 payload.

The XRT79L71 includes DS3/E3 Framing, Line Interface Unit with Jitter Attenuator that supports mapping of HDLC framed data. A flexible parallel microprocessor interface is provided for configuration and control.

GENERAL FEATURES:

- Integrated T3/E3 Line Interface Unit
- Integrated Jitter Attenuator that can be selected either in Receive or Transmit path
- HDLC Controller that provides the mapping/extraction of either bit or byte mapped encapsulated packet from DS3/E3 Frame.
- 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 M13 and C-Bit Parity Framing Formats
- Supports DS3/E3 Clear-Channel Framing.
- Includes PRBS Generator and Receiver
- Supports Line, Cell, and PLCP Loop-backs
- Interfaces to 8 Bit wide Intel, Motorola, PowerPC, and Mips µPs
- Low power 3.3V, 5V Input Tolerant, CMOS
- Available in 208 STBGA Package
- JTAG Interface

LINE INTERFACE UNIT

- On chip Clock and Data Recovery circuit for high input jitter tolerance
- Meets E3/DS3 Jitter Tolerance Requirements
- Detects and Clears LOS as per G.775.
- Receiver Monitor mode handles up to 20 dB flat loss with 6 dB cable attenuation
- Compliant with jitter transfer template outlined in ITU G.751, G.752, G.755 and GR-499-CORE,1995 standards
- Meets ETSI TBR 24 and GR-499 Jitter Transfer Requirements
- On chip B3ZS/HDB3 encoder and decoder that can be either enabled or disabled
- On-chip clock synthesizer provides the appropriate rate clock from a single 12.288 MHz Clock
- On chip advanced crystal-less Jitter Attenuator
- Jitter Attenuator can be selected in Receive or Transmit paths
- 16 or 32 bits selectable FIFO size
- Meets the Jitter and Wander specifications described in T1.105.03b,ETSI TBR-24, Bellcore GR-253 and GR-499 standards
- Jitter Attenuator can be disabled
- Typical power consumption 1.3W

DS3/E3 FRAMER

- DS3 framer supports both M13 and C-bit parity.
- DS3 framer meets ANSI T1.107 and T1.404 standards.
- Detects OOF,LOF,AIS,RDI/FERF alarms.
- Generation and Insertion of FEBE on received parity errors supported.
- Automatic insertion of RDI/FERF on alarm status.
- E3 framer meets G.832,G.751 standards.
- Framers can be bypassed.
- Maps HDLC data stream directly into DS3 or E3 frame
- Extracts in-band messaging packets
- Supports CRC-16/32, HDLC flag and Idle sequence generation

RECEIVE PACKET PROCESSING

- Extracts HDLC data stream from DS3 or E3 frame
- Inserts in-band messaging packets
- Detects and removes HDLC flags

SERIAL INTERFACE

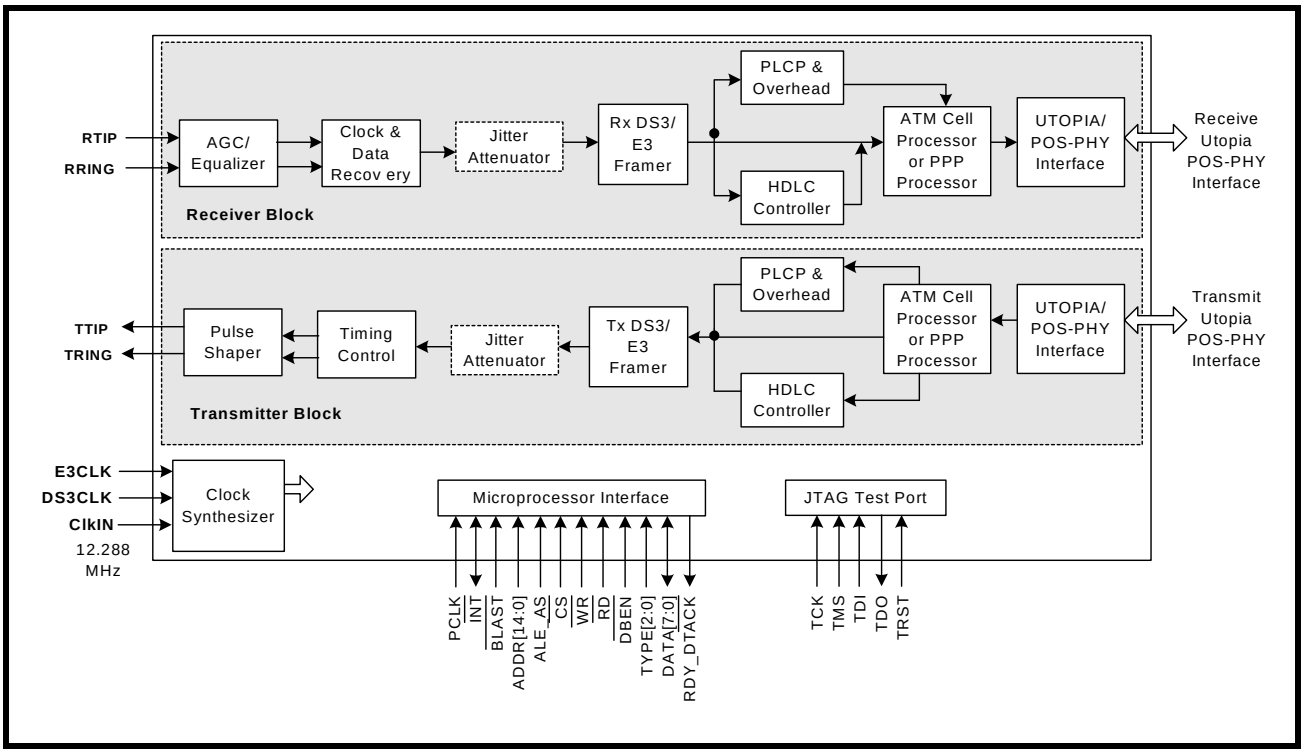
- Serial clock and data interface for accessing DS3/E3 framer

- Serial clock and data interface for accessing cell/packet processor

APPLICATIONS

- Digital Access and Cross Connect Systems
- 3G Base Stations
- DSLAMs

FIGURE 1. BLOCK DIAGRAM OF THE XRT79L71



PRODUCT ORDERING INFORMATION

PRODUCT NUMBER	PACKAGE TYPE	OPERATING TEMPERATURE RANGE
XRT79L71IB	17X17 mm 208 Ball Shrink Thin Ball Grid Array	-40°C to +85°C

TABLE 1: PIN OUT OF THE XRT79L71 (TOP VIEW)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T	TXUADDR_1	TXUADDR_0	TXUDATA_0	TXUDAT_3	TXUDATA_7	TXUDATA_12	GPIO_0	DMO_0	TCK	TRING	TTIP	TXAVDD	RXAVDD	RXAGND	JAGND	JAAVDD
R	TXUADDR_3	TXUADDR_2	TXUPRTY	TXUDATA_2	TXUDATA_6	TXUDATA_11	TXUDATA_15	GPIO_3	TMS	TRST	NC	REFAVDD	RRING	RTIP	TXON	CLKVDD
P	TXUCLKO	TXUADDR_4	TXUSOC	TXUDATA_1	TXUDATA_5	TXUDATA_9	TXUDATA_14	GPIO_2	TDI	MTIP	MRING	REFAGND	ANAIO1	ANAIO2	ICTB	DS3CLK
N	TXPEOP	RXPDVAL	TXUCLAV	TXUDATA_10	TXUDATA_4	TXUDATA_8	TXUDATA_13	GPIO_1	TDO	TXDGND	TXDVDD	TXAGND	OVDD	VDD	GND	CLKGND
M	TXUCLK	TXPER	TXMOD	TXUEN_L									OGND	RESET_L	TESTMODE	E3CLK
L	RXMOD	RXPERR	RXUCLK	TSX_TSOF									GPI_2	GPI_1	GPI_0	NIBBLEINTF
K	RXUADDR_4	RXUCLKO	RXPEOP	RSX_RSOF									GPO_2	GPO_1	GPO_0	CLKOUT
J	RXUADDR_0	RXUADDR_1	RXUADDR_2	RXUADDR_3									PDBEN_L	PTYPE_2	PTYPE_1	PTYPE_0
H	RXUCLAV	RXUSOC	RXUPRTY	RXUEN_L									DA_SEL	VDD	GND	PCLK
G	RXUDATA_1	RXUDATA_0	RXUDATA_3	RXUDATA_6									DPADDR_7	DPADDR_6	DPADDR_5	DPADDR_4
F	RXUDATA_2	RXUDATA_4	RXUDATA_7	RXUDATA_10									DPADDR_3	DPADDR_2	DPADDR_1	DPADDR_0
E	RXUDATA_5	RXUDATA_8	RXUDATA_11	RXUDATA_14									PADDR_6	PADDR_5	PADDR_4	PADDR_3
D	RXUDATA_9	RXUDATA_12	RXUDATA_15	RXCP	RXPOOF	RXNIB_1	RXLOS	TXNIB_1	TXNIBCLK	TXOHINS	PDATA	PDATA_4	PINT_L	PCS_L	PADDR_1	PADDR_2
C	RXUDATA_13	TXGFCCLK	RXGFC	RXPOHFRAME	RXNIB_0	RXOHIND	RXOH	TXNOB_2	TXSER	TXINCLK	TXOH	PDATA_1	PDATA_5	PRDY_L	PRD_L	PADDR_0
B	RXGFCMSB	RXPRED	TXPOHCLK	RXNIB_3	RXOUTCLK	RXFRAME	RXOHENABLE	TXNOB_3	TXOHIND	TXFRAME	TXOHFRAME	TXOHCLK	PDATA_2	PDATA_6	PBLAST_L	PWR_L
A	RXGFCCLK	RXPLOF	TXPOHFRAME	RXNIB_2	RXSER	RXCLK	RXOHCLK	RXOHFRAME	TXNIB_0	TXNIBFRAME	TXFRAMEREF	TXOHENABLE	TXAISEN	PDATA_3	PDATA_7	PAS_L

TABLE OF CONTENTS

1.0 BRIEF XRT79L71 ARCHITECTURE DESCRIPTION

The XRT79L71 can be configured to operate in any of the following modes.

- The Clear-Channel DS3/E3 Framer Mode
- The High-Speed HDLC Controller over DS3/E3 Mode
- The ATM UNI over DS3/E3 Mode
- The PPP over DS3/E3 Mode

The detailed functional description of the XRT79L71, that covers these four modes of operation, can be found in various documents as presented below in Table 2.

TABLE 2: LISTING OF ARCHITECTURAL/FUNCTIONAL DESCRIPTION DOCUMENTS FOR THE XRT79L71

MODE OF OPERATION	ARCHITECTURAL/FUNCTIONAL DESCRIPTION DOCUMENT
DS3/E3 Clear-Channel Framer Mode	**79L71_Arch_Descr_CC.pdf** Architectural/Functional Description of the XRT79L71, 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel with LIU, IC - Clear-Channel Framer Applications
DS3/E3 High-Speed HDLC Controller Mode	**79L71_Arch_Descr_HDLC.pdf -** Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel with LIU IC - High-Speed HDLC Controller Mode Applications
DS3/E3 ATM UNI Mode	**79L71_Arch_Descr_ATM.pdf** Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - ATM UNI Mode Applications
DS3/E3 PPP Mode	**79L71_Arch_Descr_PPP.pdf** Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - PPP Mode Applications

If the XRT79L71 is to be operated in a particular mode, obtain the appropriate document, which is presented in Table 2. However, a brief functional/architectural description of the XRT79L71, when it is configured to operate in any of these modes will be presented below.

1.1 BRIEF FUNCTIONAL ARCHITECTURE DESCRIPTION OF THE XRT79L71 - CLEAR-CHANNEL DS3/E3 FRAMER MODE

If the XRT79L71 has been configured to operate in the Clear-Channel DS3/E3 Framer Mode, then it will have the Functional Architecture as is presented below in Figure 2.

FIGURE 2. THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE CLEAR-CHANNEL DS3/E3 FRAMER MODE

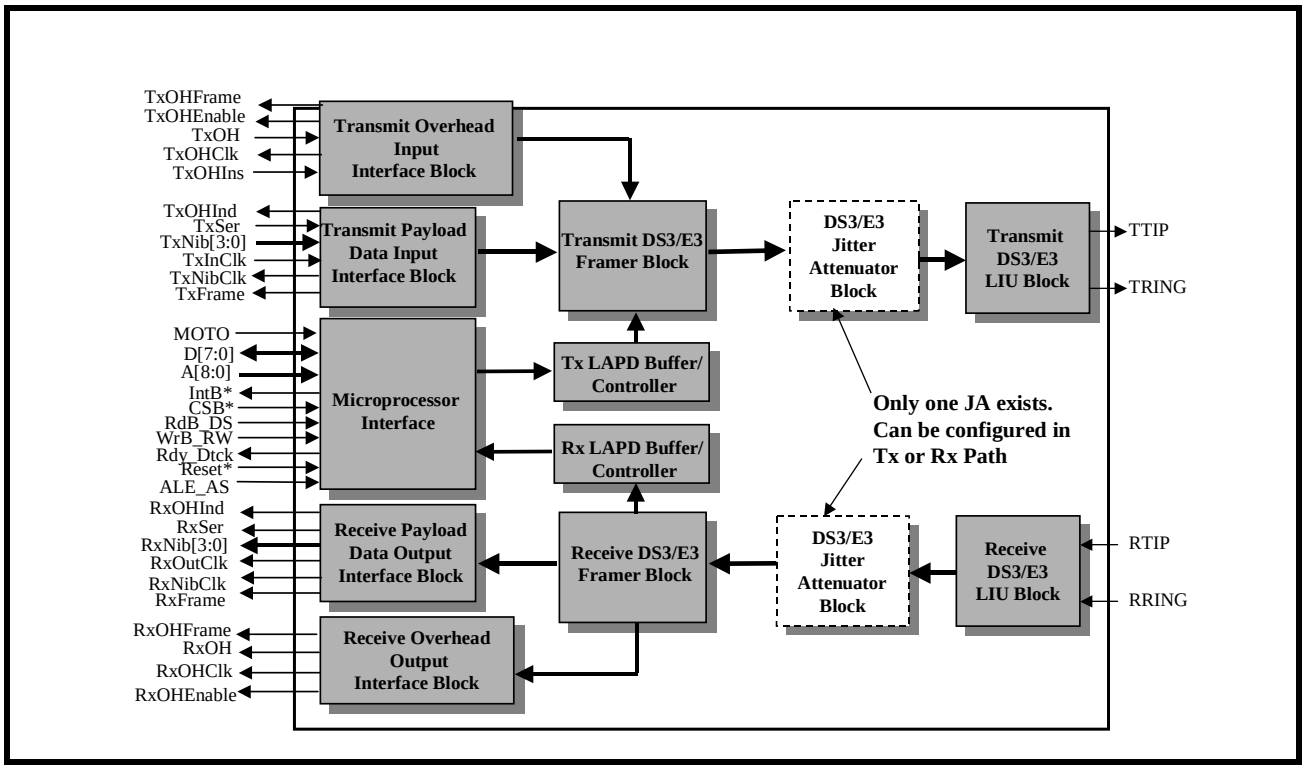


Figure 2 indicates that the XRT79L71 consists of the following functional blocks.

- The Transmit Payload Data Input Interface Block
- The Transmit Overhead Data Input Interface Block
- The Transmit LAPD Controller Block
- The Transmit FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)
- The Transmit SSM Controller Block (E3, ITU-T G.832 Applications Only)
- The Transmit DS3/E3 Framer Block
- The Transmit DS3/E3 LIU Block
- The Receive DS3/E3 LIU Block
- The Receive DS3/E3 Framer Block
- The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)
- The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)
- The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Receive LAPD Controller Block
- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block

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

1.1.1 The Transmit Payload Data Input Interface Block

The purpose of the Transmit Payload Data Input Interface block is to accept outbound payload data either via a Serial or Nibble-Parallel interface, and to route this data to the Transmit DS3/E3 Framer block (where this data will ultimately be mapped into the payload bit-positions within each outbound DS3/E3 frame).

1.1.2 The Transmit Overhead Data Input Interface Block

The purpose of the Transmit Overhead Data Input Interface block is to permit the user to externally insert the users own value for overhead bits into the outbound DS3/E3 data-stream.

NOTE: *This particular feature is very valuable in those applications in which the XRT79L71 is processing a Channelized DS3 signal that is of the M23-framing format (where it is imperative to preserve the contents of the C-bits within the DS3 data-stream).*

1.1.3 The Transmit LAPD Controller Block

The purpose of the Transmit LAPD Controller block is to permit the user to transmit LAPD/PMDL (Path Maintenance Data Link) Messages to the remote terminal equipment. The Transmit LAPD Controller block comes with a LAPD Controller/Transmitter and 90 bytes of on-chip RAM (for storage of outbound PMDL Messages), prior to transmission.

1.1.4 The Transmit FEAC Controller Block (for DS3, C-Bit Parity Applications only)

The purpose of the Transmit FEAC Controller block is to permit the user to transmit FEAC (Far-End Alarm & Control) Messages to the remote terminal equipment.

NOTE: *The Transmit FEAC Controller Block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.1.5 The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit Trail-Trace Message Controller block is to permit a given terminal equipment to repeatedly transmit a "Trail-Trace" Message to the remote terminal equipment, via the TR bytes within each outbound E3, ITU-T G.832 frame

1.1.6 The Transmit SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit SSM Controller block is to permit a given terminal equipment to repeatedly transmit the "Synchronization Status Message" to the remote terminal equipment, via the MA byte, within each outbound E3, ITU-T G.832 frame.

1.1.7 The Transmit DS3/E3 Framer Block

The purpose of the Transmit DS3/E3 Framer block is to accept data from the Transmit Payload Data Input Interface block, and Transmit Overhead Data Input Interface, the Transmit LAPD Controller and the Transmit FEAC Controller block and to construct a DS3/E3 data-stream for transmission to the remote terminal equipment. Additionally, the Transmit DS3/E3 Framer block can be configured to do all of the following.

- To transmit the AIS Indicator (upon Software Control)
- To automatically transmit the FERF/RDI Indicator (in response to the Receive DS3/E3 Framer block declare the LOS, LOF/OOF or AIS defect condition).
- To transmit the FERF/RDI indicator (upon Software Control)
- To automatically transmit the FEBE/REI Indicator (in response to the Receive DS3/E3 Framer block detecting Framing bit or CP-bit errors - DS3, C-bit Parity Applications).
- To automatically transmit the FEBE/REI Indicator (in response to the Receive DS3/E3 Framer block detecting BIP-8 Error - E3, ITU-T G.832 Applications).
- To transmit the FEBE/REI indicator (upon Software Control).

1.1.8 The Transmit DS3/E3 LIU Block

The purpose of the Transmit DS3/E3 LIU Block is to accept the outbound DS3/E3 data-stream from the Transmit DS3/E3 Framer block, and to perform all of the following operations on this signal.

- To encode into the B3ZS Line Code (for DS3 Applications) or the HDB3 Line Code (for E3 Applications)
- To convert this outbound DS3 or E3 data into a DS3/E3 line signal and transmit this signal to the remote terminal equipment.
- To generate and transmit DS3 pulses that complies with the Isolated Pulse Template requirements per Bellcore GR-499-CORE
- To generate and transmit E3 pulses that complies with the ITU-T G.703 Pulse Template requirements for E3 applications.

1.1.9 The Receive DS3/E3 LIU Block

The purpose of the Receive DS3/E3 LIU Block is to receive a DS3/E3 line signal from the remote terminal equipment, and to perform the following operations

- To decode this incoming signal from the B3ZS Line Code (for DS3 Applications) or the HDB3 Line Code (for E3 Applications) into a binary data-stream
- To route this binary data-stream to the Receive DS3/E3 Framer block for further processing
- To detect and flag the occurrence of LCVs (Line Code Violations) and EXZs (Excessive Zeros)
- To insure that the XRT79L71 meets all of the following Receive requirements.
  - **a.** The Receive Sensitivity requirements for DS3 Applications (e.g., able to receive a DSX-3 type of signal through at least 450 feet of cable loss)
  - **b.** The Receive Sensitivity requirements for E3 Applications (e.g., able to receive an E3 signal over 12dB of cable loss)
  - **c.** To comply with the Category I and II Jitter Tolerance Requirements per Bellcore GR-499-CORE (for DS3 Applications)
  - **d.** To comply with the Jitter Tolerance Requirements per ITU-T G.832 (for E3 Applications)
  - **e.** To comply with the Interference Margin Requirements of 20dB, per ITU-T G.703 (for E3 Applications)

1.1.10 The Receive DS3/E3 Framer Block

The purpose of the Receive DS3/E3 Framer block is to acquire and maintain Frame Synchronization with the incoming DS3/E3 data-stream that is receives from the Receive DS3/E3 LIU Block. As the Receive DS3/E3 Framer block performs this task, it will also do the following.

- It will declare and clear the LOS defect condition
- It will declare and clear the LOF/OOF defect condition
- It will declare and clear the AIS defect condition
- It will declare and clear the FERF/RDI defect condition
- It will detect and flag the occurrences of P-bit, CP-bit and Framing bit errors (DS3 Applications)
- It will detect and flag the occurrence of BIP-8 Errors (E3, ITU-T G.832 Applications)
- It will detect and flag the occurrence of BIP-4 Errors (E3, ITU-T G.751 Applications)
- It will detect and flag the occurrence of FEBE/REI Events
- It will route all PMDL data to the Receive LAPD Controller block for further processing
- It will route all Overhead bits/bytes to the Receive Overhead Data Output Interface block for further processing
- It will route all DS3/E3 data to the Receive Payload Data Output Interface block.

1.1.11 The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive SSM Controller Block is to permit a given terminal equipment to receive (and extract out) the SSM (Synchronization Status Message) from the remote terminal equipment, via the MA byte, within each inbound E3, ITU-T G.832 frame.

1.1.12 The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive Trail-Trace Message Controller block is to permit a given terminal equipment to receive (and extract out) the Trail-Trace Message from the remote terminal equipment, via the TR byte, within each inbound E3, ITU-T G.832 frame.

1.1.13 The Receive FEAC Controller Block (DS3 Applications Only)

The purpose of the Receive FEAC Controller block is to permit the user to receive FEAC (Far-End Alarm & Control) Messages from the remote terminal equipment.

NOTE: *The Receive FEAC Controller Block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.1.14 The Receive LAPD Controller Block

The purpose of the Receive LAPD Controller block is to permit the user to receive LAPD/PMDL (Path Maintenance Data Link) Messages from the remote terminal equipment. The Receive LAPD Controller block comes with a Receive LAPD Controller and 90 bytes of on-chip RAM (for storage of inbound PMDL Messages) after reception.

1.1.15 The Receive Payload Data Output Interface Block

The purpose of the Receive Payload Data Output Interface block is to output payload data (within the incoming DS3 or E3 data-stream) via either a Serial or Nibble-Parallel interface, and to route this data to the off-chip System-Side Terminal Equipment.

1.1.16 The Receive Overhead Data Output Interface Block

The purpose of the Receive Overhead Data Output Interface block is to permit the user to extract out the overhead bits (within the incoming DS3/E3 data-stream) and to route this data to some off-chip System-Side Terminal Equipment circuitry.

1.1.17 A more detailed Functional/Architectural Description of the XRT79L71 when configured to operate in the Clear Channel Controller Mode, is in this document (*79L71_Arch_Descr_CC.pdf*).

(Section 7.0- Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - Clear Channel Applications).

1.2 BRIEF FUNCTIONAL ARCHITECTURE DESCRIPTION OF THE XRT79L71 - HIGH-SPEED HDLC CONTROLLER OVER DS3/E3 MODE

If the XRT79L71 has been configured to operate in the High-Speed HDLC Controller over DS3/E3 Mode, then it will have the Functional Architecture as is presented below in **Figure 3**.

FIGURE 3. THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE HIGH-SPEED HDLC CONTROLLER OVER DS3/E3 MODE

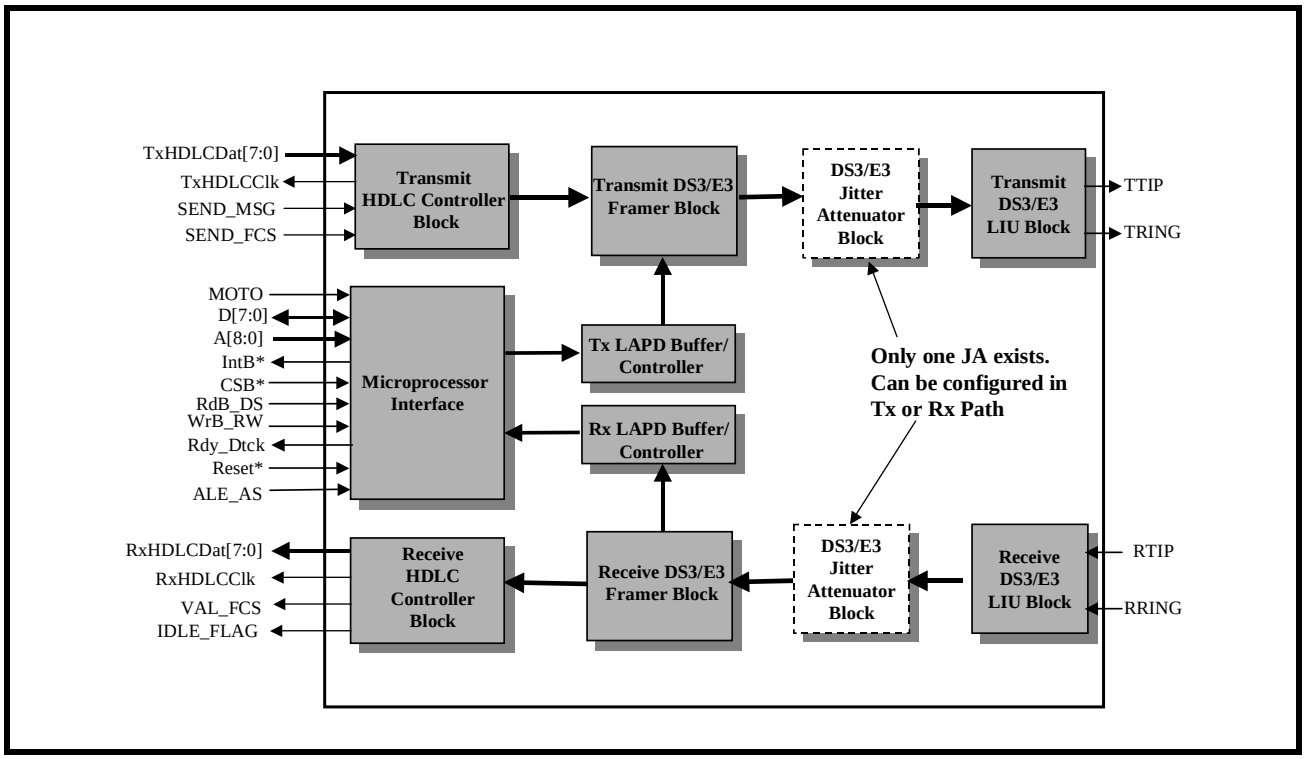


Figure 3 indicates that the XRT79L71 consists of the following functional blocks.

- The Transmit High-Speed HDLC Controller Block
- The Transmit LAPD Controller Block
- The Transmit FEAC Controller Block (DS3 Applications only)
- The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications only)
- The Transmit SSM Controller Block (E3, ITU-T G.832 Applications only)
- The Transmit DS3/E3 Framer Block
- The Transmit DS3/E3 LIU Block
- The Receive DS3/E3 LIU Block
- The Receive DS3/E3 Framer Block
- The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications only)
- The Receive SSM Controller Block (E3, ITU-T G.832 Applications only)
- The Receive FEAC Controller Block (DS3 Applications only)
- The Receive LAPD Controller Block
- The Receive High-Speed HDLC Controller Block

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

1.2.1 The Transmit High-Speed HDLC Controller Block

The purpose of the Transmit High-Speed HDLC Controller block is to perform the following tasks.

- To accept user payload data, in a byte-wide manner (via an 8-bit wide Input Data Bus)
- To (while accepting this user payload data, via this byte-wide interface, encapsulate this user data into an HDLC frame)
- To generate and transmit a repeating string of Flag Sequence octets (0x7E), whenever the Transmit High-Speed HDLC Controller block is not accepting nor processing user data
- To route these outbound HDLC frames and Flag Sequence octets to the Transmit DS3/E3 Framer block for further processing.

NOTES:

1. *Whenever the XRT79L71 has been configured to operate in the High-Speed HDLC Controller Mode, then the role of the Transmit Payload Data Input Interface block will be replaced by the Transmit High-Speed HDLC Controller block.*
2. *The Transmit High Speed HDLC Controller block should NOT be confused with the Transmit LAPD Controller block (which also exists within the XRT79L71).*

1.2.2 The Transmit Overhead Data Input Interface block (not shown in Figure 3)

If the XRT79L71 has been configured to operate in the High-Speed HDLC Controller Mode, then the Transmit Overhead Data Input Interface block will be disabled.

1.2.3 The Transmit LAPD Controller Block

The purpose of the Transmit LAPD Controller block is to permit the user to transmit LAPD/PMDL (Path Maintenance Data Link) Messages to the remote terminal equipment. The Transmit LAPD Controller block comes with a LAPD Controller/Transmitter (not to be confused with the Transmit High-Speed HDLC Controller) and 90 bytes of on-chip RAM (for storage of outbound PMDL Messages) prior to transmission.

1.2.4 The Transmit FEAC Controller Block (DS3 Applications only)

The purpose of the Transmit FEAC Controller block is to permit the user to transmit FEAC (Far-End Alarm & Control) Messages to the remote terminal equipment.

NOTE: *The Transmit FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.2.5 The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications only)

The purpose of the Transmit Trail-Trace Message Controller block is to permit a given terminal equipment to repeatedly transmit a "Trail-Trace" Message to the remote terminal equipment, via the TR bytes, within each outbound E3, ITU-T G.832 frame.

1.2.6 The Transmit SSM Controller Block (E3, ITU-T G.832 Applications only)

The purpose of the Transmit SSM Controller Block is to permit a given terminal equipment to repeatedly transmit the "Synchronization Status Message" to the remote terminal equipment, via the MA byte, within each outbound E3, ITU-T G.832 frame.

1.2.7 The Transmit DS3/E3 Framer Block

The purpose of the Transmit DS3/E3 Framer block is to accept payload data from the Transmit High-Speed HDLC Controller block, the Transmit LAPD Controller block and the Transmit FEAC Controller block and to construct a DS3/E3 data-stream for transmission to the remote terminal equipment. More specifically, in this particular application, the Transmit DS3/E3 Framer block is responsible for accepting the outbound HDLC frames and Flag Sequences octets from the Transmit High-Speed HDLC Controller block, and inserting this data into the payload bits/bytes within the outbound DS3/E3 fames. Additionally, the Transmit DS3/E3 Framer block can be configured to do all of the following.

- To transmit the AIS Indicator (upon Software Control)
- To automatically transmit the FERF/RDI Indicator (in response to the Receive DS3/E3 Framer block declaring the LOS, LOF/OOF or AIS defect condition)

- To transmit the FERF/RDI indicator (upon Software Control)
- To automatically transmit the FEBE/REI Indicator (in response to the Receive DS3/E3 Framer block detecting Frmaing bit or CP-bit errors - DS3, C-bit Parity Applications)
- To automatically transmit the FEBE/REI Indicator (response to the Receive DS3/E3 Framer block detecting BIP-8 Errors - E3, ITU-T G.832 Applications)
- To transmit the FEBE/REI Indicator (upon Software Control).

1.2.8 The Transmit DS3/E3 LIU Block

The purpose of the Transmit DS3/E3 LIU Block is to accept the outbound DS3/E3 data-stream from the Transmit DS3/E3 Framer block, and to perform all of the following operations on the signal.

- To encode into the B3ZS Line Code (for DS3 Applications) or the HDB3 Line Code (for E3 Applications)
- To convert this outbound DS3 or E3 data-stream into a DS3/E3 line signal and transmit this signal to the remote terminal equipment
- To generate and transmit DS3 pulses that complies with the Isolated Pulse Template requirements per Bellcore GR-499-CORE
- To generate and transmit E3 pulse that complies with the ITU-T G.703 Pulse Template requirements for E3 applications.

1.2.9 The Receive DS3/E3 LIU Block

The purpose of the Receive DS3/E3 LIU Block is to receive a DS3/E3 line signal from the remote terminal equipment, and to perform the following operations on this incoming line signal.

- To decode this incoming signal from the B3ZS Line Code (for DS3 Applications) or from the HDB3 Line Code (for E3 Applications) into a binary data-stream
- To route this binary data-stream to the Receive DS3/E3 Framer block for further processing.
- To insure that the XRT79L71 meets all of the following Receive requirements.
  - **a.** The Receive Sensitivity requirement for DS3 Applications (e.g., able to receive a DSX-3 type of signal through at least 450 feet of cable loss).
  - **b.** The Receive Sensitivity requirement for E3 Applications (e.g., able to receive an E3 signal over 12dB of cable loss).
  - **c.** To comply with the Category I and II Jitter Tolerance Requirements per Bellcore GR-499-CORE (for DS3 Applications)
  - **d.** To comply with the Jitter Tolerance Requirements per ITU-T G.823 (for E3 Applications)
  - **e.** To comply with the Interference Margin Requirements of 20dB, per ITU-T G.703 (for E3 Applications)

1.2.10 The Receive DS3/E3 Framer Block

The purpose of the Receive DS3/E3 Framer block is to acquire and maintain Frame Synchronization with the incoming DS3/E3 data-stream that it receives from the Receive DS3/E3 LIU Block. As the Receive DS3/E3 Framer block performs this task, it will also do the following.

- It will declare and clear the LOS defect condition
- It will declare and clear the LOF/OOF defect condition
- It will declare and clear the AIS defect condition
- It will declare and clear the FERF/RDI defect condition
- It will detect and flag the occurrences of P-bit, CP-bit and Framing bit errors (DS3 Applications)
- It will detect and flag the occurrence of BIP-8 Errors (E3, ITU-T G.832 Applications)
- It will detect and flag the occurrence of BIP-4 Errors (E3, ITU-T G.751 Applications)

- It will detect and flag the occurrence of FEBE/REI Events
- It will route all PMDL data to the Receive LAPD Controller block for further processing
- It will route all DS3/E3 payload to the Receive High-Speed HDLC Controller block.

1.2.11 The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications only)

The purpose of the Receive Trail-Trace Message Controller Block is to permit a given terminal equipment to receive (and extract out) the Trail-Trace Message (also known as "Trail-Access Point Identifier) from the incoming E3 data-stream (which is being sourced by the remote terminal equipment, via the TR byte, within each inbound E3, ITU-T G.832 frame. The Receive Trail-Trace Message Controller block will also alert the Microprocessor (by generating an interrupt) anytime it detects a change in the incoming Trail-Trace Message.

1.2.12 The Receive SSM Controller Block (E3, ITU-T G.832 Applications only)

The purpose of the Receive SSM Controller block is to permit a given terminal equipment to receive (and extract out) the SSM (Synchronization Status Message) from the remote terminal equipment, via the MA byte, within each inbound E3, ITU-T G.832 frame.

1.2.13 The Receive FEAC Controller Block (DS3 Applications only)

The purpose of the Receive FEAC Controller block is to permit the user to receive FEAC (Far-End Alarm & Control) Messages from the remote terminal equipment.

NOTE: *The Receive FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.2.14 The Receive LAPD Controller Block

The purpose of the Receive LAPD Controller block is to permit the user to receive LAPD/PMDL (Path Maintenance Data Link) Messages from the remote terminal equipment. The Receive LAPD Controller block comes with a Receive LAPD Controller (not to be confused with the Receive High-Speed HDLC Controller block) and 90 bytes of on-chip RAM (for storage of inbound PMDL Messages) after reception.

NOTE: *The role of the Receive LAPD Controller block should not be confused with that of the Receive High-Speed HDLC Controller block.*

1.2.15 The Receive High-Speed HDLC Controller Block

The purpose of the Receive High-Speed HDLC Controller block is to receive the payload data (from the incoming DS3/E3 data-stream) and perform the following tasks.

- To identify the boundaries of incoming HDLC frames (within the incoming DS3/E3 data-stream)
- To terminate the Flag Sequence octets within the incoming data-stream
- To (optionally) compute and verify the CRC-16 or CRC-32 values that have been appended to the back-end of these incoming HDLC frames (at the remote terminal equipment) and to flag any occurrences of CRC errors
- To zero-un-stuff the contents within these incoming HDLC frames
- To output this HDLC frame data (in a byte-wide manner) via an 8-bit wide Output Data Bus.

1.2.16 The Receive Overhead Data Output Interface Block (not shown in Figure 3).

If the XRT79L71 has been configured to operate in the High-Speed HDLC Controller Mode, then the Receive Overhead Data Output Interface block will be disabled.

1.2.17 A more detailed Functional/Architectural Description of the XRT79L71, when configured to operate in the High-Speed HDLC Controller Mode, is in the document (*79L71_Arch_Descr_HDLC.pdf*).

(Section 8.0 - Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - High-Speed HDLC Controller Applications).

1.3 BRIEF FUNCTIONAL/ARCHITECTURAL DESCRIPTION OF THE XRT79L71 - ATM UNI OVER DS3/E3 MODE

If the XRT79L71 has been configured to operate in the ATM UNI over DS3/E3 Mode, then it will have the Functional Architecture as is presented below in Figure 4.

FIGURE 4. THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE ATM UNI OVER DS3/E3 MODE

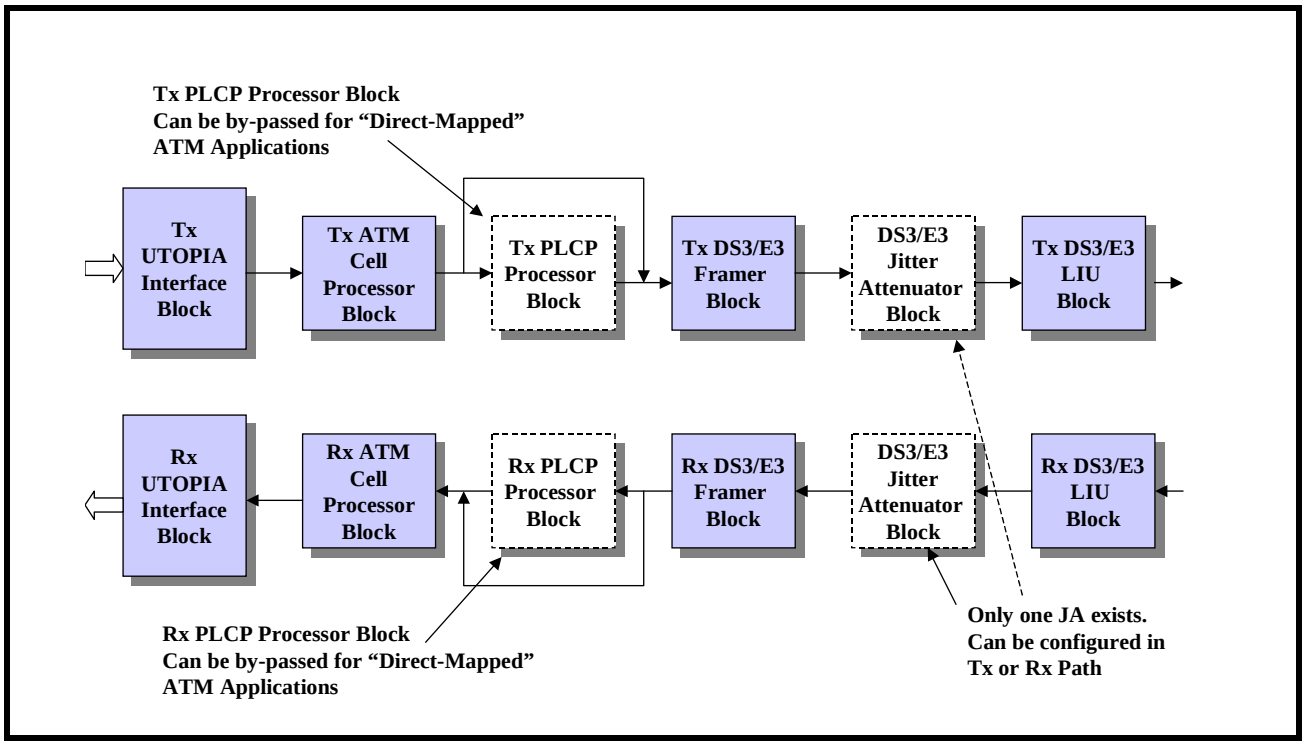


Figure 4 indicates that the XRT79L71 consists of the following functional blocks.

- The Transmit UTOPIA Interface Block
- The Transmit Overhead Data Input Interface Block
- The Transmit ATM Cell Processor Block
- The Transmit FEAC Controller Block (DS3 C-bit Parity Applications Only)
- The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)
- The Transmit SSM Controller Block (E3, ITU-T G.832 Appications Only)
- The Transmit LAPD Controller Block
- The Transmit PLCP Processor Block
- The Transmit DS3/E3 Framer Block
- The Transmit DS3/E3 LIU Block
- The Receive DS3/E3 LIU Block
- The Receive DS3/E3 Framer Block
- The Receive PLCP Processor Block
- The Receive LAPD Controller Block
- The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)

- The Receive Trail-Trace Message Controller Block (ITU-T G.832 Applications Only)
- The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Receive ATM Cell Processor Block
- The Receive Overhead Data Output Interface Block
- The Receive UTOPIA Interface Block

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

1.3.1 The Transmit UTOPIA Interface Block

The purpose of the Transmit UTOPIA Interface block is to provide a standard UTOPIA Level 1, 2 or 3 interface to the ATM Layer Processor, for writing in the contents of all Valid ATM cells, into the Transmit FIFO (TxFIFO).

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

NOTE: *The Transmit UTOPIA Interface block supports UTOPIA Level 3 from a signaling stand-point. The Transmit UTOPIA Interface Block (within the XRT79L71) still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates at clock rates of up to 50MHz (not 100MHz).*

1.3.2 The Transmit Overhead Data Input Interface block (not shown in Figure 4)

The purpose of the Transmit Overhead Data Input Interface block is to permit the user to externally insert their own values for overheads bits into the outbound DS3/E3 data-stream.

The Transmit ATM Cell Processor Block

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

- Optionally Compute and Verify the HEC byte of each cell written into the TxFIFO
- To optionally discard all incoming ATM cells that contain HEC byte errors
- To optionally compute and insert the HEC byte into the fifth octet position, within each ATM cell that is written into the TxFIFO
- To optionally filter User Cells (that are read out from the TxFIFO) by either discarding these User Cells, or by replicating them and routing the copies of these cells to the Transmit Cell Extraction Buffer
- To insert cells (residing within the Transmit Cell Insertion Buffer) into the Transmit Data Path anytime the TxFIFO is depleted of user cells
- To generate Idle Cells anytime the TxFIFO and the Transmit Cell Insertion Buffer are depleted of User cells
- To route the composite stream of valid and idle cells to either the Transmit PLCP Processor or the Transmit DS3/E3 Framer Block.

1.3.3 The Transmit FEAC Controller Block (DS3, C-bit Parity Applications Only)

The purpose of the Transmit FEAC Controller block is to permit the user to transmit FEAC (Far-End Alarm & Control) Messages to the remote terminal equipment.

NOTE: *The Transmit FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.3.4 The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit Trail-Trace Message Controller block is to permit a given terminal equipment to repeatedly transmit a "Trail-Trace" Message to the remote terminal equipment, via the TR bytes within each outbound E3, ITU-T G.832 frame.

1.3.5 The Transmit SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit SSM Controller Block is to permit a given terminal equipment to repeatedly transmit the "Synchronization Status Message" to the remote terminal equipment, via the MA byte, within each outbound E3, ITU-T G.832 frame.

1.3.6 The Transmit LAPD Controller Block

The purpose of the Transmit LAPD Controller block is to permit the user to transmit LAPD/PMDL (Path Maintenance Data Link) Messages to the remote terminal equipment. The Transmit LAPD Controller block comes with a LAPD Controller/Transmitter (not to be confused with the Transmit High-Speed HDLC Controller) and 90 bytes of on-chip RAM (for storage of outbound PMDL Messages) prior to transmission.

1.3.7 The Transmit PLCP Processor Block (Can be by-passed)

The purpose of the Transmit PLCP Processor block is to perform the following functionsl

- To accept ATM cells from the Transmit ATM Cell Processor Block and to pack 12 of these ATM cells into a PLCP frame
- As the Transmit PLCP Processor block creates these PLCP frames, it will also do the following
  - **a.** Automatically transmit the PLCP RAI (Remote Alarm Indicator) to the remote terminal whenever the corresponding Near-End Receive PLCP Processor block declares the PLCP LOF defect condition.
  - **b.** Automatically transmits the PLCP FEBE (Far-End Block Error) condition to the remote terminal whenever the corresponding near-end Receive PLCP Processor block detects PLCP B1 byte errors within its incoming PLCP data-stream.
- The Transmit PLCP Processor block can be commanded to transmit the PLCP RAI indicator upon Software command
- The Transmit PLCP Processor block can be commanded to transmit the PLCP FEBE indicator upon Software command
- Routes these outbound PLCP frames to the Transmit DS3/E3 Framer block
- Can be by-passed for Direct-Mapped ATM Applications.

1.3.8 The Transmit DS3/E3 Framer Block

The purpose of the Transmit DS3/E3 Framer block is to perform the following functions.

- To accept ATM cell data from the Transmit ATM Cell Processor block and to map this ATM cell data into the payload bits within each outbound DS3 or E3 frame (for Direct Mapped ATM Applications)
- To accept PLCP frames from the Transmit PLCP Processor block and to map these PLCP frames into the payload bits within each outbound DS3 or E3 frames (for PLCP Applications)
- To transmit this resulting framed DS3 or E3 data-stream to the Transmit DS3/E3 LIU block. In this case, the user can also configure the Transmit DS3/E3 Framer block to do the following.
  - **a.** Transmit an AIS pattern (upon software command)
  - **b.** Transmit a DS3 Idle pattern (upon software command)
  - **c.** Automatically transmit the FERF/RDI indicator whenever the Receive DS3/E3 Framer block (within this particular XRT79L71) declares the LOS, LOF/OOF or AIS defect condition.
  - **d.** Automatically transmits the FEBE/REI indicator whenever the corresponding near-end Receive DS3/E3 Framer block detects CP-bit or Framing bit errors within the incoming DS3 data-stream (for DS3 C-bit Parity Applications).
  - **e.** Automatically transmit the FEBE/REI indicator whenever the Receive DS3/E3 Framer block detects BIP-8 errors within the incoming E3 data-stream (for E3, ITU-T G.832 Applications).
  - **f.** To transmit FEAC (Far-End Alarm & Control) messages to the remote terminal equipment (DS3, C-bit Parity Applications only).
  - **g.** To transmit LAPD/PMDL (Path Maintenance Data Link) Messages to the remote terminal equipment.

h. To transmit Trail-Trace Messages to the remote terminal equipment (E3, ITU-T G.832 Applications only).

1.3.9 The Transmit DS3/E3 LIU Block

The purpose of the Transmit DS3/E3 LIU Block is to accept the outbound DS3/E3 data-stream from the Transmit DS3/E3 Framer block and to perform the following functions on this data.

- Encode this data into the B3ZS/HDB3 Line code
- Convert this data-stream into a proper DS3 or E3 line code that complies with the following requirements.
  - **a.** The Isolated Pulse Template requirements per Bellcore GR-499-CORE (for DS3 Applications).
  - **b.** The ITU-T G.703 Pulse Template requirements (for E3 Applications).

1.3.10 The Receive DS3/E3 LIU Block

The purpose of the Receive DS3/E3 LIU Block is to receive an inbound DS3/E3 line signal from the remote terminal equipment, and to perform the following functions on this data.

- To decode this data from the B3ZS/HDB3 Line code into a binary data-stream (prior to routing this data to the Receive DS3/E3 Framer block
- To detect and flag the occurrence of LCVs (Line Code Violations) and EXZs (Excessive Zeros)
- To insure that the XRT79L71 meets all of the following Receive requirements
  - **a.** The Receive Sensitivity requirements for DS3 Applications (e.g., able to receive a DSX-3 type of signal through at least 450 feet of cable loss).
  - **b.** The Receive Sensitivity requirements for E3 Applications (e.g., able to receive an E3 signal over 12dB of cable loss).
  - **c.** To comply with the Category I and II Jitter Tolerance Requirements per Bellcore GR-499-CORE (for DS3 Applications)
  - **d.** To comply with the Jitter Tolerance Requirements per ITU-T G.823 (for E3 Applications)
  - **e.** To comply with the Interference Margin Requirements of 20dB, per ITU-T G.703 (for E3 Applications)

1.3.11 The Receive DS3/E3 Framer Block

The purpose of the Receive DS3/E3 Framer block is to perform the following functions.

- To receive either a DS3 or E3 data-stream from the Receive DS3/E3 LIU Block. As the Receive DS3/E3 Framer block receives this DS3/E3 data-stream, it will do the following.
  - **a.** Compute and verify P and CP-bits (for DS3 Applications)
  - **b.** Compute and verify BIP-4 bits (for E3, ITU-T G.751 Applications)
  - **c.** Compute and verify BIP-8 bits (for E3, ITU-T G.832 Applications)
  - **d.** To declare and clear the LOS, OOF, LOF, AIS and FERF/RDI defect conditions
  - **e.** To declare and clear the DS3 Idle condition (DS3 Applications)
  - **f.** To detect FEBE/REI events.

1.3.12 The Receive PLCP Processor Block (Can be by-passed)

The purpose of the Receive PLCP Processor block is to perform the followoing functions.

- To receive the PLCP framd data-stream from the Receive DS3/E3 Framer block, and to acquire and maintain PLCP frame synchronization with this data
- To declare and clear the PLCP OOF, PLCP LOF and PLCP RAI defect conditions
- To detect and flag B1 byte errors
- To detect and flag PLCP FEBE/REI events

- To extract out the payload data from incoming PLCP frame (which consists of ATM cells) and to route this data to the Receive ATM Cell Processor block for further processing
- Can be by-passed for Direct-Mapped ATM Applications

1.3.13 The Receive LAPD Controller Block

The purpose of the Receive LAPD Controller block is to permit the user to receive LAPD/PMDL (Path Maintenance Data Link) Messages from the remote terminal equipment. The Receive LAPD Controller block comes with a Receive LAPD Controller (not to be confused with the Receive High-Speed HDLC Controller block) and 90 bytes of on-chip RAM (for storage of inbound PMDL Messages) after reception.

1.3.14 The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive SSM Controller block is to permit a given terminal equipment to receive (and extract out) the SSM (Synchronization Status Message) from the remote terminal equipment, via the MA byte, within each inbound E3, ITU-T G.832 data-stream. The Receive SSM Controller block will also alert the Microprocessor (by generating an interrupt) anytime it detects a change in the incoming SSM value.

1.3.15 The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive Trail-Trace Message Controller Block is to permit a given terminal equipment to receive (and extract out) the Trail-Trace Message (also known as "Trail-Access Pointer Identifier) from the remote terminal equipment, via the TR byte, within each inbound E3, ITU-T G.832 frame. The Receive Trail-Trace Message Controller block will also alert the Microprocessor (by generating an interrupt) anytime it detects a change in the incoming Trail-Trace Message.

1.3.16 The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)

The purpose of the Receive FEAC Controller block is to permit the user to receive FEAC (Far-End Alarm & Control) Messages from the remote terminal equipment.

NOTE: *The Receive FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.3.17 The Receive ATM Cell Processor Block

The purpose of the Receive ATM Cell Processor block is to extract out the data (being carried by the incoming DS3/E3 frame or PLCP frame data-stream) and to perform the following operations on it.

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

1.3.18 The Receive Overhead Data Output Interface block

The purpose of the Receive Overhead Data Output Interface block is to permit the user to extract out the overhead bits (within the incoming DS3/E3 data-stream) and to route this data to some off-chip System-Side Terminal Equipment circuitry.

1.3.19 The Receive UTOPIA Interface block

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

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

NOTE: The Receive UTOPIA Interface block supports UTOPIA Level 3 from a signaling stand-point. The Receive UTOPIA Interface block (within the XRT79L71) still only supports a 16-bit wide (not 32-bit wide) UTOPIA Bus and only operates up to 50MHz (not 100MHz).

1.3.20 A more detailed Functional/Architectural Description of the XRT79L71, when configured to operate in the ATM UNI Mode, can be found in the document (*79L71_Arch_Descr_ATM.pdf*).

(Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - ATM UNI Mode Applications).

1.4 FUNCTIONAL ARCHITECTURE/DESCRIPTION OF THE XRT79L71 - PPP OVER DS3/E3 MODE

If the XRT79L71 has been configured to operate in the PPP over DS3/E3 Mode, then it will have the Functional Architecture as is presented below in Figure 5.

FIGURE 5. THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 WHEN IT HAS BEEN CONFIGURED TO OPERATE IN THE PPP OVER DS3/E3 MODE

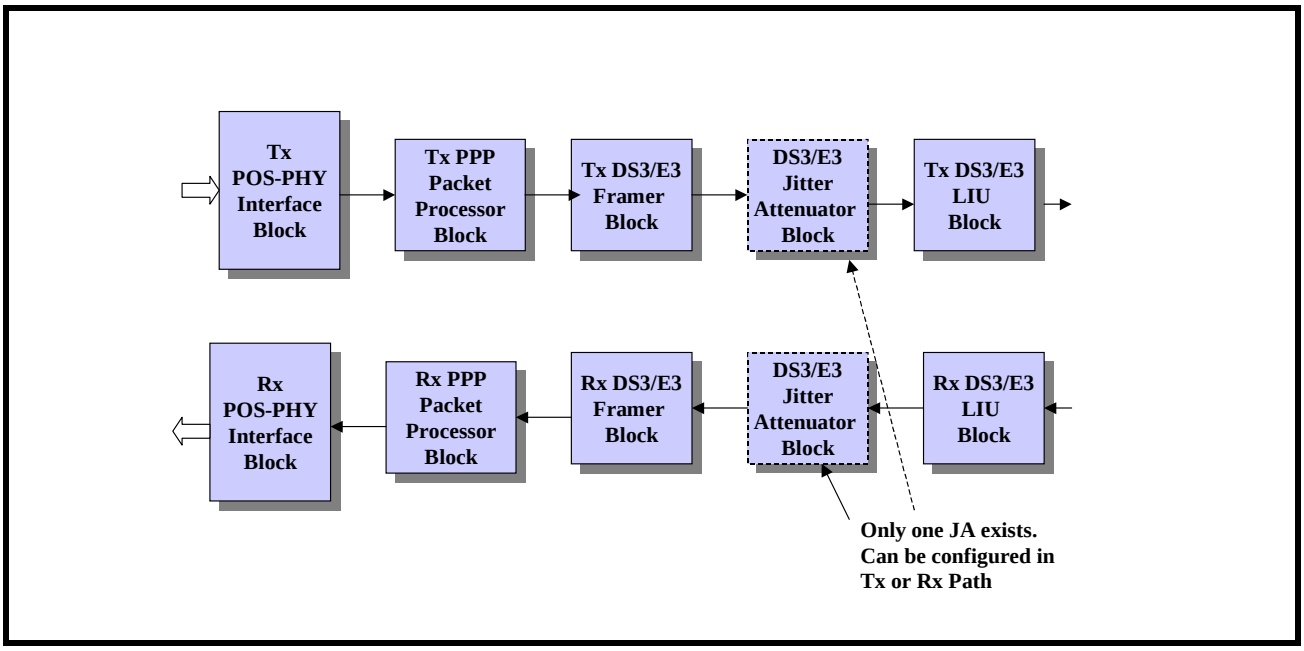


Figure 5 indicates that the XRT79L71 consists of the following functional blocks.

- The Transmit POS-PHY Interface block
- The Transmit Overhead Data Input Interface block
- The Transmit PPP Packet Processor Block
- The Transmit FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)
- The Transmit SSM Controller Block (E3, ITU-T G.832 Applications Only)
- The Transmit LAPD Controller Block
- The Transmit DS3/E3 Framer Block
- The Transmit DS3/E3 LIU Block
- he Receive DS3/E3 LIU Block

- The Receive DS3/E3 Framer Block
- The Receive LAPD Controller Block
- The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)
- The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)
- The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Receive PPP Packet Processor Block
- The Receive Overhead Data Output Interface Block
- The Receive POS-PHY Interface Block

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

1.4.1 The Transmit POS-PHY Interface Block

The purpose of the Transmit POS-PHY Interface block is to provide a standard Saturn POS-PHY™ Level 2 or 3 compliant interface to the Link Layer Processor, for writing in the contents of all outbound PPP packets, into the Transmit FIFO (TxFIFO).

The Transmit POS-PHY Interface block can be configured to operate with either an 8 or 16-bit wide Transmit POS-PHY Data Bus.

NOTES:

1. *The Transmit POS-PHY Interface Block supports POS-PHY Level 3 from a signaling stand-point. The Transmit POS-PHY Interface block (within the XRT79L71) still only supports a 16-bit wide (not 32-bit wide) POS-PHY Data Bus and only operates up to 50MHz (not 104MHz).*
2. *The Transmit POS-PHY Interface block can be configured to support either Out-of-Band Addressing or In-Band Addressing for Device Selection to WRITE operations. However, since the XRT79L71 is a single-channel device, we strongly recommend that the user only use Out-of-Band Addressing for Device Selection whenever it is designed into a Multi-PHY system in which multiple PHY Layer devices are sharing the same POS-PHY Bus.*

1.4.2 The Transmit Overhead Data Input Interface Block

The purpose of the Transmit Overhead Data Input Interface block is to permit the user to externally insert their own value for overhead bits into the outbound DS3/E3 data-stream.

1.4.3 The Transmit PPP Packet Processor Block

The purpose of the Transmit PPP Packet Processor block is to read out the contents of the PPP Packets that have been written into the TxFIFO (via the Transmit POS-PHY Interface block) and perform the following functions.

- Compute and verify the Transmit POS-PHY Interface Parity value for each byte or (16-bit) word of each incoming PPP Packet
- To Parse through the contents of each outbound packet for any occurrence of the value 0x7E and 0x7D and to character-stuff (or replace) these values with strings of values 0x7D5E and 0x7D5D, respectively
- To compute and append either a CRC-16 or CRC-32 value to the back-end of each outbound PPP Packet
- To repeatedly generate and transmit the Flag Sequence Octet, anytime the Transmit PPP Packet Processor block is NOT processing any PPP Packet data from the TxFIFO (e.g., whenever the TxFIFO is depleted)
- To route this composite stream of PPP packet and Flag Sequence octets to the Transmit DS3/E3 Framer block for further processing.

1.4.4 The Transmit FEAC Controller Block (DS3, C-bit Parity Applications Only)

The purpose of the Transmit FEAC Controller block is to permit the user to transmit FEAC (Far-End Alarm & Control) Messages to the remote terminal equipment.

NOTE: The Transmit FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.

1.4.5 The Transmit Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit Trail-Trace Message Controller block is to permit a given terminal equipment to repeatedly transmit a "Trail-Trace" Message to the remote terminal equipment, via the TR bytes within the outbound E3, ITU-T G.832 data-stream.

1.4.6 The Transmit SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Transmit SSM Controller Block is to permit a given terminal equipment to repeatedly transmit the "Synchronization Status Message" to the remote terminal equipment, via the MA byte, within each outbound E3, ITU-T G.832 frame.

1.4.7 The Transmit LAPD Controller Block

The purpose of the Transmit LAPD Controller block is to permit the user to transmit LAPD/PMDL (Path Maintenance Data Link) Messages to the remote terminal equipment. The Transmit LAPD Controller block comes with a LAPD Controller/Transmitter (not to be confused with the Transmit High-Speed HDLC Controller) and 90 bytes of on-chip RAM (for storage of outbound PMDL Messages) prior to transmission.

1.4.8 The Transmit DS3/E3 Framer Block

The purpose of the Transmit DS3/E3 Framer block is to perform the following functions.

- To accept PPP packet and Flag Sequence octets from the Transmit PPP Packet Processor block and to map all of this data into the payload bits within each outbound DS3 or E3 frame.
- To transmit the resulting framed DS3 or E3 data-stream to the Transmit DS3/E3 LIU Block. In this case, the user can also configure the Transmit DS3/E3 Framer block to do the following.
  - **a.** Transmit an AIS Pattern (upon software command)
  - **b.** Transmit a DS3 Idle Pattern (upon software command)
  - **c.** Automatically transmit the FERF/RDI whenever the Receive DS3/E3 Framer block (within the very same XRT79L71) declares the LOS, LOF/OOF or AIS defect conditions.
  - **d.** Automatically transmit the FEBE/REI indicator whenever the Receive DS3/E3 Framer block detects CP-bit or Framing bit errors within the incoming DS3 data-stream (for DS3, C-bit Parity Applications only)
  - **e.** Automatically transmit the FEBE/REI indicator whenever the Receive DS3/E3 Framer block detects BIP-8 errors within the incoming E3 data-stream (for E3, ITU-T G.832 Applications).

1.4.9 The Transmit DS3/E3 LIU Block

The purpose of the Transmit DS3/E3 LIU Block is to accept the outbound DS3/E3 data-stream from the Transmit DS3/E3 Framer block and to perform the following functions on this data.

- Encode this data into the B3ZS Line Code (for DS3 Applications) or the HDB3 Line Code (for E3 Applications)
- To convert this outbound DS3 or E3 data-stream into a DS3/E3 line signal and transmit this signal to the remote terminal equipment
- To generate and transmit DS3 pulses that comply with the Isolated Pulse Template requirements per Bellcore GR-499-CORE
- To generate and transmit E3 pulses that comply with the ITU-T G.703 Pulse Template requirements for E3 Applications.

1.4.10 The Receive DS3/E3 LIU Block

The purpose of the Receive DS3/E3 LIU Block is to receive a DS3/E3 line signal from the remote terminal equipment, and perform the following operations on this incoming line signal.

- To decode this incoming signal from the B3ZS Line Code (for DS3 Applications) or from the HDB3 Line Code (for E3 Applications) into a binary data-stream
- To route this binary data-stream to the Receive DS3/E3 Framer block for further processing
- To insure that the XRT79L71 meets all of the following Receive requirements.
  - **a.** The Receive Sensitivity requirements for DS3 Applications (e.g., able to receive a DSX-3 type of signal through at least 450 feet of cable loss)
  - **b.** The Receive Sensitivity requirements for E3 Applications (e.g., able to receive a E3 siganl over 12dB of cable loss)
  - **c.** To comply with the Category I and II Jitter Tolerance Requirements per Bellcore GR-499-CorE (for DS3 Applications)
  - **d.** To comply with the Jitter Tolerance Requirements per ITU-T G.823 (for E3 Applications)
  - **e.** To comply with the Interference Margin Requirements of 20dB, per ITU-T G.703 (for E3 Applications).

1.4.11 The Receive DS3/E3 Framer Block

The purpose of the Receive DS3/E3 Framer block is to acquire and maintain Frame Synchronization with the incoming DS3/E3 data-stream that it receives from the Receive DS3/E3 LIU Block. As the Receive DS3/E3 Framer block performs this task, it will also do the following.

- It will declare and clear the LOS defect condition
- It will declare and clear the LOF/OOF defect condition
- It will declare and clear the AIS defect condition
- It will declare and clear the FERF/RDI defect condition
- It will detect and flag the occurrence of P-bit, CP-bit and Framing bit errors (DS3 Applications)
- It will detect and flag the occurrence of BIP-8 Errors (E3, ITU-T G.832 Applications)
- It will detect and flag the occurrence of BIP-4 Errors (E3, ITU-T G.751 Applications)
- It will detect and flag the occurrences of FEBE/REI Events
- It will detect and flag any occurrences of LCVs (Line Code Violations) and EXZs (Excessive Zero) events within the incoming DS3/E3 line signal
- It will extract the payload bits (out from each incoming DS3 or E3 frame) and it will route this data to the Receive PPP Packet Processor block for further processing.

1.4.12 The Receive LAPD Controller Block

The purpose of the Receive LAPD Controller block is to permit the user to receive LAPD/PMDL (Path Maintenance Data Link) Messages from the remote terminal equipment. The Receive LAPD Controller block comes with a Receive LAPD Controller (not to be confused with the Receive High-Speed HDLC Controller block) and 90 bytes of on-chip RAM (for storage of inbound PMDL Messages) after reception.

1.4.13 The Receive SSM Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive SSM Controller block is to permit a given terminal equipment to receive (and extract out) the SSM (Synchronization Status Message) from the remote terminal equipment, via the MA byte, within each inbound E3, ITU-T G.832 frame. The Receive SSM Controller block will also alert the Microprocessor (by generating an interrupt) anytime it detects a change in the incoming SSM value.

1.4.14 The Receive Trail-Trace Message Controller Block (E3, ITU-T G.832 Applications Only)

The purpose of the Receive Trail-Trace Message Controller Block is to permit a given terminal equipment to receive (and extract out) the Trail-Trace Message from the remote terminal equipment, via the TR byte, within each inbound E3, ITU-T G.832 frame. The Receive Trail-Trace Message Controller block will also alert the Microprocessor (by generating an interrupt) anytime it detects a change in the incoming Trail-Trace Message.

1.4.15 The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)

The purpose of the Receive FEAC Controller block is to permit the user to receive FEAC (Far-End Alarm & Control) Messages from the remote terminal equipment.

NOTE: *The Receive FEAC Controller block is only active if the XRT79L71 is configured to operate in the DS3, C-bit Parity Framing format.*

1.4.16 The Receive PPP Packet Processor Block

The purpose of the Receive PPP Packet Processor block is to extract out the payload data (being carried by the incoming DS3/E3 data-stream) and to perform the following operations on it.

- Identification/Location of boundaries of incoming PPP packets
- Computation and Verification of either the CRC-16 or CRC-32 values within the incoming PPP Packets
- To Parse through the contents of each inbound packet for any occurrences of the values 0x7D5E and 0x7D5D and to character de-stuff (or replace) these values with strings of 0x7E and 0x7D, respectively
- To terminate any incoming Flag Sequence octets
- To flag the occurrence of any incoming RUNT packets
- To flag the occurrence of any incoming Aborted Packets

1.4.17 The Receive Overhead Data Output Interface Block

The purpose of the Receive Overhead Data Output Interface block is to permit the user to extract out the overhead bits (within the incoming DS3/E3 data-stream) and to route this data to some off-chip System-Side Terminal Equipment circuitry.

1.4.18 The Receive POS-PHY Interface Block

The purpose of the Receive POS-PHY Interface block is to provide a standard Saturn POS-PHY™ Level 2 or 3 compliant interface to the Link Layer Processor, for reading in the contents of all inbound PPP Packets, from the Receive FIFO (RxFIFO).

The Receive POS-PHY Interface block can be configured to operate with either an 8 or 16-bit wide Receive POS-PHY Data Bus.

NOTES:

1. *The Receive POS-PHY Interface Block supports POS-PHY Level 3 from a signaling stand-point. The Receive POS-PHY Interface block (within the XRT79L71) still only supports a 16-bit wide (not 32-bit wide) POS-PHY Data Bus and only operates up to 50MHz (not 104MHz).*
2. *The Receive POS-PHY Interface Block can be configured to support either Out-of-Band Addressing or In-Band Addressing for Device Selection to READ. However, since the XRT79L71 is a single-channel device, we strongly recommend that the user only use Out-of-Band Addressing for Device Selection whenever it is designed into a Multi-PHY systemin which multiple PHY Layer devices are sharing the same POS-PHY Bus.*

1.4.19 A more detailed Functional/Architectural Description of the XRT79L71, when configured to operate in the PPP Mode can be found in the document (*79L71_Arch_Descr_PPP.pdf)*

(Architectural/Functional Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer with LIU IC - PPP Mode Applications).

2.0 MICROPROCESSOR INTERFACE

The Microprocessor Interface of the XRT79L71 can be configured to support a wide variety of modes. These modes are listed below.

- Intel-Asynchronous Mode
- Motorola-Asynchronous Mode
- Intel X86 Mode

- Intel i960 Mode
- IDT3051/52 (MIPS) Mode
- Power PC 403 Mode

Of these various "Microprocessor Interface" Modes, the two most popular modes are the "Intel-Asynchronous" and the "Power PC 403" Modes. Each of these modes will be discussed in considerable detail in the Sections below.

THE PINS OF THE MICROPROCESSOR INTERFACE

Table 1a presents a list and a brief description of each of the pins that make up the Microprocessor Interface block, within the XRT79L71.

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
A[14:0]	E13 E14 E15 E16 D16 D15 C16 G13 G14 G15 G16 F13 F14 F15 F16	I	**Address Bus Input Pins:** This input pins permit the Microprocessor to identify on-chip registers and Buffer/Memory locations (within the XRT79L71) whenever it performs READ and WRITE operations with the XRT79L71.
D[7:0]	A15 B14 C13 D12 A14 B13 C12 D11	I/O	**Bi-Directional Data Bus pins:** This pins are used to drive and receive data over the bi-directional data bus, whenever the Microprocessor performs READ or WRITE operations with the Microprocessor Interface of the XRT79L71.

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
ALE/$\overline{AS}$	A16	I	**Address Latch Enable/Address Strobe:** The function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in, as described below. **Intel-Asynchronous Mode - ALE** If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then this active-high input pin is used to latch the data (residing on the Address Bus, A[14:0]) into the Microprocessor Interface circuitry of the XRT79L71 and to indicate the start of a READ or WRITE cycle. Pulling this input pin "high" enables the input bus drivers for the Address Bus input pins. The contents of the Address Bus will be latched into the Microprocessor Interface circuitry, upon the falling edge of this input signal. **Motorola-Asynchronous (68K) Mode - $\overline{AS}$** If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this active-low input pin is used to latch the data (residing on the Address Bus, A[14:0]) into the Microprocessor Interface circuitry of the XRT79L71 Pulling this input pin "low" enables the input bus drivers for the Address Bus input pins. The contents of the Address Bus will be latched into the Microprocessor Interface circuitry, upon the rising edge of this input signal. **Power PC 403 Mode - No Function - Tie to GND:** If the Microprocessor Interface has been configured to operate in the Power PC 403 Mode, then this input pin has no role nor function and should be tied to GND.
$\overline{CS}$	D14	I	**Chip Select Input:** The user must assert this active low signal in order to select the Microprocessor Interface for READ and WRITE operations between the Microprocessor and the XRT79L71 on-chip registers and buffer/memory locations.
$\overline{INT}$	D13	O	**Interrupt Request Output:** This active-low output signal will be asserted (pulled to a logic low level) whenever the XRT79L71 is requesting interrupt service from the Microprocessor. This output pin should typically be connected to the "Interrupt Request" input of the Microprocessor.

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
$\overline{RD}$/$\overline{DS}$/$\overline{WE}$	C15	I	**READ Strobe/Data Strobe:** The function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in. **Intel-Asynchronous Mode - $\overline{RD}$ - READ Strobe Input:** If the Microprocessor Interface is operating in the Intel-Asynchronous Mode, then this input pin will function as the $\overline{RD}$ (Active Low Read Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the XRT79L71 will place the contents of the addressed register (or buffer location) on the Microprocessor Interface Bi-directional data bus (D[7:0]). When this signal is negated, then the Data Bus will be tri-stated. **Motorola-Asynchronous (68K) Mode - $\overline{DS}$ - Data Strobe:** If the Microprocessor Interface is operating in the Motorola-Asynchronous Mode, then this input pin will function as the $\overline{DS}$ (Data Strobe) input signal. **Power PC 403 Mode - $\overline{WE}$ - Write Enable Input:** If the Microprocessor Interface is operating in the Power PC 403 Mode, then this input pin will function as the $\overline{WE}$ (Write Enable) input pin. Anytime the Microprocessor Interface samples this active-low input signal (along with $\overline{CS}$ and WR/$\overline{R}$/$\overline{W}$) also being asserted (at a logic low level) upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents on the Bi-Directional Data Bus (D[7:0]) into the "target" on-chip register or buffer location within the XRT79L71.

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION

RDY/$\overline{\text{DTACK}}$	C14	O	**READY or $\overline{\text{DTACK}}$ Output:**
			The exact function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in, as described below.
			Intel-Asynchronous Mode - RDY - Ready Output:
			If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then this output pin will function as the "active-low" READY output.
			During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic low level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic "low" level, then it is now safe for it to move on and execute the next READ or WRITE cycle.
			If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "high" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it detects this output pin being toggled to the logic low level.
			Motorola-Asynchronous Mode - $\overline{\text{DTACK}}$ - Data Transfer Acknowledge Output
			If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this output pin will function as the "active-low" DTACK output.
			During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic low level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic "low" level, then it is now safe for it to move on and execute the next READ or WRITE cycle. If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "high" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it detects this output pin being toggled to the logic low level.
			Power PC 403 Mode - RDY - Ready Output:
			If the Microprocessor Interface has been configured to operate in the Power PC 403 Mode, then this output pin will function as the "active-high" READY output.
			During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic high level, ONLY when the Microprocessor Interface is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has sampled this signal being at the logic "high" level (upon the rising edge of PCLK), then it is now safe for it to move on and execute the next READ or WRITE cycle.
			If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "low" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it samples this output pin being at the logic low level.
			NOTE: *The Microprocessor Interface will update the state of this output pin upon the rising edge of* μPCLK.

$\overline{\text{RESET}}$	M14	I	**Hardware Reset Input:** When this "active-low" signal is asserted, the XRT79L71 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.

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
µPCLK	H16	I	**Microprocessor Interface Clock Input:** This clock input signal is only used if the Microprocessor Interface has been configured to operate in one of the Synchronous Modes (e.g., Power PC 403 Mode). If the Microprocessor Interface is configured to operate in one of these modes, then it will use this clock signal to do the following. • To sample the $\overline{CS}$, $\overline{WR}$/R/$\overline{W}$, A[14:0], D[7:0], $\overline{RD}$/$\overline{DS}$ and DBEN input pins, and • To update the state of the D[7:0] and the RDY/$\overline{DTACK}$ output signals. ***NOTES:*** *1. The Microprocessor Interface can work with µPCLK frequencies ranging up to 33MHz.* *2. This pin is inactive if the Microprocessor interface has been configured to operate in either the Intel-Asynchronous or the Motorola-Asynchronous Modes. In this case, tie this pin to GND.*

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
$\overline{WR}$/R/$\overline{W}$	B16	I	**Write Strobe/Read-Write Operation Identifier:** The function of this input pin depends upon which mode the Microprocessor Interface has been configured to operate in. **Intel-Asynchronous Mode - $\overline{WR}$ - Write Strobe Input:** If the Microprocessor Interface is configured to operate in the Intel-Asynchronous Mode, then this input pin functions as the $\overline{WR}$ (Active Low WRITE Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the input buffers (associated with the Bi-Directional Data Bus pin, D[7:0]) will be enabled. The Microprocessor Interface will latch the contents on the Bi-Directional Data Bus (into the "target" register or address location, within the XRT79L71) upon the rising edge of this input pin. **Motorola-Asynchronous Mode - R/$\overline{W}$ - Read/Write Operation Identification Input Pin:** If the Microprocessor Interface is operating in the "Motorola-Asynchronous Mode", then this pin is functionally equivalent to the "R/$\overline{W}$" input pin. In the Motorola Mode, a "READ" operation occurs if this pin is held at a logic "1", coincident to a falling edge of the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin. Similarly a WRITE operation occurs if this pin is at a logic "0", coincident to a falling edge of the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin. **Power PC 403 Mode - R/$\overline{W}$ - Read/Write Operation Identification Input:** If the Microprocessor Interface is configured to operate in the Power PC 403 Mode, then this input pin will function as the "Read/Write Operation Identification Input" pin. Anytime the Microprocessor Interface samples this input signal at a logic low (while also sampling the $\overline{CS}$ input pin "low") upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents of the Address Bus (A[14:0]) into the Microprocessor Interface circuitry, in preparation for this forthcoming READ operation. At some point (later in this READ operation) the Microprocessor will also assert the $\overline{DBEN}$/$\overline{OE}$ input pin, and the Microprocessor Interface will then place the contents of the "target" register (or address location within the XRT79L71) upon the Bi-Directional Data Bus pins (D[7:0]), where it can be read by the Microprocessor Anytime the Microprocessor Interface samples this input signal at a logic high (while also sampling the $\overline{CS}$ input pin a logic "low") upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents of the Address Bus (A[14:0]) into the Microprocessor Interface circuitry, in preparation for the forthcoming WRITE operation. At some point (later in this WRITE operation) the Microprocessor will also assert the $\overline{RD}$/$\overline{DS}$/$\overline{WE}$ input pin, and the Microprocessor Interface will then latch the contents of the Bi-Directional Data Bus (D[7:0]) into the contents of the "target" register or buffer location (within the XRT79L71).

Table 3: List and Brief Description of the Microprocessor Interface Pins

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
PTYPE[2:0]	J14J15J16	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]** / **Microprcessor Interface mode** 000 — Intel-Asynchronous Mode 001 — Motorola-Asynchronous Mode - (Motorola 68K) 010 — Intel X86 011 — Intel i960 100 — IDT3051/52 (MIPS) 101 — Power PC 403 Mode
$\overline{\text{DBEN}}$	J13	I	**Bi-directional Data Bus Enable Input pin:** This input pin permits the user to either enable or tri-state the Bi-Directional Data Bus pins (D[7:0]). Setting this input pin "low" enables the Bi-directional Data bus. Setting this input "high" tri-states the Bi-directional Data Bus.
$\overline{\text{BLAST}}$	B15	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. ***NOTE:*** *If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous, the Motorola-Asynchronous or the Power PC 403 Mode, then tie this input pin to GND.*

2.1 Operating the Microprocessor Interface in the Intel-Asynchronous Mode

If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then the following Microprocessor Interface pins will assume the role that is described below in Table 1b.

TABLE 4: THE ROLES OF VARIOUS MICROPROCESSOR INTERFACE PINS, WHEN CONFIGURED TO OPERATE IN THE INTEL-ASYNCHRONOUS MODE

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
ALE/AS	A16	I	**Address Latch Enable Input - ALE** If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then this active-high input pin is used to latch the data (residing on the Address Bus, A[14:0]) into the Microprocessor Interface circuitry of the XRT79L71 and to indicate the start of a READ or WRITE cycle. Pulling this input pin "high" enables the input bus drivers for the Address Bus input pins. The contents of the Address Bus will be latched into the Microprocessor Interface circuitry, upon the falling edge of this input signal.
$\overline{RD}/\overline{DS}/\overline{WE}$	C15	I	**Read Strobe Input - $\overline{RD}$** If the Microprocessor Interface is operating in the Intel-Asynchronous Mode, then this input pin will function as the $\overline{RD}$ (Active Low Read Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the XRT79L71 will place the contents of the addressed register (or buffer location) on the Microprocessor Interface Bi-directional data bus (D[7:0]). When this signal is negated, then the Data Bus will be tri-stated.
$\overline{RDY}/DTACK/\overline{RDY}$	C14	O	**Active Low Ready Output - $\overline{RDY}$** If the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then this output pin will function as the "active-low" READY output. During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic low level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic "low" level, then it is now safe for it to move on and execute the next READ or WRITE cycle. If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "high" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it detects this output pin being toggled to the logic low level.
µPCLK	H16	I	NONE - Tie to GND
$\overline{WR}/R/\overline{W}$	B16	I	**Write Strobe Input - $\overline{WR}$** If the Microprocessor Interface is configured to operate in the Intel-Asynchronous Mode, then this input pin functions as the $\overline{WR}$ (Active Low WRITE Strobe) input signal from the Microprocessor. Once this active-low signal is asserted, then the input buffers (associated with the Bi-Directional Data Bus pin, D[7:0]) will be enabled. The Microprocessor Interface will latch the contents on the Bi-Directional Data Bus (into the "target" register or address location, within the XRT79L71) upon the rising edge of this input pin.

TABLE 4: THE ROLES OF VARIOUS MICROPROCESSOR INTERFACE PINS, WHEN CONFIGURED TO OPERATE IN THE INTEL-ASYNCHRONOUS MODE

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
$\overline{\text{DBEN}}$	J13	I	**Data Bus Enable Input:** For Intel-Asynchronous Mode operation, either tie this pin to a logic "low" or assert this pin (e.g., toggle it to a logic "low") anytime a READ operation is being performed with the Microprocessor Interface of the XRT79L71.
$\overline{\text{BLAST}}$	B15	I	NONE - Tie this pin to GND

Configuring the Microprocessor Interface to operate in the Intel-Asynchronous Mode

Configure the Microprocessor Interface to operate in the Intel-Asynchronous Mode by tying the PTYPE[2:0] pins/balls (e.g., Ball Numbers J14, J15 and J16, respectively) to GND.

Finally, if the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then it will perform READ and WRITE operations as described below.

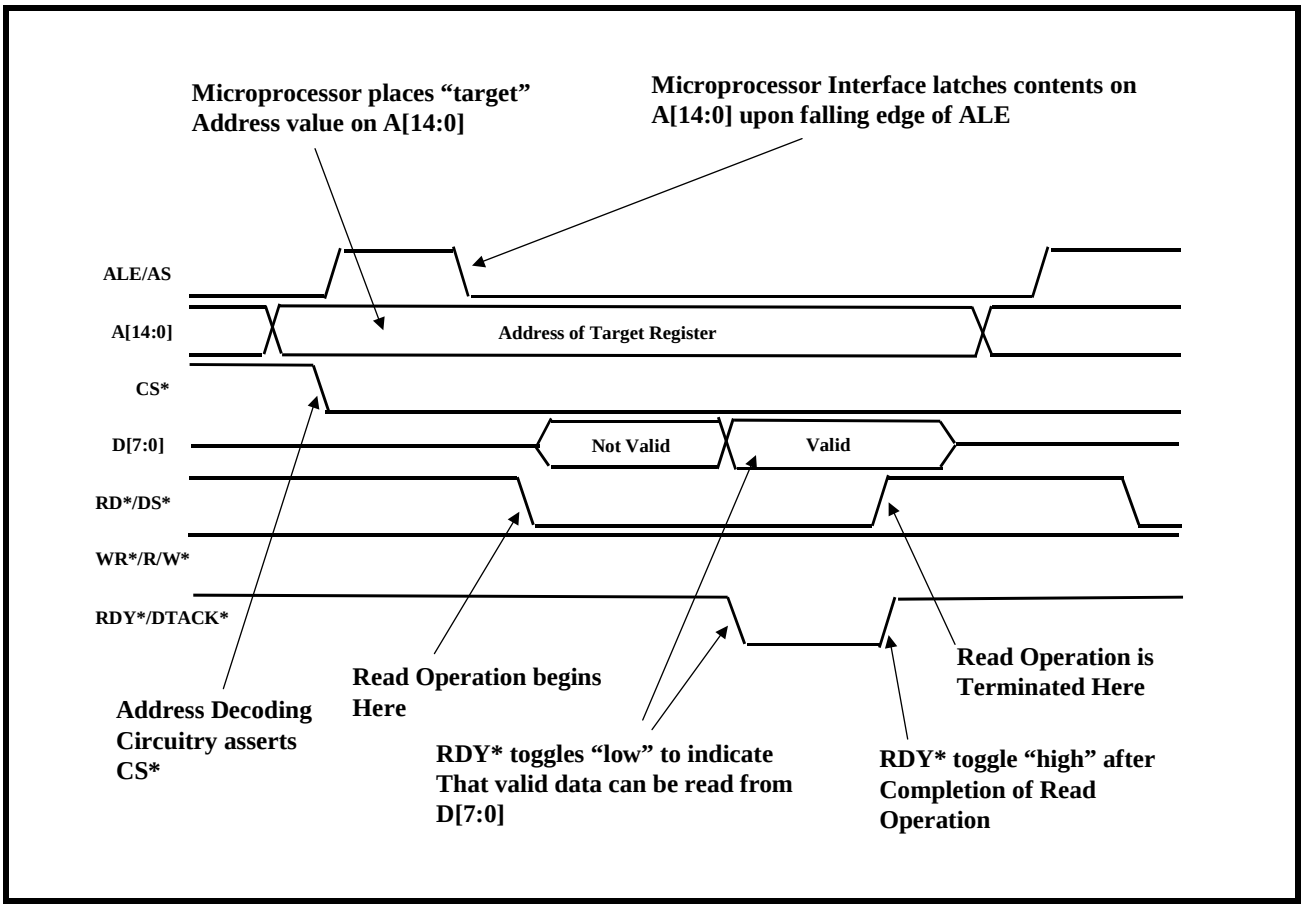
2.1.1 The Intel-Asynchronous Read-Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the Intel-Asynchronous Mode, then the Microprocessor should do all of the following, anytime it wishes to read out the contents of a register or some location within the Receive LAPD Message buffer, the Receive Cell Extraction Memory or the Transmit Cell Extraction Memory, (within the XRT79L71).

1. Place the address of the "target" register or buffer location (within the XRT79L71) on the Address Bus input pins A[14:0].
2. While the µC/µP is placing this address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{\text{CS}}$ (Chip Select) pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface block.
3. Toggle the ALE/AS (Address Latch Enable) input pin "high". This step enables the "Address Bus" input drivers, within the Microprocessor Interface block of the XRT79L71.
4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address" Data Setup time"), the µC/µP should toggle the ALE/AS pin "low". This step causes the XRT79L71 to "latch" the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer locations (within the XRT79L71) has now been selected.
5. Next, the µC/µP should indicate that this current bus cycle is a "Read" Operation by toggling the $\overline{\text{RD}}/\overline{\text{DS}}$ (Read Strobe) input pin "low". This action also enables the bi-directional data bus output drivers of the XRT79L71. At this point, the "bi-directional" data bus output drivers will proceed to drive the contents of the "latched addressed" register (or buffer/memory location) onto the bi-directional data bus, D[7:0].
6. Immediately after the µC/µP toggles the "Read Strobe" ($\overline{\text{RD}}/\overline{\text{DS}}$) signal "low", the XRT79L71 will continue to drive the $\overline{\text{RDY}}/\overline{\text{DTACK}}$ output pin "high". The XRT79L71 does this in order to inform the µC/µP that the data (to be read from the data bus) is "NOT READY" to be "latched" into the µC/µP. In this case, the µC/µP should continue to hold the "Read Strobe" ($\overline{\text{RD}}/\overline{\text{DS}}$) signal "low" until it detects the $\overline{\text{RDY}}/\overline{\text{DTACK}}$ output pin toggling low.
7. After some settling time, the data on the "bi-directional" data bus will stabilize and can be read by the µC/µP. At this time, the XRT79L71 will indicate that this data can be read by toggling the $\overline{\text{RDY}}/\overline{\text{DTACK}}$ (READY) signal "low".
8. After the µC/µP detects the $\overline{\text{RDY}}/\overline{\text{DTACK}}$ signal (from the XRT79L71) toggling "low", it can then terminate the Read Cycle by toggling the $\overline{\text{RD}}/\overline{\text{DS}}$ (Read Strobe) input pin "high".

Figure 6 presents a timing diagram that illustrates the behavior of the Microprocessor Interface signals, during an "Intel-Asynchronous" Mode Read Operation.

FIGURE 6. BEHAVIOR OF MICROPROCESSOR INTERFACE SIGNALS DURING AN "INTEL-ASYNCHRONOUS" READ OPERATION.



2.1.2 The Intel-Asynchronous Write Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the Intel-Asynchronous Mode, then the Microprocessor should do all of the following, anytime it wishes to write a byte of data into a register or some location within the Transmit LAPD Message buffer, the Transmit Cell Insertion Memory or the Receive Cell Insertion Memory (within the XRT79L71), it should do the following.

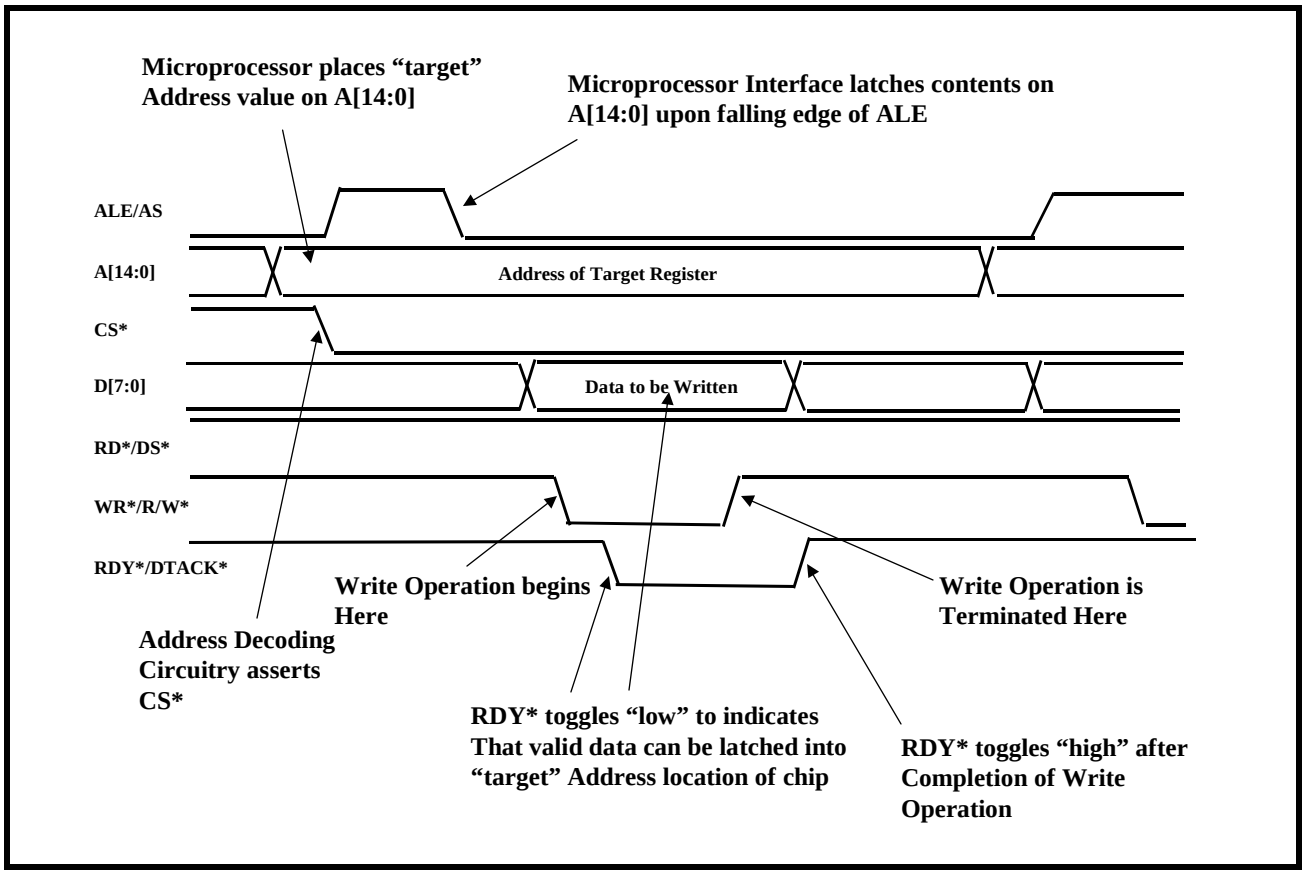
1. Place the address of the "target" register or buffer location (within the XRT79L71) on the Address Bus input pins, A[14:0].
2. While the µC/µP is placing the address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{CS}$ (Chip Select) input pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface.
3. Toggle the ALE/AS (Address Latch Enable) input pin "high". This step enables the "Address Bus" input drivers, within the Microprocessor Interface block of the XRT79L71.
4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time) the µC/µP should toggle the ALE/AS input pin "low". This step causes the XRT79L71 to "latch" the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer locations (within the XRT79L71) has now been selected.

5. Next, the mC/uP should then place the byte that it intends to write into the "target" register (or buffer/ memory location) into the XRT79L71, on the bi-directional data bus pins (D[7:0]).
6. Afterwards, the µC/µP should then indicate that this current bus cycle is a "Write" Operation; by toggling the WR/R/W (Write Strobe) input pin "low". This action also enables the "bi-directional" data bus input drivers of the XRT79L71. At this point, the "bi-directional" data bus input drivers will proceed to drive the contents (currently residing on the Bi-Directional Data bus into the register (or buffer/memory location) that corresponds with the "latched address".
7. Immediately after the µC/µP toggles the "Write Strobe" (WR/R/W) signal "low", the XRT79L71 will continue to drive the RDY/DTACK output pin "high". The XRT79L71 does this in order to inform the µC/ µP that the data (to be written into the "target" address location (within the XRT79L71) is "NOT READY" to be latched into the µC/µP. In this case, the µC/µP should continue to hold the "Write Strobe" (WR/R/W) input pin "low" until it detects the RDY/DTACK output pin toggling high.
8. After waiting the appropriate amount of time, for the data (on the bi-directional data bus) to stabilize and can be safely accepted by the µC/µP. At this time, the XRT79L71 will indicate that this data can be latched into the "target" address location, by toggling the RDY/DTACK output pin "low".
9. After the µC/µP detects the RDY/DTACK signal (from the XRT79L71) toggling "low", it can then terminate the Write Cycle by toggling the WR/R/W (Write Strobe) input pin "high".

NOTE: *Once the user toggles the "WR/R/W (Write Strobe) input pin "high", then the Microprocessor Interface (of the XRT79L71) will latch the contents of the bi-directional data bus (D[7:0]) into the "target" address location within the chip.*

Figure 7 presents a timing diagram that illustrates the behavior of the Microprocessor Interface signals, during an "Intel-Asynchronous" Mode Write Operation.

FIGURE 7. BEHAVIOR OF THE MICROPROCESSOR INTERFACE SIGNALS, DURING AN "INTEL-ASYNCHRONOUS" WRITE OPERATION.



2.2 Operating the Microprocessor Interface in the Motorola-Asynchronous Mode

If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then the following Microprocessor Interface pins will assume the role that is described below in Table 1c.

TABLE 5: THE ROLES OF VARIOUS MICROPROCESSOR INTERFACE PINS, WHEN CONFIGURED TO OPERATE IN THE MOTOROLA-ASYNCHRONOUS MODE

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
ALE/$\overline{AS}$	A16	I	**Address Strobe Input - $\overline{AS}$** If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this active-low input pin is used to latch the data (residing on the Address Bus, A[14:0]) into the Microprocessor Interface circuitry of the XRT79L71. Pulling this input pin "low" enables the input bus drivers for the Address Bus input pins. The contents of the Address Bus will be latched into the Microprocessor Interface circuitry, upon the rising edge of this input signal.
$\overline{RD}$/$\overline{DS}$/$\overline{WE}$	C15	I	**Data Strobe Input - $\overline{RD}$** If the Microprocessor Interface is operating in the Motorola-Asynchronous Mode, then this input pin will function as the $\overline{DS}$ (Data Strobe) input signal.
$\overline{RDY}$/$\overline{DTACK}$/RDY	C14	O	**Data Transfer Acknowledge Output - $\overline{DTACK}$** If the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then this output pin will function as the "active-low" DTACK output. During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic low level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has determined that this input pin has toggled to the logic "low" level, then it is now safe for it to move on and execute the next READ or WRITE cycle. If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "high" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it detects this output pin being toggled to the logic low level.
µPCLK	H16	I	NONE - Tie to GND
$\overline{WR}$/R/$\overline{W}$	B16	I	**Read/Write Operation Identification Input - R/$\overline{W}$** If the Microprocessor Interface is operating in the "Motorola-Asynchronous Mode", then this pin is functionally equivalent to the "R/$\overline{W}$" input pin. In the Motorola Mode, a "READ" operation occurs if this pin is held at a logic "1", coincident to a falling edge of the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin. Similarly a WRITE operation occurs if this pin is at a logic "0", coincident to a falling edge of the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin.
$\overline{DBEN}$	J13	I	**Data Bus Enable Input:** For Intel-Asynchronous Mode operation, either tie this pin to a logic "low" or assert this pin (e.g., toggle it to a logic "low") anytime a READ operation is being performed with the Microprocessor Interface of the XRT79L71.
BLAST	B15	I	NONE - Tie this pin to GND

Configuring the Microprocessor Interface to operate in the Motorola-Asynchronous Mode

Configure the Microprocessor Interface to operate in the Intel-Asynchronous Mode by tying the PTYPE2 and PTYPE1 pins/balls (e.g., Ball Numbers J14 and J15, respectively) to GND, and by tying PTYPE 0 (Ball Number J16) to a logic "high".

Finally, if the Microprocessor Interface has been configured to operate in the Motorola-Asynchronous Mode, then it will perform READ and WRITE operations as described below.

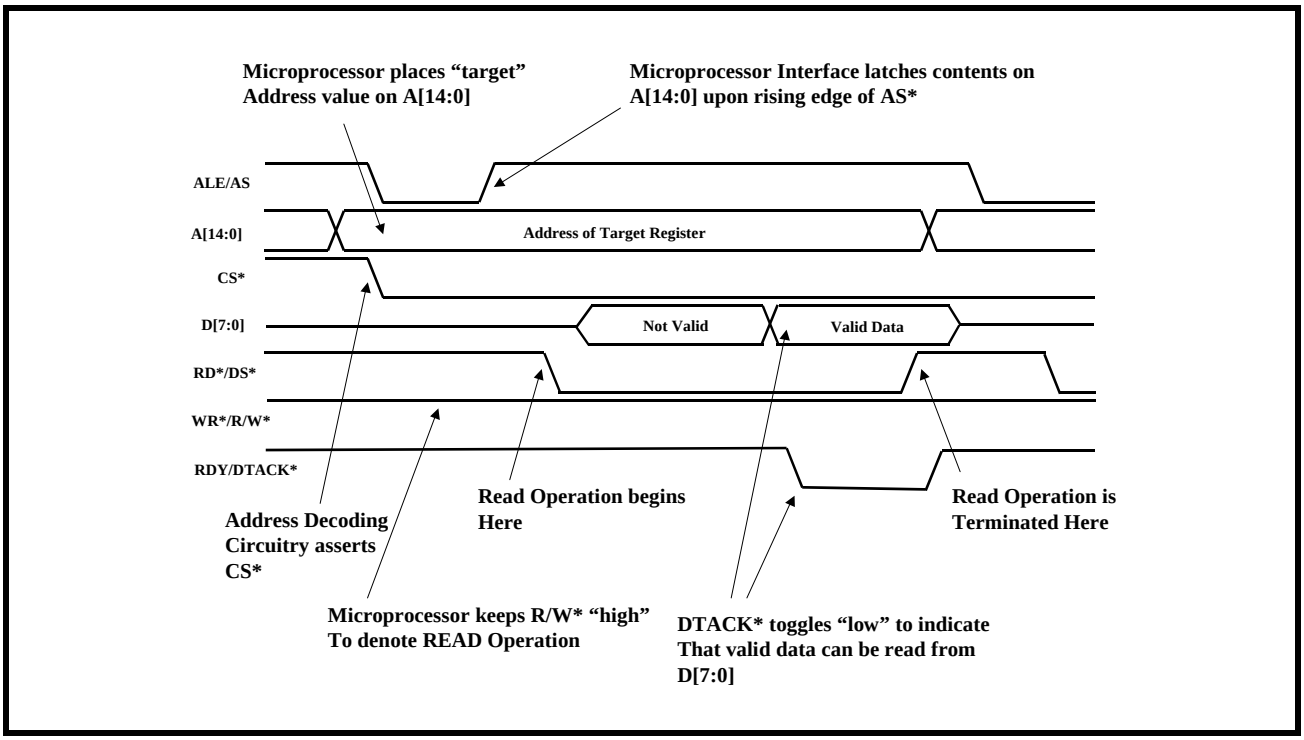
2.2.1 The Motorola-Asynchronous Read-Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the Motorola-Asynchronous Mode, then the Microprocessor should do all of the following, anytime it wishes to read out the contents of a register or some location within the Receive LAPD Message buffer, the Receive Cell Extraction Memory or the Transmit Cell Extraction Memory (within the XRT79L71), it should do the following.

1. Place the address of the "target" register (or buffer location) within the XRT79L71, on the Address Bus Input pins, A[14:0].
2. While the µC/µP is placing the address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{CS}$ (Chip Select) pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface block.
3. Assert the ALE/AS (Address-Strobe) input pin by toggling it low. This step enables the Address Bus input drivers, within the Microprocessor Interface Block of the XRT79L71.
4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time), the µC/µP should toggle the ALE/AS input pin "high". This step causes the XRT79L71 to latch the contents of the "Address Bus" into its internal circuitry. At this point, the address of the register or buffer location (within the XRT79L71) has now been selected.
5. Afterwards, the µC/µP should indicate that this cycle is a "Read" cycle by setting the $\overline{WR}$/R/$\overline{W}$ (R/$\overline{W}$) input pin "high".
6. Next the µC/µP should initiate the current bus cycle by toggling the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin "low". This step enables the bi-directional data bus output drivers, within the XRT79L71. At this point, the bi-directional data bus output drivers will proceed to driver the contents of the "Address" register onto the bi-directional data bus, D[7:0].
7. Immediately after the µC/µP toggles the "Data Strobe" ($\overline{RD}$/$\overline{DS}$) signal "low", the XRT79L71 will continue to drive the $\overline{RDY}$/$\overline{DTACK}$ output pin "high". The XRT79L71 does this in order to inform the µC/µP that the data (to be read from the data bus) is "NOT READY" to be latched into the µC/µP. In this case, the µC/µP should continue to hold the "Data Strobe" ($\overline{RD}$/$\overline{DS}$) signal "low" until it detects the $\overline{RDY}$/$\overline{DTACK}$ output pin toggling "low".
8. After some settling time, the data on the "bi-directional" data bus will stabilize and can be read by the µC/µP. The XRT79L71 will indicate that this data can be read by asserting the $\overline{RDY}$/$\overline{DTACK}$ (DTACK) output signal (by toggling it "low").
9. After the µC/µP detects the $\overline{RDY}$/$\overline{DTACK}$ signal (from the XRT79L71) toggling "low", it can terminate the Read Cycle by toggling the "$\overline{RD}$/$\overline{DS}$" (Data Strobe) input pin "high".

Figure 8 presents a timing diagram that illustrates the behavior of the Microprocessor Interface signals during a "Motorola-Asynchronous" Read Operation.

FIGURE 8. ILLUSTRATION OF THE BEHAVIOR OF MICROPROCESSOR INTERFACE SIGNALS, DURING A "MOTOROLA-ASYNCHRONOUS" READ OPERATION.



2.2.2 The Motorola-Asynchronous Write-Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the Motorola-Asynchronous Mode, then the Microprocessor should do all of the following, anytime it wishes to write a byte of data into a register or some location within the Transmit LAPD Message buffer, the Transmit Cell Insertion Memory or the Receive Cell Insertion Memory (within the XRT79L71).

1. Place the address of the "target" register or buffer location (within the XRT79L71) on the Address Bus input pins, A[14:0].
2. While the µC/µP is placing the address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{CS}$ (Chip Select) input pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface.
3. Assert the ALE/AS (Address Strobe) input pin by toggling it "low". This step enables the "Address Bus" input drivers, within the Microprocessor Interface block of the XRT79L71.
4. After allowing the data on the Address Bus pins to settle (by waiting the appropriate "Address Setup" time), the µC/µP should toggle the ALE/AS input pin "high". This step causes the XRT79L71 to "latch" the contents of the "Address Bus" into its internal circuitry. At this point, the Address of the register or buffer location (within the XRT79L71) has now been selected.
5. Afterwards, the µC/µP should indicate that this current bus cycle is a "Write" operation by toggling the $\overline{WR}$/R/$\overline{W}$ (R/$\overline{W}$) input pin "low".
6. The µC/µP should then place the byte or word that it intends to write into the "target" register, on the bi-directional data bus, D[7:0].
7. Next, the µC/µP should initiate the bus cycle by toggling the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin "low". When the XRT79L71 senses that the WRB_RW (R/$\overline{W}$) input pin is "high" and that the $\overline{RD}$/$\overline{DS}$ (Data Strobe) input pin has toggled "low", it will enable the "input drivers" of the bi-directional data bus, D[7:0].

8. Immediately after the µC/µP toggles the RD/DS (Data Strobe) signal "low", the XRT79L71 will continue to drive the RDY/DTACK output pin "high". The XRT79L71 does this in order to inform the µC/µP that the data (to be written into the "target" address location, within the XRT79L71) is "NOT READY" to be latched into the µC/µP. In this case, the µC/µP should continue to hold the "Data Strobe" (RD/DS) input pin "low" until it detects the RDY/DTACK output pin toggling "high".
9. After waiting the appropriate time, for the data (on the bi-directional data bus) to settle and can be safely accepted by the µC/µP. At this time, the XRT79L71 will indicate that this data can be latched into the "target" address location by toggling the RDY/DTACK output pin "low".
10. After the µC/µP detects the RDY/DTACK signal (from the XRT79L71) toggling "low", it can then terminate the Write Cycle by toggling the "RD/DS" (Data Strobe) input pin "high".

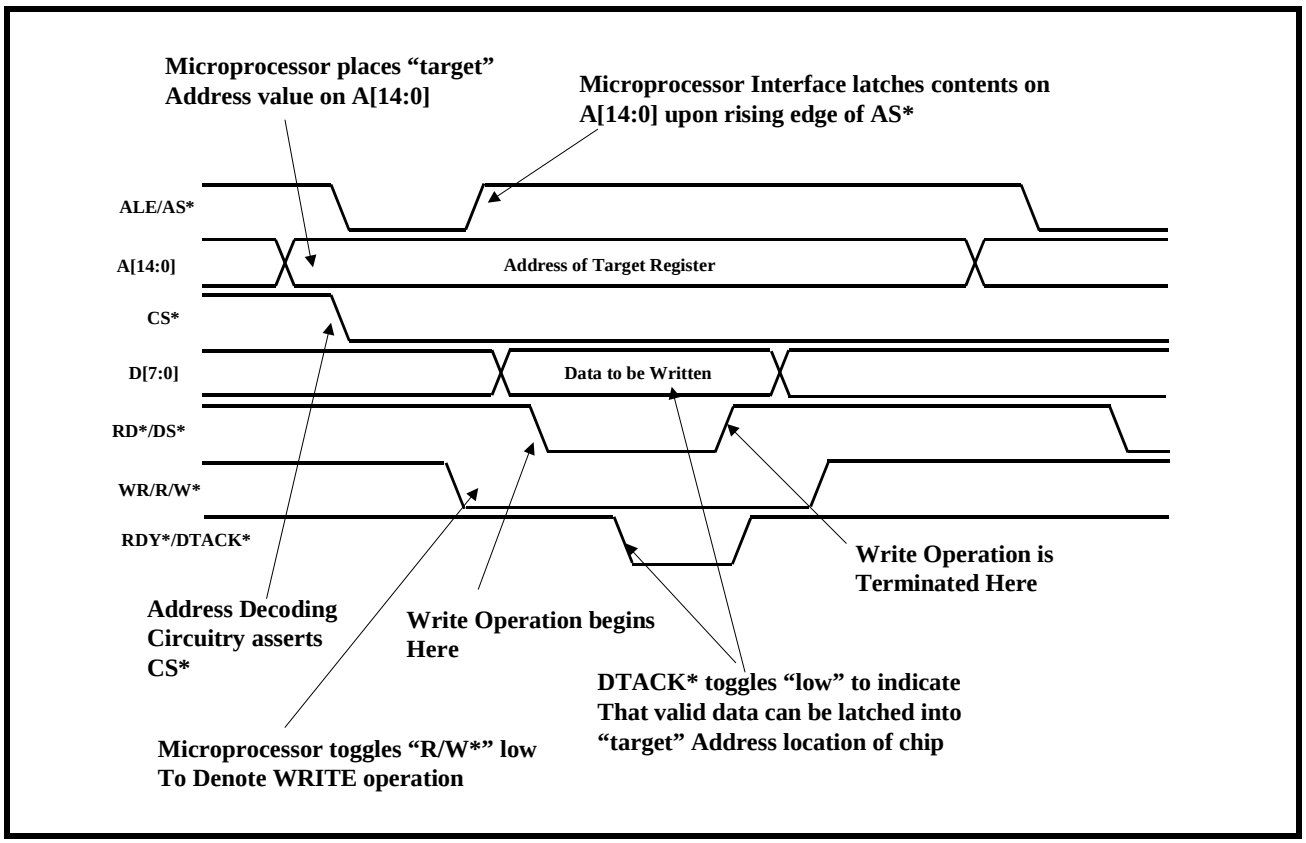
NOTE: *Once the user toggles the RD/DS (Data Strobe) input pin "high", then the following two things will happen.*

a. The XRT79L71 will latch the contents of the bi-directional data bus into the Microprocessor Interface block.

b. The XRT79L71 will terminates the "Write" cycle.

Figure 9 presents a timing diagram that illustrates the behavior of the Microprocessor Interface signals, during a "Motorola-Asynchronous" Write Operation.

FIGURE 9. ILLUSTRATION OF THE BEHAVIOR OF THE MICROPROCESSOR INTERFACE SIGNAL, DURING A "MOTOROLA-ASYNCHRONOUS" WRITE OPERATION.



2.3 Operating the Microprocessor Interface in the PowerPC 403 Mode

If the Microprocessor Interface has been configured to operate in the PowerPC 403 Mode, then the following Microprocessor Interface pins will assume the role that is described below in Table 1d.

TABLE 6: THE ROLES OF VARIOUS MICROPROCESSOR INTERFACE PINS, WHEN CONFIGURED TO OPERATE IN THE POWERPC 403 MODE

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
ALE/AS	A16	I	No Function - Tie to GND:
$\overline{RD}$/$\overline{DS}$/$\overline{WE}$	C15	I	**Write Enable Input - $\overline{WE}$** If the Microprocessor Interface is operating in the Power PC 403 Mode, then this input pin will function as the $\overline{WE}$ (Write Enable) input pin. Anytime the Microprocessor Interface samples this active-low input signal (along with $\overline{CS}$ and $\overline{WR}$/R/$\overline{W}$) also being asserted (at a logic low level) upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents on the Bi-Directional Data Bus (D[7:0]) into the "target" on-chip register or buffer location within the XRT79L71.
$\overline{RDY}$/$\overline{DTACK}$/RDY	C14	O	**Active High READY Output - RDY** If the Microprocessor Interface has been configured to operate in the Power PC 403 Mode, then this output pin will function as the "active-high" READY output. During a READ or WRITE cycle, the Microprocessor Interface block will toggle this output pin to the logic high level, ONLY when it (the Microprocessor Interface) is ready to complete or terminate the current READ or WRITE cycle. Once the Microprocessor has sampled this signal being at the logic "high" level (upon the rising edge of PCLK), then it is now safe for it to move on and execute the next READ or WRITE cycle. If (during a READ or WRITE cycle) the Microprocessor Interface block is holding this output pin at a logic "low" level, then the Microprocessor is expected to extend this READ or WRITE cycle, until it samples this output pin being at the logic low level. ***NOTE:*** *The Microprocessor Interface will update the state of this output pin upon the rising edge of µPCLK.*
µPCLK	H16	I	**Microprocessor Interface Clock Input:** This clock input signal is only used if the Microprocessor Interface has been configured to operate in one of the Synchronous Modes (e.g., Power PC 403 Mode). If the Microprocessor Interface is configured to operate in one of these modes, then it will use this clock signal to do the following. • To sample the $\overline{CS}$, $\overline{WR}$/R/$\overline{W}$, A[14:0], D[7:0], $\overline{RD}$/$\overline{DS}$ and DBEN input pins, and· • To update the state of the D[7:0] and the RDY/$\overline{DTACK}$ output signals. ***NOTE:*** *The Microprocessor Interface can work with µPCLK frequencies ranging up to 33MHz.*

TABLE 6: THE ROLES OF VARIOUS MICROPROCESSOR INTERFACE PINS, WHEN CONFIGURED TO OPERATE IN THE POWERPC 403 MODE

PIN NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
WR/R/W	B16	I	**Read/Write Operation Identification Input - R/W** If the Microprocessor Interface is configured to operate in the Power PC 403 Mode, then this input pin will function as the "Read/Write Operation Identification Input" pin. Anytime the Microprocessor Interface samples this input signal at a logic low (while also sampling the CS input pin "low") upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents of the Address Bus (A[14:0]) into the Microprocessor Interface circuitry, in preparation for this forthcoming READ operation. At some point (later in this READ operation) the Microprocessor will also assert the DBEN/OE input pin, and the Microprocessor Interface will then place the contents of the "target" register (or address location within the XRT79L71) upon the Bi-Directional Data Bus pins (D[7:0]), where it can be read by the Microprocessor. Anytime the Microprocessor Interface samples this input signal at a logic high (while also sampling the CS input pin a logic "low") upon the rising edge of µPCLK, then the Microprocessor Interface will (upon the very same rising edge of µPCLK) latch the contents of the Address Bus (A[14:0]) into the Microprocessor Interface circuitry, in preparation for the forthcoming WRITE operation. At some point (later in this WRITE operation) the Microprocessor will also assert the RD/DS/WE input pin, and the Microprocessor Interface will then latch the contents of the Bi-Directional Data Bus (D[7:0]) into the contents of the "target" register or buffer location (within the XRT79L71).
DBEN	J13	I	**Data Bus Enable Input:** For PowerPC Mode operation, the user should tie this pin to the OE output (from the MPC860/8260 Microprocessor, or similar pin). This input pin will be sampled upon the rising edge of µPCLK.
BLAST	B15	I	NONE - Tie this pin to GND

Configuring the Microprocessor Interface to operate in the PowerPC 403 Mode

Configure the Microprocessor Interface to operate in the Intel-Asynchronous Mode by tying the PTYPE2 and PTYPE0 pins/balls (e.g., Ball Numbers J14 and J16, respectively) to a logic "high", and by tying PTYPE 1 (Ball Number J15) to GND.

Finally, if the Microprocessor Interface has been configured to operate in the Intel-Asynchronous Mode, then it will perform READ and WRITE operations as described below.

2.3.1 The PowerPC 403 Read-Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the PowerPC 403 Mode, then the Microprocessor should do all of the following, anytime it wishes to read out the contents of a register or some location within the Receive LAPD Message buffer, the Receive Cell Extraction Memory or the Transmit Cell Extraction Memory (within the XRT79L71).

1. As the Microprocessor executes all of the following steps, it should designate this particular bus cycle as a READ Operation by making sure that the WR/R/W (R/W) input pin is held at a logic "high".
2. Place the address of the "target" register or buffer location (within the XRT79L71) on the Address Bus input pin, A[14:0].

NOTE: *As the Microprocessor places this address value, on the Address Bus, the user should make sure that the Microprocessor respects the "Address to Rising edge of* µPCLK *Set-up time" requirements.*

3. While the µC/µP is placing this address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{CS}$ (Chip Select) pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface block.

NOTE: As the Microprocessor/Address Decoding logic asserts the $\overline{CS}$ signal, the user should make sure that the Microprocessor/Address Decoding circuitry respects the "$\overline{CS}$ to Rising edge of µPCLK Set-up time" requirements.

4. At some time later, the Microprocessor should toggle the "$\overline{DBEN}$" ($\overline{OE}$) input pin "low". This step will enable the output drivers of the Bi-directional Data Bus pins (D[7:0]). Once the Microprocessor does this, (and once the Microprocessor Interface samples the "$\overline{OE}$" input pin being at a logic "low" upon a given rising edge of µPCLK) then the Microprocessor Interface (of the XRT79L71) will proceed to place the contents of the "target" address location (within the XRT79L71) onto the Bi-Directional Data Bus.

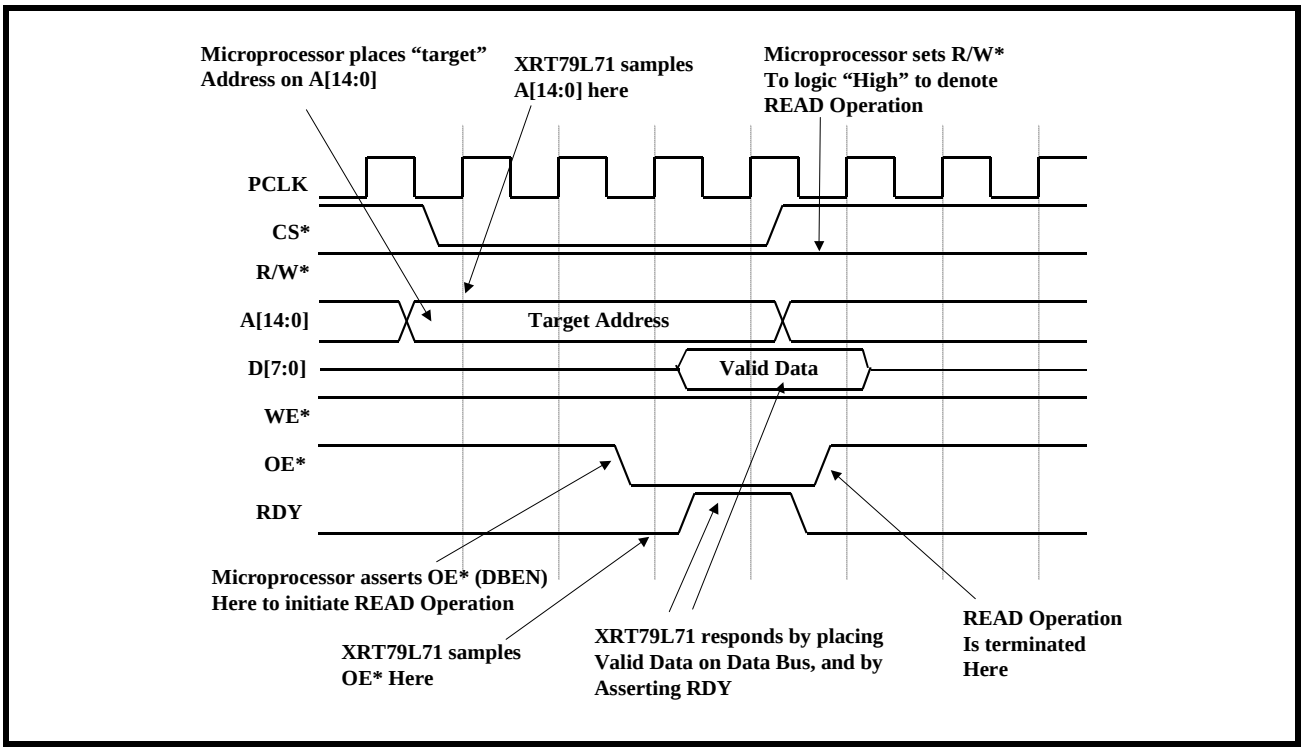
5. Immediately after the µC/µP toggles the "$\overline{DBEN}$" ($\overline{OE}$) input pin "low", the XRT79L71 will continue to drive the "$\overline{RDY}$/$\overline{DTACK}$/RDY output pin "low". The XRT79L71 does this in order to inform the µC/µP that the data (to be read from the data bus) is "NOT READY" to be latched into the µC/µP. In this, case the µC/µP should continue to hold the "$\overline{DBEN}$" input pin "low" until it samples the "$\overline{RDY}$/$\overline{DTACK}$/RDY" output pin being at a logic "high".

6. After some settling time, the data on the bi-directional data bus will stabilize and can be read by the µC/µP. The XRT79L71 will indicate that this data can be read by asserting the $\overline{RDY}$/$\overline{DTACK}$/RDY (READY) output signal (by toggling it "low"). NOTE: The Microprocessor Interface will update the state of the RDY signal upon the rising edge of µPCLK.

7. After the µC/µP detects the $\overline{RDY}$/$\overline{DTACK}$/RDY signal (from the XRT79L71) toggling "low" it can terminate the Read Cycle by toggling the "$\overline{DBEN}$" ($\overline{OE}$) input pin "high".

Figure 10 presents a timing diagram that illustrates the behavior of the Microprocessor Interface signals during a "PowerPC 403" Read Operation

FIGURE 10. ILLUSTRATION OF THE BEHAVIOR OF MICROPROCESSOR INTERFACE SIGNALS, DURING A "POWERPC 403" READ OPERATION



2.3.2 The PowerPC 403 Write-Cycle

If the Microprocessor Interface (of the XRT79L71) has been configured to operate in the PowerPC 403 Mode, then the Microprocessor should do all of the following, anytime it wishes to write a byte of data into a register or some location within the Transmit LAPD Message buffer, the Transmit Cell Insertion Memory or the Receive Cell Insertion Memory (within the XRT79L71).

1. Designate that this particular bus cycle is a WRITE operation by toggling the "$\overline{WR}$/R/$\overline{W}$" (R/$\overline{W}$) input pin "low".

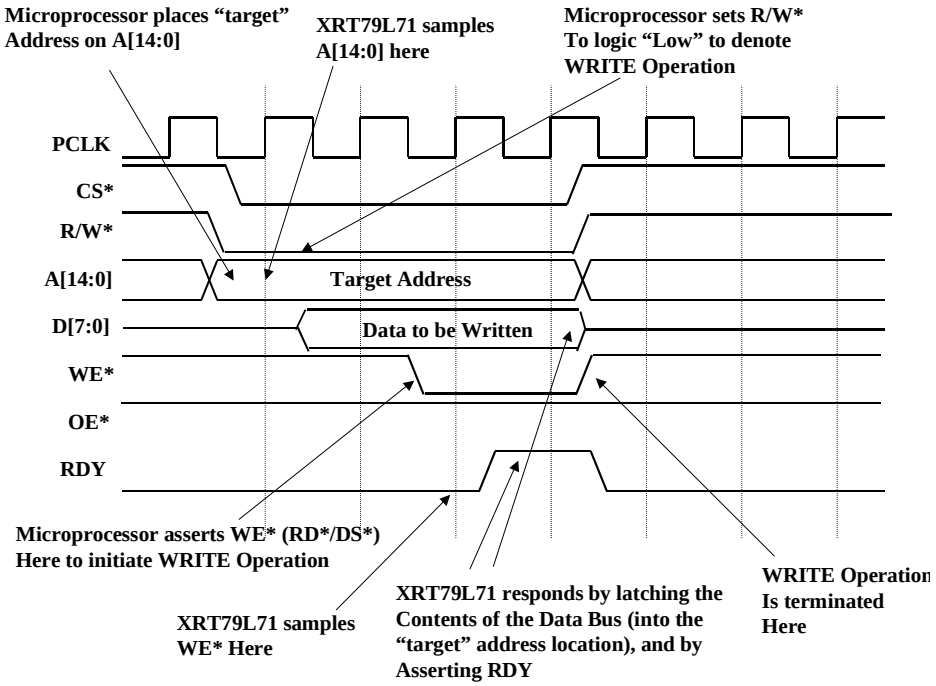
NOTE: *As the Microprocessor/Address Decoding logic asserts the $\overline{WR}$/R/$\overline{W}$ signal, the user should make sure that the Microprocessor/Address Decoding circuitry respects the "R/$\overline{W}$ to Rising edge of* µPCLK *Set-up time" requirements.*

2. Place the address of the "target" register or buffer location (within the XRT79L71) on the Address Bus input pins, A[14:0].

NOTE: *As the Microprocessor places this address value, on the Address Bus, the user should make sure that the Microprocessor respects the "Address to Rising edge of* µPCLK *Set-up time" requirements.*

3. While the µC/µP is placing the address value on the Address Bus, the Address Decoding circuitry (within the user's system) should assert the $\overline{CS}$ (Chip Select) input pin of the XRT79L71, by toggling it "low". This action enables further communication between the µC/µP and the XRT79L71 Microprocessor Interface. (NOTE: As the Microprocessor/Address Decoding logic asserts the $\overline{CS}$ signal, the user should make sure that the Microprocessor/Address Decoding circuitry respects the "$\overline{CS}$ to Rising edge of µPCLK Set-up time" requirements.)
4. The µC/µP should then place the byte or word that it intends to write into the "target" register, on the bi-directional data bus, D[7:0].
5. Next, the µC/µP should initiate the bus cycle by toggling the $\overline{RD}$/$\overline{DS}$/$\overline{WE}$ (Write Enable) input pin "low". When the XRT79L71 samples the $\overline{CS}$, $\overline{WR}$/R/$\overline{W}$, and the $\overline{WE}$ input pins being low (upon a given rising edge of µPCLK), then it will enable the "input drivers" of the bi-directional data bus, D[7:0].
6. Immediately after the µC/µP toggles the "$\overline{RD}$/$\overline{DS}$/$\overline{WE}$ (Write Enable) signal "low", the XRT79L71 will continue to drive the "$\overline{RDY}$/DTACK/RDY output pin "low". The XRT79L71 does this in order to inform the µC/µP that the data (to be written into the "target" address location, within the XRT79L71) is "NOT READY" to be latched into the µC/µP. In this case, the µC/µP should continue to hold the "Write Enable" input pin "low" until it samples the ""$\overline{RDY}$/$\overline{DTACK}$/RDY" output pin being at a logic "high".
7. After waiting the appropriate time (e.g., number of µPCLK periods), for the data (on the bi-directional data bus) to settle and can be safely accepted by the µC/µP. At this time, the XRT79L71 will indicate that this data can be latched into the "target" address location by toggling the "$\overline{RDY}$/$\overline{DTACK}$/RDY" output pin "high". (NOTE: The Microprocessor Interface will update the state of the "RDY" output pin upon the rising edge of µPCLK).

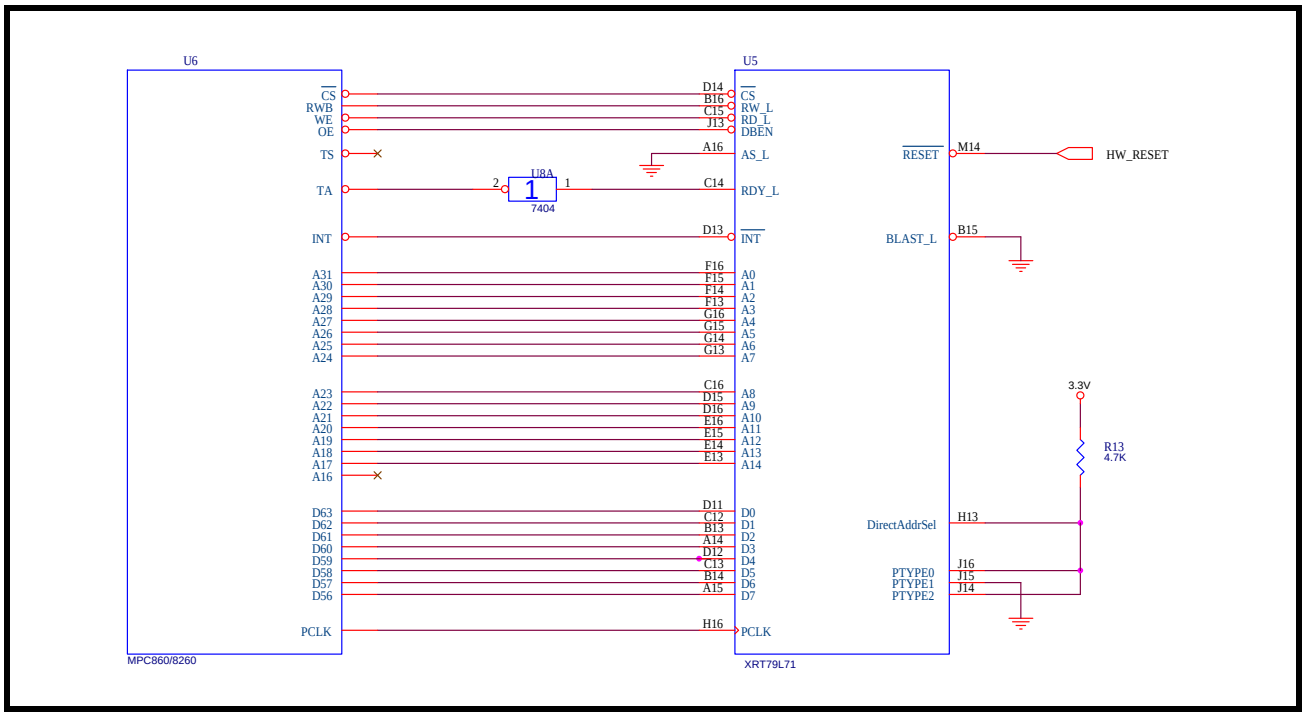
FIGURE 11. ILLUSTRATION OF THE BEHAVIOR OF MICROPROCESSOR INTERFACE SIGNALS, DURING A "POWERPC 403" WRITE OPERATION



2.3.3 Interfacing the Microprocessor Interface to the MPC860 Microprocessor, when configured to operate in the PowerPC 403 Mode

Figure 12 presents a schematic design on how we recommend that one interface the Microprocessor Interface (of the XRT79L71, when it is configured to operate in the PowerPC 403 Mode) to an MPC860/8260 type of Microprocessor.

FIGURE 12. ILLUSTRATION ON RECOMMENDATION ON HOW TO INTERFACE THE MICROPROCESSOR INTERFACE OF THE XRT79L71 TO THE MPC860, WHEN CONFIGURED TO OPERATE IN THE POWERPC 403 MODE



2.4 The need for TxInClk, in order to operate the Microprocessor Interface

In Revision A Silicon (which, at the time that this version of the Data Sheet was written, is the only revision of the XRT79L71 that exists), the user is required to supply a clock signal at the TxInClk in order to permit the Microprocessor Interface to function (e.g., support to READ and WRITE operations). The Microprocessor Interface (internally) uses the TxInClk signal to update the registers within the XRT79L71. If a clock signal is not applied to the TxInClk input pin, the Microprocessor Interface will NOT support READ or WRITE operations AT ALL.

If the user is having problems performing READ and WRITE operations with the XRT79L71, check and verify that a clock signal is present at the TxInClk input pin.

NOTES:

1. *The Microprocessor Interface needs for a clock signal to be present at the TxInClk input pin, even if the part has been configured to operate in the Loop-Timing Mode (and if the System-Side Terminal Equipment is using the RxOutClk as timing source for the 44.736MHz or 33.368MHz clock signal).*
2. *This requirement will NOT be true for Revision B (and beyond) silicon. In the case of future silicon, the user will need to make sure that the Receive DS3/E3 LIU Block is provided with either a Recovered Clock signal (from the incoming line signal) or a Reference Clock Signal (which is derived from either a 12.288MHz, 34.368MHz or a 44.736MHz clock signal, via the SFM Synthesizer Block). Please see Section 3.3.1.5 for more details on the SFM Synthesizer Block.*

2.5 Reading out the DS3/E3 Framer Block Performance Monitor Registers

The DS3/E3 Framer Block PMON (Performance Monitor) Registers (below) are 16-bit "RESET-upon-READ" registers. However, the manner in which these particular PMON Registers are to be read is listed below.

As mentioned earlier, these PMON Registers are 16-bits in length. More specifically each of these PMON Registers will consist of a "MSB" (Most Significant Byte) 8-bit register, and a "LSB" (Least Significant Byte) register. Since the Microprocessor Interface of the XRT79L71 contains an eight-bit wide bi-directional data bus, the user will have to execute two consecutive read operations in order to obtain the full 16-bit content of a given PMON register. As the contents of these PMON Registers are being read out, the following restrictions apply.

- During the first (of the two) read operations (to a given PMON Register), the user can read out either the "MSB" or the "LSB" Register.
- However, as the user executes this "first" read operation, the entire 16-bit contents of this particular PMON register will be cleared to "0x0000". The XRT79L71 will store the contents of the "un-read" register into the "PMON Holding Register (Address = 0x116C).
- Therefore, during the second (of the two) read operations (to a given PMON Register), the user MUST obtain the contents of the "un-read" byte, from the PMON Holding Register.
- This method for reading out the PMON Registers, applies to the following PMON Registers.
    - **a.** PMON Excessive Zero Count Registers
    - **b.** PMON Line Code Violation Count Registers
    - **c.** PMON Framing Bit/Byte Error Count Registers
    - **d.** PMON Parity/P-Bit Count Registers
    - **e.** PMON FEBE Event Count Registers
    - **f.** PMON CP-Bit Error Count Registers
    - **g.** PRBS Error Count Registers

3.0 INTERRUPT STRUCTURE WITHIN THE XRT79L71

The XRT79L71 is equipped with a sophisticated Interrupt Servicing Structure. This Interrupt Structure includes an Interrupt Request output pin ($\overline{INT}$), numerous Interrupt Enable Registers and numerous Interrupt Status Registers. The Interrupt Servicing Structure, within the XRT79L71 IC contains two levels of hierarchy. The top level is at the Functional Block level (e.g., the Receive ATM Cell Processor Block, the Receive PPP Packet Processor Block, the Receive DS3/E3 Framer block, etc). The lower hierarchical level is at the individual or source level. Each hierarchical level consists of a complete set of Interrupt Status Registers/bits and Interrupt Enable Registers/bits, as will be discussed below.

Most of the functional blocks within the XRT79L71 are capable of generating Interrupt Requests to the μC/µP. The XRT79L71 Interrupt Structure has been carefully designed to allow the user to quickly determine the exact source of the interrupt (with a minimum number of read operations, and, in-turn, minimal latency) which will aid the μC/µP in determine the appropriate interrupt service routine to call up in order to either eliminate, or properly respond to the condition(s) causing the interrupt.

Table 7 lists all of the possible conditions that can generate interrupts, within each functional block of the XRT79L71.

TABLE 7: LIST OF ALL OF THE POSSIBLE CONDITIONS THAT CAN GENERATE INTERRUPTS WITHIN THE XRT79L71 WHEN CONFIGURED TO OPERATE IN THE CLEAR-CHANNEL FRAMER MODE

FUNCTIONAL BLOCK	INTERRUPT CONDITION
DS3/E3 Framer Block (Consists of both the Transmit DS3/E3 Framer Block, the Transmit PLCP Processor Block, the Receive DS3/E3 Framer Block and the Receive PLCP Processor Block)	**Transmit DS3/E3 Framer Block Interrupts** • Completion of Transmission of FEAC Message (DS3,C-bit Parity Only) • Completion of Transmission of LAPD/PMDL Message **Transmit PLCP Processor Block Interrupts** • None **Receive DS3/E3 Framer Block Interrupts** • Change of LOS (Loss of Signal) Defect Condition • Change of OOF (Out of Frame) Defect Condition • Change of AIS Defect Condition • Change in Trail-Trace Buffer Message (E3, ITU-T G.832 only) • Change of FERF/RDI (Yellow Alarm) Defect Condition • Detection of P-Bit Errors (DS3 Applications only) • Detection of CP-Bit Errors (DS3, C-bit Parity Applications only)· Detection of BIP-4 Error (E3, ITU-T G.751 only) • Detection of BIP-8 Error (E3, ITU-T G.832 only)· Detection of FEBE (Far-End Block Error) Event • Validation of FEAC Message (DS3, C-bit Parity Only)· • Removal of FEAC Message (DS3, C-bit Parity Only) • Receipt of New LAPD/PMDL Message • One Second Interrupt **Receive PLCP Processor Block Interrupts** • Change of PLCP OOF Defect Condition • Change of PLCP LOF Defect Condition • Change of PLCP RAI Defect Condition
DS3/E3 LIU Block	• Change of FL (Jitter Attenuator FIFO Limit Alarm) Condition • Change of LOL (Receive Loss of Lock) Condition • Change of LOS (Loss of Signal) Condition • Change of DMO (Transmit Drive Monitor Output) Condition

TABLE 7: LIST OF ALL OF THE POSSIBLE CONDITIONS THAT CAN GENERATE INTERRUPTS WITHIN THE XRT79L71 WHEN CONFIGURED TO OPERATE IN THE CLEAR-CHANNEL FRAMER MODE

FUNCTIONAL BLOCK	INTERRUPT CONDITION
Receive ATM Cell/PPP Packet Processor Block	**Receive ATM Cell Processor Block Interrupts** • Receive Cell Extraction Event • Receive Cell Insertion Event • Receive Cell Insertion Memory Overflow Condition • Receive Cell Extraction Memory Overflow Condition • RxFIFO Overflow • Detection of Correctable HEC Byte Error • Detection of Uncorrectable HEC Byte Error • Declaration of the LCD (Loss of Cell Delineation) Defect Condition • Clearances of the LCD Defect Condition **Receive PPP Packet Processor Block Interrupts** • Detection of Receive FIFO Overflow Condition • Detection of FCS (Frame Check Sequence) Error· • Detection of ABORT Sequence • Detection of RUNT Packet
Transmit ATM Cell/PPP Packet Processor Block	**Transmit ATM Cell Processor Block Interrupts** • Transmit Cell Extraction Event • Transmit Cell Insertion Event· • Transmit Cell Insertion Memory Overflow Condition • Transmit Cell Extraction Memory Overflow Condition • TxFIFO Overflow • Detection of HEC Byte Error • Detection of Transmit UTOPIA Parity Error • Transmit PPP Processor Block Interrupts • Detection of Transmit FIFO Overflow Condition· • Detection of Transmit POS-PHY Interface Parity Error

The XRT79L71 ATM UNI/PPP/Clear-Channel DS3/E3 Framer device comes equipped with the following registers to support the servicing of this wide array of potential interrupt request sources. Table 8 lists these registers and their corresponding addresses, within the XRT79L71.

TABLE 8: A LISTING OF THE XRT79L71 ATM UNI/PPP/CLEAR-CHANNEL DS3/E3 FRAMER DEVICE INTERRUPT BLOCK REGISTERS - CLEAR-CHANNEL FRAMER APPLICATIONS

REGISTER	ADDRESS LOCATION
Operation Control Register	0x0101
Operation Interrupt Status Register - Byte 1	0x0112
Operation Interrupt Status Register - Byte 0	0x0113
Operation Interrupt Enable Register - Byte 1	0x0116
Operation Interrupt Enable Register - Byte 0	0x0117

TABLE 8: A LISTING OF THE XRT79L71 ATM UNI/PPP/CLEAR-CHANNEL DS3/E3 FRAMER DEVICE INTERRUPT BLOCK REGISTERS - CLEAR-CHANNEL FRAMER APPLICATIONS

REGISTER	ADDRESS LOCATION
Framer Block Interrupt Enable Register	0x1104
Framer Block Interrupt Status Register	0x1105
RxDS3 Interrupt Enable RegisterRxE3 Interrupt Enable Register # 1 - ITU-T G.751RxE3 Interrupt Enable Register # 1 - ITU-T G.832	0x1112
RxDS3 Interrupt Status RegisterRxE3 Interrupt Enable Register # 2 - ITU-T G.751RxE3 Interrupt Enable Register # 2 - ITU-T G.832	0x1113
RxE3 Interrupt Status Register # 1 - ITU-T G.751RxE3 Interrupt Status Register # 1 - ITU-T G.832	0x1114
RxE3 Interrupt Status Register # 2 - ITU-T G.751RxE3 Interrupt Status Register # 2 - ITU-T G.832	0x1115
RxDS3 FEAC Interrupt Enable/Status Register	0x1117
RxDS3/E3 LAPD Control Register	0x1118
Transmit DS3 FEAC Configuration & Status Register	0x1131
Transmit DS3/E3 LAPD Status/Interrupt Register	0x1134
RxPLCP Interrupt Enable Register	0x1191
RxPLCP Interrupt Status Register	0x1192
LIU Interrupt Enable Register	0x1301
LIU Interrupt Status Register	0x1302
Receive ATM Cell Processor Block - Receive ATM Interrupt Status Register - Byte 1	0x170A
Receive ATM Cell Processor Block - Receive ATM Interrupt Status Register - Byte 0	0x170B
Receive ATM Cell Processor Block - Receive ATM Interrupt Enable Register - Byte 1	0x170E
Receive ATM Cell Processor Block - Receive ATM Interrupt Enable Register - Byte 0	0x170F
Transmit ATM Cell Processor Block - Transmit ATM Interrupt Status Register	0x1F0B
Transmit ATM Cell Processor Block - Transmit ATM Interrupt Enable Register	0x1F0F

3.1 General Flow of XRT79L71 ATM UNI/PPP/Clear-Channel DS3/E3 Framer Device Interrupt Servicing

Whenever any of the conditions, presented in **Table 7** occur (if their Interrupt is enabled), then the XRT79L71 will generate an interrupt request to the µC/µP by asserting the active-low interrupt request output pin, $\overline{INT}$. Shortly after the µC/µP has detected the activated $\overline{INT}$ signal, it will enter into the appropriate user-supplied interrupt service routine. The first task, for the µC/µP, while running this interrupt service routine, may be to isolate the source of the interrupt request down to the device level (e.g., the XRT79L71 ATM UNI/PPP/Clear-Channel DS3/E3 Framer device), if multiple peripheral devices exist in the user's system.

However, once the interrupting peripheral device has been identified and determined to be the XRT79L71, the next task for the µC/µP is to identify the functional block (within the XRT79L71) that requested the interrupt. Finally, the µC/µP will need to proceed further and identify the exact condition(s) causing the interrupt to be generated by the XRT79L71.

The procedure for servicing the XRT79L71 Interrupts is best achieved by executing the following steps.

STEP 1 - Determine the Functional Block(s) requesting the Interrupt

If the interrupting device turns out to be the XRT79L71 ATM UNI/PPP/Clear-Channel DS3/E3 Framer IC, then the µC/µP must determine which functional block requested the interrupt. Hence, upon reaching this state, one of the very first things that the µC/µP must do within the user supplied XRT79L71 Interrupt Service Routine, is to perform a read of both of the following registers.

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

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

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

Operation Interrupt Status Register - Byte 0 (Address = 0x0113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Receive ATM Cell/ PPP Processor 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

NOTE: *For Clear-Channel Framer Applications, it is not necessary to read out the contents of the "Operation Interrupt Status Register - Byte 0".*

Each of the Operation Block Interrupt Status Register presents the interrupt-request status of each of the functional blocks within the chip. The purpose of these two registers is to help the µC/µP identify which functional block(s) has requested the interrupt. Whichever bit(s) are asserted, in this register, identifies which block(s) has requested the interrupt. Whichever bit(s) are asserted, in this register, identifies which block(s) have experienced an interrupt-generating condition as presented in **Table 7**. Once the µC/µP has read this register, it can determine which branch within the interrupt service routine that it must follow in order to properly service this interrupt.

The XRT79L71 ATM UNI/PPP/Clear-Channel DS3/E3 Framer IC further supports the Operational Block hierarchy by providing the Operation Block Interrupt Enable Register - Bytes 1 and 0. The bit format of these two registers are identical to that for the Operation Block Interrupt Status Registers - Byte 1 and 0, and are presented below for the sake of completeness.

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 Inter-rupt 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

Operation Interrupt Enable Register - Byte 0 (Address = 0x0117)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Receive ATM Cell/ PPP Proces-sor BlockIn-terrupt Enable	Unused			Transmit ATM Cell/ PPP Proces-sor Block Interrupt Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	0

These Operation Block Interrupt Enable registers permit the user to individually enable or disable the interrupt requesting capability of the functional blocks within the XRT79L71. If a particular bit-field, within this register contains the value "0", then the corresponding functional block has been disabled for generating any interrupt requests. Conversely, if that bit-field contains the value "1", then the corresponding functional block has been enabled for interrupt generation (e.g., those potential interrupts, within the enabled functional block that are enabled at the source level are now enabled). The user should be aware of the fact that each functional block, within the XRT79L71 contains multiple potential interrupt sources. Each of these lower lever interrupt sources contain their own set of interrupt enable bits and interrupt status bits, existing in various on-chip registers.

STEP 2 - Interrupt Service Routing Branching: After reading the Operation Block Interrupt Status Registers

The contents of the Operation Block Interrupt Status Registers permit the user to identify which of the four (4)functional blocks (within the XRT79L71 IC) have requested interrupt service. The µC/µP should use this information in order to determine where, within the Interrupt Service Routine, program control should branch to. The following table can be viewed as an interrupt service routine guide. It lists each of the Functional Blocks that contain bit-fields in the Operation Block Interrupt Status and Enable registers. Additionally, this table also presents a list and addresses of the corresponding on-chip Registers that the Interrupt Service Routine should branch to and read, based upon the Interrupting Functional Block.

TABLE 9: INTERRUPT SERVICE ROUTINE GUIDE FOR THE XRT79L71

INTERRUPT FUNCTIONAL BLOCK	THE NEXT REGISTER TO BE READ DURING THE INTERRUPT SERVICE ROUTINE	ADDRESS LOCATION
DS3/E3 Framer Block	Framer Block Interrupt Status Register	0x1105
DS3/E3 LIU/JA Block	LIU Interrupt Status Register	0x1302

TABLE 9: INTERRUPT SERVICE ROUTINE GUIDE FOR THE XRT79L71

INTERRUPT FUNCTIONAL BLOCK	THE NEXT REGISTER TO BE READ DURING THE INTERRUPT SERVICE ROUTINE	ADDRESS LOCATION
Receive ATM Cell/PPP Packet Processor Block	Receive ATM Cell Processor Block - Receive ATM Interrupt Status Register - Byte 1	0x170A
	Receive ATM Cell Processor Block - Receive ATM Interrupt Status Register - Byte 0	0x170B
Transmit ATM Cell/PPP Packet Processor Block	Transmit ATM Cell Processor Block - Transmit ATM Interrupt Status Register - Byte 1	0x1F0B

NOTES:

1. *Registers associated within each functional block are specified in ascending order (based upon the on-chip Address Location). No other inferences should be made regarding the order in which these registers are presented in this table.*
2. *2.For Clear-Channel Framer Applications, the Receive ATM Cell/PPP Packet Processor and Transmit ATM Cell/ PPP Packet Processor Blocks will be disabled and do not need to be included as a part of the Interrupt Service Routine*

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

3.2 Interrupt Servicing for the DS3/E3 Framer Block

If, upon reading out the contents of the Operation Interrupt Status Registers that the interrupting block is the DS3/E3 Framer block then the user should execute a READ operation to the Framer Block Interrupt Status Register. The bit-format for the Framer Block Interrupt Status Register is as presented below.

Framer Block Interrupt Status Register (Address = 0x1105)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Status	Unused					Transmit DS3/E3 Framer Block Interrupt Status	One Second Interrupt Status
R/O	R/O	R/O	R/O	R/O	R/O	R/O	RUR
X	0	0	0	0	0	X	X

Depending upon which of these bit-fields are set to "1", the µC/µP (while executing the Interrupt Service Routine) will need to branch to reading out the appropriate register as depicted below in **Table 10**.

TABLE 10: INTERRUPT SERVICE ROUTINE GUIDE FOR THE DS3/E3 FRAMER BLOCK

BIT NUMBER ASSERTED IN REGISTER	BIT-FIELD NAME	NEXT REGISTER TO READ WITHIN INTERRUPT SERVICE ROUTINE	ADDRESS LOCATION OF NEXT REGISTER
7	Receive DS3/E3 Framer Block Interrupt Status	RxDS3 Interrupt Status Register	0x1113
		RxE3 Interrupt Status Register # 1	0x1114
		RxE3 Interrupt Status Register # 2	0x1115
		RxDS3 FEAC Interrupt Enable/Status Register	0x1117
		RxDS3/E3 LAPD Control Register	0x1118
6	Receive PLCP Processor Block Interrupt Status	RxPLCP Interrupt Status Register	0x1192
1	Transmit DS3/E3 Framer Block Interrupt Status	Transmit DS3 FEAC Configuration & Status Register	0x1131
		Transmit DS3/E3 LAPD Status/Interrupt Register	0x1134
0	One Second Interrupt Status	NONE - Cause of Interrupt was One Second Interrupt	****

A more detailed description of each of these registers will be presented in the corresponding sections describing the Receive DS3/E3 Framer and Transmit DS3/E3 Framer Blocks.

4.0 ARCHITECTURAL/FUNCTIONAL DESCRIPTION OF THE XRT79L71 1-CHANNEL DS3/E3 ATM UNI/ PPP/CLEAR-CHANNEL FRAMER WITH LIU IC - CLEAR CHANNEL FRAMER AND HIGH-SPEED HDLC CONTROLLER MODE APPLICATIONS

This particular document describes how to operate the XRT79L71 in both the DS3/E3 Clear-Channel Framer Mode. If the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, then its functional block diagram will be as depicted below in Figure 13.

Figure 13. An Illustration of the Functional Block Diagram of the XRT79L71 when it has been configured to operate in the DS3 Clear-Channel Framer Mode

Figure 13 indicates that whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, then all of the following blocks will become active.

- The Transmit Payload Data Input Interface Block
- The Transmit Overhead Data Input Interface Block
- The Transmit FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Transmit LAPD Controller Block (DS3, C-bit Parity Applications Only)
- The Transmit DS3/E3 Framer Block
- The Transmit DS3/E3 LIU Block
- The Receive DS3/E3 LIU Block
- The Receive DS3/E3 Framer Block
- The Receive LAPD Controller Block (DS3, C-bit Parity Applications Only)
- The Receive FEAC Controller Block (DS3, C-bit Parity Applications Only)
- The Receive Overhead Data Output Interface Block
- The Receive Payload Data Output Interface Block

The XRT79L71 can be configured to support any of the following framing formats.

- DS3, C-bit Parity
- DS3, M13 (also referred to as M23)
- E3, ITU-T G.751
- E3, ITU-T G.832

This document describes, (1) how to configure the XRT79L71 into each of these framing formats and (2) how to use each of the features within the XRT79L71. This section discusses "DS3 Clear-Channel Framer" Applications; where Section 5.0 and Section 6.0 discusses "E3, ITU-T G.751 Clear-Channel Framer" and "E3, ITU-T G.832 Clear-Channel Framer" Applications, respectively.

Prior to going into the details of Clear-Channel Framer operations of the XRT79L71, the user is advised that the XRT79L71 can also be configured to support ATM over DS3/E3 Applications, PPP over DS3/E3 Applications, and finally High-Speed HDLC over DS3/E3 Applications. The Architectural Description for each of these documents can be obtained separately and are listed below, by title.

- XRT79L71ATM - Architectural Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer and LIU IC - ATM UNI Applications
- XRT79L71PPP - Architectural Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer and LIU IC - PPP Applications

- XRT79L71_HDLC - Architectural Description of the XRT79L71 1-Channel DS3/E3 ATM UNI/PPP/Clear-Channel Framer and LIU IC - High-Speed HDLC Controller Applications

4.1 DESCRIPTION OF THE DS3 FRAME STRUCTURE AND OVERHEAD BITS

Prior to discussing the architecture and the role of the DS3/E3 Framer blocks, whenever the XRT79L71 has been configured to operate in the DS3 Mode, it is important to discuss the DS3 Frame structure.

The role of the various OH (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 84-bit payload packets. The Transmit DS3/E3 Framer and Receive DS3/E3 Framer blocks, within the XRT79L71, supports the following two DS3 framing formats:

- C-bit Parity
- M13/M23

Figure 14 and Figure 15 present the DS3 Frame Format for C-bit Parity and M13/M23, respectively.

FIGURE 14. DS3 FRAME FORMAT FOR C-BIT PARITY

X	I	F1	I	AIC	I	F0	I	NA	I	F0	I	FEAC	I	F1	I
X	I	F1	I	UDL	I	F0	I	UDL	I	F0	I	UDL	I	F1	I
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 = multi-frame 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 = C-bit parity

FEBE = far end block error

UDL = User Data Link

FIGURE 15. DS3 FRAME FORMAT FOR M13/M23

X	I	F1	I	C11	I	F0	I	C12	I	F0	I	C13	I	F1	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**

Configuring the XRT79L71 to operate in the DS3 Clear-Channel Framer Mode

The XRT79L71 can be configured to operate in the DS3 Clear-Channel Framer Mode, by executing the following three steps.

STEP 1 - Configure the XRT79L71 to operate in the Clear-Channel Framer/HDLC Controller Mode.

This can be accomplished by setting Bit 0 (Configuration Control), within the Operation Control Register - Byte 3 to "1" as depicted below.

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

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Configura-tion Control
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	1

This step configures the XRT79L71 to operate in either the Clear-Channel Framer or in the "High-Speed HDLC Controller" Modes.

STEP 2 - Configure the XRT79L71 to operate in the Clear-Channel Framer Mode

The user can accomplish this by setting Bit 6 (HDLC Controller Enable), within the Payload HDLC Control Register, to "0" as depicted below.

Payload HDLC Control Register, Address = 0x1182

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer By-Pass	HDLC Controller Enable	HDLC CRC-32	Unused	HDLC Loop-back	Unused		
R/W	R/W	R/W	R/O	R/W	R/O	R/O	R/O
0	0	0	0	0	0	0	0

STEP 3 - Configure the XRT79L71 to operate in the DS3 Mode

The user can choose between these two DS3 framing formats, by writing the appropriate data into Bits 2 (Frame Format) and 6 (IsDS3) within the Framer Operating Mode Register (Address = 0x1100), as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

The following table lists the relationship between the value of these bit-fields and the resulting framing format for the XRT79L71.

TABLE 11: THE RELATIONSHIP BETWEEN THE CONTENTS OF BITS 2 (FRAME FORMAT) AND 6 (ISDS3) WITHIN THE FRAMER OPERATING MODE REGISTER, AND THE RESULTING FRAMING FORMAT

BIT 6 (IS DS3)	BIT 2 (FRAME FORMAT)	RESULTING FRAMING FORMAT OF XRT79L71
0	0	E3, ITU-T G.751
0	1	E3, ITU-T G.832
1	0	DS3, C-bit Parity
1	1	DS3, M13/M23

The XRT79L71 will be operating in the E3, ITU-T G.751 Framing format upon power-up or hardware reset. Therefore, the user must write a "1" into both Bits 2 and 6 within this particular register in order to configure the XRT79L71 to operate in the DS3, M13/M23 Mode.

NOTE: *These bit setting configures the framing format for both the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block.*

Each of the two DS3 Frame Formats, as presented in **Figure 14** and **Figure 15**, constitute an M-frame or a full DS3 frame. Each M-frame consists of 7 680-bit F-frames, sometimes referred to as sub-frames. In **Figure 14** and **Figure 15** 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/M23 and C-bit Parity Frame Formats

The frame formats for M13/M23 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/M23 Framing format, the C-bit reflect the status of stuff-opportunities that either were or were not used while multiplexing the 7 DS2 signals into this DS3 signal. If at least two of the three stuff bits, within an F-frame are set to "1", then the corresponding Si bit (not shown in Figure 15) is designated as being a stuff bit. Conversely, if at least two of the three C-bits within an F-frame are set to "0", then the corresponding Si is not interpreted as being a stuff bit. In the C-bit Parity framing format, the C-bits take on different roles as is presented below in Table 12.

TABLE 12: C-BIT FUNCTIONS FOR THE DS3, C-BIT PARITY FRAMING 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	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	Path Maintenance Data Link
C61, C62, C63, C71, C72, C73	User Data Link (undefined for DS3 Frame)

Definition of the DS3 Frame Overhead Bits

In general, the DS3 Frame Overhead (OH) bits serve the following three purposes.

1. To support Frame Synchronization between the Near-End and remote DS3 Terminals
2. To provide parity bits in order to facilitate performance monitoring and error detection within the DS3 data-stream.
3. To support the transmission of Alarms, Status, and Data Link information between the Near-End and the remote DS3 Terminals.

The DS3 Overhead bits supporting each of these purposes are further defined below.

Frame Synchronization Bits (Applies to both M13/M23 and C-bit Parity Framing Formats)

Each DS3 Frame or M-frame contains a total of 31 bits that support frame synchronization. Each DS3 M-frame contains three (3) M-bits. According to Figure 14 and Figure 15, these M-bits reside within the very first bit-fields within F-frames # 5, 6 and 7. These three bits appear in each M-frame with the repeating pattern of [0, 1, 0]. This fact is also presented in Figure 14 and Figure 15, in which these particular bit-fields are designated as M0, M1 and M0, where M0 is an M-bit that is set to "0" and M1 is an M-bit that is set to "1".

Each of the seven (7) F-frames, within a DS3 M-frame contains four (4) F-bits, which also aid in synchronization between the Near-End and remote DS3 terminals. Therefore, each DS3 M-frame consists of a total of 28 F-bits. These F-bits exhibit a repeating pattern of [1, 0, 0, 1] within each F-frame. This fact is also presented in Figure 14 and Figure 15 in which these particular bit-fields are designated as F0 and F1, where F0 is an F-bit that is set to the value "0", and F1 is an F-bit that is set to the value "1".

Each of these bit-fields is used by the Receive DS3/E3 Framer block to perform Frame Acquisition and Frame Maintenance functions. For more information on how the Receive DS3/E3 Framer block uses these bit-fields, please see Section 4.3.2.2.

Performance Monitoring Bits (Parity)

The DS3 Frame includes numerous bits that are used to support performance monitoring of a DS3 signal, as it is transmitted from one DS3 terminal to another over coaxial cable or optical fiber, typically when mapped into a SONET/SDH signal. Depending upon the framing format chosen, each DS3 frame contains at least one type of parity bit and may contain two types of parity bits. P-bits are available in both the M13/M23 and C-bit Parity Framing formats. However, the C-bit Parity Framing format also includes three additional CP or Path Parity bits. Each of these types of parity bits are described in some detail below.

P-Bits (Applies to M13/M23 and C-bit Parity Framing 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. As a Transmitting DS3 Terminal assembles a DS3 frame, prior to transmitting this DS3 signal to the remote terminal equipment, it computes the EVEN parity over all 4704 payload bits within a given DS3 frame. The Transmitting DS3 Terminal then inserts the resulting parity information into the two P-bit fields within the very next outbound DS3 frame. These two P-bits are set to "1" if the payload data within the previous DS3 frame consists of an odd number of "1s". Conversely, the two P-bits are set to "0" if the payload data within the previous DS3 frame consists of an even number of "1s".

As a Receiving DS3 Terminal receives a given incoming DS3 frame, it locally computes its value for the P-bits. Afterwards, this Receiving DS3 Terminal compares its locally-computed P-bits with the value within the P-bit positions within the very next incoming DS3 frame. If these two P-bit values match, then the Receiving DS3 Terminal presumes that the first of these two incoming DS3 frames was received in an error-free manner. If these two P-bit values DO NOT match, then the Receiving DS3 Terminal presumes that the first of these two DS3 frames was received in an erred manner.

For information on how the Receive DS3/E3 Framer block handles P-bits within the DS3 data-stream that it receives, please see Section 4.3.2.7.

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

Each DS3 M-frame consists of three (3) CP-bits. Just like the P-bits these three CP-bits carry the information of the previous DS3 frame for performance monitoring. These three CP-bits must be identical, within a given DS3 frame. As a Transmitting DS3 Terminal assembles a DS3 frame, prior to transmitting this DS3 signal to the remote terminal equipment, it computes the EVEN parity over all 4704 payload bits within a given DS3 frame. The Transmitting DS3 Terminal then inserts the resulting parity information into the three CP-bit fields within the very next outbound DS3 frame. These three CP-bits have a role that is very similar to that of P-bits. However, in 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 terminal equipment to an adjacent piece of terminal equipment (e.g., a single DS3 link between two terminals).
- CP-bits, which are often times referred to as DS3 Path Parity bits, are used to support error detection within a DS3 data-stream, as it travels from the Source Terminal equipment where this particular DS3 data-stream originated, to the Sink Terminal equipment where this particular DS3 data-stream is terminated. As a consequence, once these CP-bits are computed and inserted into the outbound DS3 data-stream at the Source Terminal Equipment, these CP-bits cannot be altered by any Mid-Network equipment until this DS3 data-stream is terminated at the Sink Terminal Equipment. In contrast, the values of P-bits within this same DS3 data-stream can be re-computed and potentially changed by any Mid-Network equipment.

How CP-bits are Processed throughout the Network

The following section describes how CP-bits are processed at three locations within the network.

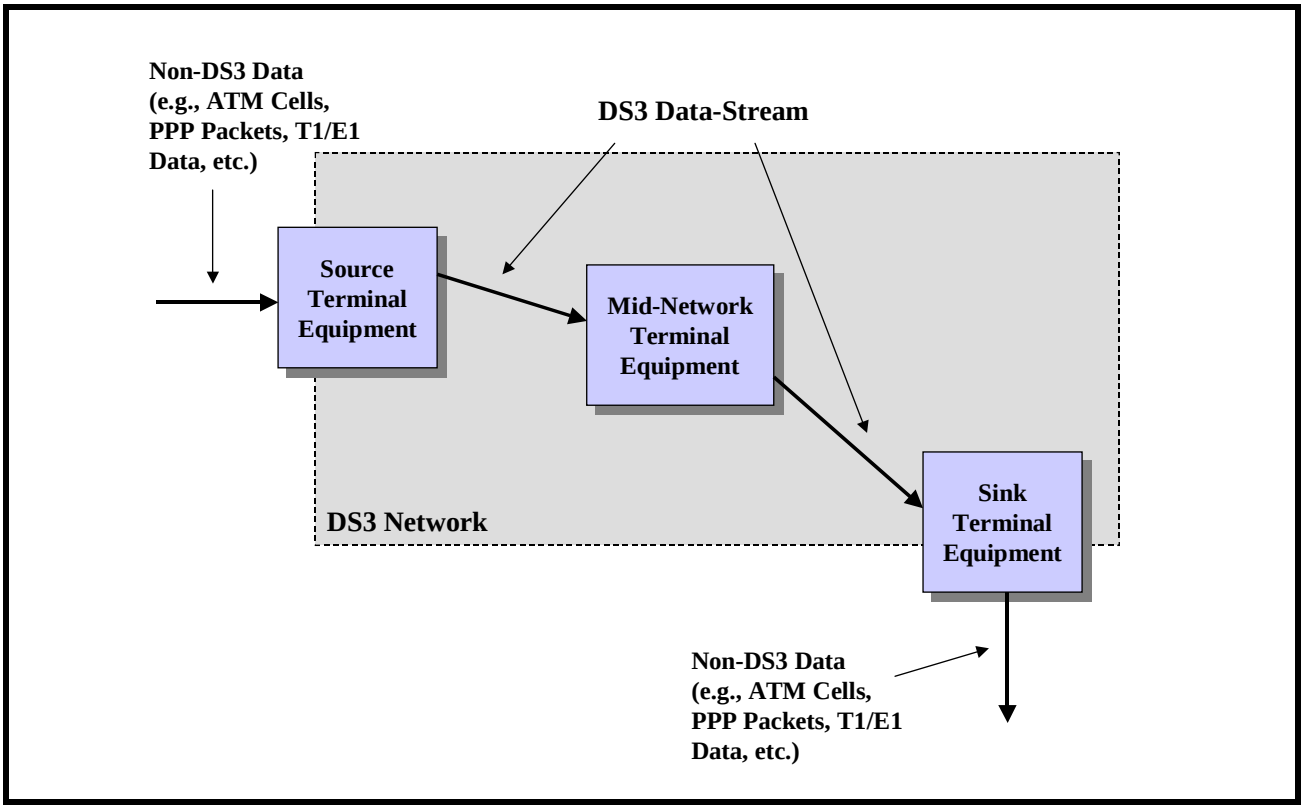
- The Source Terminal Equipment
- The Mid-Network Terminal Equipment
- The Sink Terminal Equipment

NOTES:

1. *Examples of Source and Sink Terminal Equipment would be DS3 ATM UNIs, DS3 POS-PHY Interfaces and M13 Multiplexers/De-Multiplexers.*
2. *This transmission path from the Source Terminal Equipment to the Sink Terminal Equipment may involve the transmission through numerous network elements such as DS3 repeaters and SONET/SDH Mappers, etc.*

Figure 16 presents a simple illustration of a DS3 network that consists of these three types of Terminal Equipment.

FIGURE 16. A SIMPLE ILLUSTRATION OF THE LOCATIONS OF THE SOURCE, MID-NETWORK AND SINK TERMINAL EQUIPMENT (FOR CP-BIT PROCESSING)



Processing at the Source Terminal Equipment

The purpose of the Source Terminal Equipment is to accept some non-DS3 data from some entity (e.g., ATM cells, PPP packets, DS1 or E1 signals, etc.) and to map this data into payload bits within an outbound DS3 data-stream. As the Source Terminal Equipment constructs each outbound DS3 data-stream, it computes the EVEN parity value over all of the payload bits, within this outbound DS3 data-stream. This resulting parity value will be inserted into the two (2) P-bit fields and the three (3) CP-bit-fields within the very next outbound DS3 frame.

Hence, both the P-bit and CP-bit values within a given DS3 frame will originate from the Source Terminal Equipment.

Processing at the Mid-Network Terminal Equipment

The Mid-Network Terminal Equipment has the task of doing the following.

- Receiving and doing performance monitoring on a given DS3 data-stream from some entity or piece of equipment (e.g., another remote terminal equipment, SONET/SDH Mapper, a DS3 Source Terminal Equipment, etc.)
- Transmitting this same DS3 data-stream out to some other entity or piece of equipment.

Figure 17 presents a simple illustration of the basic architecture of a Mid-Network Terminal Equipment.

FIGURE 17. A SIMPLE ILLUSTRATION OF AN EXAMPLE OF A MID-NETWORK TERMINAL EQUIPMENT

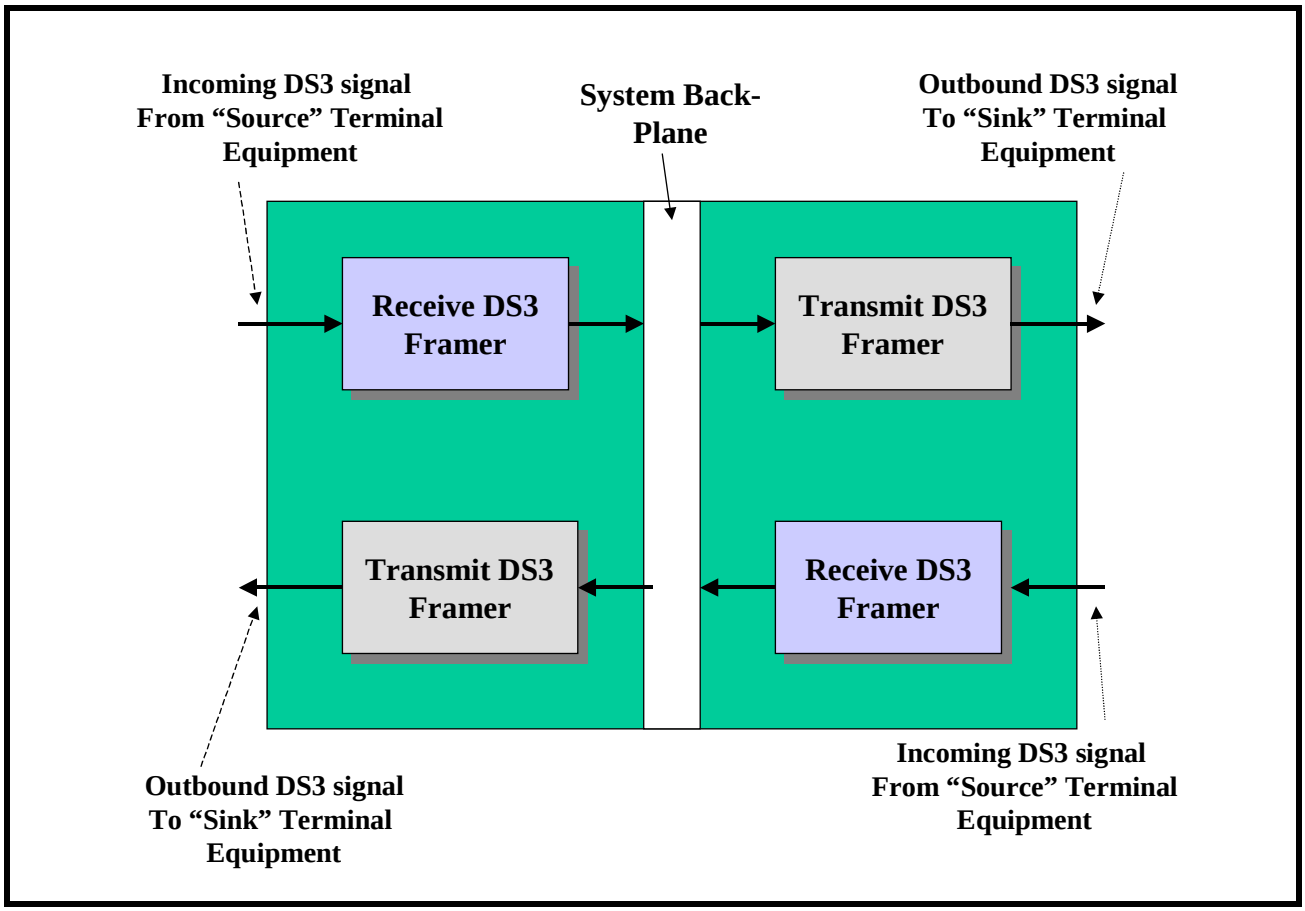


Figure 17 presents an illustration of two DS3 Boards that are connected to each other via a System Back-Plane. Each of these DS3 Boards contains a Receive DS3 Framer and a Transmit DS3 Framer entity. In this system, a given Receive DS3 Framer will receive and do performance monitoring (e.g., compute and verify both the P and CP bits) on a given DS3 signal that it has received from a given Source Terminal Equipment. After this incoming DS3 signal has passed through the Receive DS3 Framer it will then pass through the System Back-plane and will arrive at the Transmit DS3 Framer on the other board. The Transmit DS3 Framer will accept this DS3 data-stream and will process this DS3 data-stream by re-computing and inserting the DS3 overhead bits within DS3 data-stream.

NOTE: *As this Transmit DS3 Framer re-computes and inserts the DS3 overhead bits back into this outbound DS3 data-stream, it is imperative that this Transmit DS3 Framer NOT re-compute or potentially alter the state of the CP-bits. Once the CP-bits are computed and inserted into an outbound DS3 data-stream, by the Source Terminal Equipment, these CP-bit values must be retained as the DS3 data-stream traverses the network until it reaches the Sink Terminal Equipment.*

After this DS3 data-stream has passed through the Transmit DS3 Framer it will then be output to a remote Sink Terminal Equipment.

In summary, the role of the Receive DS3 Framer within the Mid-Network Terminal is listed below.

- To compute and verify the P-bits of each incoming DS3 frame
- To compute and verify the CP-bits of each incoming DS3 frame
- To output both payload and overhead bits to the System Back-Plane for transmission to the other board within the Mid-Network Terminal Equipment.

Further, in summary, the role of the Transmit DS3 Framer within the Mid-Network Terminal is listed below.

- To accept both the payload and overhead bits from the other board within the Mid-Network Terminal Equipment via the System Back-Plane
- To extract out the CP-bit values within this DS3 data-stream that it accepts from the System Back-plane and to re-insert these values into the CP-bit fields, within the outbound DS3 data-stream
- To compute the EVEN parity over all payload bits, within a given outbound DS3 frame, and insert this value into the P-bit positions within the very next outbound DS3 frame
- To transmit this DS3 data-stream to the remote terminal equipment.

SOME REQUIREMENTS ASSOCIATED WITH THE MID-NETWORK TERMINAL EQUIPMENT

Once again, the key differences between the P and the CP-bits are listed below.

- The values of P-bits are verified and can be recomputed as the DS3 signal passes through a Mid-Network Terminal equipment which is neither a Source nor a Sink terminal equipment.
- The values of the CP-bits as generated by the Source Terminal equipment MUST be preserved as a DS3 frame travels to the Sink Terminal equipment through any number of Mid-Network Terminal equipment.

Processing at the Sink Terminal Equipment

The purpose of the Sink Terminal Equipment is to accept and terminate a DS3 data-stream from the remote terminal equipment, and to extract out or de-map non-DS3 data, such as ATM cells, PPP Packets, DS1 or E1 signals, etc., from this signal. As the Sink Terminal Equipment receives this DS3 data-stream, it will also do the following.

- Compute and verify the P-bits within each inbound DS3 frame
- Compute and verify the CP-bits within each inbound DS3 frame.

Alarm and Signaling-Related Overhead Bits

The DS3 frame includes numerous bit-fields that are used to support the handling of alarms/defects and signaling information. Each of these bit-fields is defined below.

The AIC (Application Identification Channel) Bit

The AIC bit-field is located in the third C-bit, within F-Frame # 1, as depicted in **Figure 14**. The purpose of the AIC bit within the DS3 data-stream is to permit a given Terminal Equipment to determine, though not conclusively, if it is receiving a DS3 signal that is of the M13/M23 framing format or is of the C-bit Parity framing format.

For C-bit Parity Applications a given Transmitting DS3 Terminal Equipment will set the AIC bit to "1". However, for Channelized M13/M23 applications, the AIC bit-position within a given DS3 frame is simply a C-bit that is either set to "0" or "1" in order to denote stuff opportunities that either were or were not taken whenever this particular DS3 signal was created by multiplexing the seven (7) lower tributary DS2 signals.

Therefore, if a given DS3 Terminal Equipment receives a DS3 signal, in which the AIC bit is set to "0", then the Terminal Equipment can definitely conclude that it is receiving a DS3 signal that is of the M13/M23 Framing format. However, if a given DS3 Terminal Equipment receives a DS3 signal in which the AIC bit is always set to "1", then the Terminal Equipment can largely assume, but not definitely conclude, that it is receiving a DS3 signal that is of the C-bit Parity Framing format.

For information on how the Receive DS3/E3 Framer block handles the AIC bit, please see Section 4.3.2.9.

Alarm Indication Signal (AIS) Transmission

The Alarm Indication Signal (AIS) pattern is an alarm signal that is inserted into the outbound DS3 data-stream whenever a service affecting defect or failure condition is detected within the upstream traffic by the Local Terminal Equipment. **Figure 18** presents an illustration in which the transmission of the AIS indicator would be necessary.

FIGURE 18. AN ILLUSTRATION OF A CONDITION IN WHICH AIS WOULD NEED TO BE TRANSMITTED

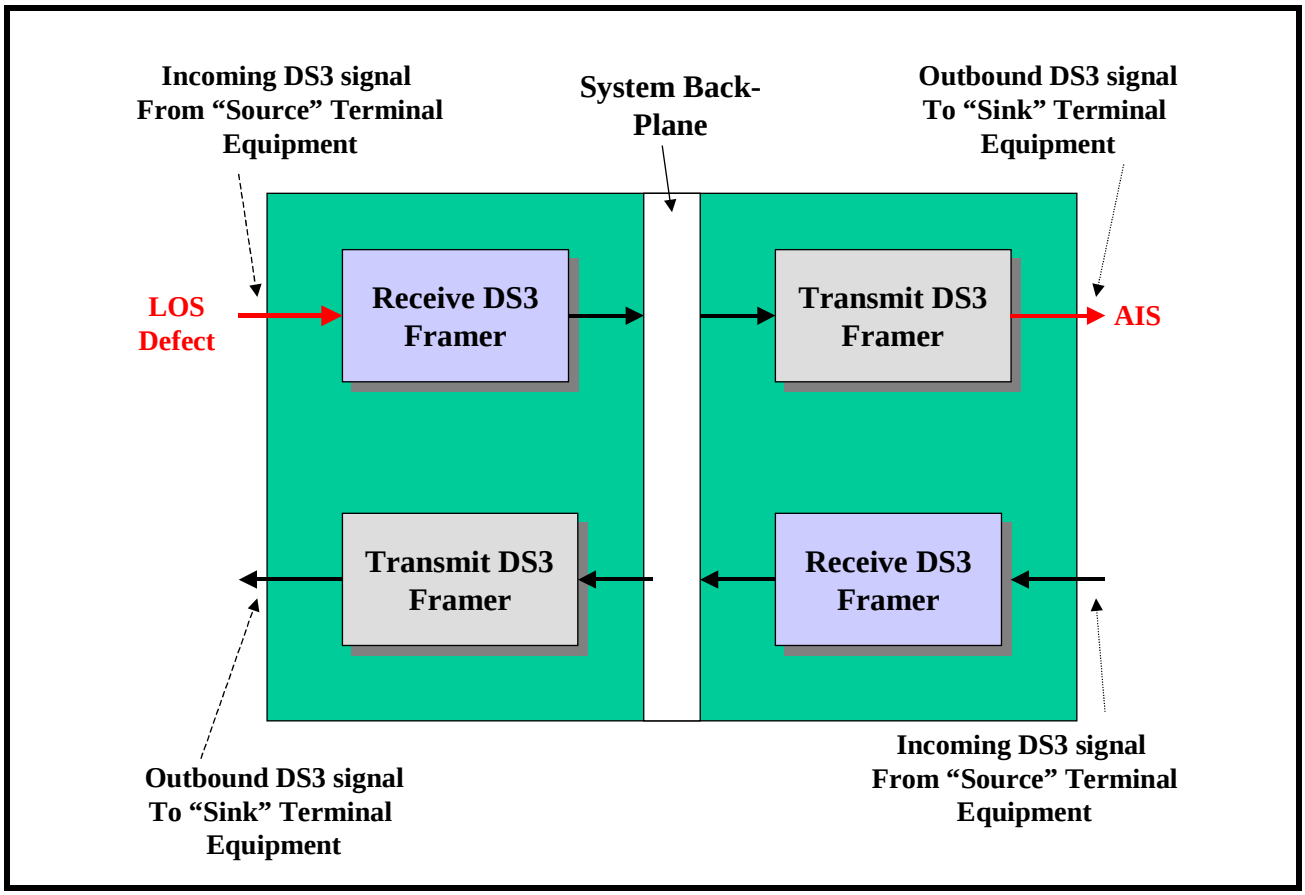


Figure 18 presents an illustration of two DS3 Boards that are connected to each other via a System Back-Plane. Each of these DS3 Boards contains a Receive DS3 Framer and a Transmit DS3 Framer entity. In one of these boards, the Receive DS3 Framer declares the LOS (Loss of Signal) Defect condition with the DS3 signal that it is receiving from the remote terminal equipment. In this particular system, the correct thing to do is for the System-Level Software to respond to this declaration of the LOS Defect condition, by commanding the Transmit DS3 Framer block on the other DS3 Board to transmit the DS3 AIS indicator in the down-stream direction.

This transmission of the AIS indicator serves the following two purposes.

a. It fills a given DS3 data-stream, in which the contents have been wiped-out by the up-stream defect condition, such as LOS, with a defined pattern.

b. It notifies the down-stream circuitry of this service-affecting defect condition.

The transmission of the AIS indicator would also be appropriate if the Receive DS3 Framer (in Figure 18) were to declare the LOF/OOF or AIS defect conditions.

The XRT79L71 can be configured to transmit and receive the following types of DS3 AIS patterns.

The Bellcore GR-499-CORE Compliant DS3 AIS Pattern

This particular AIS pattern has the following characteristics

- Valid M-bits, F-bits and P-bits
- All C-bits are set to "0"
- All X-bits are set to "1"

- A repeating [1, 0, 1, 0, …] pattern is written into the payload of each DS3 frame.

The Relaxed Bellcore GR-499-CORE Compliant DS3 AIS Pattern

This particular AIS pattern has the following characteristics

- Valid M-bits, F-bits and P-bits
- All X-bits are set to "1"
- A repeating [1, 0, 1, 0, …] pattern is written into the payload of each DS3 frame
- The C-bits are NOT required to be set to "0"

The Unframed All Ones Pattern

THE DS3 IDLE CONDITION SIGNAL

The DS3 Idle Condition signal is often used to indicate that the DS3 Channel is functionally sound, but has not yet been assigned any traffic. In other cases, the DS3 Idle pattern may be transmitted whenever a given piece of up-stream test equipment is operating in a loop-back mode, and is performing some sort of test diagnostic operation. The XRT79L71 can be configured to transmit and receive (detect) the following types of DS3 Idle Patterns.

1. The Bellcore GR-499-CORE Compliant DS3 Idle Pattern

This particular Idle pattern has the following characteristics

- Valid M-bits, F-bits and P-bits
- The three CP-bits within F-frame # 3 are each set to "0"
- The X-bits are each set to "1"
- A repeating [1, 1, 0, 0, …] pattern is written into the payload of the DS3 frames.

2. The user-specified, unframed DS3 Idle Pattern

In this case, the user can specify an unframed 4-bit repeating pattern which can be programmed by the user as being the DS3 Idle Pattern.

FEAC (Far-End Alarm & Control) Messages (Only available for the C-bit Parity Framing format)

The third C-bit within the first F-frame (e.g., C13 or FEAC) is used as the Far-End Alarm & Control (FEAC) channel between the Near-End DS3 terminal and the remote DS3 terminal. The FEAC channel is typically used to carry the following types of information.

- Alarm and Status Information
- Loopback commands to initiate and deactivate DS3 and DS1 loop-backs at the remote terminals.

The FEAC messages are encoded into a repeating 16-bit string that is of the following form.

FIGURE 19. THE BIT-FORMAT OF THE FEAC MESSAGE

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

Each of the dn bits can be either "1" or "0" with the rightmost bit transmitted first.

NOTE: *The FEAC Message consists of a six-bit code word ([d5, d4, d3, d2, d1, d0]) which is encapsulated along with 10 framing bits to form the 16-bit FEAC Message. Since each DS3 frame carries only one FEAC bit, 16 DS3 frame periods are required to deliver one complete FEAC Message. This six-bit code word can represent up-to 64 distinct messages, of which 43 have been defined in Bellcore GR-499-CORE. For a more detailed discussion on the transmission and reception of FEAC Messages, please see Section 4.2.3 and Section 4.3.4, respectively.*

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

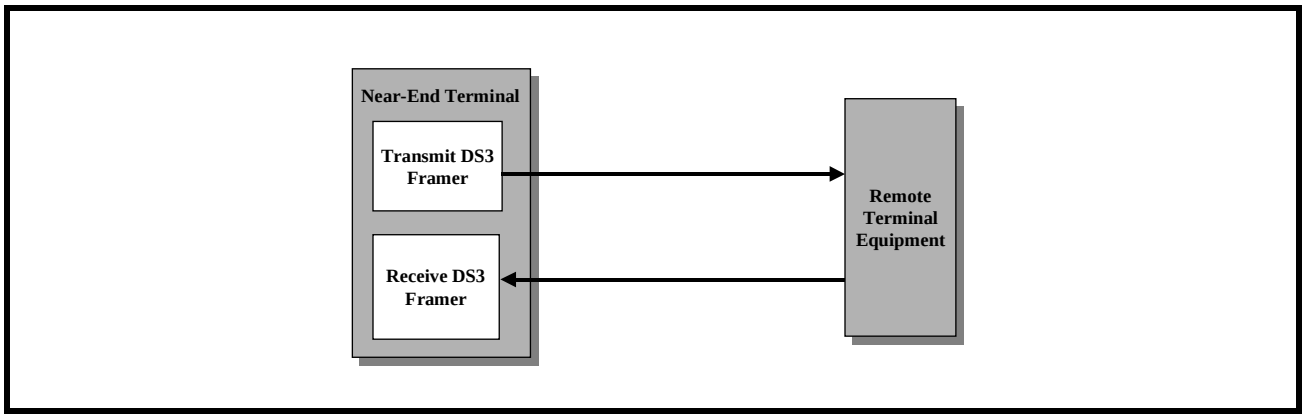
F-Frame # 4 consists of 3 bit fields for the FEBE (Far-End Block Error) channel. If the Near-End XRT79L71 detects CP-bit errors or a framing bit error within the incoming (received) DS3 data-stream, then it will inform the remote terminal of this fact by setting the three FEBE bits within the outgoing, or return DS3 frame, to any pattern other than [1, 1, 1] to indicate an erred condition. The Transmit Section of the XRT79L71 will typically set the FEBE bits to the value [1, 1, 1] in order to denote an un-erred condition, only if both of the following conditions are true.

- The XRT79L71 has not detected any M-bit or F-bit errors within its incoming (received) DS3 data-stream
- The XRT79L71 has not detected any CP-bit errors within its incoming (received) DS3 data-stream

This concept of transmitting the FEBE indicator in response to these error conditions is reinforced in Figure 20 through Figure 23, below.

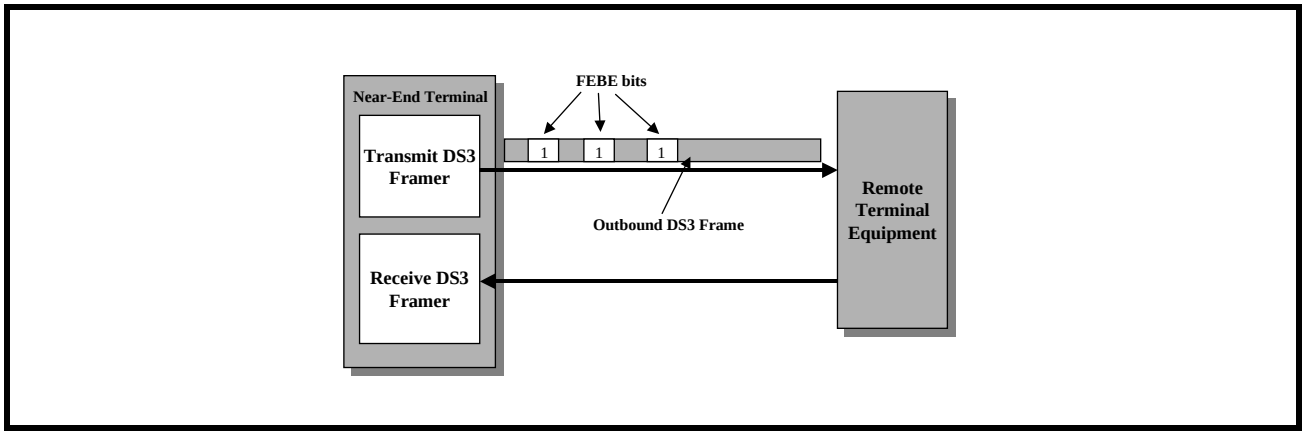
Figure 20 presents an illustration of a given Near-End Terminal that is exchanging DS3 data with the remote terminal, in an un-erred manner.

FIGURE 20. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING DS3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



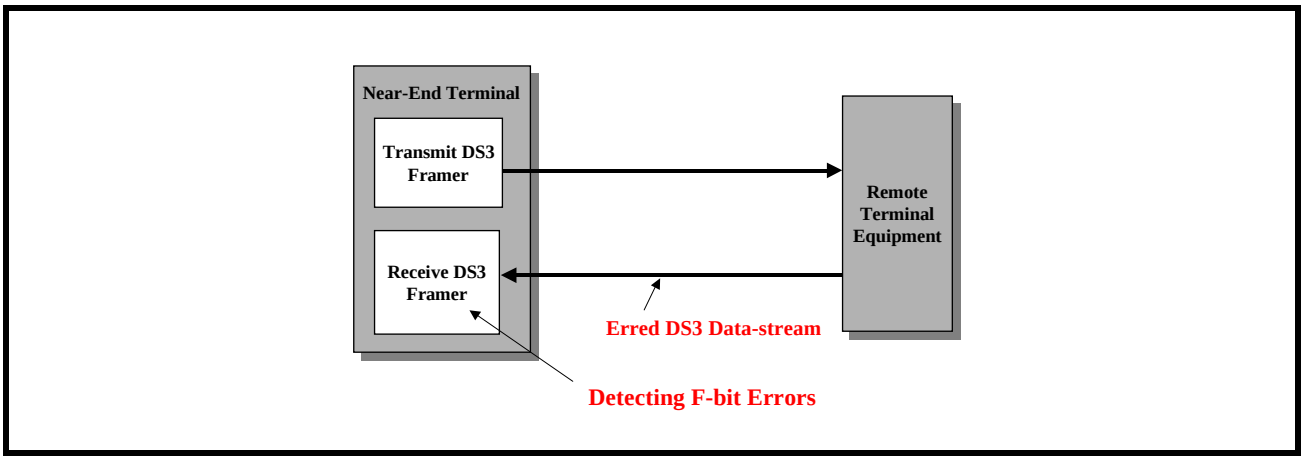
In response to this un-erred condition, the Transmit DS3 Framer within the Near-End Terminal will respond by setting each of the three (3) FEBE bit-fields to "1". Figure 21 presents an illustration of the Transmit DS3 Framer sending this un-erred indication to the Remote Terminal Equipment.

FIGURE 21. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



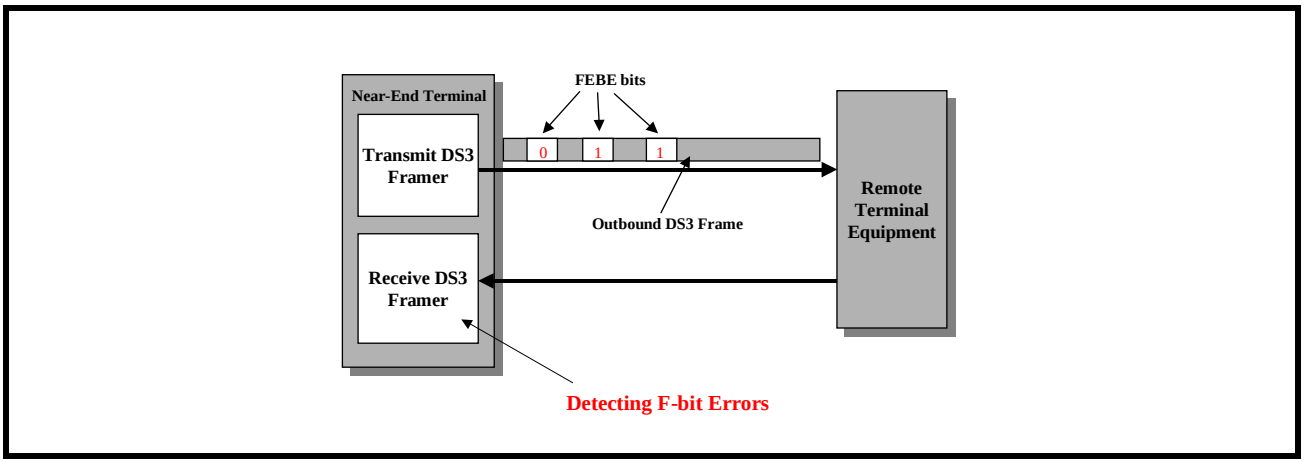
Next, Figure 22 presents an illustration of a given Near-End Terminal that is detecting framing (F-bit) errors within its incoming DS3 signal.

FIGURE 22. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DETECTING FRAMING BIT ERRORS WITHIN ITS INCOMING DS3 SIGNAL



In response to this erred condition, the Transmit DS3 Framer within the Near-End Terminal will respond by setting the three (3) FEBE bits, within each of its outbound DS3 frames to some value other than [1, 1, 1] in order to denote a FEBE (Far-End Block Error) or REI (Remote Error Indicator) event. More specifically, the Transmit DS3 Framer within the Near-End Termina) will transmit this FEBE/REI indicator (e.g., a DS3 frame with the FEBE bit-fields set to some value other than [1, 1, 1]) each time the corresponding Receive DS3 Framer, also within the Near-End Terminal, detects either a CP-bit or Framing bit error within a DS3 frame. Figure 23 presents an illustration of the Transmit DS3 Framer sending the FEBE/REI indicator to the Remote Terminal Equipment.

FIGURE 23. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FEBE/REI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block, within the XRT79L71, handles the FEBE bits, can be found in Section 4.2.5.6 and Section 4.3.2.10.

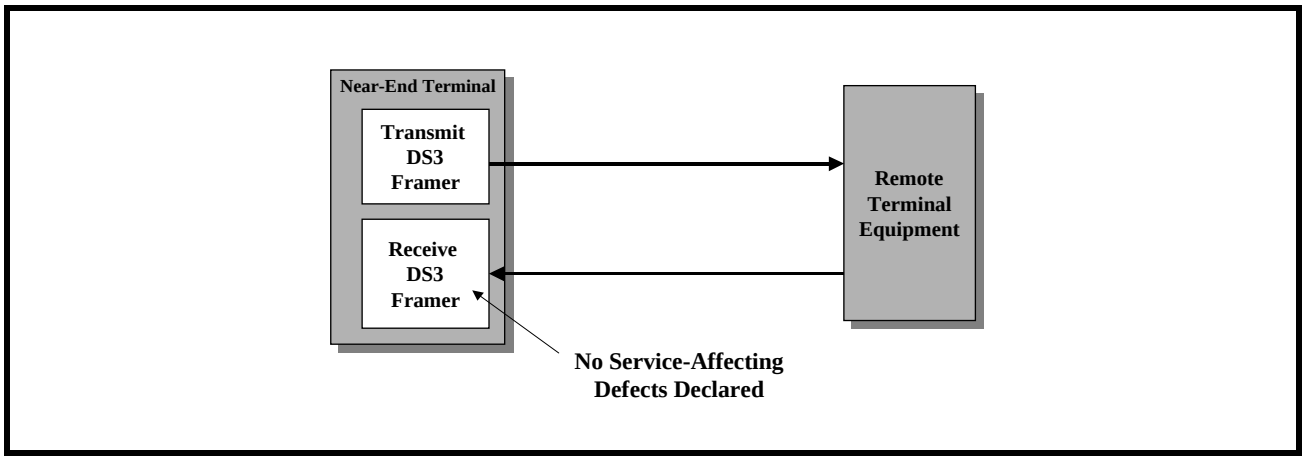
The FERF/RDI (Far-End Receive Failure/Remote Defect Indicator) - X-Bits

The purpose of the X-bits within the DS3 frame is to permit a given DS3 Terminal Equipment to transmit the FERF/RDI indicator to the remote terminal equipment. Whenever a Near-End DS3 Terminal declares a service-affecting defect condition (e.g., the LOS, LOF/OOF or the AIS defect condition) within its incoming DS3 data-stream, then it will inform the remote terminal equipment (e.g., the source of this defective DS3 signal) of this fact by transmitting the FERF/RDI indicator back out to the Remote Terminal Equipment via the outbound (returning) DS3 signal. This Near-End DS3 Terminal will indicate the FERF/RDI condition by setting both of the X-bits, within each outbound DS3 frame to "0", for the duration that the Receive-related/Service-affecting

defect condition exists. Conversely, this Near-End DS3 Terminal will indicate that it is NOT transmitting the FERF/RDI indicator by setting both of the X-bits within each outbound DS3 frame to "1" for the duration that no Receive-related/Service-affecting defect conditions exists. This concept of transmitting the FERF/RDI indicator in response to certain defect conditions is reinforced in Figure 24 through Figure 27, below.

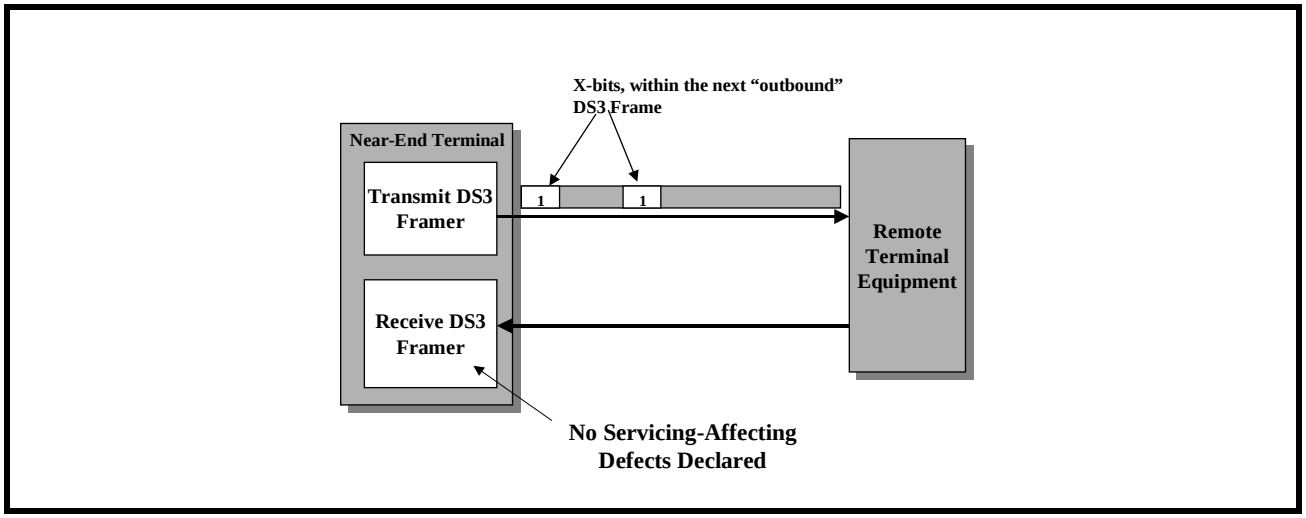
Figure 24 presents an illustration of a given Near-End Terminal that is exchanging DS3 data with a remote terminal, in an un-erred manner.

FIGURE 24. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING DS3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



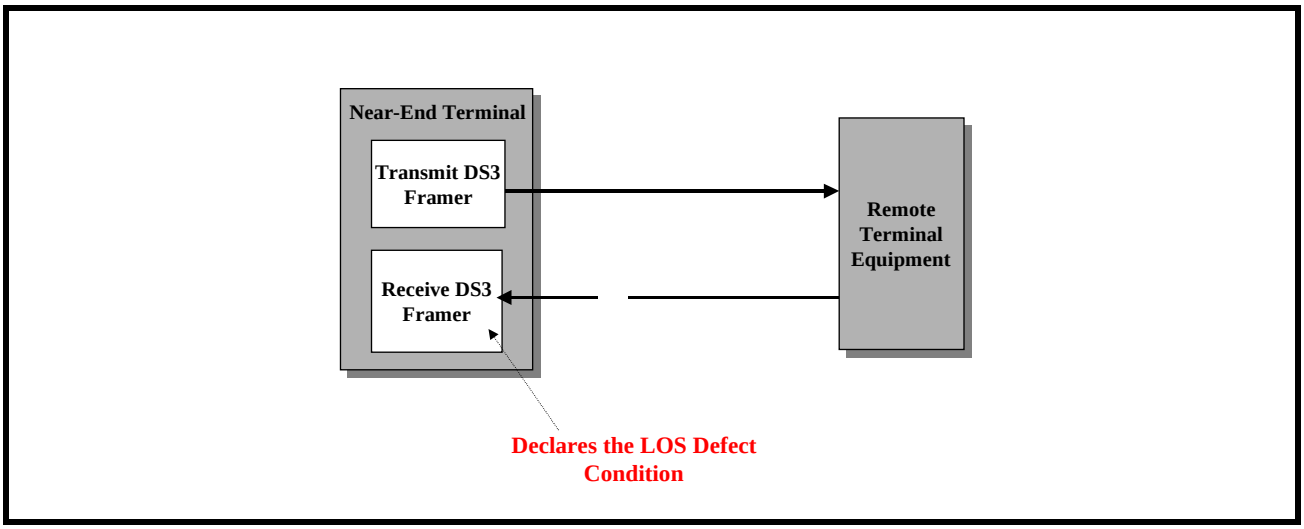
In response to this un-erred condition, the Transmit DS3 Framer within the Near-End Terminal will respond by setting the X-bits within each of its outbound DS3 frames, to "1" in order to denote an un-erred condition. Figure 25 presents an illustration of Transmit DS3 Framer sending the un-erred indication to the Remote Terminal Equipment.

FIGURE 25. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION THE REMOTE TERMINAL EQUIPMENT



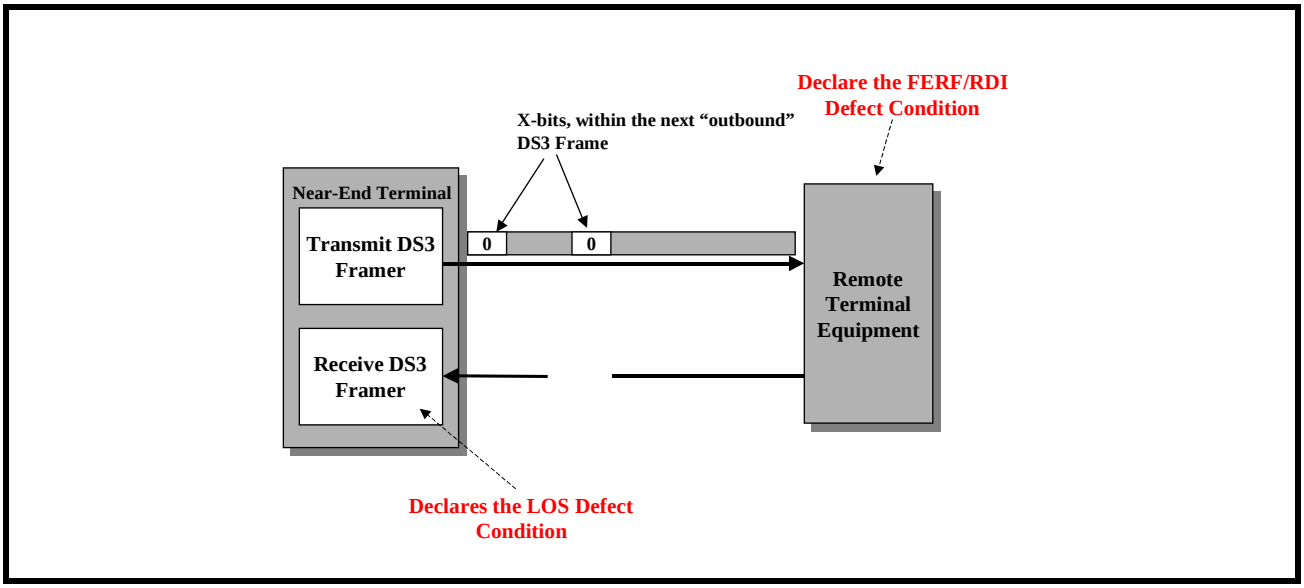
Next, Figure 26 presents an illustration of a given Near-End Terminal that is declaring the LOS defect with its incoming DS3 signal.

FIGURE 26. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DECLARING THE LOS DEFECT CONDITION WITH ITS INCOMING DS3 SIGNAL



In response to this erred condition, the Transmit DS3 Framer within the Near-End Terminal will respond by setting the X-bits within each of its outbound DS3 frames to "0", in order to denote a FERF (Far-End-Receive Failure) or RDI (Remote Defect Indicator) condition. Figure 27 presents an illustration of the Transmit DS3 Framer sending the FERF/RDI indicator to the Remote Terminal Equipment.

FIGURE 27. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block, within the XRT79L71, handle the FERF/RDI indication will be described in Section 4.2.5.4 and Section 4.3.2.6, respectively.

Data Link Related Overhead Bits

UDL - User Data Link Bits (C-bit Parity Framing format only)

Bellcore GR-499-CORE does not define any purpose for these bit-fields. As a consequence, these bit-fields may be used for the transmission of a proprietary data link between two pieces of terminal equipment. These

bit-fields are not processed by the Receive DS3/E3 Framer block, within the XRT79L71, and the Transmit DS3/E3 Framer block will automatically set these particular bit-fields to "1" within each outbound DS3 frame. However, to support such a proprietary data link, use the Transmit Overhead Data Input Interface block and the Receive Overhead Data Output Interface block in order to support the transmission/reception of data via these particular bit-fields. Section 4.2.2 and Section 4.3.5 discuss the Transmit Overhead Data Input Interface block and the Receive Overhead Data Output Interface block in detail, respectively.

DL - Path Maintenance Data Link Bits (C-bit Parity Framing Format only)

The purpose of these bit-fields is to support the transmission/reception of the Path Maintenance Data Link between any two pieces of DS3 Terminal Equipment. The transmission/reception of the PMDL data will be handled by the Transmit LAPD Controller and the Receive LAPD Controller blocks, respectively.

For information on how the Transmit LAPD Controller and the Receive LAPD Controller handle LAPD/PMDL messages, please see Section 4.2.4 and Section 4.3.3, respectively.

4.2 THE TRANSMIT DIRECTION - DS3 CLEAR-CHANNEL FRAMER APPLICATIONS

Now that the basics of the DS3 frame structure have been discussed, the next several sections present an in-depth functional description of all of the blocks that are operating in the Transmit Direction, within the XRT79L71, when configured to operate in the Clear-Channel DS3 Framer Mode. Figure 28 presents a functional block diagram of the Transmit Direction circuitry within the XRT79L71.

FIGURE 28. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY WHEN THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE

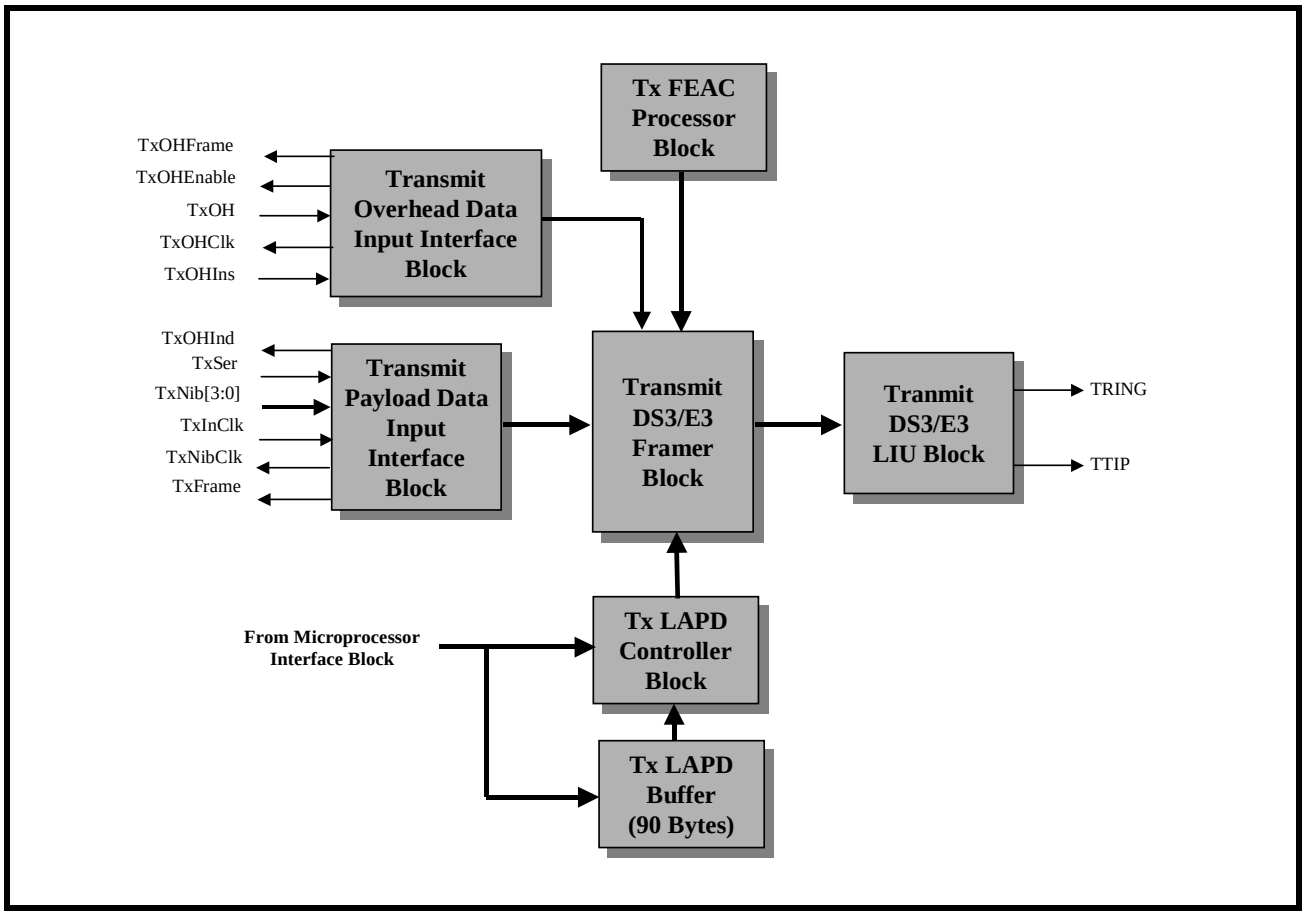


Figure 28 indicates that the Transmit Direction circuitry consists of the following functional blocks.

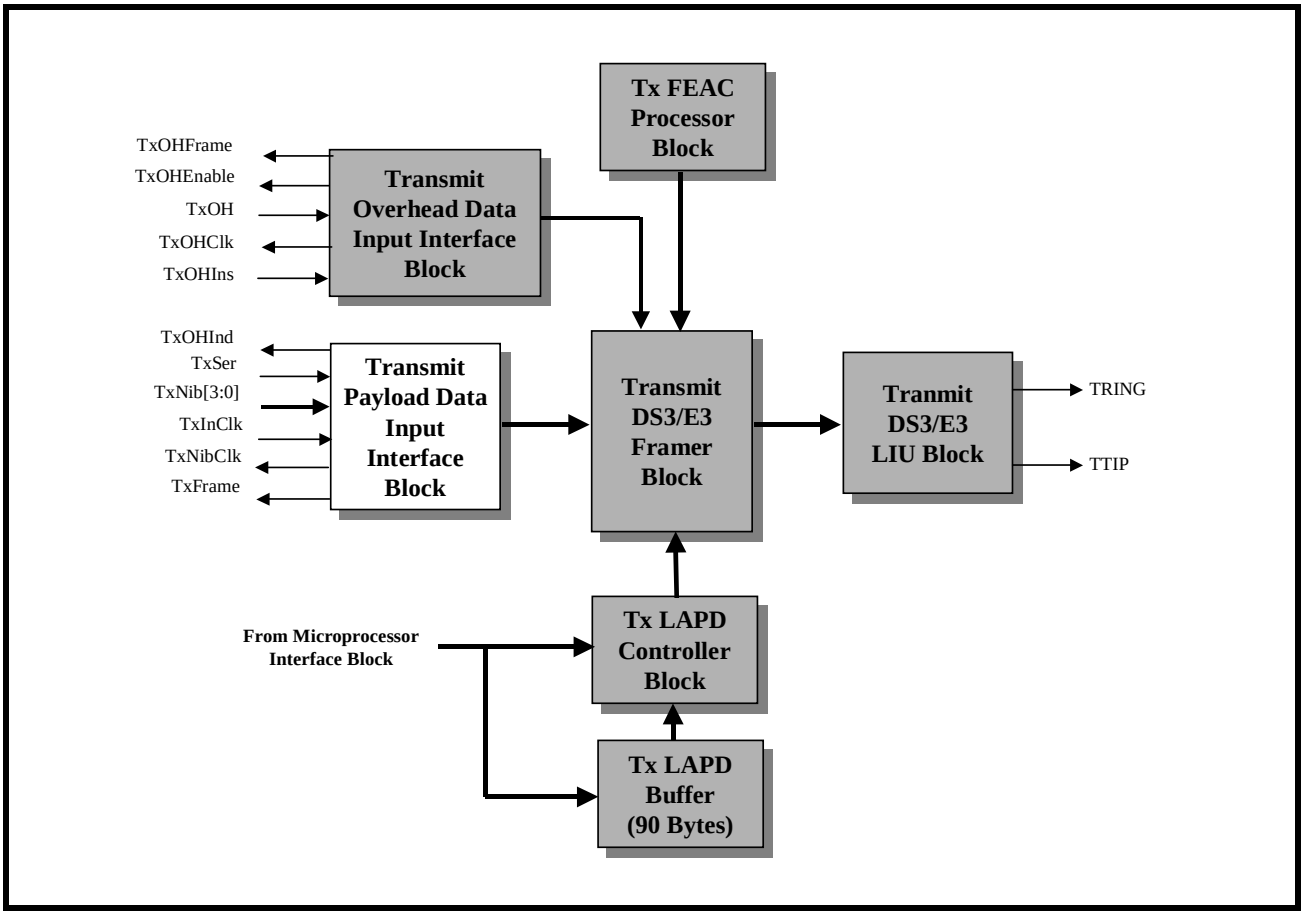
- The Transmit Payload Data Input Interface block

- The Transmit Overhead Data Input Interface block
- The Transmit LAPD Controller block
- The Transmit FEAC Controller block
- The Transmit DS3 Framer block
- The Transmit DS3 LIU Interface block

4.2.1 TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

The Transmit Payload Data Input Interface block is the very first functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 29** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit Payload Data Input Interface block highlighted.

FIGURE 29. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHEN THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK HIGHLIGHTED



The purpose of the Transmit Payload Data Input Interface block is to accept payload data from some system-side or up-stream source and to pass this payload data along to the Transmit DS3 Framer block that will ultimately map this payload data into the payload bits within each outbound DS3 frame.

In order to accomplish this, the Transmit Payload Data Input Interface block has numerous input and output pins. **Table 13** presents a list and a brief definition of each of these pins.

TABLE 13: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
TxSer	C9	I	**Transmit Serial Payload Data Input Pin:** If the Transmit Payload Data Input Interface block is operated in the Serial Mode, then the System-Side equipment is expected to apply the payload data that is to be transported via the outbound DS3 data-stream to this input pin, in a serial manner. The XRT79L71 samples the data that is on this input pin upon the rising edge of either the RxOutClk for loop-timing applications or the TxInClk signal or local-timing applications. ***NOTE:*** *This signal is only active if the NibIntf input pin is pulled "Low".*
TxNib[3:0]	B8 C8 D8 A9	I	**Transmit Nibble-Parallel Payload Data Input pins:** If the Transmit Payload Data Input Interface block is operated in the Nibble-Parallel Mode, then the System-Side equipment is expected to apply the payload data that is to be transported via the outbound DS3 data-stream to these input pins, in a nibble-parallel manner. The XRT79L71 samples the data that is placed on these input pins upon the third rising edge of TxInClk, following a given rising edge of the TxNibClk output pin. ***NOTE:*** *These signals are only active if the NibIntf input pin is pulled "High".*
TxNibFrame	A10	O	**Transmit End of Frame Output Indicator - Nibble Mode:** The Transmit Section of the XRT79L71 pulses this output pin "High" for one nibble period whenever the Transmit Payload Data Input Interface block is processing the very last nibble within a given DS3 frame. The purpose of this output pin is to alert the System-Side Terminal Equipment that it needs to begin transmission of a new DS3 frame to the Transmit Payload Data Input Interface of the XRT79L71. ***NOTE:*** *This output pin is only active if the XRT79L71 has been configured to operate in the Nibble-Parallel Mode*
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the Transmit Circuitry within the XRT79L71. Additionally, if the XRT79L71 has been configured to operate in both the Serial and Local-Timing Mode, then the XRT79L71 will sample the data, residing on the TxSer input pin, upon the rising edge of this input clock signal.
TxNibClk	D9	O	Transmit Nibble-Mode Clock Output SignalIf the XRT79L71 has been configured to operate in the Nibble-Parallel Mode, then the XRT79L71 will derive this output clock signal from either the TxInClk or the LIU Recovered Clock signal. ***NOTE:*** *The frequency of this clock output signal is approximately one-fourth of the TxInClk or the RxOutClk signals.* The user is advised to update the Nibble-Parallel data via the TxNib[3:0] output pins, upon the rising edge of this clock output signal. The XRT79L71 will sample the TxNib[3:0] input pins upon the third rising edge of the TxInClk clock input signal following a rising edge in this particular signal.

TABLE 13: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
TxOHInd/ TxGapClk	B9	O	**Transmit Overhead Bit Indicator Output/Transmit Gap-Clock Output:** The exact function of this output pin depends upon whether the XRT79L71 has been configured to operate in the Gapped-Clock Mode or Not. **Non-Gapped Clock Mode - TxOHInd:** This output pin will pulse "High" one bit period prior to the time that the Transmit Section of the XRT79L71 is processing an Overhead bit. This output pin will be held "Low" at all other times. The purpose of this output pin is to warn the System-side Terminal Equipment that during the very next bit-period, the XRT79L71 is going to be processing an Overhead bit and will be, during this very next bit-period, ignoring any data that is applied to the TxSer input pin. **Gapped-Clock Mode - TxGapClk:** If the XRT79L71 has been configured to operate in the Gapped-Clock Mode, then this particular output pin will function as a demand output clock signal. In this case, the System-Side Terminal Equipment will be expected to update the data on the TxSer input pin, upon the rising edge of this particular output signal. The XRT79L71 will sample and latch the TxSer data, upon the falling edge of the TxGapClk signal. ***NOTE:*** *In the Gapped-Clock Mode, the XRT79L71 will only generate a clock edge via this output pin whenever the Transmit Payload Data Input Interface is processing payload data. The XRT79L71 will NOT generate a clock edge via this output pin whenever the Transmit Payload Data Input Interface is processing an overhead bit.*
TxFrame	B10	O	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT79L71 pulses this output pin "High" for one bit-period coincident to whenever the Transmit Payload Data Input Interface block is processing the last bit of a given DS3 frame. The purpose of this output pin is to alert the System-side Terminal Equipment that it needs to begin transmission of a new DS3 frame to the Transmit Payload Data Input Interface block of the XRT79L71 (e.g., to permit the XRT79L71 to maintain Transmit DS3 Framing alignment control over the System-Side Terminal Equipment).

TABLE 13: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
TxFrameRef	A11	I	**Transmit DS3 Frame Reference Input:** The XRT79L71 permits the user to configure this input pin to function as the Transmit DS3 Frame Generation Reference Input. Iinvoking this particular configuration option, will cause the Transmit DS3 Framer block, within the XRT79L71, to initiate DS3 frame generation anytime it detects a rising edge on this input pin. ***NOTE:*** *If tthis configuration option is implemented, it is imperative that this particular input signal is synchronous with the TxInClk input signal. Failure to do so will result in the transmission of erred DS3 frames to the remote terminal equipment.*
RxOutClk	B5	O	**Loop-Timing Reference Clock Output Pin:** If the XRT79L71 is configured to operate in the Loop-Timing mode, then the Transmit Section of the XRT79L71 will be configured to use the LIU Recovered Clock signal as its timing source. In this case, the XRT79L71 will output a 44.736MHz clock signal via this particular output pin. In this configuration, the TxInClk signal will be inactive and will NOT be used to sample and latch the data on the TxSer input pin. In this case, the XRT79L71 will now be configured to sample the TxSer input pin upon the rising edge of the RxOutClk signal. ***NOTE:*** *This output pin will always be active, in the sense that it will always generate a 44.736MHz clock signal, even when the XRT79L71 is NOT configured to operate in the Loop-Timing Mode. However, the XRT79L71 will only use this particular clock signal to sample and latch the data on the TxSer input pin whenever the XRT79L71 has been configured to operate in the Loop-Timing Mode.*

OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

The Transmit Payload Data Input Interface block permits the user to configure it to operate in the following combination of modes.

- The Serial or Nibble-Parallel Interface Mode
- The Loop-Timing or Local-Timing Mode

If the XRT79L71 is configured to operate in the Local-Timing Mode, then there are two additional sub-options.

- The Frame-Master or Frame-Slave Mode

With these three sets of configuration options, the Transmit Payload Data Input Interface block can be configured to operate in any one of the following six (6) Modes.

- Mode 1 - Serial/Loop-Timing Mode
- Mode 2 - Serial/Local-Timing/Frame Slave Mode
- Mode 3 - Serial/Local-Timing/Frame Master Mode
- Mode 4 - Nibble-Parallel/Loop-Timing Mode
- Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode
- Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode

Table 8A presents a "Quick-Look" Summary of each of these Modes.

TABLE 14: A SUMMARY OF THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" MODES

MODE	NIBBLE-PARALLEL/ SERIAL MODE	SOURCE OF "SYSTEM-SIDE TERMINAL EQUIPMENT" CLOCK	FRAMING ALIGNMENT TIMING SOURCE
1	Serial	**Loop-Timing Mode:** The XRT79L71 will output a 44.736MHz clock signal via the "RxOutClk" output pin. This clock signal is ultimately derived from Recovered Line Clock (from Receive DS3/E3 LIU Block).	Asynchronous upon Power up.
2	Serial	**Local-Timing Mode:** The user is expected to apply a 44.736MHz clock signal to the TxInClk Input pin	TxFrameRef Input
3	Serial	Local-Timing Mode: The user is expected to apply a 44.736MHz clock signal to the TxInClk Input pin	Asynchronous upon Power up
4	Nibble-Parallel	**Loop-Timing Mode:** The XRT79L71 will output an 11.184MHz "Nibble-Clock" signal (via the "TxNibClk" output). This clock signal is ultimately derived from the Recovered Line Clock (from the Receive DS3/E3 LIU Block).	Asynchronous upon Power up
5	Nibble-Parallel	**Local-Timing Mode:** The user is expected to apply a 44.736MHz clock to the TxInClk input pin. The XRT79L71 will use the TxInClk signal to derive the 11.184MHz clock signal (which is output via the "TxNibClk" output pin).	TxFrameRef Input pin
6	Nibble-Parallel	**Local-Timing Mode:** The user is expected to apply a 44.736MHz clock signal to the TxInClk input pin. The XRT79L71 will use the TxInClk signal to derive the 11.184MHz clock signal (which is output via the "TxNibClk" output pin).	Asynchronous upon Power up

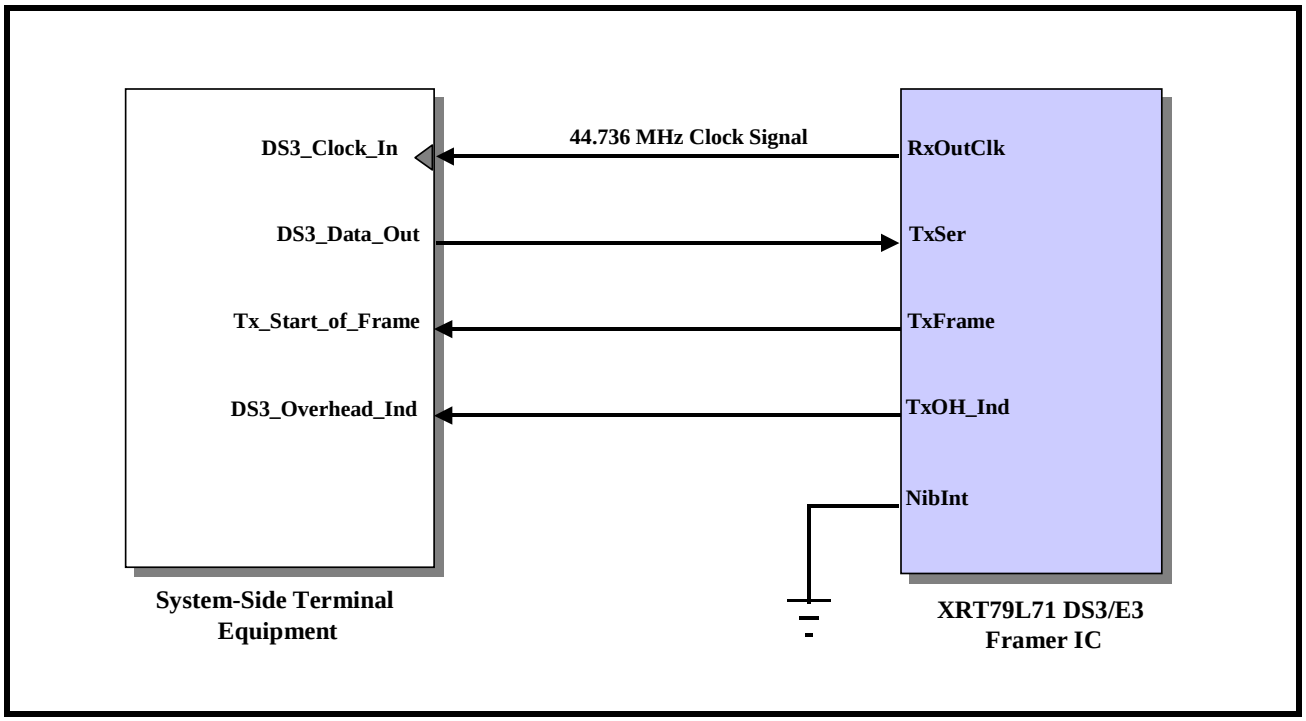
4.2.1.1 Mode 1 - Serial/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 1 then all of the following is true.

- The XRT79L71 will be configured to operate in the Loop-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal from the Receive DS3 LIU Block as its timing source.
- Since the XRT79L71 is configured to operate in the Serial-Mode, it will sample and latch the data, being applied to the TxSer input pin upon the rising edge of the RxOutClk output signal.
- The XRT79L71 will pulse the TxFrame output pin "High" for one bit period coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given DS3 frame.

Figure 30 presents an illustration of how to interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71, for Mode 1 operation.

FIGURE 30. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR MODE 1 (SERIAL/LOOP-TIMING) OPERATION



Mode 1 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of the 44.736MHz clock signal via the RxOutClk output signal.

NOTE: *The "RxOutClk" signal is a buffered version of the "Recovered Line Clock" signal, from the Receive DS3/E3 LIU Block).*

This clock signal is used as the System-Side Terminal Equipment clock source by both the Transmit Payload Data Input Interface block of the XRT79L71 and the System-Side Terminal Equipment device or circuitry.

The System-Side Terminal Equipment should serially output the payload data, that is to be transported via the outbound DS3 data-stream, via its DS3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out output pin upon the rising edge of the 44.736MHz clock signal at its DS3_Clock_In input pin as depicted below in **Figure 31**.

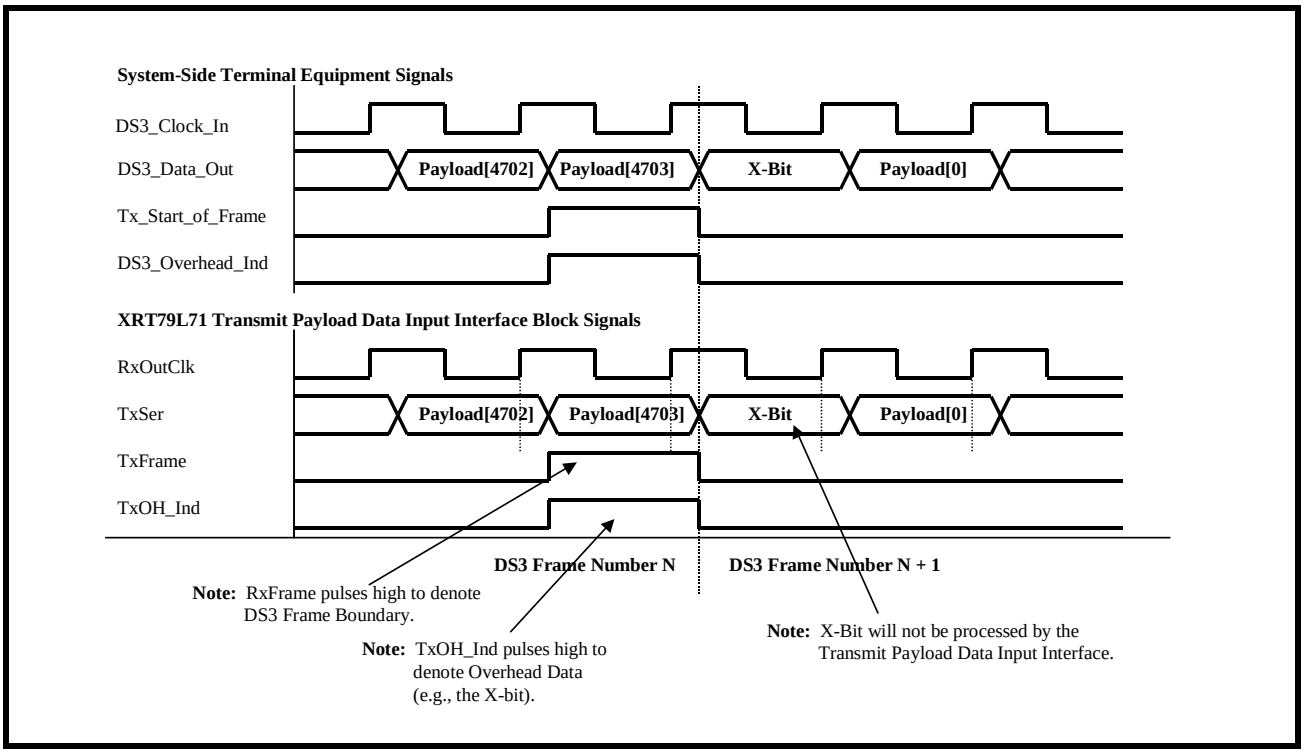
The XRT79L71 will latch the contents of the TxSer input pin, upon the rising edge of the RxOutClk signal. The XRT79L71 will indicate that it is processing the very last bit of a given DS3 frame by pulsing its TxFrame output pin "High" for one bit-period. The TxFrame output pin will be held "Low" at all other times. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_End_of_Frame input pin, then it is expected to begin the transmission of the contents of the very next outbound DS3 frame, via the DS3_Data_Out output or the TxSer input pin.

Finally, the Transmit Payload Data Input Interface block, within the XRT79L71, will indicate that it is about to process an overhead bit by pulsing the TxOH_Ind output pin "High" for one period prior to its processing. In **Figure 30**, the TxOH_Ind output pin of the XRT79L71 is connected to the DS3_Overhead_Ind input pin of the System-Side Terminal Equipment circuitry. Whenever the DS3_Overhead_Ind input pin is pulsed "High" the System-Side Terminal Equipment is expected to NOT transmit a DS3 payload bit upon the very next rising edge of DS3_Clock_In. Instead, the System-Side Terminal Equipment is expected to delay its transmission of the very next payload bit by one RxOutClk clock period.

NOTE: *For information on operating the "Transmit Payloads Data Input Interface" block in the "Gapped-Clock" Mode, see Section 4.2.1.7.*

Figure 31 presents an illustration of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 1 Operation.

FIGURE 31. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 1 (SERIAL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 1 (Serial/Loop-Timing)

The user can configure the XRT79L71 to operate in Mode 1 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 31.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

NOTE: *This step also configures the Receive Payload Data Output Interface block to operate in the Serial Mode.*

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Mode Operating Mode Register to [0, 0] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	0	0

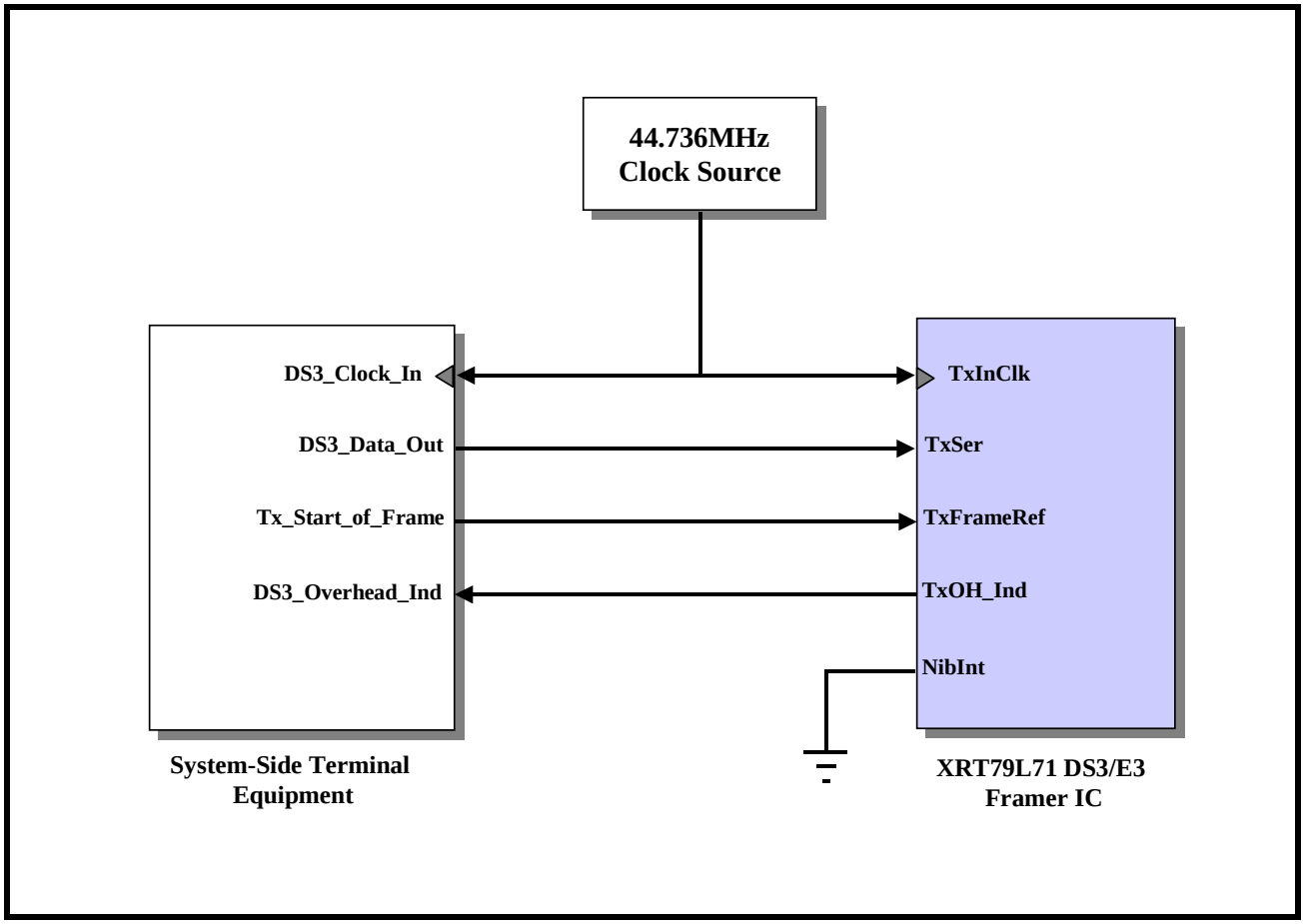
4.2.1.2 Mode 2 - Serial/Local-Timing/Frame Slave Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 2, then all of the following are true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- Since the XRT79L71 is configured to operate in the Serial-Mode, it will sample and latch the data, being applied to the TxSer input pin upon the rising edge of the TxInClk input signal.
- The Transmit Section of the XRT79L71 will initiate the generation and transmission of a new DS3 frame anytime it detects a rising edge at the TxFrameRef input pin.
- The XRT79L71 will still pulse the TxFrame output pin coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given DS3 frame.

Figure 32 presents an illustration of the behavior of the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 2 operation.

FIGURE 32. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 2 (SERIAL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Mode 2 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 44.736MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More specifically, this 44.736MHz clock signal will be applied to both the DS3_Clock_In input of the System-Side Terminal Equipment and the TxInClk input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data, that is to be transported via the outbound DS3 data-stream, via its DS3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out output pin upon the rising edge of the 44.736MHz clock signal at its DS3_Clock_In input pin, as depicted below in **Figure 33**.

The XRT79L71 will latch the contents of the TxSer input pin, upon the rising edge of the TxInClk signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a Framing Reference signal to the XRT79L71 by pulsing its TxFrameRef input pin "High" for one bit-period, coincident with the first bit of a new outbound DS3 frame being applied to the TxSer input pin. Once the XRT79L71 detects the rising edge of the input at its TxFrameRef input pin, it will begin to generate and transmit a new DS3 frame.

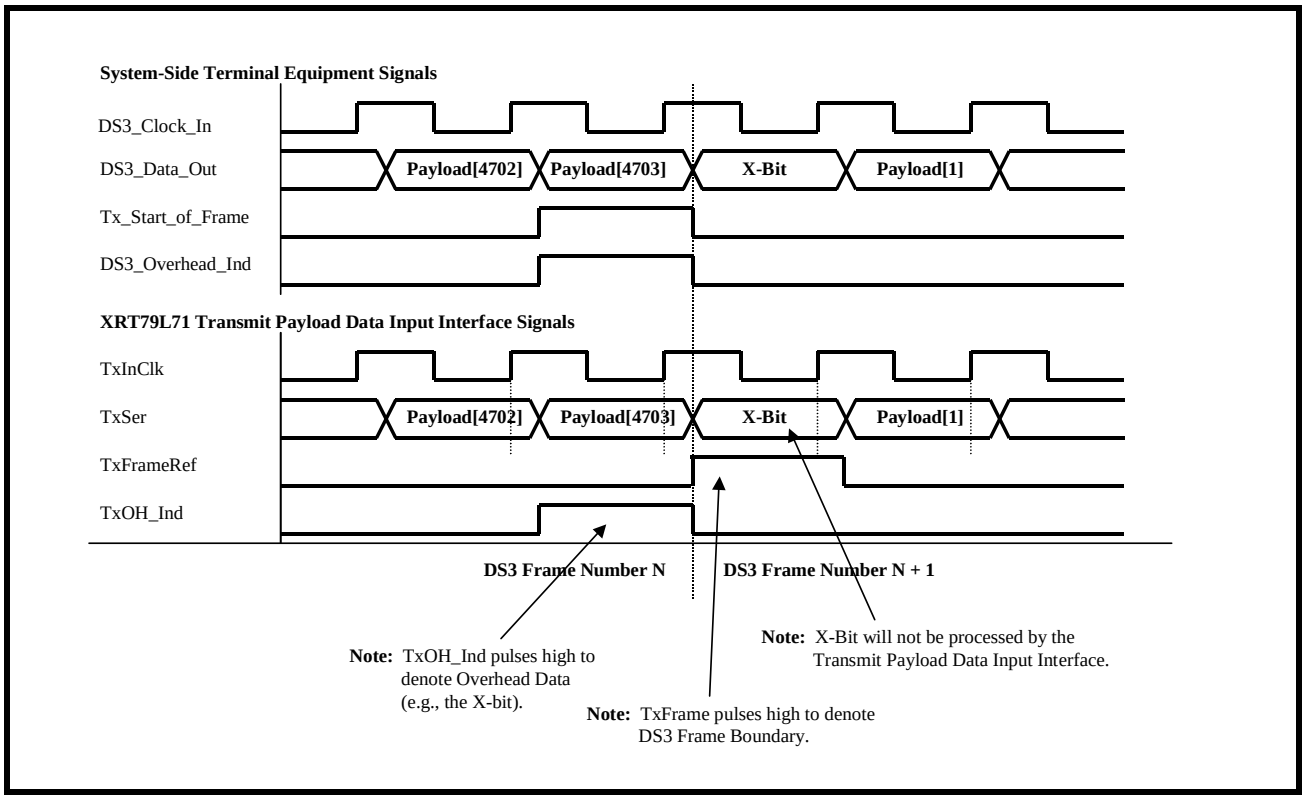
NOTES:

1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of Frame Generation and is referred to as the Frame Master. Since the XRT79L71 does not control or dictate the instant that it will generate a new DS3 frame, but is driven by the System-Side Terminal Equipment, it is referred to as the Frame Slave.*
2. *If the XRT79L71 is configured to operate in Mode 2 then it is imperative that the Tx_Start_of_Frame or TxFrameRef signal is synchronized to the TxInClk input clock signal. If the user fails to insure that the TxFrameRef input signal is synchronized to the "TxInClk" input clock signal, then the XRT79L71 will transmit erred DS3 data to the remote terminal equipment.*

Finally, the XRT79L71 pulses its TxOH_Ind output pin "High" one bit-period prior to it processing a given overhead bit within the outbound DS3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the DS3_Overhead_Ind input pin of the System-Side Terminal Equipment whenever the XRT79L71 pulses its TxOH_Ind output pin "High", it will also drive the DS3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay transmission of the very next DS3 payload bit by one TxInClk clock period.

Figure 33 presents an illustration of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 2 Operation.

FIGURE 33. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 2 (SERIAL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 2 (Serial/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 2 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 32.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 1] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

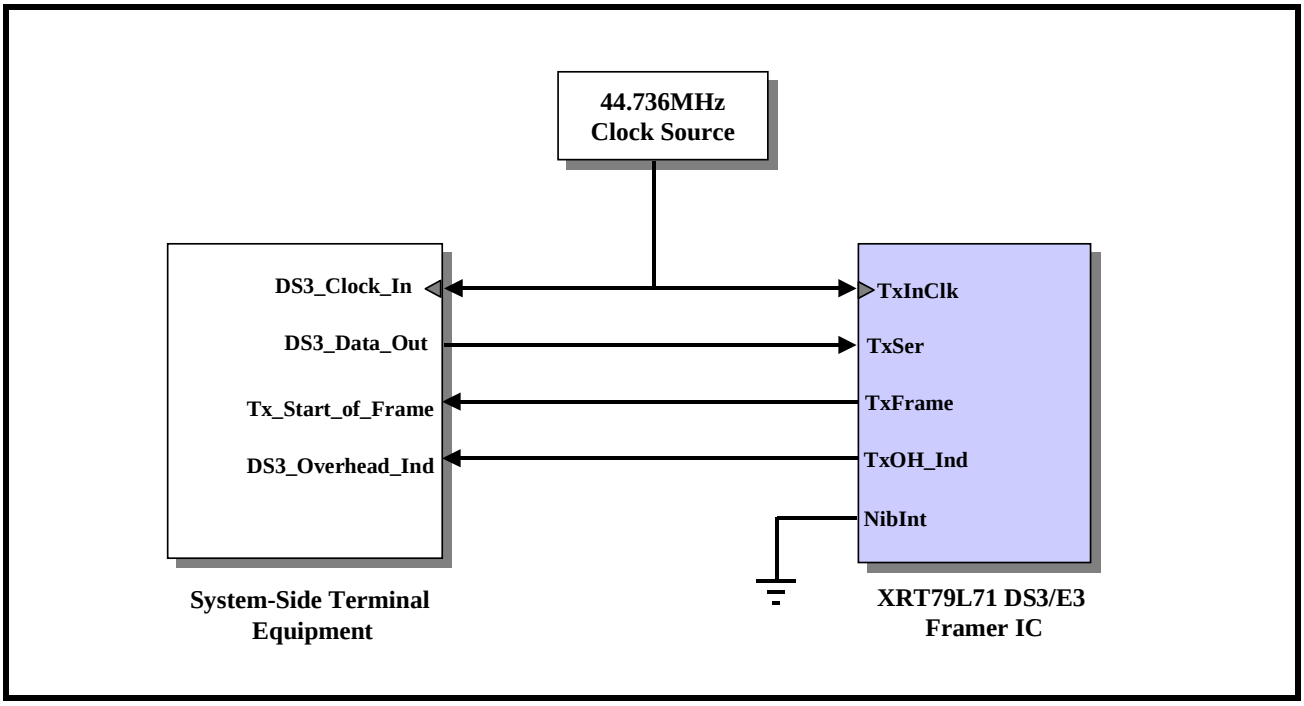
4.2.1.3 Mode 3 - Serial/Local-Timing/Frame Master Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 3 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- Since the XRT79L71 is configured to operate in the Serial-Mode, it will sample and latch the data, being applied to the TxSer input pin upon the rising edge of the TxInClk input signal.
- The XRT79L71 will still pulse the TxFrame output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given DS3 frame.

Figure 34 presents an illustration of how to interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 3 operation.

FIGURE 34. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Mode 3 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 44.736MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More specifically, this 44.736MHz clock signal will be applied to both the DS3_Clock_In input of the System-Side Terminal Equipment and the TxInClk input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data that is to be transported via the outbound DS3 data-stream via its DS3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out output pin upon the rising edge of the 44.736MHz clock signal at its DS3_Clock_In input pin as depicted below in Figure 35.

The XRT79L71 will latch the contents of the TxSer input pin upon the rising edge of the TxInClk signal. The XRT79L71 will indicate that it is processing the very last bit of a given DS3 frame by pulsing its TxFrame output pin "High" for one bit-period. The TxFrame output pin will be held "Low" at all other times. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_End_of_Frame input pin, then it is expected to begin the transmission of a contents of the very next outbound DS3 frame, via the DS3_Data_Out output or the TxSer input pin.

NOTES:

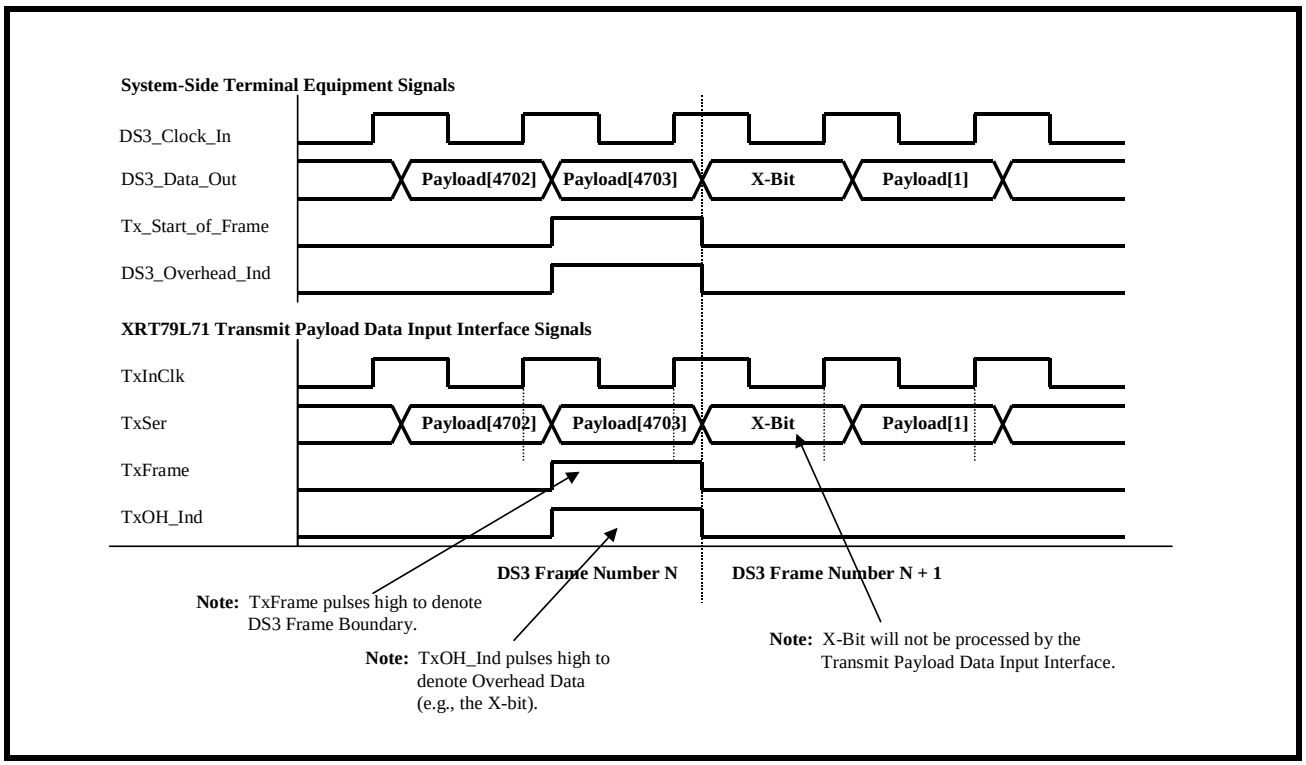
1. *In this particular mode, the Transmit Direction circuitry (within the XRT79L71) is dictating the instant that it will initiate the generation of a new DS3 frame; and is referred to as the "Frame Master". As a consequence, this particular mode is referred to as the "Frame Master" Mode.*
2. *In contrast to "Mode 2" operation, if the XRT79L71 is configured to operate in "Mode 3", then it is NOT imperative that the "TxFrameRef" input be synchronized to the "TxInClk" input clock signal. In this case, we recommend that the user tie the "TxFrameRef" input pin to GND.*

Finally, the XRT79L71 pulses its TxOH_Ind output pin "High" one bit-period prior to it processing a given overhead bit within the outbound DS3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the DS3_Overhead_Ind input pin of the System-Side Terminal Equipment whenever the XRT79L71 pulses its TxOH_Ind output pin "High", it will also drive the DS3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay transmission of the very next DS3 payload bit by one TxInClk clock period.

NOTE: *If operating the "Transmit Payload Data Input Interface" block in the "Gapped-Clock" Mode; trefer to Section 4.2.1.7.*

Figure 35 presents an illustration of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 3 Operation.

FIGURE 35. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 3 (Serial/Local-Timing/Frame-Master Mode)

The user can configure the XRT79L71 to operate in Mode 3 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 34.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [1, X] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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

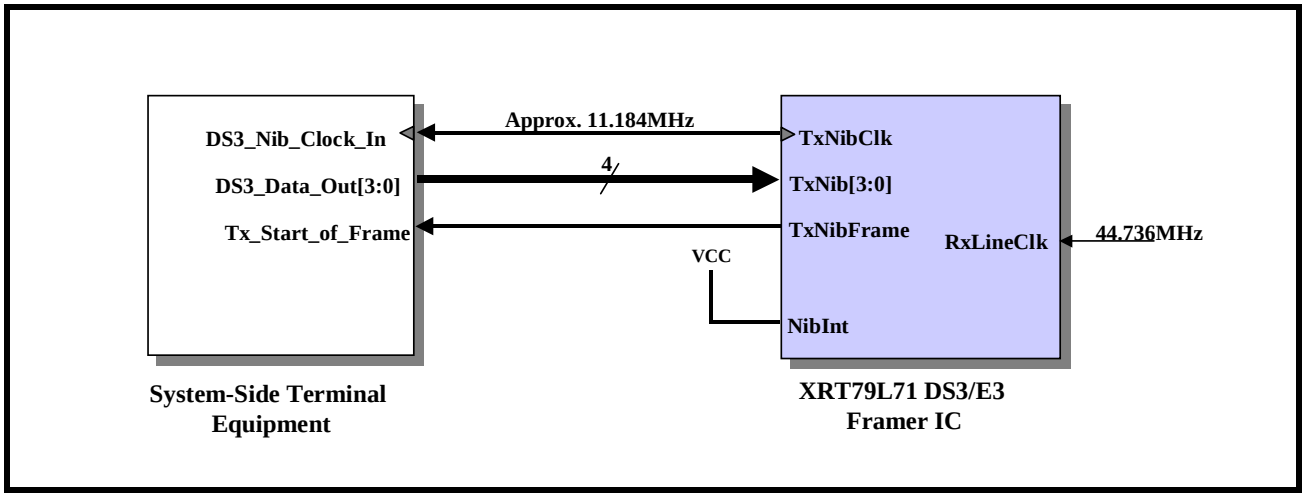
4.2.1.4 Mode 4 - Nibble-Parallel/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 4 then all of the following is true.

- The XRT79L71 will be configured to operate in the Loop-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal from the Receive DS3 LIU Block as its timing source.
- In this mode, the XRT79L71 will use the LIU Recovered Clock signal to derive the TxNibClk signal.
- For DS3 Applications, the TxNibClk frequency is approximately one-fourth of the LIU Recovered Clock signal or 11.184MHz. The reason for the TxNibClk frequency not being exactly 11.184MHz will be explained later in this section.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pins upon the third rising edge of the RxOutClk output clock signal, following a given rising edge of the TxNibClk output clock signal.
- The XRT79L71 will pulse the TxNibFrame output pin "High" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble within a given DS3 frame.

Figure 36 presents an illustration of how to interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71, for Mode 4 operation.

FIGURE 36. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMING) OPERATION



Mode 4 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of both the 44.736MHz clock signal via the RxOutClk output signal and a Nibble Clock signal via the TxNibClk output signal.

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound DS3 data-stream in a Nibble-Parallel manner via its DS3_Data_Out[3:0] output pins. The user is advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its DS3_Nib_Clock_In input pin, as depicted below in Figure 37.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the RxOutClk signal following a given rising edge in the TxNibClk signal. The XRT79L71 will indicate that it is processing the very last nibble of a given DS3 frame by pulsing its TxNibFrame output pin "High" for one nibble-period. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_Start_of_Frame input pin, then it is

expected to begin the transmission of the contents of the very next outbound DS3 frame, via the DS3_Data_Out[3:0] output or TxNib[3:0] input pins.

The Transmit Payload Data Input Interface block's handling of DS3 Overhead bits when configured to operate in the Nibble-Parallel Mode

If one reviews the DS3 framing formats for both M13 and C-bit Parity in **Figure 14** and **Figure 15**, one will quickly note that the DS3 framing format is a bit-oriented framing format. More specifically, each of the DS3 framing formats consists of multiple strings of 84 consecutive bits of payload data that are separated from each other by a DS3 overhead bit. As a consequence, there will never be a case in which the Transmit Payload Data Input Interface, within the XRT79L71, will be processing a DS3 overhead nibble. In other words, the TxOH_Ind output pin has no meaning and will NOT be active whenever the XRT79L71 is configured to operate in both the DS3 and Nibble-Parallel Modes.

Whenever the user configures the Transmit Payload Data Input Interface to operate in the Nibble-Parallel Mode, then it will only handle or process DS3 payload bits. This statement brings us to the next topic.

The Frequency of TxNibClk for DS3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Transmit Payload Data Input Interface has been configured to operate in the Nibble-Parallel Mode, it will NOT process the DS3 overhead bits. Only DS3 payload data is processed through the Transmit Payload Data Input Interface (e.g., via the TxNib[3:0] input pins). As a consequence, the frequency of the TxNibClk signal will NOT simply be 44.736MHz/4 or 11.184MHz.

If we were to look at this issue another way, we would recall that each DS3 frame consists of 4760 bits. Of these bits, 56 are overhead bits and the remaining 4704 bits are payload bits. This means that there are 4704 bits/4 = 1176 nibbles of payload bits within each DS3 frame.

The frame repetition rate (for DS3) is 9.398kHz. Therefore, if one performs the following multiplication:

9398 Frames/sec · 1176 Nibble/Frame = 11.052MHz (for the Average Frequency of the TxNibClk output signal).

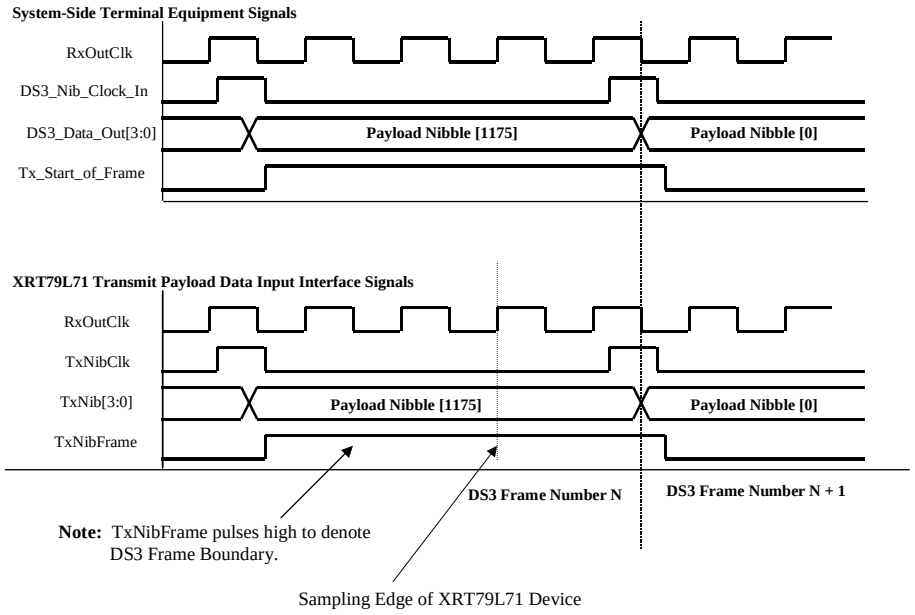
How 1176 Clock Edges within the TxNibClk output signal are distributed throughout a DS3 frame.

In general, for 1120 TxNibClk periods, the instantaneous frequency of the TxNibClk output clock signal will be 11.184MHz (e.g., each of these TxNibClk clock periods will corresponding to exactly 4 RxOutClk clock periods). However, for the remaining 56 of these TxNibClk periods, the periods of these clock signals will be lengthened to five (5) of these RxOutClk clock periods.

For this reason, if one were to monitor the TxNibClk signal via a scope the user would notice what appears to be a considerable amount of jitter within this particular clock signal.

Figure 37 presents an illustration of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 4 Operation.

FIGURE 37. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 4 (Nibble-Parallel/Loop-Timing) Mode

The user can configure the XRT79L71 to operate in Mode 4 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 36.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 0] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

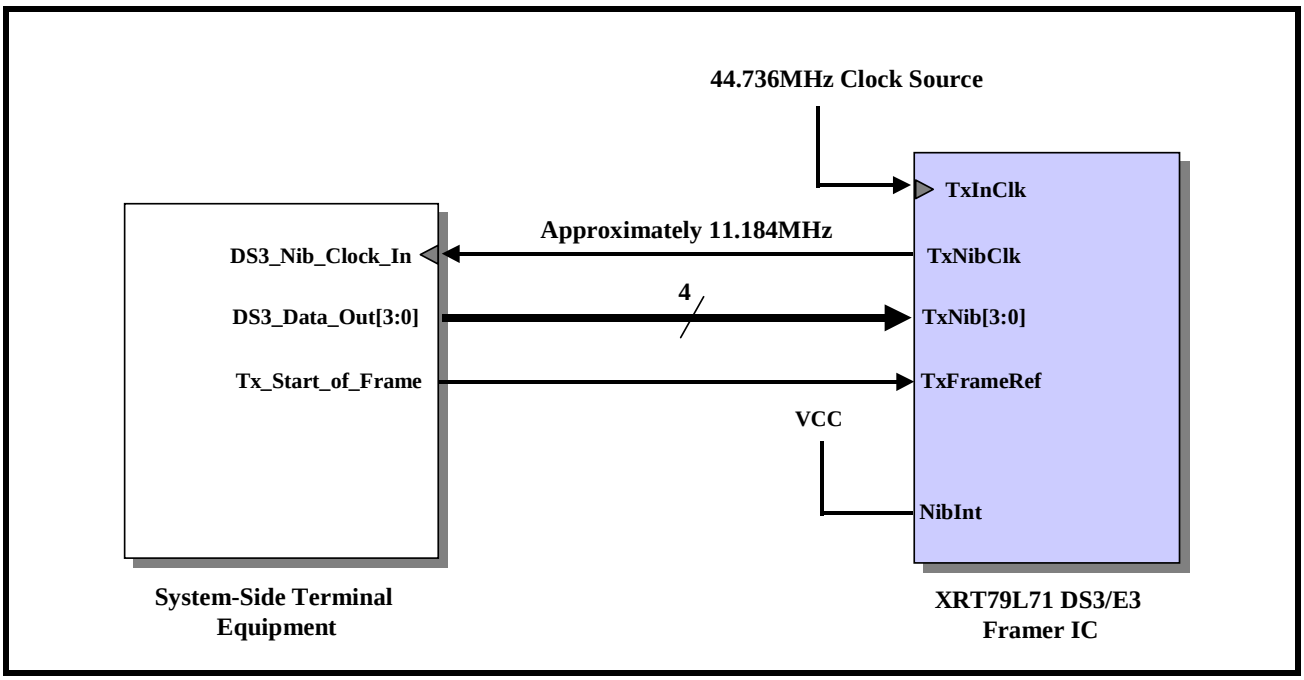
4.2.1.5 Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode Operation for the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 5 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- In this mode, the XRT79L71 will use the TxInClk signal to derive the TxNibClk signal.
- For DS3 Applications, the TxNibClk frequency is approximately one-fourth of the TxInClk clock input signal or 11.184MHz. The reason for the TxNibClk frequency not being exactly 11.184MHz will be explained later in this section.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pins upon the third rising edge of the TxInClk input clock signal, following a given rising edge of the TxNibClk output clock signal.
- The Transmit Section of the XRT79L71 will initiate the generation and transmission of a new DS3 frame anytime it detects a rising edge in the TxFrameRef input pin.
- The XRT79L71 will pulse the TxNibFrame output pin "High" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble within a given DS3 frame.

Figure 38 presents an illustration of how to interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 5 operation.

FIGURE 38. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Mode 5 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble Clock signal via the TxNibClk output signal.

NOTE: *For "Mode 5" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound DS3 data-stream, in a Nibble-Parallel manner via its DS3_Data_Out[3:0] output pins. The user is

advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its DS3_Nib_Clock_In input pin, as depicted below in **Figure 39**.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the TxInClk signal following a given rising edge in the TxNibClk signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a Framing Reference signal to the XRT79L71 by pulsing its TxFrameRef input pin "High" for one nibble-period, coincident with the first nibble of a new outbound DS3 frame being applied to the TxNib[3:0] input pins. Once the XRT79L71 detects the rising edge of the input at its TxFrameRef input pin, it will begin to generate and transmit a new DS3 frame.

NOTES:

1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of Frame Generation and is referred to as the Frame Master. Since the XRT79L71 does not control or dictate the instant that it will generate a new DS3 frame, but is driven by the System-Side Terminal Equipment, it is referred to as the Frame Slave.*
2. *If the XRT79L71 is configured to operate in Mode 5 then it is imperative that the Tx_Start_of_Frame or TxFrameRef signal is synchronized to the TxInClk input clock signal. Failure to do this will result in the transmission of erred DS3 data to the remote terminal equipment.*

The Transmit Payload Data Input Interface block's handling of DS3 Overhead bits when configured to operate in the Nibble-Parallel Mode

If one reviews the DS3 framing formats for both M13 and C-bit Parity in **Figure 14** and **Figure 15**, one will quickly note that the DS3 framing format is a bit-oriented framing format. More specifically, each of the DS3 framing formats consists of multiple strings of 84 consecutive bits of payload data that are separated from each other by a DS3 overhead bit. As a consequence, there will never be a case in which the Transmit Payload Data Input Interface, within the XRT79L71, will be processing a DS3 overhead nibble. In other words, the TxOH_Ind output pin has no meaning and will NOT be active whenever the XRT79L71 is configured to operate in both the DS3 and Nibble-Parallel Modes.

Whenever the user configures the Transmit Payload Data Input Interface block to operate in the Nibble-Parallel Mode, then it will only handle or process DS3 payload bits. This statement brings us to the next topic.

The Frequency of TxNibClk for DS3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Transmit Payload Data Input Interface has been configured to operate in the Nibble-Parallel Mode, it will NOT process the DS3 overhead bits. Only DS3 payload data is processed through the Transmit Payload Data Input Interface (e.g., via the TxNib[3:0] input pins). As a consequence, the frequency of the TxNibClk signal will NOT simply be 44.736MHz/4 or 11.184MHz.

If we were to look at this issue another way, we would recall that each DS3 frame consists of 4760 bits. Of these bits, 56 are overhead bits and the remaining 4704 bits are payload bits. This means that there are 4704 bits/4 = 1176 nibbles of payload bits within each DS3 frame.

The frame repetition rate (for DS3) is 9.398kHz. Therefore, if one performs the following multiplication:

(9398 Frames/sec X 1176 Nibble/Frame) = 11.052MHz (for the Average Frequency of the TxNibClk output signal).

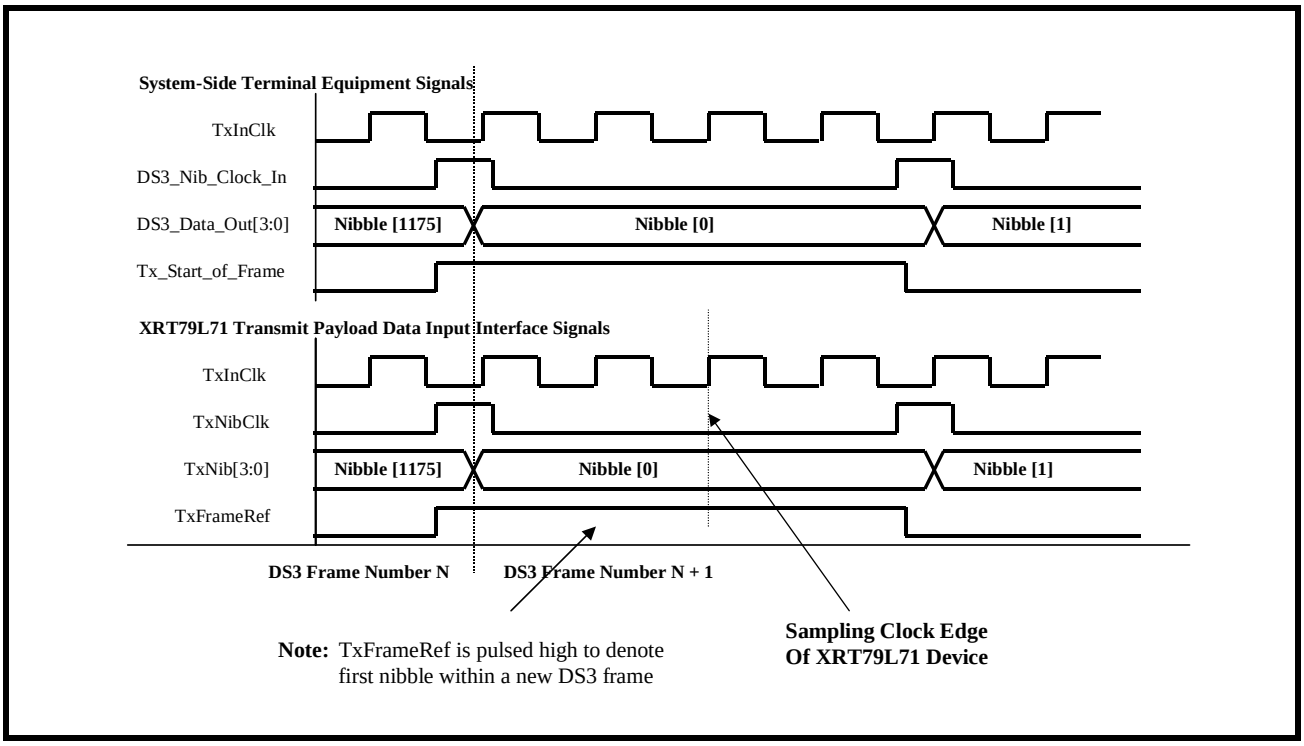
How 1176 Clock Edges within the TxNibClk output signal are distributed throughout a DS3 frame

In general, for 1120 TxNibClk periods, the instantaneous frequency of the TxNibClk output clock signal will be 11.184MHz (e.g., each of these clock periods will correspond to exactly 4 TxInClk clock periods). However, for the remaining 56 of these TxNibClk periods, the periods of these clock signals will be lengthened to five (5) of these TxInClk clock periods.

For this reason, if the TxNibClk signal was monitored via a scope what appears to be a considerable amount of jitter would be observed within this particular clock signal.

Figure 39 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 5 Operation.

FIGURE 39. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 5 (Nibble-Parallel/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 5 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 38.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 1] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

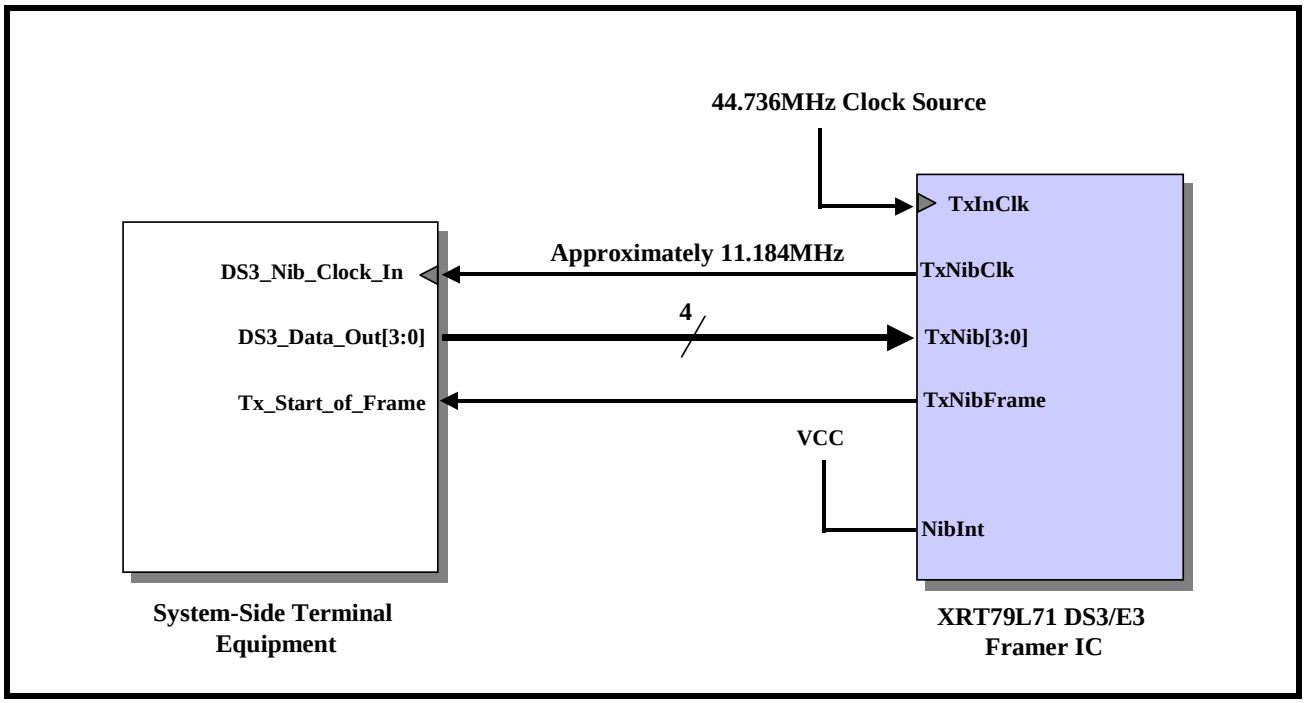
4.2.1.6 Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode Operation for the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 6 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- In this mode, the XRT79L71 will use the TxInClk signal to derive the TxNibClk signal.
- For DS3 Applications, the TxNibClk frequency is approximately one-fourth of the TxInClk clock input signal or 11.184MHz. The reason for the TxNibClk frequency not being exactly 11.184MHz will be explained later in this section.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pin upon the third rising edge of the TxInClk input clock signal, following a given rising edge of the TxNibClk output clock signal.
- The XRT79L71 will pulse the TxNibFrame output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given DS3 frame.

Figure 40 presents an illustration of how to interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 6 operation.

FIGURE 40. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Mode 6 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble-Clock signal via the TxNibClk output signal.

NOTE: *For "Mode 6" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound DS3 data-stream, in a Nibble-Parallel manner via its DS3_Data_Out[3:0] output pins. The user is advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the DS3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its DS3_Nib_Clock_In input pin, as depicted below in Figure 41.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the TxInClk signal following a given rising edge in the TxNibClk signal. The XRT79L71 will indicate that it is processing the very last nibble of a given DS3 frame by pulsing its TxNibFrame output pin "High" for one nibble-period.

Whenever the System-Side Terminal Equipment detects this pulse at its Tx_Start_of_Frame input pin, then it is expected to begin the transmission of the contents of the very next outbound DS3 frame, via the DS3_Data_Out[3:0] output or TxNib[3:0] input pins.

NOTES:

1. *1. In this particular mode, the Transmit Direction circuitry (within the XRT79L71) dictates the instant that it will generate a new DS3 frame; and is (therefore) referred to as the "Frame Master". As a consequence, this particular mode is referred to as the "Frame Master" Mode.*
2. *In contrast to "Mode 5" operation, if the XRT79L71 is configured to operate in "Mode 6", then it is NOT imperative that the "TxFrameRef" input be synchronized to the "TxInClk" clock signal. In this case, we recommend that the user tie "TxFrameRef" to GND.*

The Transmit Payload Data Input Interface block's handling of DS3 Overhead bits when configured to operate in the Nibble-Parallel Mode

If one reviews the DS3 framing formats for both M13 and C-bit Parity in **Figure 14** and **Figure 15**, one will quickly note that the DS3 framing format is a bit-oriented framing format. More specifically, each of the DS3 framing formats consists of multiple strings of 84 consecutive bits of payload data that are separated from each other by a DS3 overhead bit. As a consequence, there will never be a case in which the Transmit Payload Data Input Interface, within the XRT79L71, will be processing a DS3 overhead nibble. In other words, the TxOH_Ind output pin has no meaning and will NOT be active whenever the XRT79L71 is configured to operate in both the DS3 and Nibble-Parallel Modes.

Whenever the user configures the Transmit Payload Data Input Interface block to operate in the Nibble-Parallel Mode, then it will only handle or process DS3 payload bits. This statement brings us to the next topic.

The Frequency of TxNibClk for DS3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Transmit Payload Data Input Interface has been configured to operate in the Nibble-Parallel Mode, it will NOT process the DS3 overhead bits. Only DS3 payload data is processed through the Transmit Payload Data Input Interface (e.g., via the TxNib[3:0] input pins). As a consequence, the frequency of the TxNibClk signal will NOT simply be 44.736MHz/4 or 11.184MHz.

If we were to look at this issue another way, we would recall that each DS3 frame consists of 4760 bits. Of these bits, 56 are overhead bits and the remaining 4704 bits are payload bits. This means that there are 4704 bits/4 = 1176 nibbles of payload bits within each DS3 frame.

The frame repetition rate (for DS3) is 9.398kHz. Therefore, if one performs the following multiplication:

(9398 Frames/sec X 1176 Nibble/Frame) = 11.052MHz (for the Average Frequency of the TxNibClk output signal).

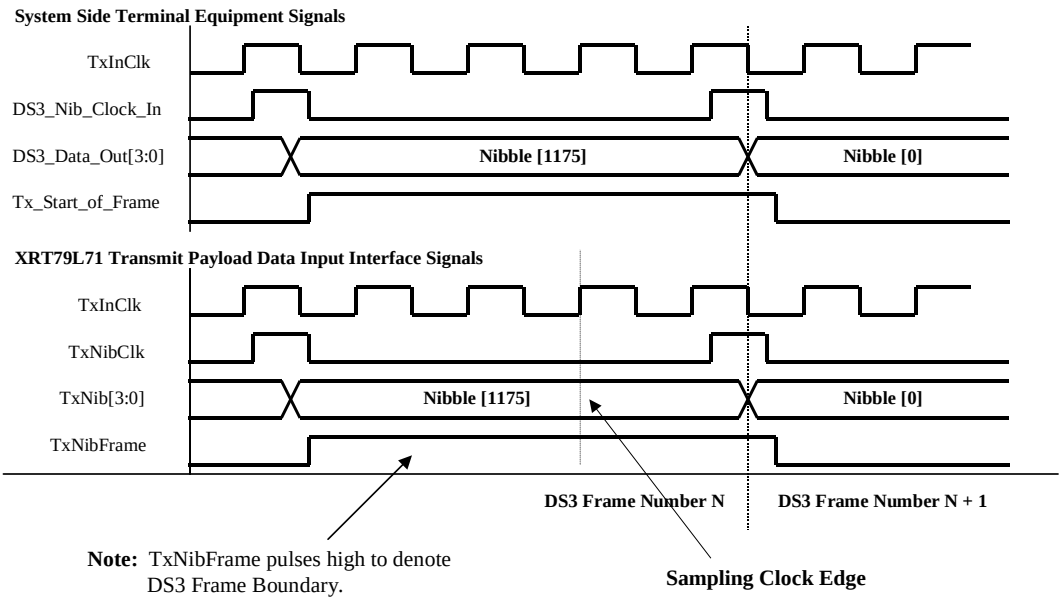
How 1176 Clock Edges within the TxNibClk output signal are distributed throughout a DS3 frame

In general, for 1120 TxNibClk periods the instantaneous frequency of the TxNibClk output clock signal will be 11.184MHz (e.g., each of these clock periods will correspond to exactly 4 TxInClk clock periods). However, for the remaining 56 of these TxNibClk periods, the periods of these clock signals will be lengthened to five (5) of these TxInClk clock periods.

For this reason, if the TxNibClk signal was monitored via a scope, what appears to be a considerable amount of jitter would be observed within this particular clock signal.

Figure 41 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 6 Operation.

FIGURE 41. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 6 (Nibble-Parallel/Local-Timing/Frame Master Mode)

The user can configure the XRT79L71 to operate in Mode 6 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 40.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [1, X] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	X

4.2.1.7 Operating the Transmit Payload Data Input Interface in the Gapped Clock Mode

If the Transmit Payload Data Input Interface, within the XRT79L71, has been configured to operate in any one of the Serial Modes (e.g., Modes 1 through 3, as described in Section 4.2.1.1, Section 4.2.1.2 and Section

4.2.1.3), then we have recommended that the user design or configure their System-Side Terminal Equipment to perform the following procedure when supplying payload data to the TxSer input pin.

- Check the state of the TxOH_Ind output pin from the XRT79L71 upon the falling edge of either the TxInClk or the RxOutClk signal.
- Perform either of the following actions, depending upon the sampled state of the TxOH_Ind output pin, as described below.

If TxOH_Ind is sampled "Low"

Then the System-Side Terminal Equipment should proceed to place the very next payload bit on the TxSer input pin upon the very next rising edge of either the TxInClk or RxOutClk signal.

If TxOH_Ind is sampled "High"

Then the System-Side Terminal Equipment should NOT proceed to place the very next payload bit on the TxSer input pin, upon the very next rising edge of either the TxInClk or RxOutClk signal. In this case, the System-Side Terminal Equipment should hold or NOT advance the very next payload bit to the TxSer input pin of the XRT79L71 until it samples the TxOH_Ind output "Low" once again.

In this particular approach, the user must design in the appropriate State Machine circuitry within the System-Side Terminal Equipment in order to properly respond to the state of the TxOH_Ind output pin, while providing the payload data to the Transmit Payload Data Input Interface. While designing such a State Machine into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode.

If the Transmit Payload Data Input Interface block has been configured to operate in the Gapped-Clock Mode

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the role of the TxOH_Ind output pin will change from being the Overhead Indicator output pin, to now being a demand-clock output pin. In other words, If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the Transmit Payload Data Input Interface block will generate a clock pulse via the TxOH_Ind output pin if and only if it is ready to accept and process a payload bit. If the Transmit Payload Data Input Interface block is about to process an overhead bit, then it will not generate a clock pulse via the TxOH_Ind output pin. This action will result in the Transmit Payload Data Input Interface block generating a gapped clock signal via the TxOH_Ind output pin, hence the term Gapped-Clock Mode.

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the System-Side Terminal Equipment will be expected to update the data on the TxSer input pin of the XRT79L71, upon the rising edge of the TxOH_Ind output signal. The XRT79L71 will sample and latch the TxSer data, upon the falling edge of the TxOH_Ind output signal. In this case, there is no need to check the state of a certain output pin, and then gate the placement of the next payload bit on the TxSer input pin with the sampled state of this particular signal. The System-Side Terminal Equipment only needs to respond to the rising edge of this particular Gapped-Clock signal.

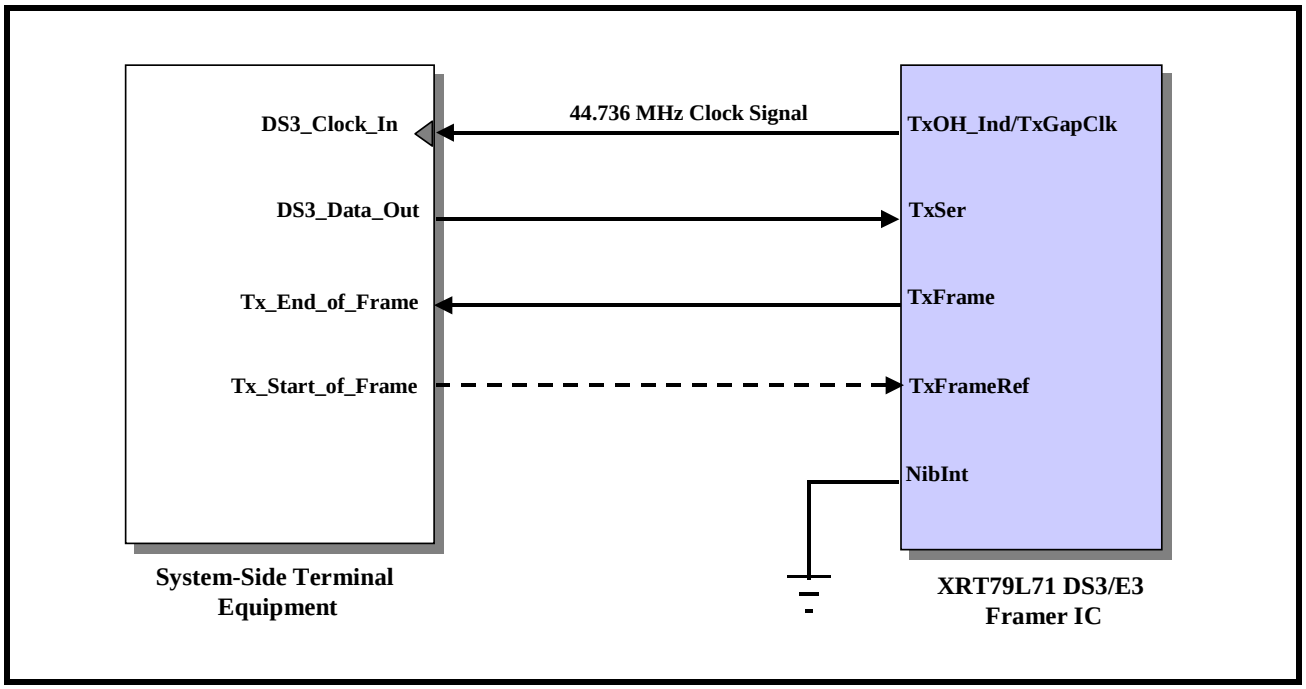
If the XRT79L71 has been configured to operate in the Loop-Timing Mode (e.g., Mode 1), then this Gapped-Clock signal from the TxOH_Ind output pin will be derived from the LIU Recovered Clock signal from the Receive DS3 LIU Block. Similarly, if the XRT79L71 has been configured to operate in the Local-Timing Mode, then this Gapped-Clock signal will be derived from the TxInClk input signal.

Configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode

To configure the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode do all of the following.

STEP 1 - The user should interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block, in a manner as indicated below in Figure 42.

FIGURE 42. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR GAPPED-CLOCK MODE OPERATIONS



STEP 2 - Set Bit 5 (TxGapped Clock Mode Enable), within the Framer Test Register to "1" as depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	0	1	0	0	0	0	0

NOTE: *This setting only configures the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode. Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode is discussed separately in Section 4.3.6.1.2.*

4.2.1.8 Accepting and Inserting DS3 Overhead Bits via the Transmit Payload Data Input Interface

By default, the Transmit DS3 Framer block will be configured to internally generate and insert all of the overhead bits within its outbound DS3 data-stream. More specifically, the DS3 Frame Generator block will internally generate the DS3 overhead bits by doing all of the following, as presented below in Table 15 and Table 16, for C-Bit Parity and the M13/M23 Framing formats, respectively.

TABLE 15: HOW THE TRANSMIT DS3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS - C-BIT PARITY APPLICATIONS

BIT NAME	BIT DESCRIPTION	HOW THE OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT DS3 FRAMER BLOCK
X-Bits (2)	FERF/Yellow Alarm Indicator Bits	Either Software Controlled or automatically set to "0" whenever the corresponding Receive DS3 Framer block declares the LOS, LOF/OOF or AIS defect condition.
F1 Bits (14)	F-Frame Framing Alignment bits that are of the value "1"	Set to the value of "1".
F0 Bits (14)	F-Frame Framing Alignment bits that are of the value "0".	Set to the value of "0".
M1 Bit (1)	M-Frame Framing Alignment bits that are of the value "1	Set to the value of "1".
M0 Bits (2)	M-Frame Framing Alignment bits that are of the value "0"	Set to the value of "0".
P-bits (2)	Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the two P-bit positions within the very next DS3 frame.
CP-bits (3)	Path Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the three CP-bit positions within the very next DS3 frame.
AIC bit (1)	Application Identical Channel	Set to the value "1" in order to denote C-bit Parity framing format.
UDL bits (9)	User Data Link Bits	Set to the value "1".
DL bits (3)	Path Maintenance Data Link (PMDL) bits	These bits carry the PMDL/LAPD Message that is generated by the LAPD Transmitter within the Transmit Section of the XRT79L71. However, if the Transmit LAPD Controller block is not being used, then these bits will each be set to "1".
FEAC bit (1)	Far-End Alarm & Control Bit	This bit carries the FEAC Message that is generated by the Transmit FEAC Processor within the Transmit Section of the XRT79L71. However, if the Transmit FEAC Controller Block is not being used, then this bit will be set to "1"
FEBE bits (3)	Far-End Block Error Bits	These bits are set to [1, 1, 1] whenever the corresponding Receive DS3 Framer block detects no CP nor framing F and M bit errors within its incoming DS3 data-stream. These bits are set to values other than [1, 1, 1] whenever the corresponding Receive DS3 Framer block detects CP or framing bit errors within its incoming DS3 data-stream. ***NOTE:*** *These bit-fields can also be software-controlled.*

TABLE 16: HOW THE TRANSMIT DS3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS - M13/M23 APPLICATIONS

BIT NAME	BIT DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT DS3 FRAMER BLOCK
X-Bits (2)	FERF/Yellow Alarm Indicator Bits	Either Software Controlled or automatically set to "0" whenever the corresponding Receive DS3 Framer block declares the LOS, LOF/OOF or AIS defect condition.
F1 Bits (14)	F-Frame Framing Alignment bits that are of the value "1"	Set to the value of "1".
F0 Bits (14)	F-Frame Framing Alignment bits that are of the value "0"	Set to the value of "0".
M1 Bit (1)	M-Frame Framing Alignment bits that are of the value "1".	Set to the value of "1".
M0 Bit (2)	M-Frame Framing Alignment bits that are of the value "0".	Set to the value of "0".
P-bits (2)	Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the two P-bit positions within the very next DS3 frame.
C-bits (21)	DS2 to DS3 Multiplexing Stuff Indicator bits	Set to the value of "0".

However, the Transmitter Section of the XRT79L71 can also be configured to externally accept values via a certain input port and to insert this data into certain select overhead bits, within the outbound DS3 data-stream. In this case, these "externally inserted" values for these overhead bits will overwrite that which has been internally generated by the Transmit DS3/E3 Framer block. The XRT79L71 permits the user to implement this overhead bit insertion by either of the following two methods.

- By configuring the Transmit Section of the XRT79L71 to accept DS3 overhead data via the Transmit Overhead Input Interface (this option will be discussed in Section 4.2.2).
- By configuring the Transmit Section of the XRT79L71 to accept DS3 overhead data via the Transmit Payload Data Input Interface block.

Configuring the Transmit Section of the XRT79L71 to Accept DS3 Overhead Bits via the Transmit Payload Input Interface

This section describes how one can configure the XRT79L71 to do the following.

1. To accept a DS3 data-stream that consists of both payload data and some DS3 overhead bits, and
2. To configure the Transmit Section of the XRT79L71 to accept a user specified set of overhead bits from the Transmit Payload Data Input Interface block and insert these overhead bits into the outbound DS3 data-stream, without modification.

For DS3 applications, this insertion of overhead data can be performed for all DS3 overhead bits, with the exception of the P, F and M bits. The Transmit DS3 Framer block will unconditionally internally generate these particular overhead bits.

The procedure for configuring the Transmit Framer block to externally accept and insert DS3 Overhead data Via the Transmit Payload Data Input Interface and insert this data into the specified overhead bit position is presented below.

STEP 1 - Set Bit 7 (TxOHSrc), within the Test Register to "1", as depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOHSrc	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	RxPRBS Lock	RxPRBS Enable	TxPRBS Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
1	0	0	0	0	0	0	0

Once the user executes this step, then the Transmit Section of the XRT79L71 will know that the Transmit Payload Data Input Interface - Serial Input is now a potential source of overhead bits. Further, this step automatically changes the functionality of the bits within the F-Bit Mask Registers as described below.

STEP 2 - Specify the overhead bits that the Frame Generator block should externally accept (via the Transmit Payload Data Input Interface - Serial Input) and insert into the outbound DS3 data-stream.

This is accomplished by setting the appropriate bits within the Transmit DS3 F-Bit Mask Registers to the value "0". The bit-format for the Transmit DS3 F-Bit Mask Registers whenever the TxOHSrc bit-field is set to "1" is presented below.

Transmit DS3 F-Bit Mask # 1 Register (Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				UDL Bit # 9 (C73)	UDL Bit # 8 (C72)	UDL Bit # 7 (C71)	Unused
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	X	X	X	0

Transmit DS3 F-Bit Mask # 2 Register (Address = 0x1137)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UDL Bit # 6 (C63)	UDL Bit # 5 (C62)	UDL Bit # 4 (C61)	Unused	DL Bit # 3 (C53)	DL Bit # 2 (C52)	DL Bit # 1 (C51)	Unused
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	0

Transmit DS3 F-Bit Mask # 3 Register (Address = 0x1138)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FEBE Bit 3 (C43)	FEBE Bit 2 (C42)	FEBE Bit 1 (C41)	Unused	CP Bit # 3 (C33)	CP Bit # 2 (C32)	CP Bit # 1 (C31)	Unused
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	X	X	X	0

Transmit DS3 F-Bit Mask # 4 Register (Address = 0x1139)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UDL Bit # 3 (C23)	UDL Bit # 2 (C22)	UDL Bit # 1 (C21)	X Bit # 2	FEAC Bit (C13)	NA Bit (C12)	AIC Bit (C11)	X Bit # 1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

Setting these bit-fields to "0" configures the Transmit Section of the XRT79L71 to accept the corresponding overhead bit(s) via the Transmit Payload Data Input Interface - Serial Input and to insert it/them into the corresponding overhead bits, within the outbound DS3 data-stream. Conversely, setting these bit-fields to "1" configures the Transmit Section of the XRT79L71 to (1) ignore value(s) of the corresponding overhead bits within the data stream (that is being accepted by the Transmit Payload Data Input Interface block) and (2) to allow the Transmit DS3/E3 Framer block to internally generate the corresponding overhead bit(s).

Why take advantage of this feature?

suppose that the user wishes to pass a Channelized DS3 data-stream that is of the M23 framing format through the Transmit Section of the XRT79L71. If this DS3 signal is of the channelized type (e.g., it is carrying either 28 DS1 or 21 E1 signals) and is of the M23 framing format, then the C-bits within this DS3 data-stream contains important bit-stuffing information that was generated by the Source or M13 Multiplexer terminal where the seven (7) DS2 signals were multiplexed together thereby creating this DS3 signal. If the value of these C-bits are lost or overwritten by the time that this particular DS3 signal reaches the M13 De-Multiplexer terminal, then the M13 De-Multiplexer terminal will not be able to de-multiplex out the seven DS2 signals from this DS3 signals. As a consequence, the 28 DS1 or 21 E1 signals that are being carried within this DS3 signal will be lost. Therefore, whenever a user is handling a channelized DS3 signal that is of the M23 framing format, the user must make sure that that value of the C-bits within this DS3 data-stream are not altered.

Problem with the Transmit DS3 Framer Block

As our channelized, M23 framing format DS3 signal is routed through the Transmit Section of the XRT79L71, it will first pass through the Transmit Payload Data Input Interface block. Afterwards, this signal will now pass through the Transmit DS3 Framer block.

According to **Table 16**, as this DS3 signal passes through the Transmit DS3 Framer block, then all of the following operations will occur.

- The Framing (F and M) bits will be re-inserted into this outbound DS3 data-stream.
- P-bits will be computed and will be re-inserted into this outbound DS3 data-stream.
- The X-bits will be set to the appropriate value depending upon whether the corresponding block is declaring the LOS, OOF/LOF or AIS defects, or not.
- The C-bits will be set to the value "0".

The last item in the above list (e.g., The C-bits will be set to the value "0") creates a problem for our Channelized, M23 Framing format DS3 signal.

What can be done about this?

Fortunately, there is a way to preserve the value of the C-bits, as this Egress Direction DS3 signal passes through the Transmit DS3 Framer block. In short, the user can preserve the value of these C-bits by configuring the Transmit DS3 Framer block to accept and externally insert certain overhead bits that it receives from the Transmit Payload Interface block into its outbound DS3 data-stream.

More specifically, this requirement can easily be achieved by taking advantage of this ability to externally insert the overhead bits into the outbound DS3 data-stream. In this specific case, the user can configure the Transmit DS3 Framer block to simply externally accept the C bits within the DS3 data-stream that it receives,

via the up-stream path from the Transmit Payload Data Input Interface block, and to insert these same C bits values into its outbound DS3 data-stream. This can be accomplished by executing the following steps.

STEP 1 - Write the value "1" into Bit 7 (TxOHSrc), within the Test Register at depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TXOH SOURCE	RECEIVE GAPPED CLOCK MODE ENABLE	TRANSMIT GAPPED CLOCK MODE ENABLE	RECEIVE PRBS LOCK	RECEIVE PRBS DETECTOR ENABLE	TRANSMIT PRBS GENERATOR ENABLE	UNUSED	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
1	0	0	0	0	0	0	0

STEP 2 - Write the value [0, 0, 0] into Bits 1 through 3 (C71 through C73) within the Transmit DS3 F-Bit Mask # 1 Register, as depicted below.

Transmit DS3 F-Bit Mask # 1 Register (Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				UDL Bit # 9(C73)	UDL Bit # 8(C72)	UDL Bit # 7(C71)	Unused
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 3 - Write the value [0, 0, 0] into Bits 1 through 3 (C51 through C53) and the value [0, 0, 0] into Bits 5 through 7 (C61 through C63) within the Transmit DS3 F-Bit Mask # 2 Register, as depicted below.

Transmit DS3 F-Bit Mask # 2 Register (Indirect Address = 0xnE, 0x37; Direct Address = 0x1137)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UDL Bit # 6 (C63)	UDL Bit # 5 (C62)	UDL Bit # 4 (C61)	Unused	DL Bit # 3 (C53)	DL Bit # 2 (C52)	DL Bit # 1 (C51)	Unused
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 4 - Write the value [0, 0, 0] into Bits 1 through 3 (C31 through C33), and the value [0, 0, 0] into Bits 5 through 7 (C41 through C43) within the Transmit DS3 F-Bit Mask # 3 Register, as depicted below.

Transmit DS3 F-Bit Mask # 3 Register (Indirect Address = 0xnE, 0x37; Direct Address = 0x1138)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FEBE Bit 3 (C43)	FEBE Bit 2 (C42)	FEBE Bit 1 (C41)	Unused	CP Bit # 3 (C33)	CP Bit # 2 (C32)	CP Bit # 1 (C31)	Unused
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	X	0	0	0	X

STEP 5 - Write the value [0, 0, 0] into Bits 1 through 3 (C11 through C13) and the value [0, 0, 0] into Bits 5 through 7 (C21 through C23) within the Transmit DS3 F-Bit Mask # 4 Register, as depicted below.

Transmit DS3 F-Bit Mask # 4 Register (Indirect Address = 0xnE, 0x39; Direct Address = 0x1139)

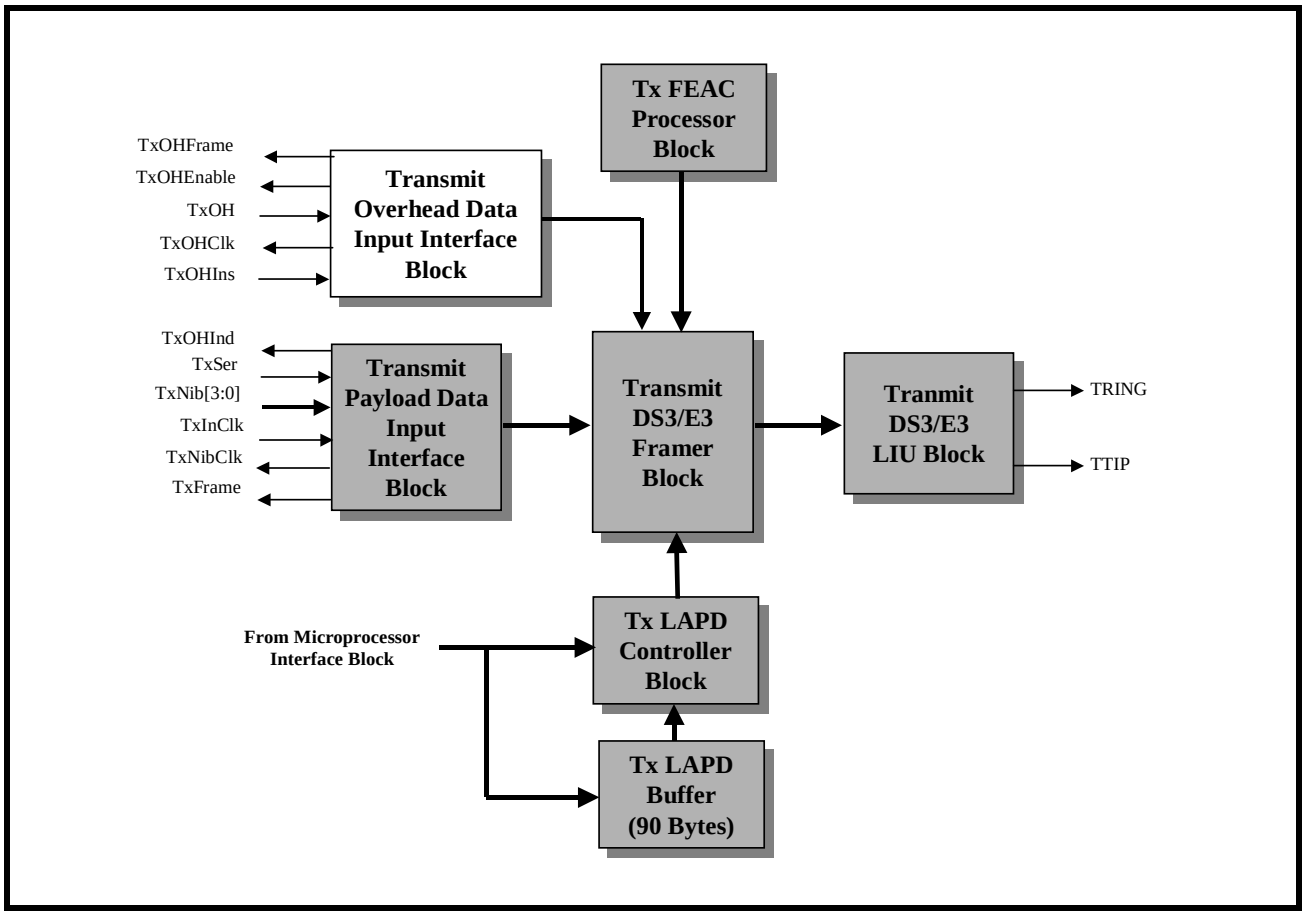
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UDL Bit # 3 (C23)	UDL Bit # 2 (C22)	UDL Bit # 1 (C21)	X Bit # 2	FEAC Bit (C13)	NA Bit (C12)	AIC Bit (C11)	X Bit # 1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	1

Once the user executes these five steps, then the Transmit DS3 Framer block will simply (1) accept the C-bit values from the DS3 data-stream that it receives from the Transmit Payload Interface block, and (2) will insert these same values into the C-bit positions within its outbound DS3 data-stream.

4.2.2 TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

The Transmit Overhead Data Input Interface block is the second functional block within the Transmit Direction of the XRT79L71 that we will discuss for DS3 Clear-Channel Framer Applications. **Figure 43** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit Overhead Data Input Interface block highlighted.

FIGURE 43. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK HIGHLIGHTED)



Some Background Information

In order to fully understand the role of the Transmit Overhead Data Input Interface, some background information needs to be discussed first.

As mentioned in Section 4.1, the DS3 frame consists of 4760 bits. Of these bits, 4704 bits are payload bits and the remaining 56 bits are overhead bits. The XRT79L71 has been designed to handle and process both the payload type and overhead type of bits for each DS3 frame. Within the XRT79L71, the Transmit Payload Data Input Interface Block (which was discussed in considerable detail in Section 4.2.1) has been designed to accept the payload data from the System-Side Terminal Equipment. Likewise, the Transmit Overhead Data Input Interface block has been designed to handle and process the overhead bits.

Accepting and Inserting DS3 Overhead Bits via the Transmit Overhead Data Input Interface

By default, the Transmit DS3 Framer block will be configured to internally generate and insert all of the overhead bits within its outbound DS3 data-stream. More specifically, the Transmit DS3 Framer block will internally generate the DS3 overhead bits by doing all of the following, as presented below in Table 17 and Table 18 for C-Bit Parity and the M23 Framing formats, respectively.

TABLE 17: HOW THE TRANSMIT DS3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS - C-BIT PARITY APPLICATIONS

BIT NAME	BIT DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT DS3 FRAMER BLOCK
X-Bits (2)	FERF/Yellow Alarm Indicator Bits	Either Software Controlled or automatically set to "0" whenever the corresponding Receive DS3 Framer block declares the LOS, LOF/OOF or AIS defect condition.
F1 Bits (14)	F-Frame Framing Alignment bits that are of the value "1"	Set to the value of "1".
F0 Bits (14)	F-Frame Framing Alignment bits that are of the value "0".	Set to the value of "0".
M1 Bit (1)	M-Frame Framing Alignment bits that are of the value "1"	Set to the value of "1".
M0 Bits (2)	M-Frame Framing Alignment bits that are of the value "0"	Set to the value of "0".
P-bits (2)	Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the two P-bit positions within the very next DS3 frame.
CP-bits (3)	Path Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the three CP-bit positions within the very next DS3 frame.
AIC bit (1)	Application Identical Channel	Set to the value "1" in order to denote C-bit Parity framing format.
UDL bits (9)	User Data Link Bits	Set to the value "1".
DL bits (3)	Path Maintenance Data Link (PMDL) bits	These bits carry the PMDL/LAPD Message that is generated by the LAPD Transmitter within the Transmit Section of the XRT79L71. However, if the Transmit LAPD Controller block is not being used, then these bits will each be set to "1".

TABLE 17: HOW THE TRANSMIT DS3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS - C-BIT PARITY APPLICATIONS

BIT NAME	BIT DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT DS3 FRAMER BLOCK
FEAC bit (1)	Far-End Alarm & Control Bit	This bit carries the FEAC Message that is generated by the Transmit FEAC Processor within the Transmit Section of the XRT79L71. However, if the Transmit FEAC Controller block is not being used, then these bits will each be set to "1".
FEBE bits (3)	Far-End Block Error Bits	These bits are set to [1, 1, 1] whenever the corresponding Receive DS3 Framer block detects no CP nor framing F and M bit errors within its incoming DS3 data-stream. These bits are set to values other than [1, 1, 1] whenever the corresponding Receive DS3 Framer block detects CP or framing bit errors within its incoming DS3 data-stream. ***NOTE:*** *These bit-fields can also be software-controlled.*

TABLE 18: HOW THE TRANSMIT DS3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS - M23 APPLICATIONS

BIT NAME	BIT DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT DS3 FRAMER BLOCK
X-Bits (2)	FERF/Yellow Alarm Indicator Bits	Either Software Controlled or automatically set to "0" whenever the corresponding Receive DS3 Framer block declares the LOS, LOF/OOF or AIS defect condition.
F1 Bits (14)	F-Frame Framing Alignment bits that are of the value "1"	Set to the value of "1".
F0 Bits (14)	F-Frame Framing Alignment bits that are of the value "0"	Set to the value of "0".
M1 Bit (1)	M-Frame Framing Alignment bits that are of the value "1".	Set to the value of "1".
M0 Bit (2)	M-Frame Framing Alignment bits that are of the value "0".	Set to the value of "0".
P-bits (2)	Parity Bits	Transmit DS3 Framer block computes the even parity value over the payload bits within a given DS3 frame. The results of this calculation are inserted into the two P-bit positions within the very next DS3 frame.
C-bits (21)	DS2 to DS3 Multiplexing Stuff Indicator bits	Set to the value of "0".

However, the Transmit Section of the XRT79L71 can also be configured to externally accept values via a certain input port and to insert this data into certain select overhead bits, within the outbound DS3 data-stream. In this case, these "externally inserted" values for these overhead bits will overwrite that which has been internally generated by the Transmit DS3/E3 Framer block. The XRT79L71 permits the user to implement this overhead bit insertion by either of the following two methods.

- By configuring the Transmit Section of the XRT79L71 to accept DS3 overhead data via the Transmit Payload Data Input Interface block (as was discussed in Section 4.2.1.8).
- By configuring the Transmit Section of the XRT79L71 to accept DS3 overhead data via the Transmit Overhead Data Input Interface (This approach will be discussed below).

The purpose of the Transmit Overhead Data Input Interface block is to accept overhead data from some system-side or up-stream source and to overwrite the contents of the overhead bits, within the outbound DS3 data-stream with these particular overhead bit values.

In order to accomplish this, the Transmit Overhead Data Input Interface block has numerous input and output pins. **Table 19** presents a list and a brief definition of each of these pins.

TABLE 19: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxOHEnable	A12	O	**Transmit Overhead Data - Enable Output Pin:** The Transmit Overhead Data Input Interface will assert this signal (e.g., pulse it "High") for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface block is processing an overhead bit. This output pin will remain "Low" at all other times.
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input Pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the Transmit Circuitry within the XRT79L71.
TxOHFrame	B11	O	**Transmit Overhead Input Interface Enable Input Pin:** This output pin pulses "High" whenever the Transmit Overhead Data Input Port is processing the last bit within a given DS3 frame.
TxOHIns	D10	I	**Transmit Overhead Data - Insert Enable Input Pin:** Asserting this input pin (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the System-Side Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data on the TxOH input pin, upon the falling edge of either the TxInClk or the TxOHClk clock signals. Setting this input pin "Low" configures the Transmit Overhead Data Input Interface block to NOT sample (e.g., ignore) the data on the TxOH input pin upon the falling edge of either the TxInClk or the TxOHClk clock signals. ***NOTE:*** *If the System-Side Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface block (e.g., if the System-Side Terminal Equipment asserts the TxOHIns input pin at a time whenever one of the non-insertable overhead bits are being processed), then this particular insertion effort will be ignored.*

TABLE 19: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxOH	C11	I	**Transmit Overhead Data Input Interface - Overhead Data Input Pin:** The Transmit Overhead Data Input Interface block accepts the overhead data via this input pin and inserts this data into the appropriate overhead bit-position within the very next outbound DS3 frame. If the TxOHIns input pin is pulled "High", then the Transmit Overhead Data Input Interface block will sample the data, residing on this input pin, upon either the rising edge of TxInClk or the falling edge of TxOHClk depending upon the Insertion Method used. If the TxOHIns input pin is pulled "Low", then the Transmit Overhead Data Input Interface block will NOT sample the data, residing on this input pin, upon the rising edge of TxInClk nor on the falling edge of TxOHClk.
TxOHClk	B13	O	**Transmit Overhead Clock Output:** This output pin functions as the "Transmit Overhead Data Input Interface" clock signal. If the Transmit Overhead Data Input Interface block is enabled by asserting the TxOHIns input pin, then the Transmit Overhead Data Input Interface block will sample and latch the data (residing on the "TxOH" input pin) upon the falling edge of this signal.

The Two Methods of Inserting Overhead Data Into the Transmit Overhead Data Input Interface Block

There are two methods that can be used to insert the overhead data into the Transmit Overhead Data Input Interface Block. One Method is referred to as Method 1 or the TxOHClk Method, and the other method is referred to as Method 2 or the TxInClk Method. Each of these methods is described in considerable detail below.

4.2.2.1 Operating the Transmit Overhead Data Input Interface using Method 1 - The TxOHClk Method

This particular method is referred to as the TxOHClk method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHClk clock output signal from the Transmit Overhead Data Input Interface block to clock overhead data onto the TxOH input pin.

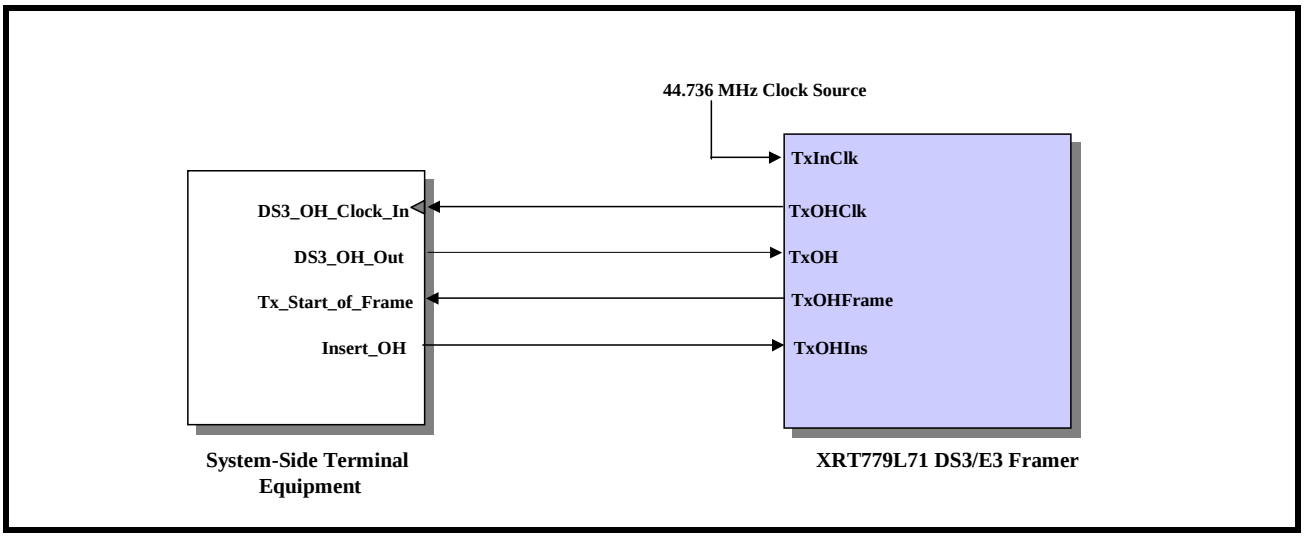
b. The Transmit Overhead Data Input Interface block will use the falling edge of the TxOHClk clock output signal to sample and latch the data residing on the TxOH input pin.

If Method 1 is used, the System-Side Terminal Equipment will need to interface to the following Transmit Overhead Data Input Interface pins.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

If Method 1 is used the users system must be designed such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in **Figure 44**.

FIGURE 44. ILLUSTRATION ON HOW ONE SHOULD INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 1



Method 1 Operation of the Transmit Overhead Data Input Interface Block

To operate the Transmit Overhead Data Input Interface block, design/configure the System-Side Terminal Equipment to continuously execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the state of the TxOHFrame output pin from the XRT79L71 upon the rising edge of the TxOHClk clock signal which is also output from the XRT79L71. Whenever the System-Side Terminal Equipment samples the TxOHFrame output pin "High" then it will know that the Transmit Overhead Data Input Interface block is ready to accept overhead bits for a new DS3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHFrame signal, it must also keep track of the number of rising edges within the TxOHClk signal that have occurred since the last time TxOHFrame was sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bit is currently being processed by the Transmit Overhead Data Input Interface block at any given TxOHClk clock period. When the System-Side Terminal Equipment knows which overhead bit is being processed within a given TxOHClk clock period, it can decide when to insert the appropriate bit-value into Transmit Overhead Data Input Interface block and in-turn, force the Transmit DS3/E3 Framer block to insert this bit into the appropriate overhead bit-position within the outbound DS3 data-stream. From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin of the XRT79L71.

Table 20 relates the number of rising clock edges, within the TxOHClk output signal, since the TxOHFrame output signal was sampled "High" to the DS3 Overhead Bit being processed by the Transmit Overhead Data Input Interface block. The user can use this table as a guide for inserting the appropriate overhead bits, within the outbound DS3 data-stream, for Method 1.

TABLE 20: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71 AND INSERTED INTO THE OUTBOUND DS3 DATA-STREAM?
0 (TxOHClk Clock Edge is coincident with the TxOHFrame signal being sampled "High")	X Bit # 1	YES
1	F1	NO
2	AIC (C11)	YES
3	F0	NO
4	NA (C12)	YES
5	F0	NO
6	FEAC (C13)	YES
7	F1	NO
8	X Bit # 2	YES
9	F1	NO
10	UDL Bit # 1 (C21)	YES
11	F0	NO
12	UDL Bit # 2 (C22)	YES
13	F0	NO
14	UDL Bit # 3 (C23)	YES
15	F1	NO
16	P	NO
17	F1	NO
18	CP Bit # 1 (C31)	YES
19	F0	NO
20	CP Bit # 2 (C32)	YES
21	F0	NO
22	CP Bit # 3 (C33)	YES
23	F1	NO
24	P	NO
25	F1	NO
26	FEBE # 1 (C41)	YES
27	F0	NO
28	FEBE # 2 (C42)	YES

TABLE 20: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71 AND INSERTED INTO THE OUTBOUND DS3 DATA-STREAM?
29	F0	NO
30	FEBE # 3 (C43)	YES
31	F1	NO
32	M0	NO
33	F1	NO
34	DL Bit # 1 (C51)	YES
35	F0	NO
36	DL Bit # 2 (C52)	YES
37	F0	NO
38	DL Bit # 3 (C53)	YES
39	F1	NO
40	M1	NO
41	F1	NO
42	UDL Bit # 4 (C61)	YES
43	F0	NO
44	UDL Bit # 5 (C62)	YES
45	F0	NO
46	UDL Bit # 6 (C63)	YES
47	F1	NO
48	M0	NO
49	F1	NO
50	UDL Bit # 7 (C71)	YES
51	F0	NO
52	UDL Bit # 8 (C72)	YES
53	F0	NO
54	UDL Bit # 9 (C73)	YES
55	F1	NO

NOTES:

1. *The shaded rows designate those Overhead bits that the XRT79L71 will permit the user to insert into the outbound DS3 data-streamvia the Transmit Overhead Data Input Interface block.*
2. *The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound DS3 data-stream.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of clock edges from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal by pulling it "High". Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit to be inserted into the outbound DS3 data-stream onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data residing on the TxOH input pin upon the very next falling edge of the TxOHClk output signal.

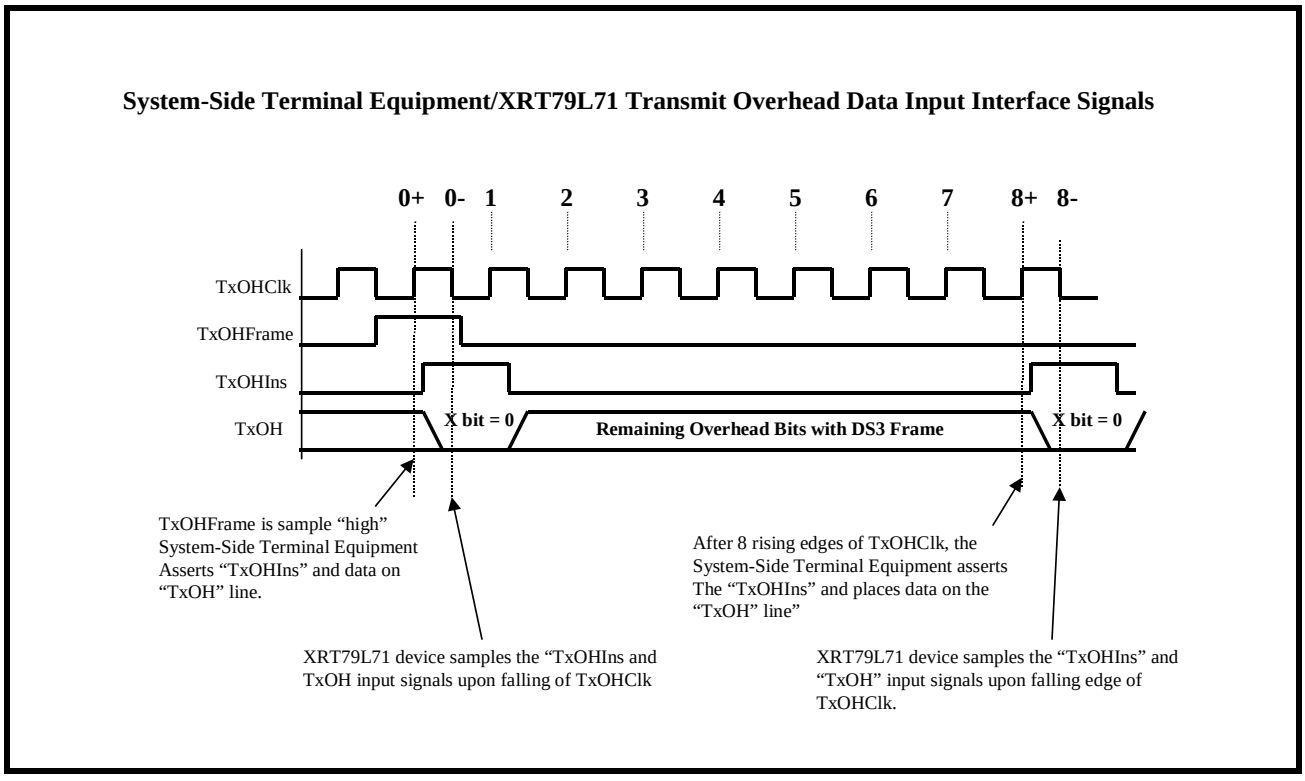
TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until the next rising edge of TxOHClk is detected. Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHClk period comes up, for inserting an overhead bit into the outbound DS3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 1 in order to transmit the FERF (Far-End Receive Failure) indicator to the remote terminal equipment.

For DS3 applications, the FERF indication is transmitted by setting both the X-bits within each outbound DS3 frame to "0".

If we were to assume that the connection between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface block of the XRT79L71 is as illustrated in **Figure 44**, then **Figure 45** presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface, when using Method 1.

FIGURE 45. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 1)



In Figure 45 the System-Side Terminal Equipment samples the TxOHFrame signal being "High" during Rising Clock Edge # 0+ with the TxOHClk signal. At this point, the System-Side Terminal Equipment knows that the XRT79L71 is just about to process the very first overhead bit within a given outbound DS3 frame. According to Table 20, the very first overhead bit to be processed within a given DS3 frame is the first X bit. In order to facilitate the transmission of the FERF indicator, the System-Side Terminal Equipment must set this particular bit-field to "0". The System-Side Terminal Equipment begins this process by implementing the following two tasks concurrently.

TASK 1 - The System-Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "0".

After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample the TxOHIns input pin being "High" and the TxOH input pin being set "Low" during the very next falling edge of TxOHClk (Clock Edge # 0-, in Figure 45). Once the XRT79L71 has sampled these two signals, it will then insert a "0" into the very first X bit-position, within the outbound DS3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as Clock Edge # 1, in Figure 45), the System-Side Terminal Equipment will negate or de-assert the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface block until Rising Clock Edge # 8. According to Table 20, the occurrence of Rising Clock Edge # 8+ indicates that the XRT79L71 is just about ready to process the second X bit, within the outbound DS3 frame. In order to facilitate this transmission of the FERF indicator, this particular X bit must also be set to "0". The System-Side Terminal Equipment begins this process by implementing the following two tasks concurrently.

TASK 1 - The System-Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "0".

After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample the TxOHIns' input pin being "High" and the TxOH input pin being set "Low" during the very next falling edge of TxOHClk (Clock Edge # 8- in Figure 45). Once the XRT79L71 has sampled these two signals, it will then insert a "0" into the second X bit-position within the outbound DS3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal, the System-Side Terminal Equipment will negate or de-assert the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface block until it samples the TxOHFrame output pin "High". Afterwards, the System-Side Terminal Equipment will repeat all of the steps that have been outlined in this case study.

4.2.2.2 Operating the Transmit Overhead Data Input Interface block using Method 2 - The TxInClk/ TxOHEnable Method

This particular method is referred to as the TxInClk/TxOHEnable method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHEnable output pin from the Transmit Overhead Data Input Interface block to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface at any given time.

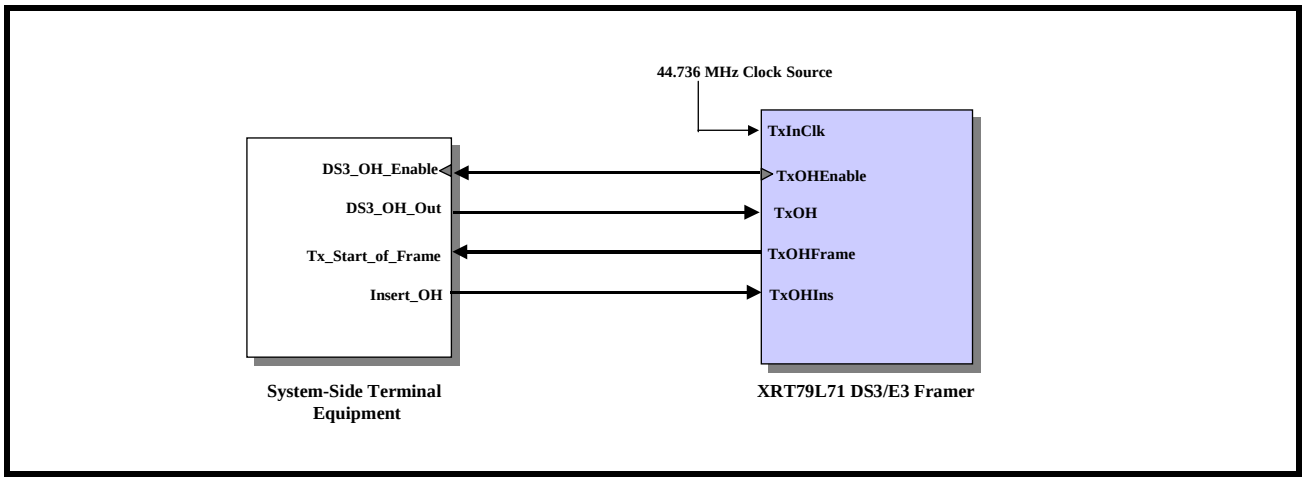
b. The Transmit Overhead Data Input Interface block will use the rising edge of TxInClk in order to sample and latch the data residing on the TxOH input pin.

If Method 2 is used, then the System-Side Terminal Equipment will need to interface to the following Transmit Overhead Data Input Interface pins.

- TxOH
- TxOHFrame
- TxInClk
- TxOHEnable
- TxOHIns

If Method 2 is used the users system must be designed such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in Figure 46.

FIGURE 46. ILLUSTRATION ON HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 2



Method 2 Operation of the Transmit Overhead Data Input Interface Block

Tto operate the Transmit Overhead Data Input Interface block per Method 2, design/configure the System-Side Terminal Equipment to continuously execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the states of both the TxOHFrame and the TxOHEnable output pins from the XRT79L71 upon the falling edge of the TxInClk clock input signal. The XRT79L71 will pulse the TxOHEnable output pin "High" for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface block is processing an overhead bit. Therefore, if the System-Side Terminal Equipment samples the TxOHEnable output pin "High", then it knows that an overhead bit insertion opportunity via the Transmit Overhead Data Input Interface block is just about to occur. If the System-Side Terminal Equipment samples both the TxOHEnable and the TxOHFrame output pins "High" at the same time, then it knows that the Transmit Overhead Data Input Interface block is just about to process the very first overhead bit (in this case an X bit) within a new DS3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHEnable and TxOHFrame signals, it must also keep track of the number of times that the TxOHEnable output pin has been sampled "High" since the last time both the TxOHEnable and the TxOHFrame output pins have been sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bits are currently being processed by the Transmit Overhead Data Input Interface block at any given TxOHEnable assertion. When the System-Side Terminal Equipment knows which overhead bit it being processed within a given TxOHEnable assertion period, it can decide when to insert the appropriate bit-value into the Transmit Overhead Data Input Interface block and in-turn, force the Transmit DS3/E3 Framer block to insert this bit into the appropriate overhead bit-position within the outbound DS3 data-stream. From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and placd the appropriate value on the TxOH input pin of the XRT79L71.

Table 21 relates the number of TxOHEnable output pulses that have occurred since both the TxOHFrame and the TxOHEnable pins were sampled "High", to the DS3 Overhead Bit being processed by the Transmit Overhead Data Input Interface block. This user can use this table as a guide for inserting the appropriate overhead bits, within the outbound DS3 data-stream, for Method 2.

TABLE 21: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND DS3 DATA-STREAM?
0 (TxOHEnable and TxOHFrame are sampled "High" simultaneously)	X Bit # 1	YES
1	F1	NO
2	AIC (C11)	YES
3	F0	NO
4	NA (C12)	YES
5	F0	NO
6	FEAC (C13)	YES
7	F1	NO
8	X Bit # 2	YES
9	F1	NO
10	UDL Bit # 1 (C21)	YES
11	F0	NO
12	UDL Bit # 2 (C22)	YES
13	F0	NO
14	UDL Bit # 3 (C23)	YES
15	F1	NO
16	P	NO
17	F1	NO
18	CP Bit # 1 (C31)	YES
19	F0	NO
20	CP Bit # 2 (C32)	YES
21	F0	NO
22	CP Bit # 3 (C33)	YES
23	F1	NO
24	P	NO
25	F1	NO
26	FEBE # 1 (C41)	YES
27	F0	NO
28	FEBE # 2 (C42)	YES

TABLE 21: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND DS3 DATA-STREAM?
29	F0	NO
30	FEBE # 3 (C43)	YES
31	F1	NO
32	M0	NO
33	F1	NO
34	DL Bit # 1 (C51)	YES
35	F0	NO
36	DL Bit # 2 (C52)	YES
37	F0	NO
38	DL Bit # 3 (C53)	YES
39	F1	NO
40	M1	NO
41	F1	NO
42	UDL Bit # 4 (C61)	YES
43	F0	NO
44	UDL Bit # 5 (C62)	YES
45	F0	NO
46	UDL Bit # 6 (C63)	YES
47	F1	NO
48	M0	NO
49	F1	NO
50	UDL Bit # 7 (C71)	YES
51	F0	NO
52	UDL Bit # 8 (C72)	YES
53	F0	NO
54	UDL Bit # 9 (C73)	YES
55	F1	NO

NOTE: *The shaded rows designate those Overhead bits that the XRT79L71 CAN insert into the outbound DS3 data-stream. The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound DS3 data-stream.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of TxOHEnable pulses from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal by pulling

it "High". Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit to be inserted into the outbound DS3 data-stream onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data residing on the TxOH input pin upon the second rising edge of TxInClk after TxOHEnable was sampled "High".

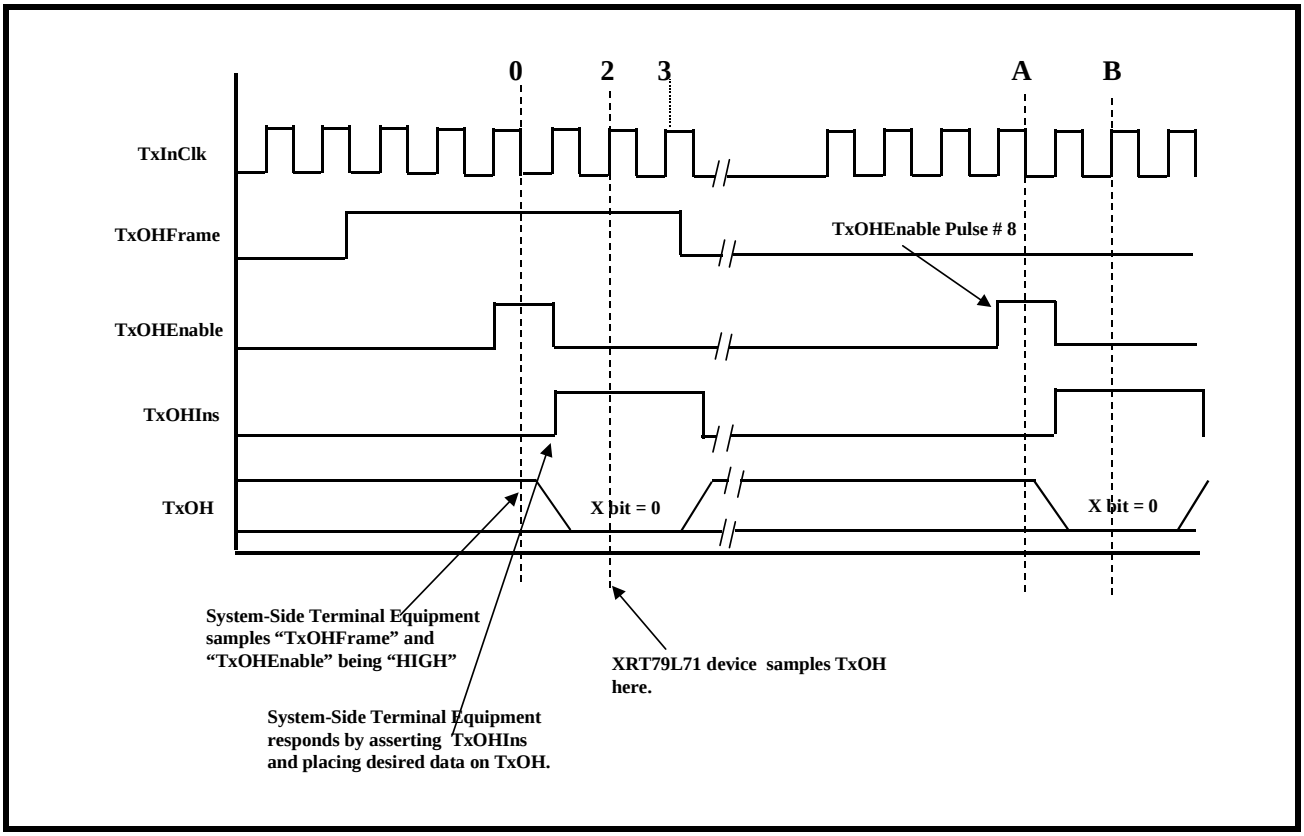
TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until TxOHEnable is sampled at a logic "high" level again. Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHEnable pulsing period comes up, for inserting an overhead bit into the outbound DS3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 2 in order to transmit the FERF (Far-End-Receive Failure) indicator to the remote terminal equipment.

For DS3 applications, the FERF indication is transmitted by setting both the X bits within each outbound DS3 frame to "0".

If we were to assume that the connection between the System-Side Terminal Equipment and the Transmit Overheads Data Input Interface block of the XRT79L71 is as illustrated in Figure 46, then Figure 47 presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface when using Method 2.

FIGURE 47. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 2)



In Figure 47 the System-Side Terminal Equipment samples the TxOHFrame and TxOHEnable output pins being "High" during Falling Clock Edge # 0 within the TxInClk signal. At this point, the System-Side Terminal Equipment knows that the XRT79L71 is just about to process the very first overhead bit within a given outbound DS3 frame. According to Table 21, the very first overhead bit to be processed within a given DS3 frame is the first X bit. In order to facilitate the transmission of the FERF indicator, the System-Side Terminal

Equipment must set this particular bit-field to "0". The System-Side Terminal Equipment begins this process by implementing the following two tasks concurrently.

TASK 1 - The System-Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "0".

After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample the TxOHIns input pin being "High" and the TxOH input pin being set "Low" during second rising edge of TxInClk after TxOHFrame and TxOHEnable were initially sampled "High" at Clock Edge # 2, in **Figure 47**. Once the XRT79L71 has sampled these two signals, it will then insert a "0" into the very first X bit-position, within the outbound DS3 frame.

Upon detection of the very next rising edge of the TxInClk clock signal (designated as Clock Edge # 3, in **Figure 47**), the System-Side Terminal Equipment will negate or de-assert the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface block until it samples the TxOHEnable output pulsing "High" eight more times (depicted in **Figure 47** as TxOHEnable Pulse # 8). According to **Table 21**, the occurrence of TxOHEnable Pulse # 8 indicates that the XRT79L71 is just about ready to process the second X bit, within the outbound DS3 frame. In order to facilitate this transmission of the FERF indicator, this particular X bit must also be set to "0". The System-Side Terminal Equipment begins this process by implementing the following two tasks concurrently.

TASK 1 - The System-Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "0".

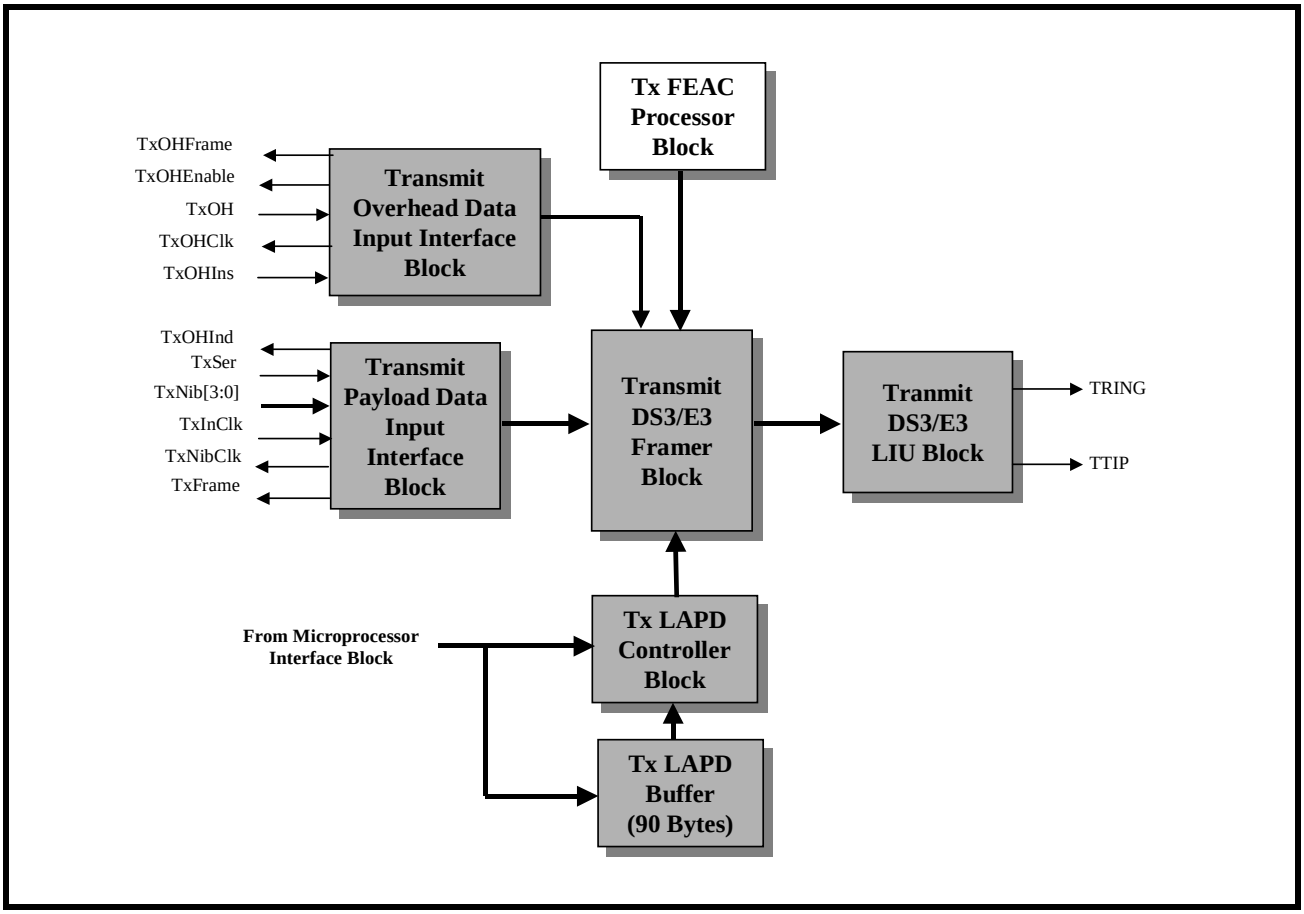
After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample the TxOHIns input pin being "High" and the TxOH input pin being set "Low" during the second rising edge TxInClk after TxOHEnable Pulse # 8 was sampled "High. Once the XRT79L71 has sampled this signal, it will then insert a "0" into the second X bit-position within the outbound DS3 frame.

Upon detection of the very next rising edge of the TxInClk clock signal (designated as Clock Edge # B, in **Figure 47**), the System-Side Terminal Equipment will negate or de-assert the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface block until it samples the TxOHEnable and the TxOHFrame output signals pulsing "High" simultaneously. Afterwards, the System-Side Terminal Equipment will repeat all of the steps that have been outlined in this case study.

4.2.3 TRANSMIT FEAC CONTROLLER BLOCK

The Transmit FEAC Controller Block is the fourth functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 48** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit FEAC Controller block highlighted.

FIGURE 48. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L91 DEVICE HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT FEAC CONTROLLER BLOCK HIGHLIGHTED)



If the XRT79L71 is operating in C-bit Parity Frame Format then the FEAC bit-field of the DS3 Frame can be used to transmit FEAC messages (See **Figure 49**). 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:

FIGURE 49. THE BIT-FORMAT OF A FEAC MESSAGE

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 6-bit 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 frame periods are required to transmit the 16 bit FEAC Code Message.

The following section describes how to use the Transmit FEAC Controller block to transmit a given FEAC Message.

Operating the Transmit FEAC Controller

In order to transmit a FEAC message to the remote terminal, the user must execute the following steps.

STEP 1 - Write the outbound 6-bit FEAC Code-word into the Transmit DS3 FEAC Register

In this step, the user must write the six bit FEAC code word into the Transmit DS3 FEAC Register. The bit format of this register is presented below.

Transmit DS3 FEAC Register (Address = 0x1132)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used	TxFEAC_Code_Word[5:0]						Not Used
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/O
0	d5	d4	d3	d2	d1	d0	0

NOTE: *The six-bit code word must be written in Bits 1 through 6 within this register. Bits 0 and 7 are in-active bits and are each forced to "0".*

STEP 2- Enable the Transmit FEAC Controller Block

In order to enable the Transmit FEAC Controller Block the user must write a "1" into bit 2 (TxFEAC Enable) within the Transmit DS3 FEAC Configuration and Status Register, as depicted below.

Transmit DS3 FEAC Configuration and Status Register (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			Tx FEAC Interrupt Enable	Tx FEAC Interrupt Status	Tx FEAC Enable	Tx FEAC Go	Tx FEAC Busy
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/O
0	0	0	X	X	1	0	0

At this point, the Transmit FEAC Controller Block will be turned on and will begin to repeatedly transmit the Idle FEAC pattern of [1, 1, 1, 1, 1, 1].

STEP 3 - Enable the Transmit FEAC Message Interrupt (Optional)

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor, as soon as the Transmit FEAC Controller block has completed its 10th transmission of the FEAC Message. The purpose of this interrupt is to alert the system Microprocessor that the Transmit FEAC Controller block has completed its 10th transmission of the most recent outbound FEAC Message, and that it is now available to transmit a different FEAC Message.

The procedure for enabling the Transmit FEAC Interrupt is actually a three-step process.

STEP 3a - Enable the DS3/E3 Framer block interrupt as the Operational Block Level

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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/W	R/W	R/O	R/W	R/W	R/W	R/O	R/O
0	0	0	0	0	1	0	0

STEP 3b - Enable the Transmit DS3/E3 Framer Block Interrupts - at the Block Level

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register to "1" as depicted below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

STEP 3c - Enable the Transmit FEAC Interrupt at the Source Level

The user can enable the Transmit FEAC Message Interrupt by setting bit 4 (Tx FEAC Interrupt Enable) within the Transmit DS3 FEAC Configuration & Status register to "1", as illustrated below.

Transmit DS3 FEAC Configuration and Status Register (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Tx FEAC Interrupt Enable	Tx FEAC Interrupt Status	Tx FEAC Enable	Tx FEAC Go	Tx FEAC Busy
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/O
0	0	0	1	X	1	0	0

If the Transmit FEAC Message Interrupt is enabled, then the channel will generate an interrupt as soon as the Transmit FEAC Controller block has completed its 10th transmission of the FEAC Message.

STEP 4 - Initiate the Transmission of the FEAC Message

The user can initiate the transmission of the FEAC message residing in the Transmit DS3 FEAC register by inducing a "0" to "1" transition in Bit 1 (Tx FEAC Go) within the Transmit DS3 FEAC Configuration and Status Register. This can be accomplished by writing a "1" to bit 1 of the Transmit DS3 FEAC Configuration and Status register, as depicted below.

Transmit DS3 FEAC Configuration and Status Register (Address = 0x1131)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Not Used			Tx FEAC Interrupt Enable	Tx FEAC Interrupt Status	Tx FEAC Enable	Tx FEAC Go	Tx FEAC Busy
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/O
0	0	0	X	X	1	0->1	0

NOTES:

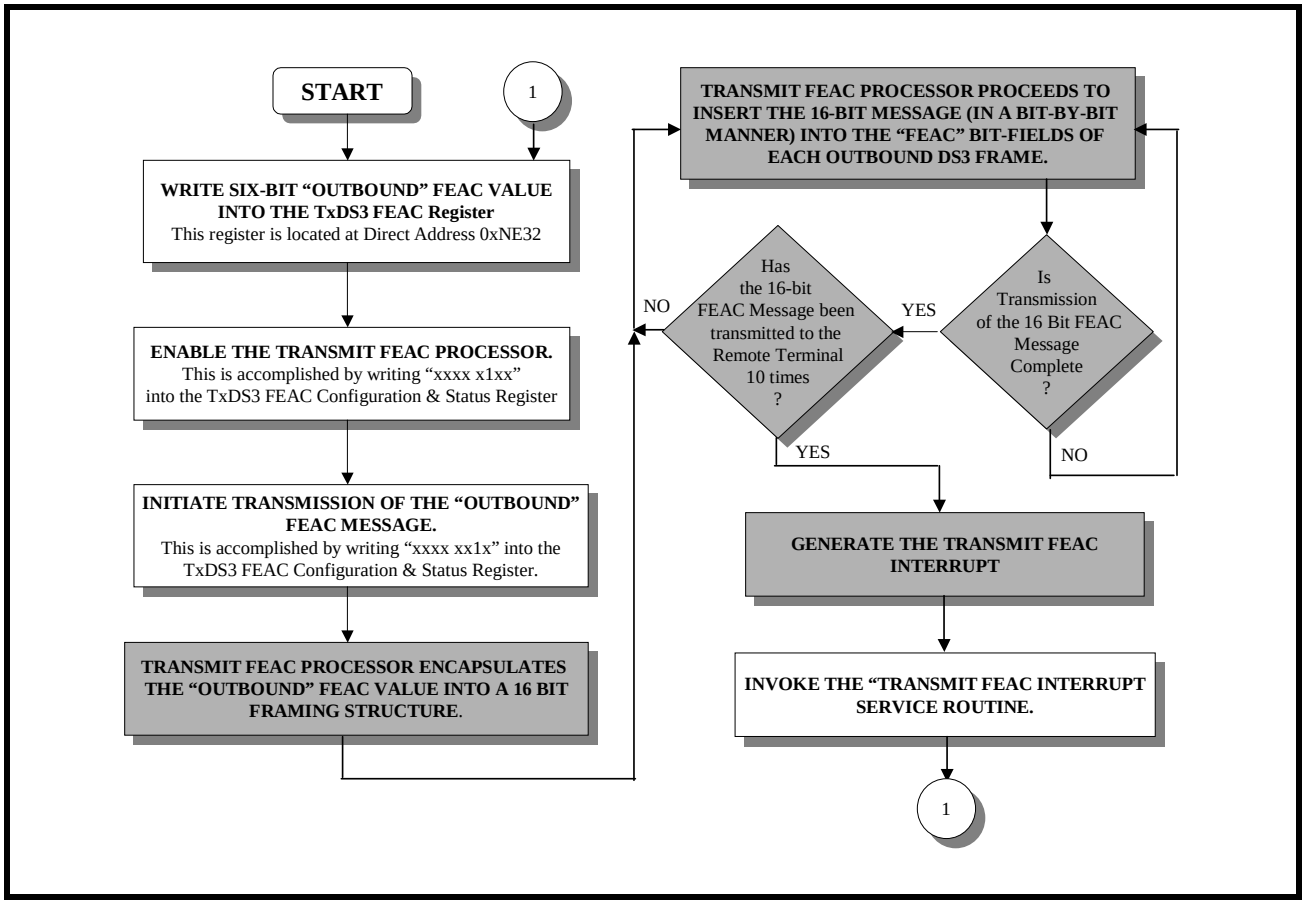
1. *While executing this particular write command, the user should write a 000xx110b to the Transmit DS3 FEAC Configuration and Status Register. The user must insure that a "1" is also being written to Bit 2 of the register, in order to keep the Transmit FEAC Controller block enabled.*
2. *The Transmit FEAC Controller block requires a "0" to "1" transition within Bit 1 (Tx FEAC Go). Therefore, sometime after setting Bit 1 to "1", the user must follow up and set this bit-field back to "0".*

At this point, the Transmit FEAC Controller block will proceed to transmit the 16 bit FEAC code via the outbound DS3 frame message repeatedly for 10 consecutive times. This process will require a total of 160 DS3 frame periods. During this process the Tx FEAC Busy bit (Bit 0) will be asserted, indicating that the

Transmit FEAC Controller block 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 Controller block 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 XRT79L71 generating this interrupt is to let the local µP/µC know that the Transmit FEAC Controller block is now available and ready to transmit a new FEAC message. Finally, once the Transmit FEAC Controller block has completed its 10th transmission of a FEAC Code Message it will continue to send the same FEAC Message for an indefinite period until the local µP/µC commands it to transmit a new FEAC message.

Figure 50 presents a flow chart depicting how to use the Transmit FEAC Controller block.

FIGURE 50. A FLOW CHART DEPICTING HOW TO TRANSMIT A FEAC MESSAGE VIA THE FEAC TRANSMITTER



NOTES:

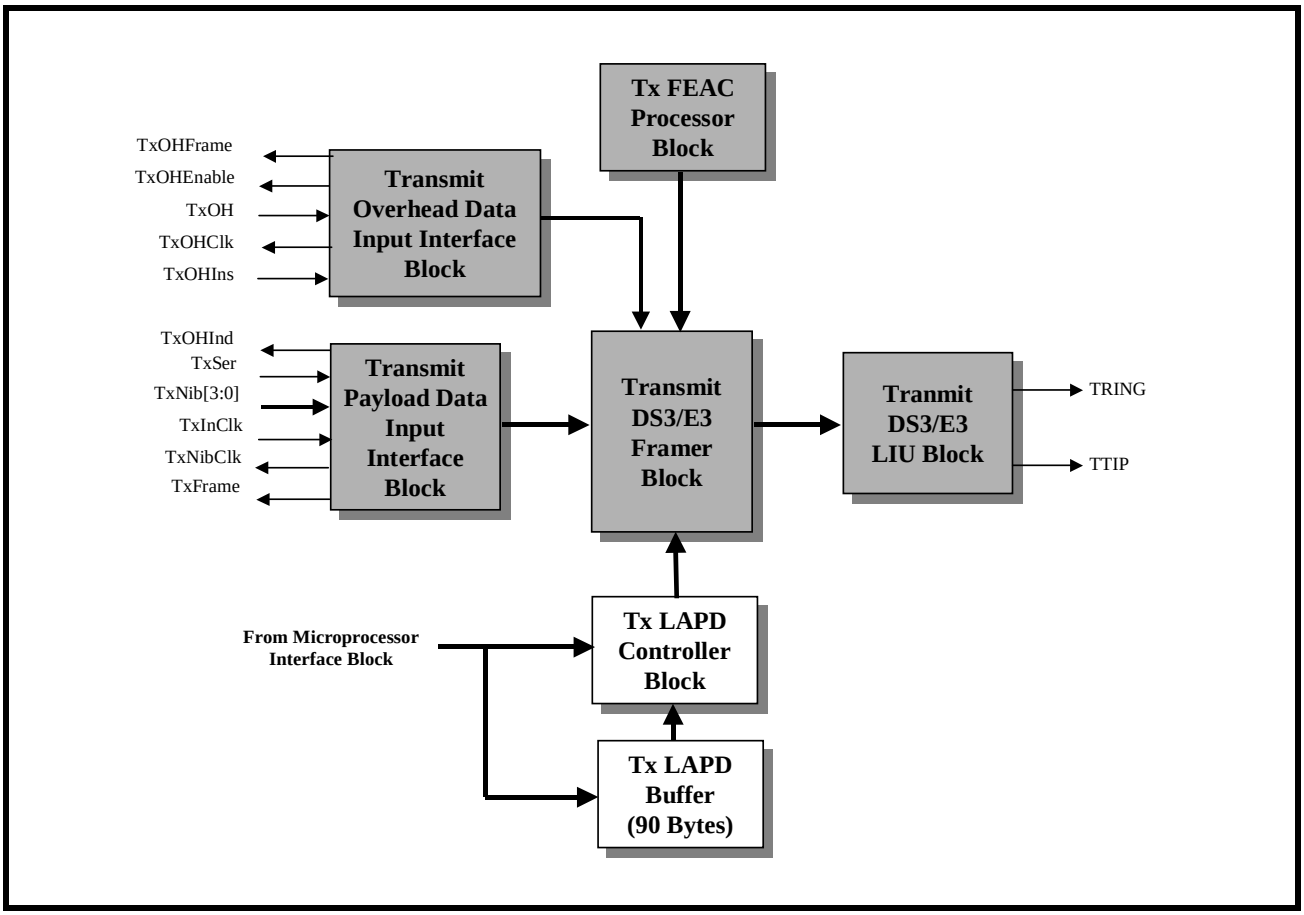
1. *The whiteboxes in Figure 50 indicate the steps that the user must employ in order to command the Transmit FEAC Controller block to transmit a FEAC message.*
2. *The shaded boxes indicate the steps that the Transmit FEAC Controller block will execute in order to transmit a FEAC message.*

For a detailed description of the Receive FEAC Controller block, please see Section 4.3.4.

4.2.4 TRANSMIT LAPD CONTROLLER BLOCK

The Transmit LAPD Controller block is the third functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 51 presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit LAPD Controller block highlighted.

FIGURE 51. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT LAPD CONTROLLER BLOCK HIGHLIGHTED)



The Transmit LAPD Controller block consists of the following sections.

- The Transmit LAPD Message Buffer
- The TransmitLAPD Controller

The Transmit LAPD Message Buffer

The purpose of the Transmit LAPD Message Buffer is to permit the user to write and temporarily store the contents of the very next outbound LAPD Message that is to be transmitted. The Transmit LAPD Message Buffer is actually a 90 byte FIFO that is located at Address location 0x11B0 within the XRT79L71 address space.

The LAPD Transmitter

The LAPD Transmitter permits the user to transmit path maintenance data link (PMDL) messages to the remote terminal equipment 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 permits the user to transmit both standard and non-standard PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer), to store the contents of the outbound PMDL message to be transmitted. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in **Figure 52**.

FIGURE 52. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-BITS)	**C/R**	**EA**
TEI (7 BITS)		**EA**
CONTROL (8-BITS)		
76 OR 82 OR ANY-SIZE BYTES OF INFORMATION (PAYLOAD)		
FCS - MSB		
FCS - LSB		
FLAG SEQUENCE (8-BITS)		

For standard Bellcore GR-499-CORE applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the LAPD Transmitter is not currently transmitting a LAPD Message, it will instead be transmitting a continuous stream of Flag Sequence bytes via the DL bits within each outbound DS3 frame.

SAPI - Service Access Point Identifier

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for standard Bellcore GR-499-CORE applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For standard Bellcore GR-499-CORE applications the user must use the Control byte the value 0x03. Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The Information Payload is the 76 bytes, 82 bytes or any number of 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 Address 0x11B0.

A Special Note about the Information Payload when transmitting standard Bellcore GR-499-CORE type LAPD Messages

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 0x11B0 within the XRT79L71. The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. **Table 22** presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT79L71 and the corresponding octet value that the user must write into the first octet position within the Transmit LAPD Message buffer.

TABLE 22: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD FOR STANDARD 76 OR 82 BYTE MESSAGES

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

NOTE: *To transmit a LAPD Message and information payload that is of a size other than 76 or 82 bytes, then there are no restrictions on the value that should be written to the first byte within the information payload.*

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, x16 + x12 + x5 + 1. Afterwards, this FCS value is inserted into the two-octet FCS value position, within the LAPD Message frame. The LAPD Receiver at the remote terminal will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see Section 4.3.3.3 on how the Receive LAPD Controller block handles and processes incoming LAPD Message frames.

Operation of the Transmit LAPD Controller

As mentioned earlier, the LAPD Transmitter permits the user to transmit either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

4.2.4.1 Transmitting Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) write the contents of an outbound PMDL Message, into the Transmit LAPD Message Buffer, and (2) command the LAPD Transmitter to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the C-Bit Parity Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "1" and that Bit 2 (Frame Format) is set to "0" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	1	X	0	X	0	X	X

STEP 2 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit DS3 LAPD Configuration Register to "1", as illustrated below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit LAPD Any	Unused			Auto Retransmit	Unused	Tx LAPD Msg Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	X	X	1

NOTES:

1. *For normal operation, it is imperative that the user also make sure that bit 4 (Reserved) and bit 7 (Transmit LAPD Any) within this register are both set to "0".*
2. *Once the user executes the above mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle (Flag) Sequence (e.g., a continuous, repeating string of octets, of the value 0x7E) via the DL bits within each outbound DS3 frame.*

STEP 3 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a LAPD/PMDL Message at one-second intervals. To configure the Transmit LAPD Controller block to do this, then the user should write a "1" into Bit 3 (Auto Retransmit) within the Transmit DS3 LAPD Configuration Register. This action is depicted below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	1	0	X	1

STEP 4 - Specify the type of LAPD Message to be transmitted.

At this stage it is necessary to specify the type of LAPD Message that is to be transmitted. The Transmit and Receive LAPD Controller blocks, within the XRT79L71 are capable of supporting the following standard types of LAPD Messages.

- Test Signal
- Idle Signal
- CL Path Identification
- ITU Path Identification

The ITU Path Identification type of PMDL Message consists of an 82-byte information payload. All of the remaining standard types of PMDL/LAPD Messages contain 76 byte information payloads. To transmit an 82 byte message (e.g., the ITU Path Identification type of LAPD Message), then the user should set Bit 1 (Transmit LAPD Message Length), within the Transmit DS3 LAPD Configuration Register to "1". For all of the remaining types of LAPD Messages which are 76 bytes in length, the user should set Bit 1 (Transmit LAPD Message Length) to "0". This operation is depicted below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	0	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle LAPD/PMDL messages that contain 76-byte sized Information Payloads.*

To transmit an ITU Path Identification type of PMDL Message, then the user needs to write the value "1" into the Transmit LAPD Message Length bit-field, as depicted below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	1	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle 82-byte sized PMDL Messages.*

STEP 5 - Load the Transmit LAPD Message Buffer

The next step is to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will

describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in Section 4.3.3.1.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 5a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer

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

Sub-STEP 5a.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

Sub-STEP 5a.2 - At Address Location 0x11C1 (the LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 5a.1, above) this will be the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

STEP 5b - Write the value for SAPI, C/R and EA0 into the second byte-position, within the Transmit LAPD Message Buffer

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

NOTE: *Bellcore mandates that the user use the values "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.*

Sub-STEP 5b.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x01".

This step will cause the internal LAPD Message Buffer pointer to point to the second byte of Indirect Address 0x01 within the Transmit LAPD Message Buffer.

Sub-STEP 5b.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write in the value "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.

This step will cause the value "0x3C" or "0x3E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 5b.1, above) this will be the second byte of Indirect Address 0x01 within the Transmit LAPD Message Buffer

STEP 5c - Write the value for TEI and EA1 into the third byte-position, within the Transmit LAPD Message Buffer

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

NOTE: *Bellcore mandates that the user use the value of "0x01" for this byte.*

Sub-STEP 5c.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x02".

This step will cause the internal LAPD Message Buffer pointer to point to the third byte of Indirect Address 0x02 within the Transmit LAPD Message Buffer.

Sub-STEP 5c.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x01".

This step will cause the value "0x01" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 5c.1, above) this will be the third byte of Indirect Address 0x02 within the Transmit LAPD Message Buffer.

STEP 5d - Write in the CONTROL BYTE value of 0x03 into the fourth byte-position, within the Transmit LAPD Message buffer

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

Sub-STEP 5d.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x03".

This step will cause the internal LAPD Message Buffer pointer to point to the fourth byte of Indirect Address 0x03 within the Transmit LAPD Message Buffer.

Sub-STEP 5d.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x03"

This step will cause the value "0x03" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 5d.1, above) this will be the fourth byte of Indirect Address 0x03 within the Transmit LAPD Message Buffer.

STEP 5e - Write in a specific value, which the remote terminal equipment can use to identify the PMDL Message type.

Therefore, the exact value that the user must write in depends upon the type of LAPD Message being transmitted to the Remote Terminal Equipment. Table 23 presents a mapping of the value to be written into this byte, with the corresponding PMDL Message type, to be transmitted.

TABLE 23: A MAPPING OF THE VALUE TO BE WRITTEN INTO INDIRECT ADDRESS LOCATION 0X11B0 AND THE CORRESPONDING PMDL MESSAGE

PMDL MESSAGE TYPE	VALUE TO BE WRITTEN INTO ADDRESS LOCATION 0X11B0 (DURING STEP 5E)
Test Signal	0x32
Idle Signal	0x34
CL Path Identification	0x38
ITU Path Identification	0x3F

All of this can be accomplished by executing the following two sub-steps.

Sub-STEP 5e.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x04".

This step will cause the internal LAPD Message Buffer pointer to point to the fifth byte of Indirect Address 0x04 within the Transmit LAPD Message Buffer.

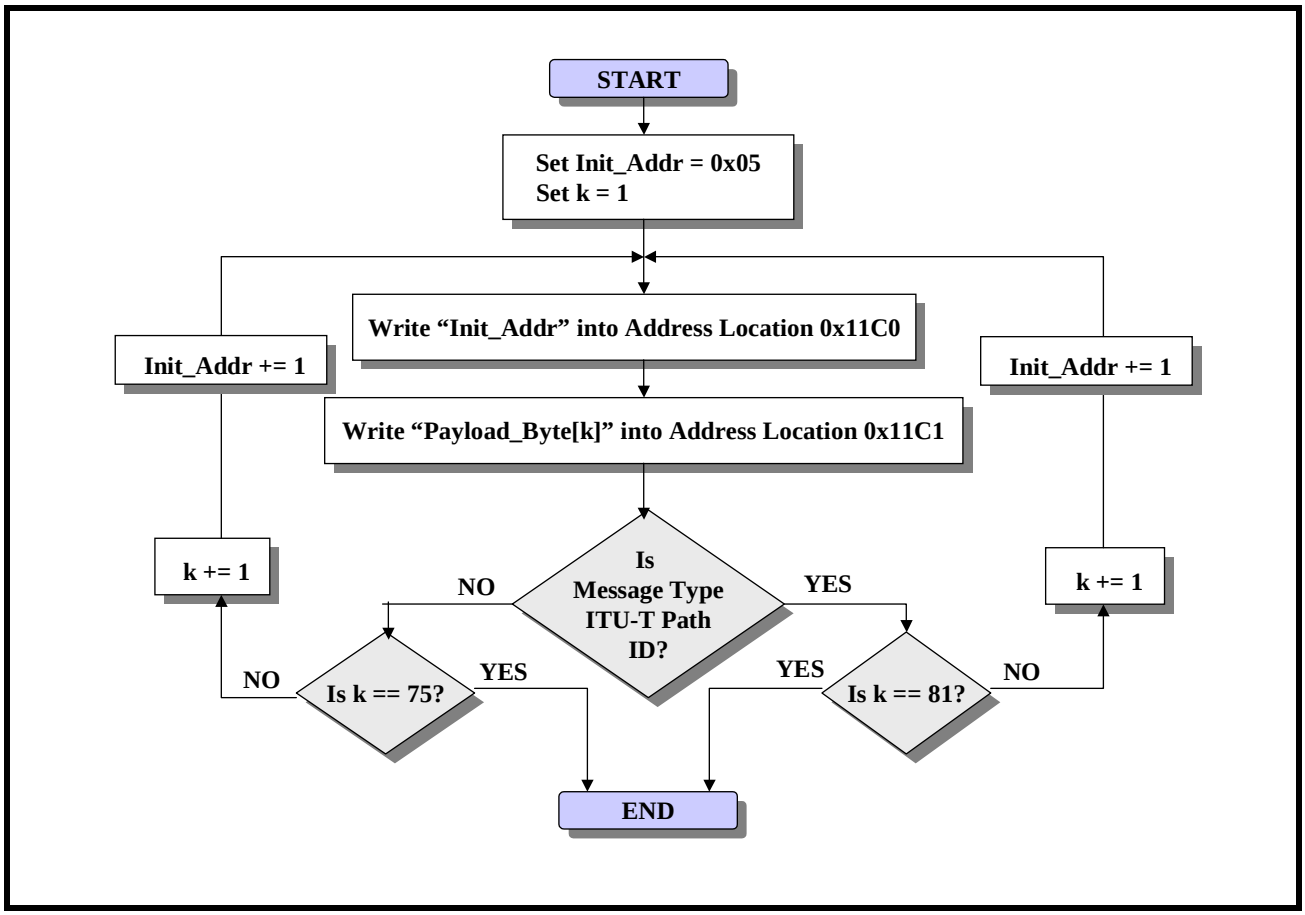
Sub-STEP 5e.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the appropriate value (per Table 23) into the location that is being identified by the LAPD Message Buffer pointer.

In this case (per Sub-STEP 5e.1, above) this will be the fifth byte of Indirect Address 0x04 within the Transmit LAPD Message Buffer.

STEP 5f - Write in the remaining 75 or 81 byte of the Information Payload within the PMDL Message into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented in the following flow-chart.

FIGURE 53. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING THE PAYLOAD PORTION OF THE LAPD/PMDL MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



STEP 6 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller has completed its transmission of a given PMDL Message. The purpose of this interrupt is to alert the system mC/mP that the Transmit LAPD Controller has completed transmitting its most recent LAPD/PMDL message, and that it is now available to transmit a different LAPD Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 6a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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/O	R/O
0			0	0	0	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 6b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/ E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

This step enables the Transmit DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 6c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit DS3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 7 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit DS3 LAPD Status/ Interrupt Register, as depicted below.

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0->1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will zero-stuff the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin in a bit-by-bit manner inserting the resulting PMDL Message into the DL bit-fields, within the outbound DS3 frames.

NOTES:

1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised some time later to execute another write operation to this register that sets the Initiate Transmission of LAPD/PMDL Message bit back to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field within the Transmit DS3 LAPD Status/Interrupt register to "1", as illustrated below.*

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will then toggle back to "0".

The Procedure for Transmitting a standard LAPD/PMDL Message presented in a Flow-Chart

Figure 54 andFigure 55 present two flow chart diagrams. Figure 54 depicts the procedure in white boxes that the user should use in order to transmit a PMDL Message via the Transmit LAPD Controller, when the Transmit LAPD Controller is configured to retransmit the LAPD Message frame, repeatedly at one-second intervals. This figure also indicates via the shaded boxes what the Transmit LAPD Controller circuitry will do before and during message transmission.

Figure 54. Flow Chart depicting how to use the Transmit LAPD Controller when the Transmit LAPD Controller is configured to re-transmit the LAPD Message frames repeatedly at one-second intervals

Figure 54 presents the procedure in white boxes that the user should use in order to transmit a PMDL Message via the Transmit LAPD Controller, when the Transmit LAPD Controller is configured to transmit the LAPD Message frame only once, and then halt transmission.

Figure 55. Flow Chart depicting how to use the Transmit LAPD Controller when the Transmit LAPD Controller is configured to re-transmit the LAPD Message frames repeatedly at one-second intervals

4.2.4.2 Transmitting Non-Standard Variable Length (e.g., up to 82 bytes) LAPD Messages

The user can (1) write the contents of the outbound PMDL Message into the Transmit LAPD Message buffer and (2) command the Transmit LAPD Controller to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the C-Bit Parity Framing format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "1" and that Bit 2 (Frame Format) is set to "0", as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	1	X	0	X	0	X	X

STEP 2 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit DS3 LAPD Configuration register to "1", as illustrated below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	X	X	X	1

NOTE: *Once the user executes the above-mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle (Flag) Sequence (e.g., a repeating string of 0x7E) to the remote terminal equipment.*

STEP 3 - Configure the Transmit LAPD Controller to transmit a non-standard size LAPD Message.

This is accomplished by setting Bit 7 (LAPD Any) within the Transmit DS3 LAPD Configuration Register to "1", as illustrated below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	X	X	X	1

STEP 4 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a PMDL Message at one-second intervals. To configure the Transmit LAPD Controller block to do this, then the user should write a "1" into Bit 3 (Auto-Retransmit) within the Transmit DS3 LAPD Configuration register. This operation is depicted below.

Transmit DS3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	1	0	X	1

STEP 5 - Specify the size of the outbound LAPD Message

This is accomplished by writing the size of the information payload in terms of number of bytes into the Transmit LAPD Byte Count Register, as depicted below.

Transmit LAPD Byte Count Register (Address = 0x1183)

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

STEP 6 - Load the Transmit LAPD Message Buffer

The next step is to load into the contents of the outbound LAPD/PMDL Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents o the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ an Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in Section 4.3.3.2.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 6a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer.

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

Sub-STEP 6a.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte of Indirect Address = 0x00 within the Transmit LAPD Message Buffer.

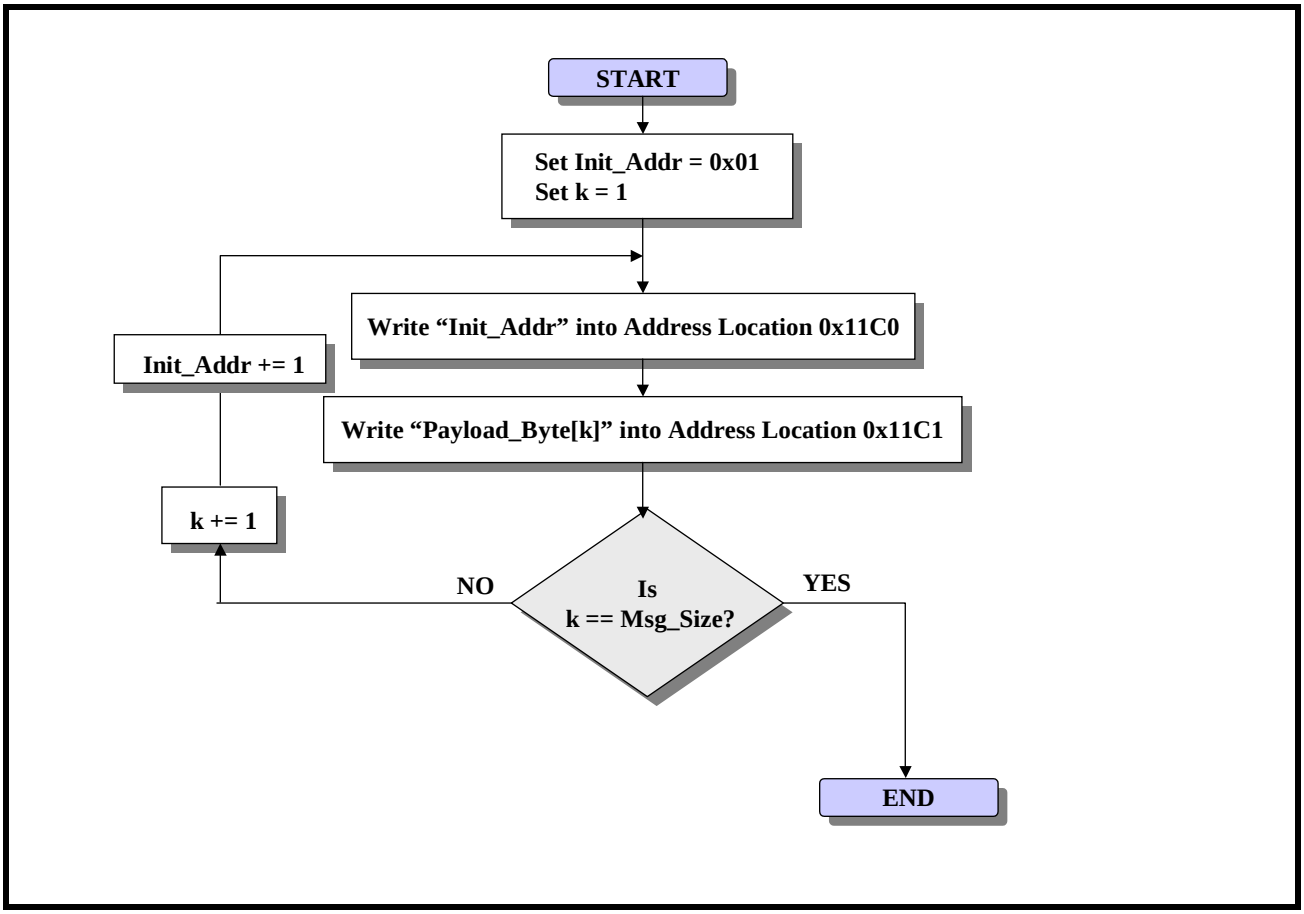
Sub-STEP 6a.2 - At Address Location 0x11C1 (the LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 6a.1, above) this will be the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

STEP 6b - Write in the remaining bytes of this outbound message into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 56. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING IN THE REMAINING BYTES OF THE OUTBOUND MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



NOTES: *About* **Figure 56**:

1. *The value Msg_Size within the Decision Diamond is the value that the user writes into the Transmit LAPD Byte Count Register, during STEP 5.*

STEP 7 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller block has completed its transmission of a given PMDL Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 7a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

STEP 7b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

STEP 7c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit DS3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/ PMDL Mes-sage	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 8 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit DS3 LAPD Status/ Interrupt Register, as depicted below.

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/ PMDL Mes-sage	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0->1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will zero-stuff the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin in a bit-by-bit manner inserting the resulting PMDL Message into the DL bit-fields, within the outbound DS3 frames.

NOTES:

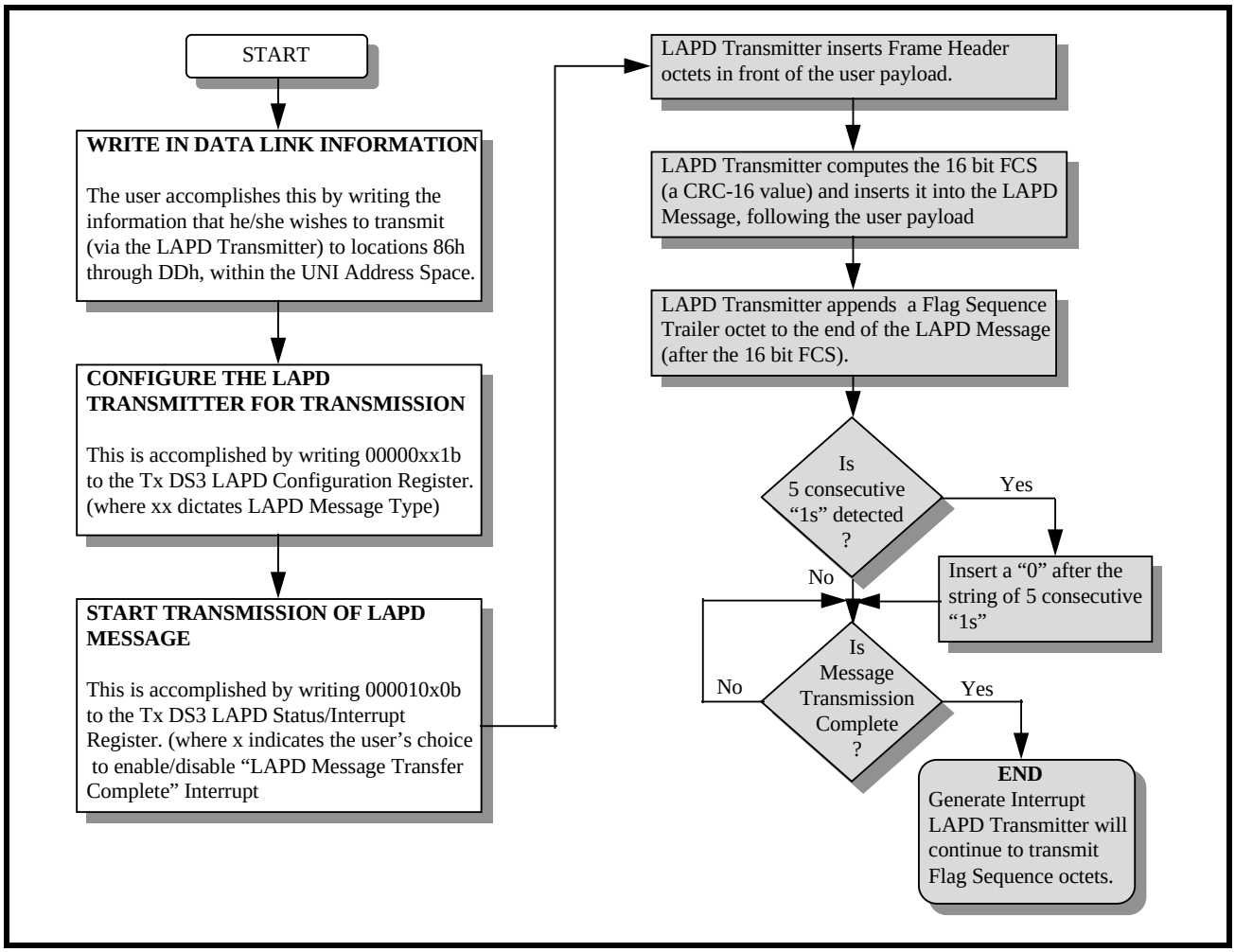
1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised some time later to execute another write operation to this register that sets the Initiate Transmission of LAPD/PMDL Message bit to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field within the Transmit DS3 LAPD Status/Interrupt register to "1", as illustrated below.*

Transmit DS3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/ PMDL Mes-sage	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will toggle back to "0".

FIGURE 57. FLOW CHART DEPICTING HOW TO USE THE TRANSMIT LAPD CONTROLLER



The Mechanics of Transmitting a New LAPD Message, if the Transmit LAPD Controller has been configured to retransmit the LAPD Message frame, repeatedly at one-second intervals

If the Transmit LAPD Controller has been configured to retransmit the LAPD Message repeatedly at one-second intervals, then it will do the following at one-second intervals.

- Parse through the contents of the Transmit LAPD Message buffer and Zero-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.

To transmit another (e.g., different) PMDL message to the remote LAPD Receiver, the user will have to write this new message into the Transmit LAPD Message buffer, via the Microprocessor Interface section of the channel. However, the user must be careful 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 LAPD Message frame transmissions, this action could interfere with these transmissions, thereby causing the Transmit LAPD Controller 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, do the following:

1. ***Configure the DS3/E3 Framer Block to automatically reset activated interrupts***

The user can do this by writing a "1" into Bit 1 (Enable Interrupt Auto-Clear) within the Operating Mode Register - Byte 2, as depicted below.

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	1	0

This action will prevent the Transmit LAPD Controller from generating its own one-second interrupts.

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 = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	0	1

3. Write the new message into the Transmit LAPD Message buffer immediately after the occurrence of the One-Second interrupt.

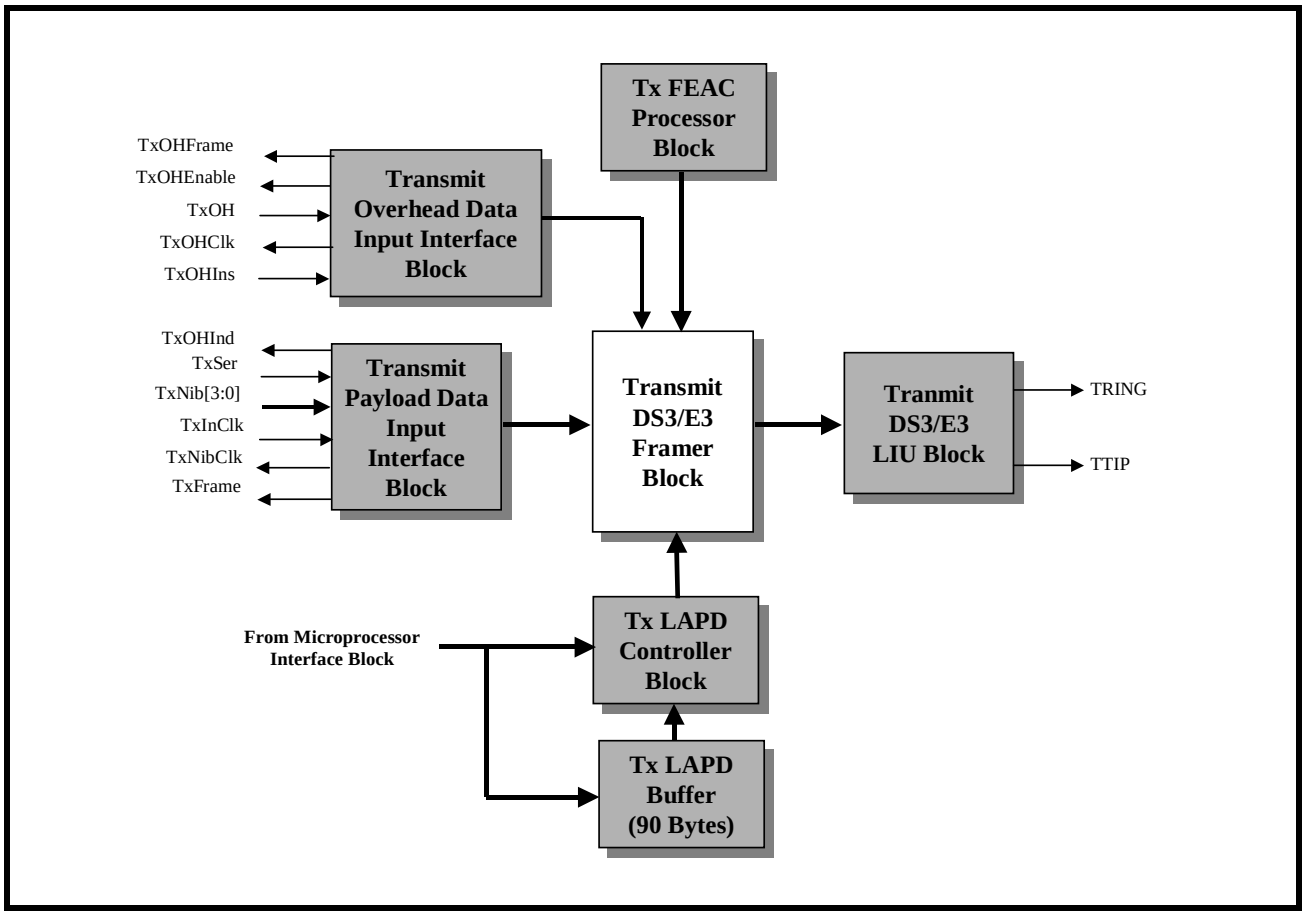
By timing the writes to the Transmit LAPD Message buffer to occur immediately after the occurrence of the One-Second interrupt, the user avoids conflicting with the one-second transmissions of the LAPD Message frame, and will transmit the correct messages to the remote LAPD Receiver.

4.2.4.3 Transmit LAPD Controller Block Interrupt

4.2.5 TRANSMIT DS3 FRAMER BLOCK

The Transmit DS3 Framer block is the fifth functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 58 presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit DS3/E3 Framer block highlighted.

FIGURE 58. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT DS3/E3 FRAMER BLOCK HIGHLIGHTED)



The purpose of the Transmit DS3 Framer block is to accept data from the Transmit Payload Data Input Interface block, the Transmit LAPD Controller and the Transmit FEAC Processor blocks, and to create and transmit a DS3 data stream. Afterwards, the Transmit DS3 Framer block will route this DS3 data-stream to the Transmit DS3 LIU Block for transmission to the remote terminal equipment. The Transmit DS3 Framer block also supports the following functions.

- Transmitting the LOS Pattern (under Software control)
- Transmitting the AIS Pattern (under Software control)
- Transmitting the Idle Pattern (under Software control)
- Transmitting the FERF (RDI) indicator (automatically and under software control)
- Forcing the X Bits to "1" (under Software control)
- ransmitting the FEBE indicator (automatically and under Software control).

4.2.5.1 TRANSMITTING THE LOS PATTERN

The XRT79L71Transmit DS3/E3 Framer block permits the user to transmit the LOS (Loss of Signal) pattern to the remote terminal equipment. The Transmit DS3/E3 Framer block provides the user with two options when transmitting the LOS pattern.

- Transmitting an All Zeros pattern
- Transmitting an All Ones pattern

The procedure for transmitting either of these types of LOS Patterns is presented below.

Transmitting an All Zeros LOS Pattern

The user can configure the Transmit DS3/E3 Framer block to transmit an LOS pattern by transmitting an All Zeros pattern. The user can accomplish this by executing the following steps.

STEP 1 - Set Bit 4 (TxLOS Pattern) within the Transmit DS3 Pattern Register, to "0", as illustrated below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	0	0

NOTE: *This step configures the Transmit DS3/E3 Framer block to transmit an All Zeros pattern, whenever the user configures it to transmit the LOS Pattern.*

STEP 2 - Set Bit 3 (TxLOS) within the Transmit DS3 Configuration Register to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	1	1

Once the user executes this step, then the Transmit DS3/E3 Framer block will override all of the outbound DS3 data payload and overhead bits, with an All Zeros pattern.

NOTE: *When this bit is set, it overrides all of the other bits in this register.*

Transmitting an All Ones LOS Pattern

The user can also configure the Transmit DS3/E3 Framer block to transmit an LOS pattern by transmitting an All Ones pattern. The user can accomplish this by executing the following steps.

STEP 1 - Set Bit 4 (TxLOS Pattern) within the Transmit DS3 Pattern Register, to "1", as illustrated below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	1	1	1	0	0

NOTE: *This step configures the Transmit DS3/E3 Framer block to transmit an All Ones pattern, whenever the user configures it to transmit the LOS Pattern.*

STEP 2 - Set Bit 3 (TxLOS) within the Transmit DS3 Configuration Register to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	1	1

Once the user executes this step, then the Transmit DS3/E3 Framer block will override all of the outbound DS3 data payload and overhead bits, with an All Zeros pattern.

NOTE: *When this bit is set, it overrides all of the other bits in this register.*

4.2.5.2 TRANSMITTING THE DS3 AIS PATTERN

The Transmit DS3/E3 Framer block provides the user with two options associated with transmitting the DS3 AIS indicator.

- Forced transmission of the DS3 AIS indicator (e.g., under software control)
- Automatic transmission of the DS3 AIS indicator

Forced Transmission of the AIS Indicator

For DS3 Applications, the Transmit DS3/E3 Framer block permits the user to transmit either of the following types of DS3 AIS patterns upon software control.

- The Bellcore GR-499-CORE Standard DS3 AIS pattern which has the following characteristics.
  - **a.** A framed, repeating 1010… pattern
  - **b.** Valid M, F and P bits
  - **c.** All C-bits are set to "0".
- The Relaxed Bellcore GR-499-CORE Standard DS3 AIS pattern which has the following characteristics
  - **a.** A framed, repeating 1010… pattern
  - **b.** Valid M, F and P bits
  - **c.** The C-bits are not necessarily set to "0"
- The Unframed, All Ones Pattern

To force the Transmit DS3/E3 Framer block to transmit any one of these types of DS3 AIS patterns, the following sequence of instructions must be executed.

Forcing the transmission of the Bellcore GR-499-CORE Compliant DS3 AIS pattern, upon Software Control

The user can force the Transmit DS3/E3 Framer block to transmit the Bellcore GR-499-CORE Compliant DS3 AIS indicator upon software command by executing the following steps.

STEP 1 - Set Bit 6 (DS3 AIS Non-Stuck Stuff), within the Transmit DS3 Pattern Register to "0", as depicted below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	0	0

This step configures the Transmit DS3/E3 Framer block to force all of the C-bits to "0" whenever it is commanded to transmit the DS3 AIS indicator.

STEP 2 - Command the Transmit DS3/E3 Framer block to transmit the DS3 AIS indicator.

This is accomplished by setting Bit 4 (Tx AIS) to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	1

NOTES:

1. *The user can command the Transmit DS3/E3 Framer block to terminate its transmission of the DS3 AIS pattern by setting Bit 4 (TxAIS) back to "0".*
2. *Bit 4 (TxAIS) is ignored for the duration that Bit 3 (TxLOS) is set to "1".*

Forcing the tranmission of the Relaxed GR-499-CORE Compliant DS3 AIS pattern, upon Software Control

The user can force the Transmit DS3/E3 Framer block to transmit the Relaxed Bellcore GR-499-CORE Compliant DS3 AIS indicator upon software command by executing the following steps.

STEP 1 - Set Bit 6 (DS3 AIS Non-Stuck Stuff), within the Transmit DS3 Pattern Register to "1", as depicted below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	1	0	0	1	1	0	0

This step configures the Transmit DS3/E3 Framer block to NOT force all of the C-bits to "0" whenever it is commanded to transmit the DS3 AIS indicator.

STEP 2 - Command the Transmit DS3/E3 Framer block to transmit the DS3 AIS indicator.

This is accomplished by setting Bit 4 (Tx AIS) to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	1

NOTES:

1. *In this setting, the Transmit DS3/E3 Framer block will not be setting the C-bits to "0", whenever it is commanded to transmit the DS3 AIS indicator. This permits the user to still be able to command the Transmit DS3/E3 Framer block to transmit FEAC and PMDL Messages even while the Transmit DS3/E3 Framer block is also transmitting the DS3 AIS indicator.*
2. *The user can command the Transmit DS3/E3 Framer block to terminate its transmission of the DS3 AIS pattern by setting Bit 4 (TxAIS) back to "0".*
3. *Bit 4 (TxAIS) is ignored for the duration that Bit 3 (TxLOS) is set to "1".*

Forcing the transmission of the Unframed, All Ones DS3 AIS pattern, upon Software Control

The user can force the Transmit DS3/E3 Framer block to transmit the Unframed, All Ones DS3 AIS pattern, upon software command by executing the following steps.

STEP 1 - Set Bit 7 (TxAIS Unframed All Ones), within the Transmit DS3 Pattern Register to "1", as depicted below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
1	0	0	0	1	1	0	0

This step configures the Transmit DS3/E3 Framer block to generate an Unframed, All Ones pattern, whenever it is commanded to transmit the DS3 AIS indicator.

STEP 2 - Command the Transmit DS3/E3 Framer block to transmit the DS3 AIS indicator.

This is accomplished by setting Bit 4 (Tx AIS) to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	1	1	1

NOTES:

1. *The user can command the Transmit DS3/E3 Framer block to terminate its transmission of the DS3 AIS pattern by setting Bit 4 (TxAIS) back to "0".*
2. *Bit 4 (TxAIS) is ignored for the duration that Bit 3 (TxLOS) is set to "1".*

4.2.5.3 TRANSMITTING THE DS3 IDLE PATTERN

The Transmit DS3/E3 Framer block permits the user to transmit the DS3 Idle Pattern to the remote terminal equipment upon software command. The Transmit DS3/E3 Framer block provides the user with two options when transmitting the Idle Pattern.

- Transmit the Bellcore GR-499-CORE standard Framed, Repeating 1, 1, 0, 0, … pattern.
- Transmit a Framed, repeating user-defined pattern.

The procedure for transmitting either of these types of Idle patterns is presented below.

Transmitting the Bellcore GR-499-CORE standard Framed, Repeating 1, 1, 0, 0, … Pattern

The user can configure the Transmit DS3/E3 Framer block to transmit an Idle Pattern by transmitting a framed, repeating 1, 1, 0, 0, … pattern. The user can accomplish this by executing the following steps.

STEP 1 - Set Bits 3 through 0 (Transmit_Idle_Pattern[3:0]) within the Transmit DS3 Pattern Register to the value 1, 1, 0, 0, as illustrated below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	0	0

NOTE: *This step configures the Transmit DS3/E3 Framer block to transmit a Framed, Repeating 1, 1, 0, 0, … pattern, whenever the user commands it to transmit the Idle Pattern.*

STEP 2 - Command the Transmit DS3/E3 Framer block to generate and transmit the DS3 Idle Pattern

This is accomplished by setting Bit 5 (TxIdle) within the Transmit DS3 Configuration Register to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	1	1	1

Once the user executes this step, then the Transmit DS3/E3 Framer block will begin to generate and transmit a Framed, Repeating 1, 1, 0, 0, … Pattern.

NOTE: *The TxIdle bit-field is ignored whenever either Bits 3 (TxLOS) or 4 (TxAIS) are set to "1".*

Transmitting the Framed, User-Defined DS3 Idle Pattern

The user can configure the Transmit DS3/E3 Framer block to transmit an Idle Pattern by transmitting a framed, user-defined pattern. The user can accomplish this by executing the following steps.

STEP 1 - Set Bits 3 through 0 (Transmit_Idle_Pattern[3:0]) within the Transmit DS3 Pattern Register to the user-defined four-bit value, as illustrated below.

Transmit DS3 Pattern Register (Address = 0x114C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxAIS - Unframed All Ones	DS3 AIS Non-Stuck Stuff	Unused	TxLOS Pattern	Transmit_Idle_Pattern[3:0]			
R/W	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	X	Y	Z	A

NOTE: *This step configures the Transmit DS3/E3 Framer block to transmit a Framed, Repeating X, Y, Z, A, .. pattern, whenever the user commands it to transmit the Idle Pattern.*

STEP 2 - Command the Transmit DS3/E3 Framer block to generate and transmit the DS3 Idle Pattern

This is accomplished by setting Bit 5 (TxIdle) within the Transmit DS3 Configuration Register to "1" as depicted below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx Yellow Alarm	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	1	1	1

Once the user executes this step, then the Transmit DS3/E3 Framer block will begin to generate and transmit a Framed, Repeating X, Y, Z, A…. pattern.

NOTE: *This bit-field is ignored whenever either Bits 3 (TxLOS) or 4 (TxAIS) are set to "1".*

4.2.5.4 TRANSMITTING THE FERF/RDI INDICATOR

The Transmit DS3 Framer block provides the user with two options associated with transmitting the FERF (Far-End-Receive Failure) condition within the outbound DS3 data-stream.

- Forced transmission of the FERF indicator (e.g., under Software control)
- Automatic transmission of the FERF indicator.

Forced Transmission of the FERF/RDI Indicator

The XRT79L71 permits the user to force the Transmit DS3 Framer block to transmit the FERF indicator to the remote terminal equipment. The user can accomplish this by setting Bit 7 (Tx FERF Indicator) within the Transmit DS3 Configuration Register, to "1" as illustrated below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Tx FERF Indicator	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	1	1	1

To transmit the FERF Indicator then each of the X-bits, in the outbound DS3 frames will be set to '0'

NOTE: *This bit-field is ignored when either the TxIDLE, TxAIS, or the TxLOS bit-fields are set to "1".*

In this case, to terminate the Transmit DS3/E3 Framer block's transmission of the FERF indicator, then the user only needs to set Bit 7 back to "0".

Automatic Transmission of the FERF/RDI Indicator

The XRT79L71 permits the user to configure the Transmit DS3 Framer block to automatically transmit the FERF/RDI indicator, in response to the following conditions.

- Whenever the corresponding near-end Receive DS3/E3 Framer block declares the LOS (Loss of Signal) defect condition
- Whenever the corresponding near-end Receive DS3/E3 Framer block is declaring the OOF (Out of Frame) defect condition
- Whenever the corresponding near-end Receive DS3/E3 Framer block is declaring the AIS defect condition.

Setting bits 0 through 2, within the Transmit DS3 Configuration Register, permits the user to configure the Transmit DS3/E3 Framer block to support the Automatic Transmission of the FERF indicator upon detection of these Red Alarm condition. These bit-fields are highlighted and defined below.

Transmit DS3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Transmit FERF/RDI Indicator	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS Defect	TxFERF upon OOF-Defect	TxFERF upon AISDe-fect
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	1	1	1

Bit Number	Name	Type	Description
2	TxFERF upon LOS Defect	R/W	Automatic Transmission of FERF upon LOS Defect:This bit-field permits the user to configure the Frame Generator block to automatically transmit the FERF Indicator to the remote terminal, upon the Receive DS3/E3 Framer block declaring the LOS defect condition. If this feature is enabled, then the Transmit DS3/E3 Framer block will transmit the FERF indicator to the remote terminal for the duration that the companion Receive DS3/E3 Framer block is declaring the LOS defect condition. The Transmit DS3/E3 Framer block will cease to transmit the FERF indicator once the Receive DS3/E3 Framer block has cleared the LOS defect condition.0 - Disables the Automatic Transmission of FERF upon declaration of the LOS defect feature.1 - Enables the Automatic Transmission of FERF upon declaration of the LOS defect feature.

1	TxFERF upon OOF Defect	R/W	Automatic Transmission of FERF upon OOF Defect:This bit-field permits the user to configure the Transmit DS3/E3 Framer block to automatically transmit the FERF indicator to the remote terminal, upon the Receive DS3/E3 Framer block declaring the OOF/LOF defect condition. If this feature is enabled, then the Transmit DS3/E3 Framer block will transmit the FERF indicator to the remote terminal for the duration that the companion Receive DS3/E3 Framer block is declaring the OOF defect condition. The Transmit DS3/E3 Framer block will cease to transmit the FERF indicator once the Receive DS3/E3 Framer block has cleared the OOF defect condition.0 - Disables the Automatic Transmission of FERF upon declaration of the OOF defect feature.1 - Enables the Automatic Transmission of FERF upon declaration of the OOF defect feature.
0	TxFERF upon AIS Defect	R/W	Automatic Transmission of FERF upon AIS Defect:This bit-field permits the user to configure the Transmit DS3/E3 Framer block to automatically transmit the FERF indicator to the remote terminal, upon the Receive DS3/E3 Framer block declaring the AIS defect condition. If this feature is enabled, then the Transmit DS3/E3 Framer block will transmit the FERF indicator to the remote terminal for the duration that the companion Receive DS3/E3 Framer block is declaring the AIS defect condition. The Transmit DS3/E3 Framer block will cease to transmit the FERF indicator once the Receive DS3/E3 Framer block has cleared the AIS defect condition.0 - Disables the Automatic Transmission of FERF upon declaration of the AIS defect feature.1 - Enables the Automatic Transmission of FERF upon declaration of the AIS defect feature.

NOTE: *Bits 2 through 0 are only active if Bit 7 (Tx FERF Indicator), within the Transmit DS3 Configuration register is set to "0".*

In summary, the XRT79L71 can be configured to transmit the FERF indicator to the remote terminal equipment anytime its Receive DS3/E3 Framer block detects either an LOS, OOF/LOF or AIS defect condition in the DS3 data that it is receiving from the remote terminal equipment. In the case of DS3 applications, the Transmit DS3/E3 Framer block will transmit the FERF indicator to the remote terminal by setting each of the X bits within each outbound DS3 frame to "0".

Conversely, if the Receive DS3/E3 Framer block within the XRT79L71 does not declare either an LOS, OOF or AIS defect condition, then the corresponding Transmit DS3/E3 Framer block will respond by NOT sending the FERF indicator to the remote terminal equipment. In this case, the Transmit DS3/E3 Framer block will NOT transmit the FERF indicator, by setting the X bits within each DS3 frame to "1".

Figure 59 through **Figure 60** illustrate this phenomenon.

Figure 59 illustrates the Near-End Receive DS3/E3 Framer block within the XRT79L71 declaring the LOS defect condition within the DS3 data-stream that it is receiving from the remote terminal equpment. **Figure 60** illustrates the subsequent action of the corresponding Near-End Transmit DS3/E3 Framer block. In this case, the Transmit DS3/E3 Framer block will set each of the X bits, within the outbound DS3 frame which is destined for the remote terminal to "0".

FIGURE 59. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK DECLARING THE LOS DEFECT CONDITION

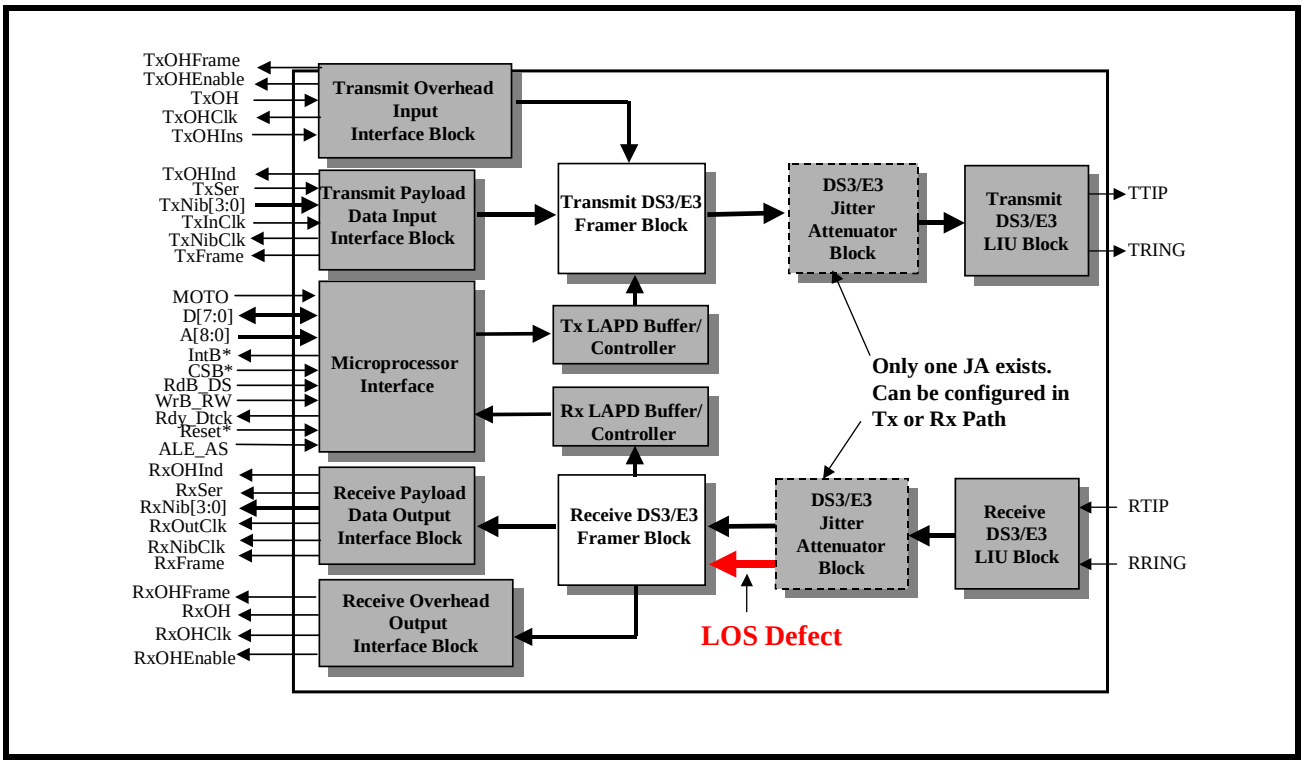


FIGURE 60. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK, TRANSMITTING A DS3 FRAME TO THE REMOTE TERMINAL WITH THE X BITS SET TO "0"

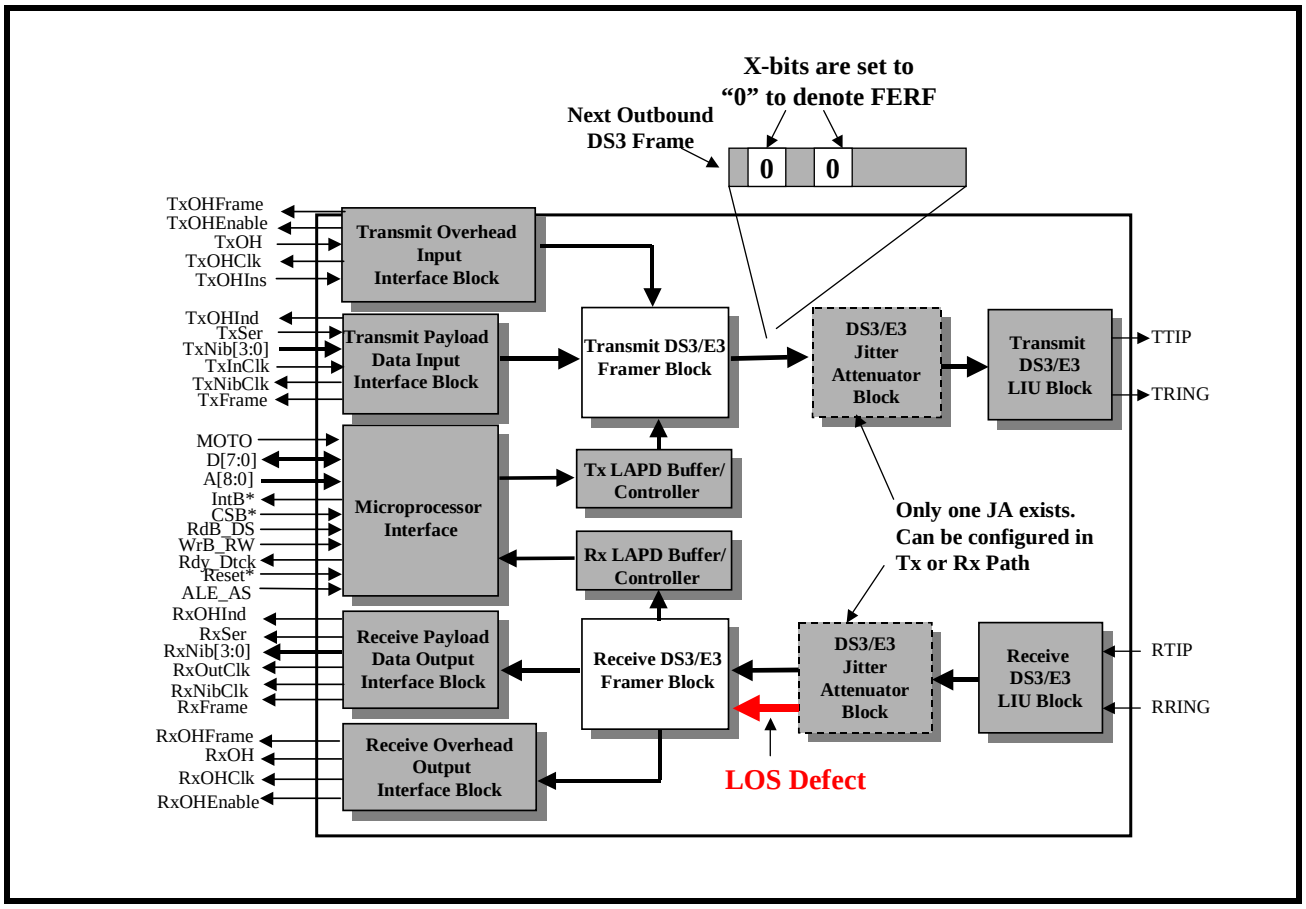


Figure 61 illustrates the Near-End Receive DS3/E3 Framer block receiving a normal DS3 data-stream from the remote terminal. Figure 62 illustrates the subsequent action of the Near-End Transmit DS3/E3 Framer block. In this case, the Transmit DS3/E3 Framer block will set each of the X bits, in the outbound DS3 frame which is destined for the remote terminal to "1".

FIGURE 61. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK RECEIVING A PROPER DS3 SIGNAL FROM THE REMOTE TERMINAL EQUIPMENT (E.G., THE LOS DEFECT CONDITION IS CLEARED)

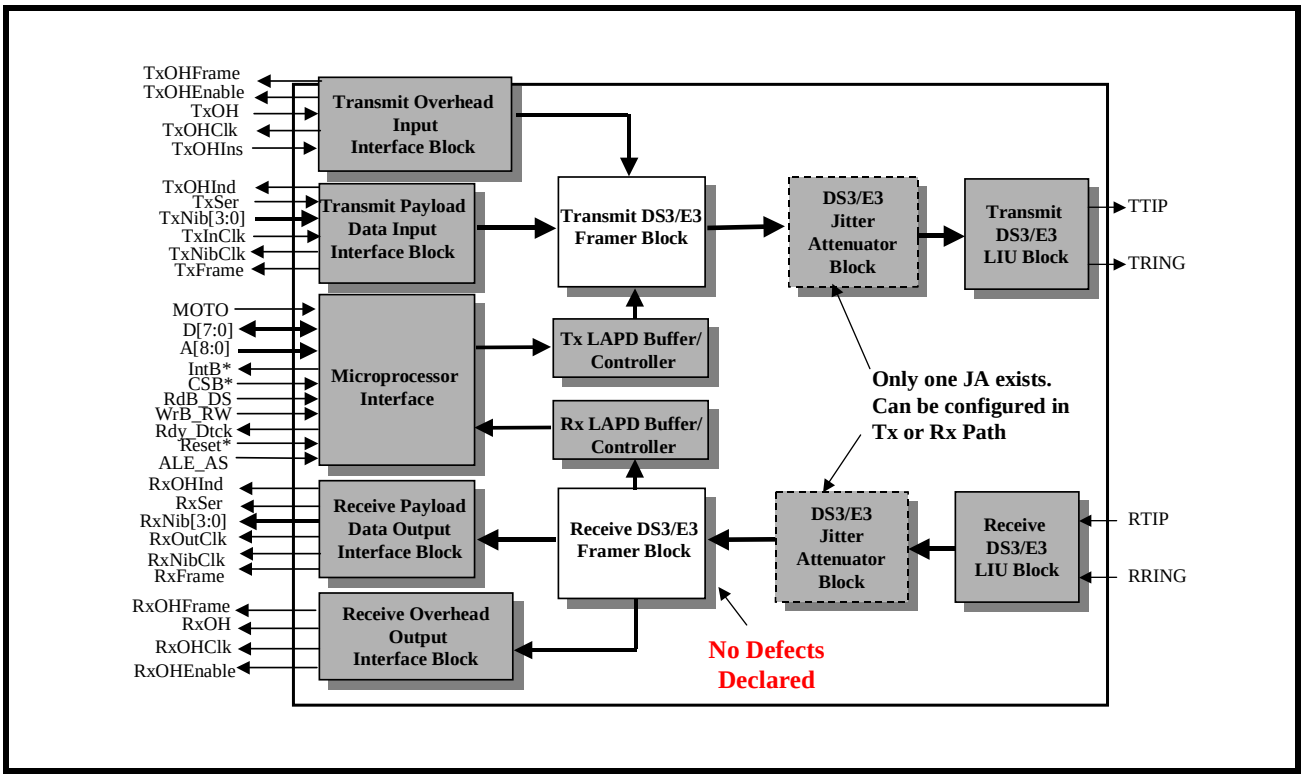
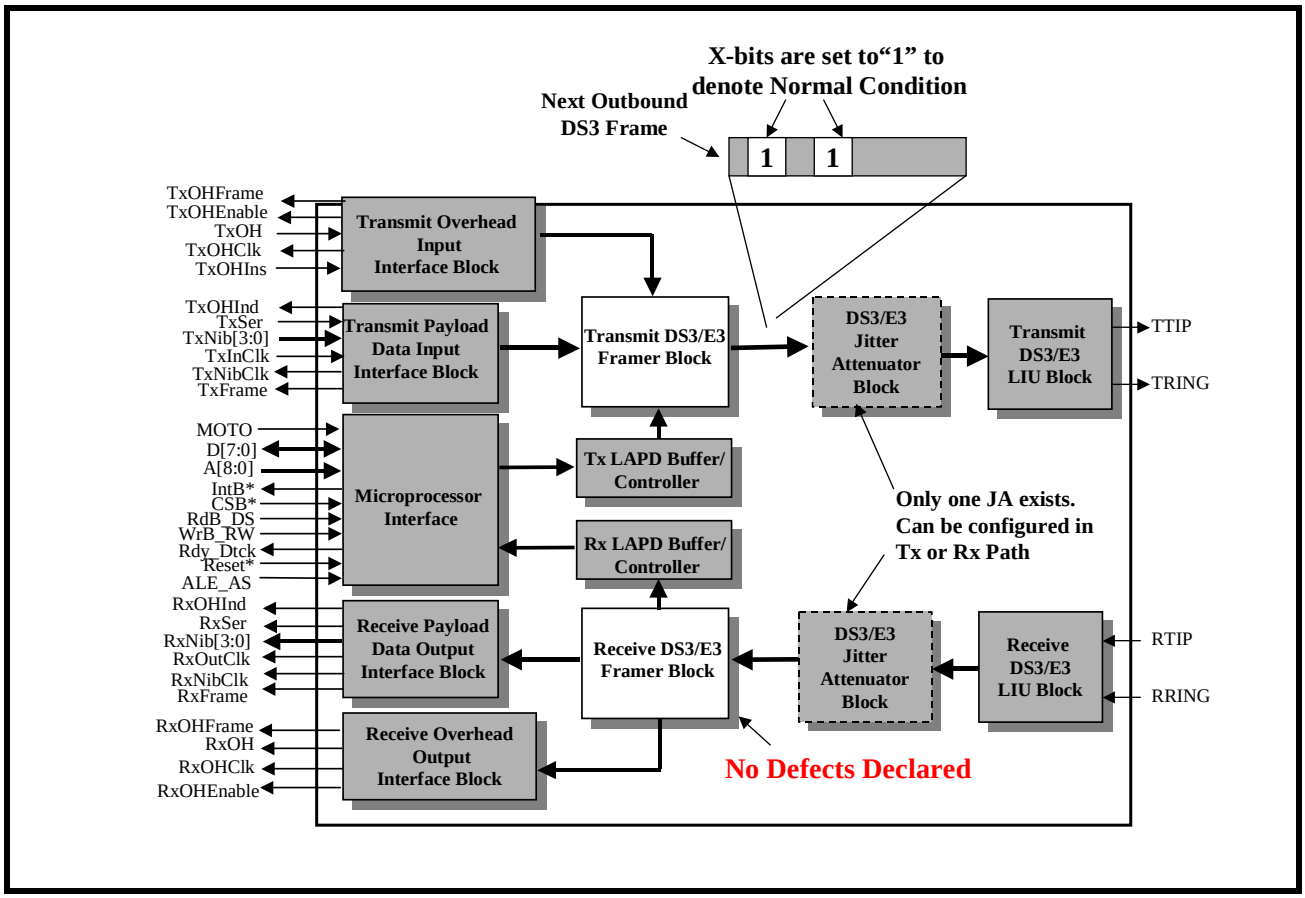


FIGURE 62. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK, TRANSMITTING A DS3 FRAME TO THE REMOTE TERMINAL EQUIPMENT WITH EACH OF THE X BITS SET TO "1"



4.2.5.5 SETTING X BITS TO "1"

The Transmit DS3 Framer block permits the user to force all of the X bits to "1". This bit field functions as the logical inverse of Bit 7 (Tx FERF Indicator). The user can accomplish this by setting Bit 6 (Tx X Bits) within the Transmit DS3 Configuration Register, to "1" as illustrated below.

Transmit DS3 Configuration Register (Indirect Address = 0xNE, 0x30; Direct Address = 0x1130)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Tx FERF Indicator	Tx X-Bits	TxIdle	TxAIS	TxLOS	TxFERF upon LOS	TxFERF upon OOF	TxFERF upon AIS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	1	1	1

This read/write bit field permits the user to force each of the X-bits, in each outbound DS3 frame, to "1" and transmit them to the remote terminal equipment.

NOTE: *This bit is ignored when either the Transmit FERF Indicator, Tx AIS, Tx IDLE, or TxLOS bit is set.*

4.2.5.6 TRANSMITTING THE FEBE (FAR-END BLOCK ERROR) INDICATOR

If the Transmit DS3/E3 Framer block is configured to support the DS3, C-bit Parity framing format, then it will be capable of transmitting the FEBE (Far-End-Block-Error) indicator to the remote terminal equipment.

The purpose of the FEBE bit-fields, within the DS3 frame is two-fold.

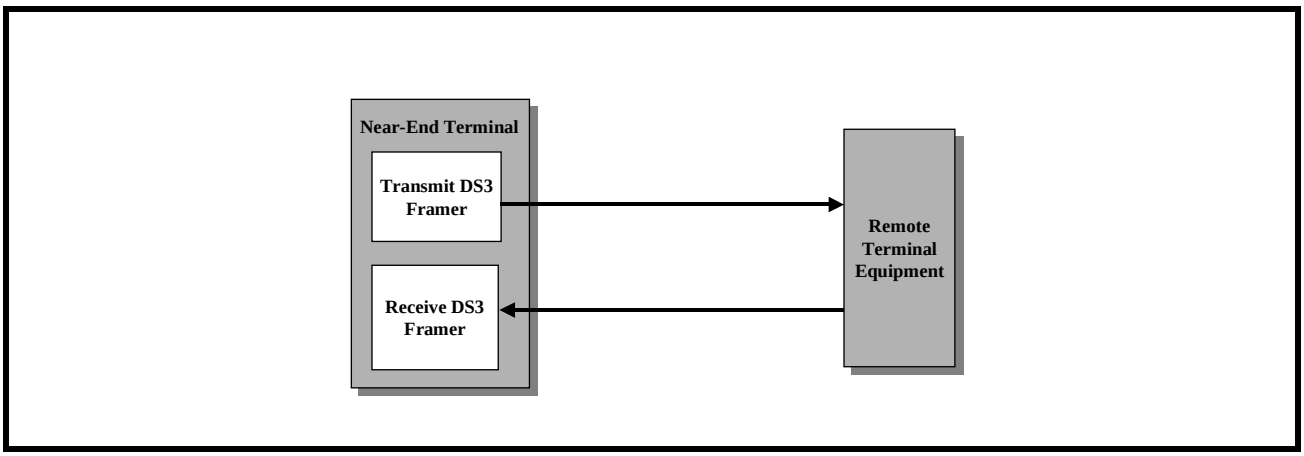
1. It permits a Terminal which is transmitting a DS3 data stream to a remote terminal to determine whether or not this remote terminal is receiving its DS3 data, in an error-free manner.
2. It permits a Terminal which is receiving a DS3 data stream from a remote terminal to inform this remote terminal when it is receiving erred DS3 frames.

The role of the FEBE bit-field is best presented in the practical example below.

Example:

Consider a Near-End terminal that is communicating with a remote terminal. This Near-End terminal consists of the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block within the XRT79L71 IC as depicted in Figure 63.

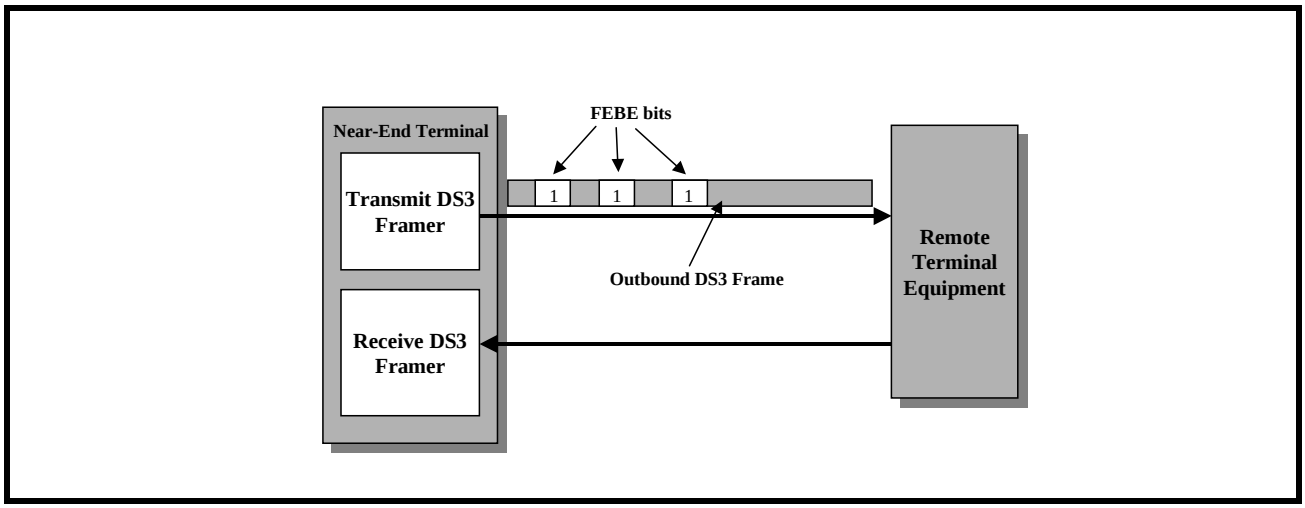
FIGURE 63. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK RECEIVING A DS3 FRAME FROM THE REMOTE TERMINAL WITH CORRECT F, M AND CP BITS



The Transmit DS3/E3 Framer block will generate and transmit DS3 frames to the remote terminal. Likewise, the Receive DS3/E3 Framer block will receive and process DS3 frames, originating from the remote terminal. The Near-End Receive DS3/E3 Framer block (e.g., the Receive DS3/E3 Framer block within this particular device) is going to verify the values of the framing (e g., the F and M) bits, as well as the CP bits, within the incoming DS3 frames from the remote terminal. If the Near-End Receive DS3/E3 Framer block detects no framing bit or CP bit errors, in the incoming DS3 frame, then it will notify the remote terminal of this fact, by forcing the Near-End Transmit DS3/E3 Framer block, set the FEBE bits within the very next outbound DS3 frame which is destined for the remote terminal to a 1, 1, 1 pattern. This phenomenon is illustrated in Figure 64 and Figure 65 below.

Figure 63 illustrates the Near-End Receive DS3/E3 Framer block receiving an error-free DS3 frame. In this figure, the locally computed CP bits of "0" matches that received from the remote terminal. Figure 64 illustrates the subsequent action of the Near-End Transmit DS3/E3 Framer block which will transmit a DS3 frame, with the FEBE bit-fields set to "1, 1, 1" to the remote terminal. This signaling indicates that the Near-End Receive DS3/E3 Framer block has received an error-free DS3 frame from the remote terminal.

FIGURE 64. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK TRANSMITTING A DS3 FRAME TO THE REMOTE TERMINAL WITH THE FEBE BITS SET TO "1, 1, 1"



Conversely, if the Near-End Receive DS3/E3 Framer block detects an error in the incoming F, M or CP bits, then it will notify the remote terminal of this fact, by forcing the Near-End Transmit DS3/E3 Framer block set the FEBE bits, within an outbound DS3 frame which is destined for the remote terminal to some pattern other than "1, 1, 1". This phenomenon is illustrated below in Figures Figure 65 and Figure 66.

Figure 65 illustrates the Near-End Receive DS3/E3 Framer block receiving an erred DS3 frame from the remote terminal. In this figure, the Near-End Receive DS3/E3 Framer block is receiving a DS3 frame, within a CP bit that is of the value "1". This value does not match the locally computed CP bit value of "0". Consequently, there is an error in this DS3 frame.

FIGURE 65. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK RECEIVING A DS3 FRAME FROM THE REMOTE TERMINAL WITH AN INCORRECT CP BIT

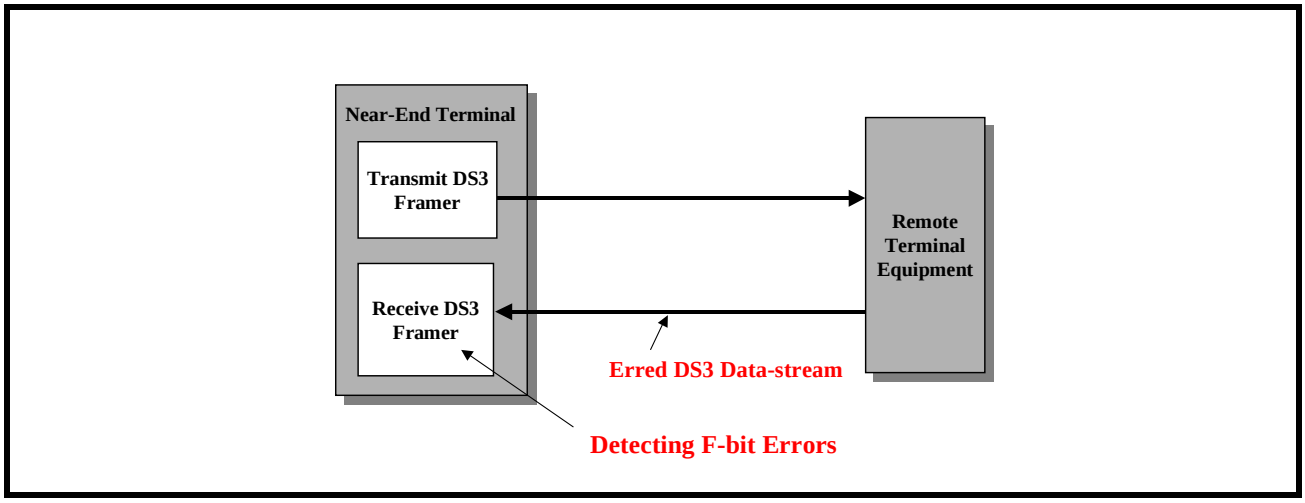
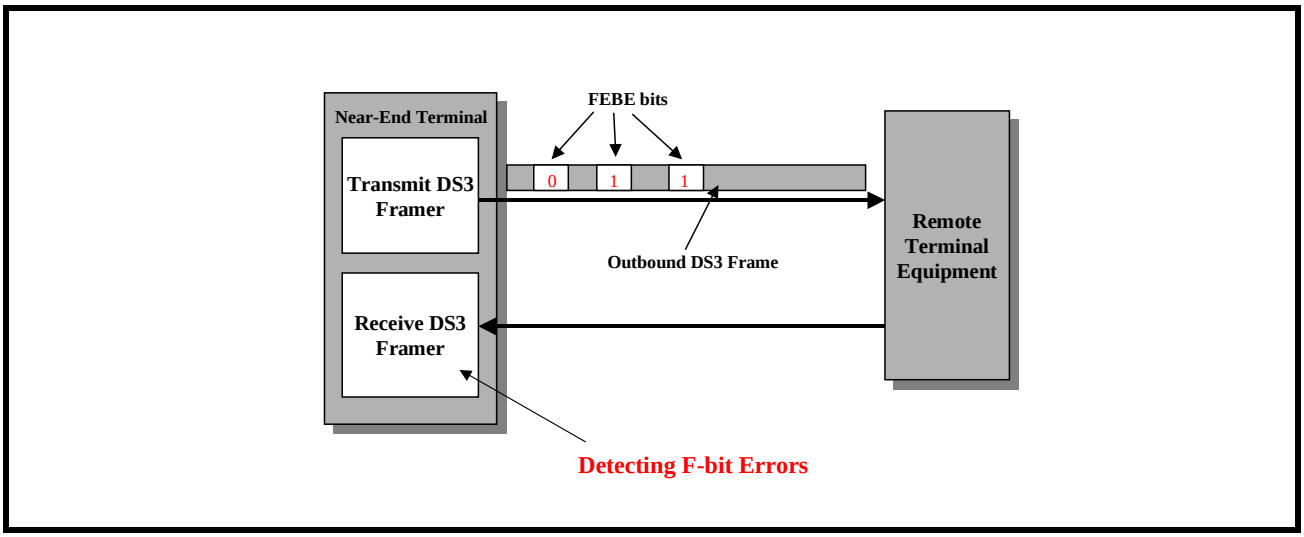


Figure 66 illustrates the subsequent action of the Near-End Transmit DS3/E3 Framer block, which will transmit a DS3 frame, with the FEBE bit-fields set to some pattern other than "1, 1, 1" to the remote terminal. In this case, the Transmit DS3/E3 Framer block is setting the FEBE bits to the value [0, 1, 1]. This signaling indicates that the Near-End Receive DS3/E3 Framer block has received an erred DS3 frame the remote terminal.

FIGURE 66. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK, TRANSMITTING A DS3 FRAME TO THE REMOTE TERMINAL WITH THE FEBE BITS SET TO "0, 1, 1"



For information on how the Receive DS3/E3 Framer block processes the FEBE bit-fields, within each incoming DS3 frame, please see Section 4.3.2.11.

The Transmit DS3 Framer block provides the user with two options associated with transmitting the FEBE indicator.

- Forced transmission of a FEBE indicator (e.g., under Software Control).
- Automatic transmission of the FEBE indicator in response to the Receive DS3/E3 Framer block detecting Framing bit or CP-bit errors

4.2.5.6.1 Forced Transmission of the FEBE Indicator.

The XRT79L71 permits the user to force the Transmit DS3/E3 Framer block to transmit any FEBE value to the remote terminal equipment. The user can accomplish this by executing the following two-step procedure.

STEP 1 - Write the desired FEBE value into Bits 5 through 7 (TxFEBEDat[2:0]), within the Transmit DS3 M-Bit Mask Register, as indicated below.

Transmit DS3 M-Bit Mask Register (Address = 0x1135)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFEBEDat[2:0]			FEBE Register Enable	Tx P-Bit Error	TxM_Bit_Mask[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	0	0	0	0	0

STEP 2 - Configure the Transmit DS3/E3 Framer block to load the value of the FEBE bit-fields within each outbound DS3 frame to the contents within the TxFEBEDat[2:0] bit-fields within the Transmit DS3 M-Bit Mask Register.

This is accomplished by setting Bit 4 (FEBE Register Enable) to "1", as illustrated below.

Transmit DS3 M-Bit Mask Register (Address = 0x1135)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFEBEDat[2:0]			FEBE Register Enable	Tx P-Bit Error	TxM_Bit_Mask[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	1	0	0	0	0

Once the user executes these two steps, then the Transmit DS3/E3 Framer block will automatically set the FEBE bit-fields within each outbound DS3 frame to the value written into Bits 7 through 5 (TxFEBEDat[2:0]) within the Transmit DS3 M-Bit Mask Register.

NOTE: *In this configuration, the value of the FEBE bit-fields that are being generated by the Transmit DS3/E3 Framer block do not reflect the health of the DS3 signal that is being received by the corresponding Near-End Receive DS3/E3 Framer block.*

4.2.5.6.2 Automatic Transmission of the FEBE Indicator

If the Forced Transmission of the FEBE Indicator feature (as described above) is not used, then the Transmit DS3 Framer block will be configured to automatically set the FEBE bit-fields, within the outbound DS3 data stream, based upon occurrences as detected by the corresponding Receive DS3/E3 Framer block.

If the companion Receive DS3/E3 Framer block does not detect any F, M or CP-bit errors, then the Transmit DS3/E3 Framer block will respond by setting all of the FEBE bit-fields within each outbound DS3 frame to "1". Hence, in this case, the FEBE value, for each outbound DS3 data stream will be set to "1, 1, 1". According to Bellcore GR-499-CORE, the FEBE value of "1, 1, 1" represents an un-erred condition.

Conversely, if at a given instant, the companion Receive DS3/E3 Framer block does detect any F, M or CP-bit errors, then the Transmit DS3/E3 Framer block will respond by setting the FEBE value within its next outbound DS3 frame to a value other than "1, 1, 1". According to Bellcore GR-499-CORE, the FEBE value of something other than "1, 1, 1" represents an erred condition. In general, the Transmit DS3/E3 Framer block will transmit an erred FEBE indicator each time that the corresponding Near-End Receive DS3/E3 Framer block receives an erred DS3 frame.

NOTE: *The Automatic Transmission of FEBE Indicator feature is enabled by setting Bit 4 (FEBE Register Enable) within the Transmit DS3 M-Bit Mask Register to "0", as illustrated below.*

Transmit DS3 M-Bit Mask Register (Address = 0x1135)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxFEBEDat[2:0]			FEBE Register Enable	Tx P-Bit Error	TxM_Bit_Mask[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

4.2.5.7 Setting the Transmit DS3 Framer Block Timing Reference

When designing your system, a decision must be made as to whether the system is going to be configured to operate in either the Local-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from a "local" [or in-system] clock source) or in the Loop-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from the "remote terminal equipment's" clock source. The XRT79L71 can be configured to support either of these applications.

In all, the XRT79L71 supports the following three (3) different "Timing Reference" Modes.

- Local-Timing/Asynchronous
- Local-Timing/TxFrameRef
- Loop-Timing

Each of these "Timing Reference" Modes is discussed in some detail below.

4.2.5.7.1 Local-Timing/Asynchronous Mode

If the XRT79L71 is configured to operate in the "Local-Timing/Asynchronous Mode, then all of the following is true.

- The Transmit DS3 Framer block will use the clock signal that is applied to the "TxInClk" input pin as its timing source, for generating and transmitting the outbound DS3 traffic to the remote terminal equipment.
- DS3 Frame Generation (e.g., the instant whenever the Transmit DS3 Framer block begins to generate and transmit a new DS3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT79L71.

Configuring the XRT79L71 to operate in the Local-Timing/Asynchronous Mode

The XRT79L71 can be configured to operate in the "Local-Timing/Asynchronous" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[1, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT79L71 is configured to operate in the "Local-Timing/Asynchronous" Mode, then do the following.

- apply a 44.736MHz clock signal to the TxInClk input pin (Ball C10).
- Tie the TxFrameRef input pin (Ball A11) to GND.

NOTE: *Configuring the XRT79L71 to operate in the "Local-Timing/Asynchronous" Mode, is synonymous with configuring the Transmit Payload Data Input Interface block into "Mode 3" (for Serial-Mode Applications - see Section 4.2.1.3) or into "Mode 6" (for Nibble-Parallel Applications - see Section 4.2.1.6).*

4.2.5.7.2 Local-Timing/TxFrameRef Mode

If the XRT79L71 is configured to operate in the "Local-Timing/TxFrameRef Mode, then all of the following is true.

- The Transmit DS3 Framer block will use the clock signal that is applied to the "TxInClk" input pin as its timing source, for generating and transmitting the outbound DS3 traffic to the remote terminal equipment.
- The Transmit DS3 Framer block will initiate the generation (and transmission) of a new DS3 frame, anytime it detects a rising edge at the TxFrameRef input pin. In this case DS3 Frame Generation (e.g., the instant whenever the Transmit DS3 Framer block begins to generate and transmit a new DS3 frame) will be synchronized to the signal that is applied to the TxFrameRef input pin.

Configuring the XRT79L71 to operate in the Local-Timing/TxFrameRef Mode

The uXRT79L71 can be configured to operate in the "Local-Timing/TxFrameRef" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT79L71 is configured to operate in the "Local-Timing/TxFrameRef" Mode, then the following must be done.

- Apply a 44.736MHz clock signal to the TxInClk input pin (Ball C10).
- Apply a 9.398kHz clock signal to the TxFrameRef input pin (Ball A11).

· This 9.398kHz clock signal (that is being applied to the TxFrameRef input pin) must be synchronous with the 44.736MHz clock signal (that is being applied to the TxInClk input pin).

NOTE: *Configuring the XRT79L71 to operate in the "Local-Timing/TxFrameRef" Mode, is synonymous with configuring the Transmit Payload Data Input Interface block into "Mode 2" (for Serial-Mode Applications - see Section 4.2.1.2) or into "Mode 5" (for Nibble-Parallel Applications - see Section 4.2.1.5).*

4.2.5.7.3 Loop-Timing Mode

If the XRT79L71 is configured to operate in the "Loop-Timing Mode, then all of the following is true.

- The Transmit DS3 Framer block will use the "Recovered" clock signal (from the Receive DS3/E3 LIU Block) as its timing source, for generating and transmitting the outbound DS3 traffic to the remote terminal equipment.
- DS3 Frame Generation (e.g., the instant whenever the Transmit DS3 Framer block begins to generate and transmit a new DS3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT79L71.

Configuring the XRT79L71 to operate in the Loop-Timing Mode

To XRT79L71 to operate in the "Local-Timing/TxFrameRef" Mode set Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 0]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Loop-Timing" Mode

If the XRT79L71 is configured to operate in the "Loop-Timing" Mode, then the following must be done.

- Insure that either one of the following conditions are true.
  - **a.** That the Receive DS3/E3 LIU Block is receive a proper DS3 line signal from the remote terminal equipment, or
  - **b.** That the SFM Synthesizer block (within the Receive DS3/E3 LIU Block) is configured to generate a 44.736MHz clock (to the remainder of the Receive DS3/E3 LIU Block circuitry) from either an externally supplied 12.288MHz or a 44.736MHz clock signal. (Please see Section 4.3.1.5 for details on how to accomplish this)
- Tie the TxFrameRef input pin (Ball A11) to GND.

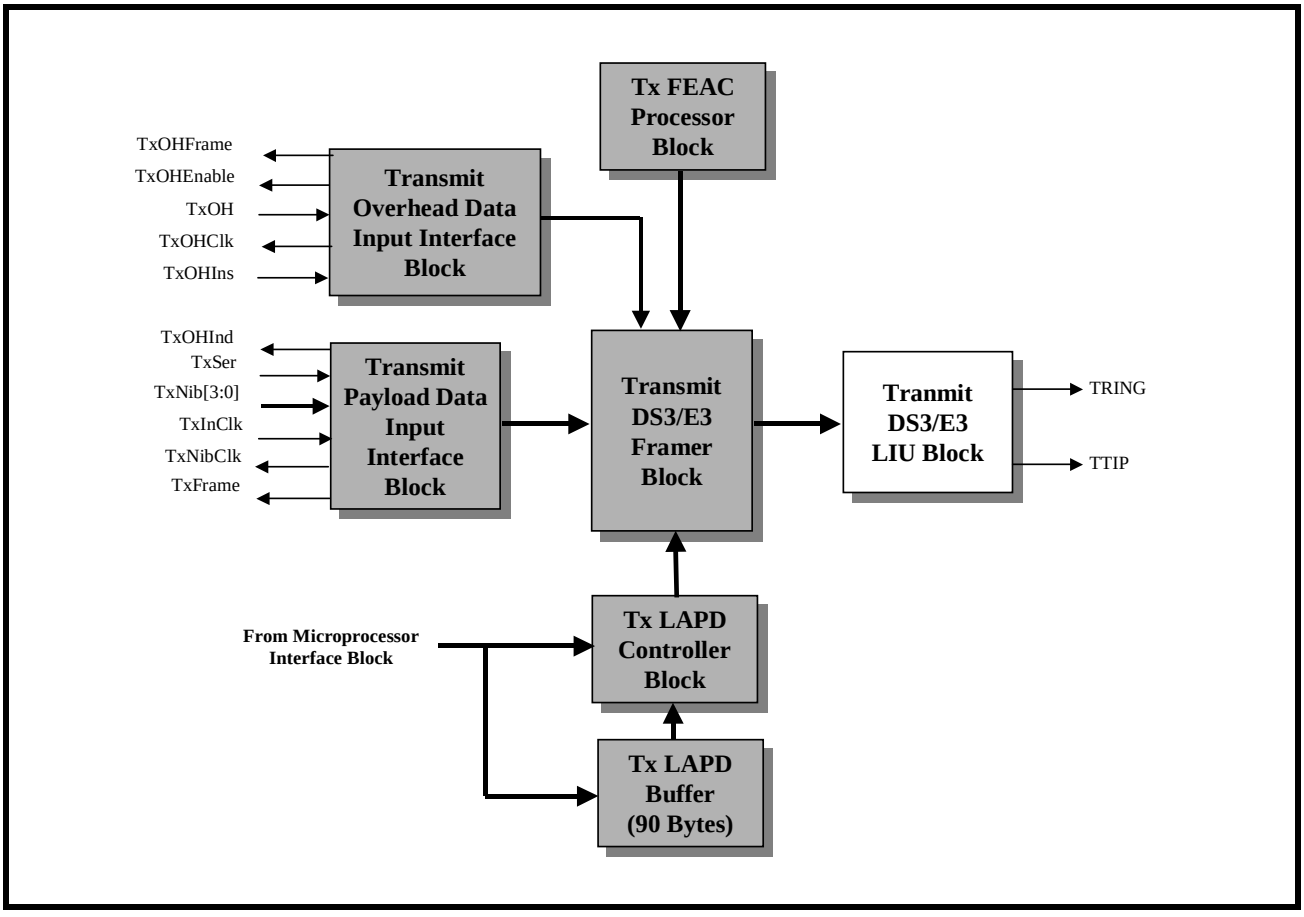
NOTES:

1. *In order to permit the Microprocessor Interface to function (for Revision A silicon) the user is still required to supply a sufficiently high frequency clock signal to the "TxInClk" input pin, even if the XRT79L71 is configured to operate in the "Loop-Timing" Mode.*
2. *Configuring the XRT79L71 to operate in the "Loop-Timing" Mode is synonymous with configuring the Transmit Payload Data Input Interface block into "Mode 1" (for Serial-Mode Applications - see SSection 4.2.1.1) or into "Mode 4" (for Nibble-Parallel Applications - see Section 4.2.1.4).*

4.2.6 TRANSMIT DS3/E3 LIU BLOCK - DS3 APPLICATIONS

The Transmit DS3/E3 LIU Block is the sixth and last functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 67 presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Transmit DS3/E3 LIU block highlighted.

FIGURE 67. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT DS3/E3 LIU BLOCK HIGHLIGHTED)



The purpose of the Transmit DS3/E3 LIU Block is to accept a DS3/E3 data-stream from the Transmit DS3/E3 Framer block and to perform all of the following operations on this data.

- To convert this outbound data into the B3ZS line code (for DS3 applications) or into the HDB3 line code (for E3 applications)
- For DS3 Applications, to further convert this data into a DS3 line signal such that each pulse that is generated by the Transmit DS3 LIU block will comply with the Isolated Pulse Template requirements per Bellcore GR-499-CORE
- For E3 Applications, to further convert this data into an E3 line signal such that each pulse that is generated by the Transmit E3 LIU block will comply with the ITU-T G.703 Pulse Template requirements for E3 applications.

This particular section will describe the functionality and configuration options of the Transmit DS3/E3 LIU Block for DS3 applications. The functionality and configuration options of the Transmit DS3/E3 LIU Block for E3 applications are discussed in Section 5.2.5.

Figure 68 presents a more detailed illustration of the Transmit DS3/E3 LIU Block within the XRT79L71.

FIGURE 68. ILLUSTRATION OF THE TRANSMIT DS3/E3 LIU BLOCK WITHIN THE XRT79L71

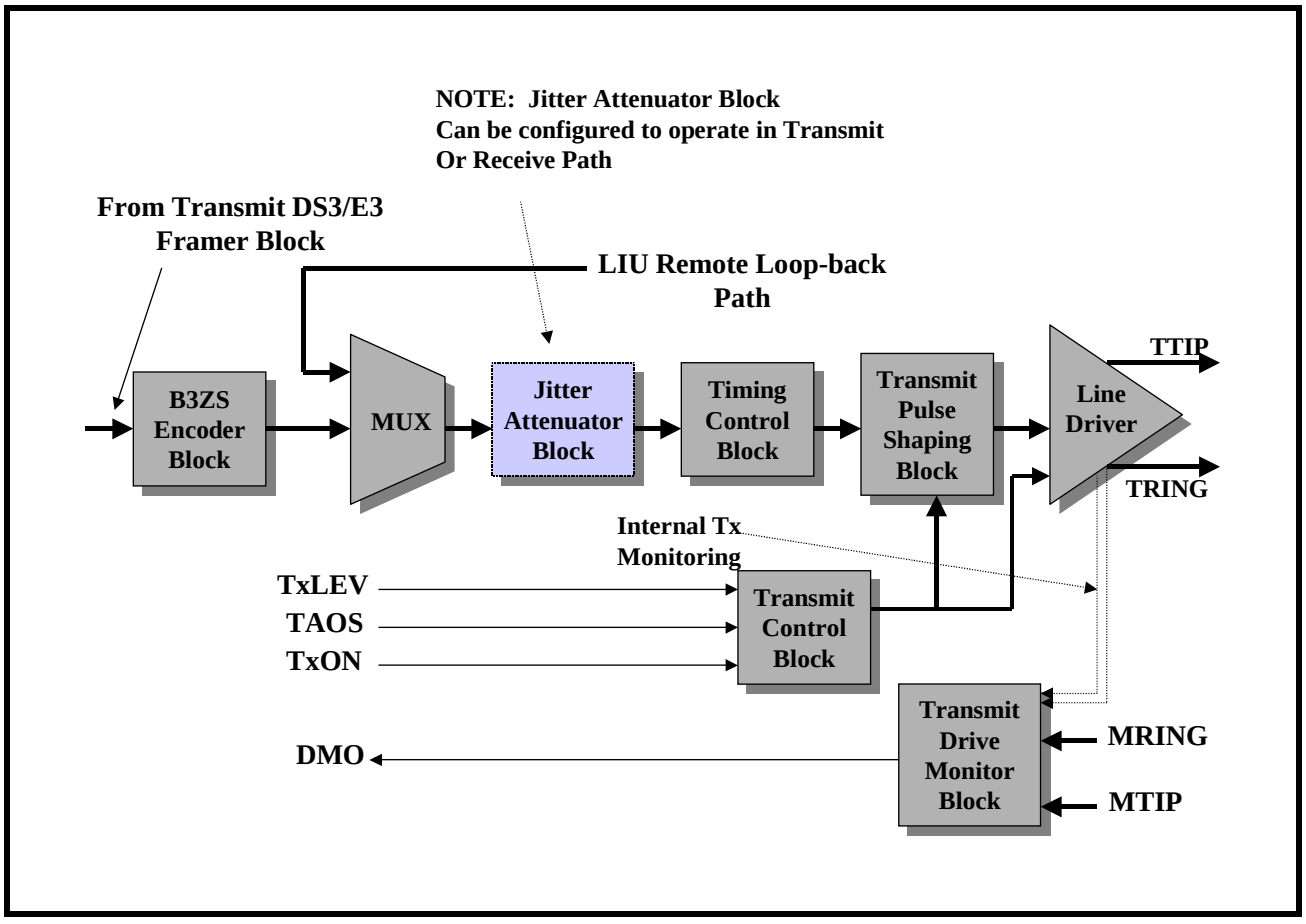


Figure 68 indicates that the Transmit DS3/E3 LIU Block consists of the following functional blocks.

- The B3ZS Encoder Block
- The Jitter Attenuator Block (which can be configured to operate in either the Transmit or Receive Directions)
- The Timing Control Block
- The Transmit Pulse Shaping Block

- The Transmit Line Driver Block
- The Transmit Drive Monitor Block

4.2.6.1 The B3ZS Encoder Block

The purpose of the B3ZS Encoder block is to encode the outbound DS3 traffic into the B3ZS Line Code. In the case of the XRT79L71, the B3ZS Encoder block will always be enabled, and the user has no ability to disable the B3ZS Encoder block.

4.2.6.2 The Jitter Attenuator Block

The XRT79L71 includes a Jitter Attenuator block that can be configured to operate in either the Transmit Direction (e.g., within the Transmit DS3/E3 LIU Block) or in the Receive Direction (e.g., within the Receive DS3/E3 LIU Block). The purpose of the Jitter Attenuator block is to permit the XRT79L71 to comply with all of the following Jitter Transfer Characteristic requirements.

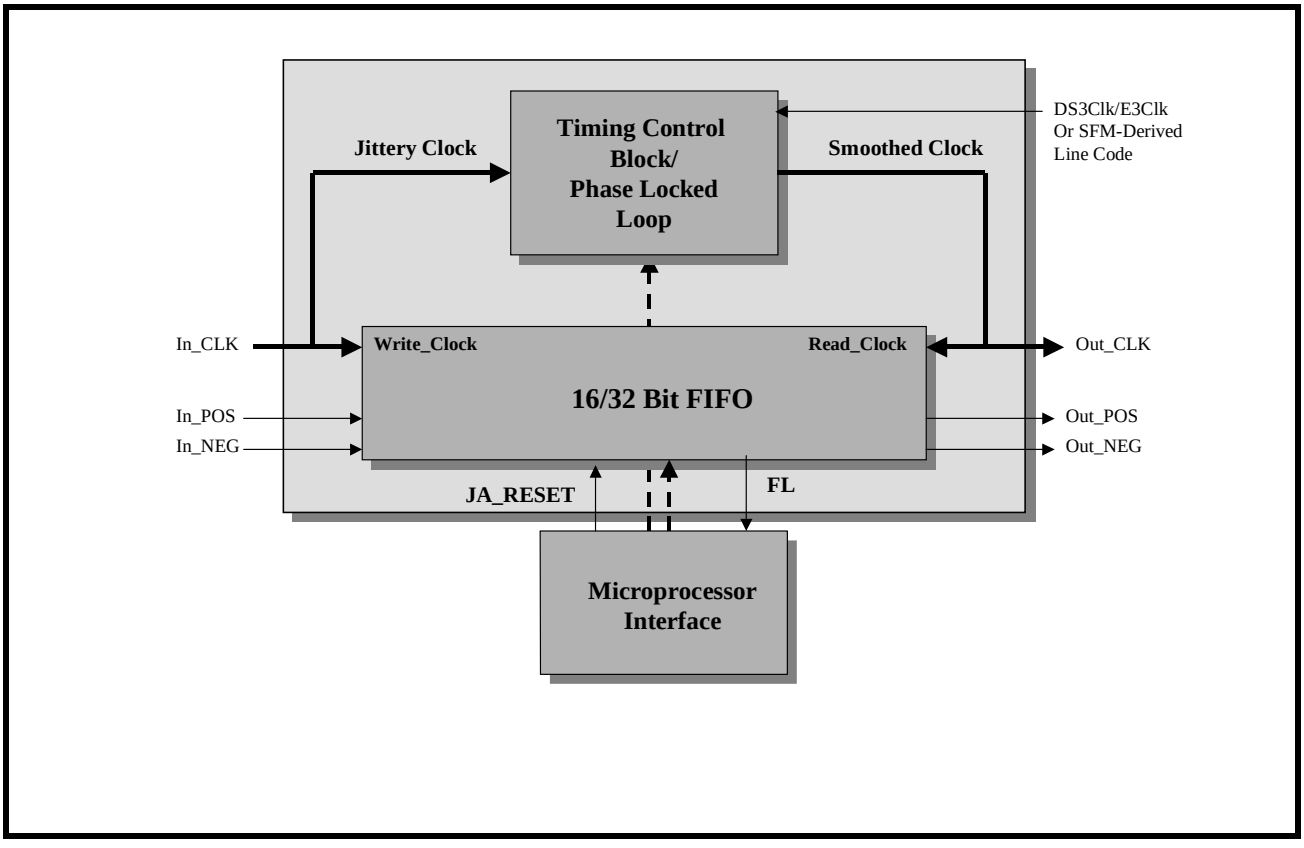
- Bellcore GR-499-CORE Category II to Category II Interfaces (DS3 Applications)
- TBR-24 34Mbps D34U and D34S System Requirements (for E3 Applications)

Each of these requirements, and how the XRT79L71 performs against these requirements is described in detail below.

4.2.6.2.1 The Basic Architecture of the Jitter Attenuator Block

Figure 69 presents an illustration of the Functional Block Diagram of the Jitter Attenuator Block

FIGURE 69. AN ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE JITTER ATTENUATOR BLOCK



According to Figure 69 the Jitter Attenuator consists of the following functional blocks.

- The Timing Control/Phase-Locked Loop Block

- The Jitter Attenuator FIFO

How the Jitter Attenuator Works

The XRT79L71 Jitter Attenuator Block will accept jittery clock (In_Clk) and data signals (In_POS and In_NEG) from either the HDB3/B3ZS Encoder Block (if the Jitter Attenuator has been configured to operate in the Transmit Direction) or from the Clock and Data Recovery Block (if the Jitter Attenuator has been configured to operate in the Receive Direction). The Jitter Attenuator block will latch the data, residing on internal signals In_POS and In_NEG into the Jitter Attenuator FIFO upon the appropriate edge of the In_Clk signal.

In parallel, the In_Clk signal will also be routed to the Timing Control/PLL Block. This block consists of a Narrow-band PLL which has the responsibility of attenuating much of the jitter within the In_Clk signal. The smoothed version of this clock signal will be output from the Jitter Attenuator block (and is designated as Out_Clk in **Figure 69**). Further, the Jittery Data (which was latched into the Jitter Attenuator FIFO via the In_Clk signal) will now be clocked out of the Jitter Attenuator block via the Out_Clk (e.g., the smoothed output clock) signal.

NOTES:

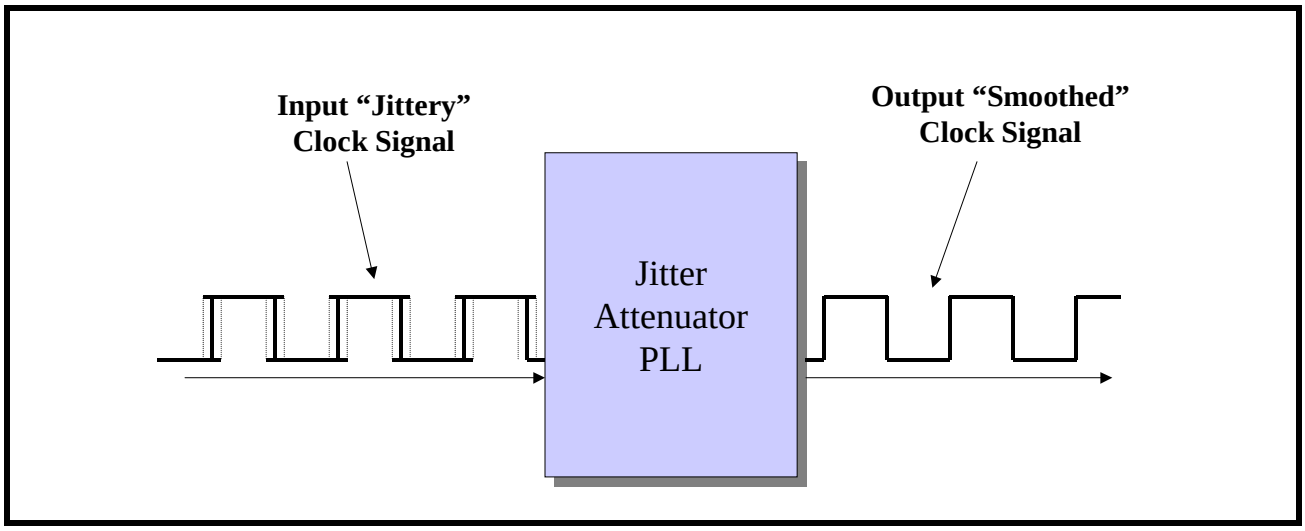
1. *If the Jitter Attenuator block has been configured to operate in the Transmit Direction and if the XRT79L71 has been configured to operate in the Local-Timing Mode, then the In_CLK signal (as depicted in **Figure 69**, above) will be a buffered version of clock signal being applied to the TxInClk input pin.*
2. *If the Jitter Attenuator block has been configured to operate in the Transmit Direction and if the XRT79L71 has been configured to operate in the Loop-Timing Mode, then the In_CLK signal (as depicted in **Figure 69**, above) will be a buffered version of the Recovered clock signal (from the Receive DS3/E3 LIU Block).*
3. *If the Jitter Attenuator block has been configured to operate in the Receive Direction, then the In_CLK signal (as depicted in **Figure 69**, above) will be a buffered version of the Recovered clock signal (from the Receive DS3/E3 LIU Block).*

Now that we have briefly described how the Jitter Attenuator block functions, we can now go into more details on how the individual functional blocks (within the Jitter Attenuator block) function.

4.2.6.2.1.1 The Jitter Attenuator PLL

The purpose of the Jitter Attenuator PLL block is to (1) receive and lock onto an input jittery clock signal, and (2) to regenerate a clock signal, of the exact same frequency, but with considerably less jitter. In performing this task, the Jitter Attenuator PLL block is said to be attenuating jitter. A very simplistic illustration of what the Jitter Attenuator PLL has been designed to accomplish is presented below in **Figure 70**.

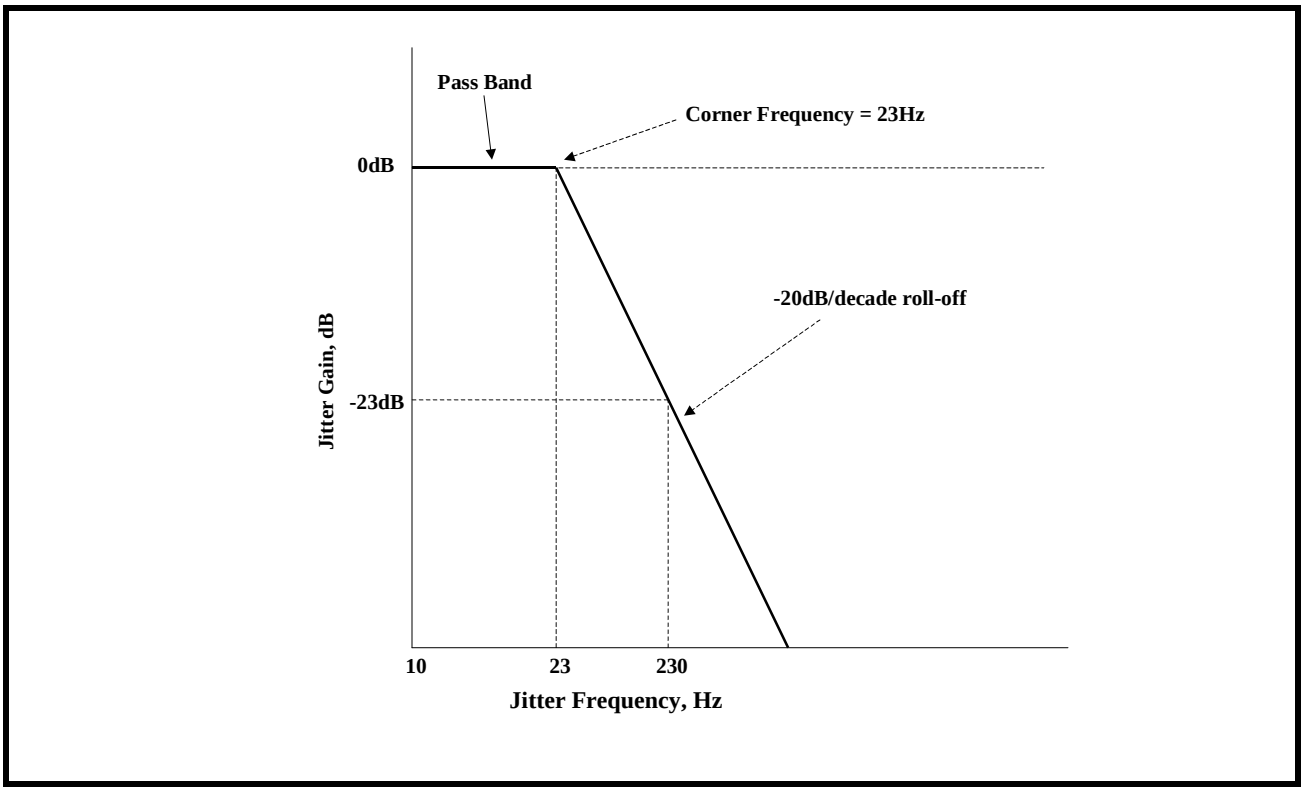
FIGURE 70. A SIMPLISTIC ILLUSTRATION OF THE ROLE/FUNCTION OF THE JITTER ATTENUATOR PLL BLOCK WITHIN THE XRT71D03 DEVICE



The Jitter Attenuator PLL receives the input jittery clock signal via the In_CLK input pin. It outputs the smoothed (e.g., clock signal with reduced jitter) clock signal via the Out_CLK output pin.

The Jitter Attenuator PLL accomplishes this Jitter Attenuation task because it is a very narrow-band PLL. The corner (e.g., -3dB) frequency of the Loop-Filter (within the Jitter Attenuator PLL) is about 23Hz. A simple illustration of the Jitter Gain (or Attenuation) Transfer Characteristics of the Jitter Attenuator PLL (within the XRT79L71) is illustrated below in **Figure 71**.

FIGURE 71. A SIMPLE ILLUSTRATION OF THE JITTER TRANSFER CHARACTERISTICS OF EACH JITTER ATTENUATOR PLL (WITHIN THE XRT79L71)



All of this means that if the Jitter Attenuator PLL receives a clock signal, that contains jitter with a frequency of about 23Hz, the Jitter Attenuator PLL will reduce the amplitude of this 23Hz jitter component by about 3dB or 50%. The Pass-Band of the Loop-filter (within the Jitter Attenuator PLL) is any frequency below 23Hz. This means that the Jitter Attenuator PLL (within the XRT79L71) will not provide much attenuation on any jitter (within the input In_CLK clock signal) that is of frequencies less than 23Hz. As a consequence, much of the low-frequency jitter that appears at the In_CLK input, will also appear at the Out_CLK output pins. The Jitter Attenuator PLL will provide more than 3dB of jitter attenuation for jitter with frequencies greater than 23Hz. The Jitter Transfer Characteristics (of the Jitter Attenuator PLL, within the XRT79L71) is such that for frequencies greater than 23Hz, it imposes a -20dB/decade roll-off in the Gain versus Frequency curve (as presented in **Figure 71**). Therefore, if the Jitter Attenuator PLL block receives a signal (via the In_CLK_n input pin) that contains jitter which is of a frequency of 230Hz, then the Jitter Attenuation PLL will reduce the amplitude of this jitter by 23dB or by 95%. In general, the higher the frequency of the jitter (within the In_CLK_n input signal), the greater the jitter amplitude will be attenuated.

DISABLING THE JITTER ATTENUATOR PLL

The XRT79L71 permits the user to disable the Jitter Attenuator PLL. If the Jitter Attenuator PLL is disabled, then the Jitter Attenuator block will perform no jitter attenuation on the In_CLK input signal. More specifically, the Jitter Attenuator FIFO Block will also be by-passed, and the signal path (through the Jitter Attenuator block) will proceed to emulate the behavior of three wires. In this mode, the In_CLK input signal (within the Jitter Attenuator block) will essentially be shorted to the Out_CLK output signal. Further, the In_POS input signal will

essentially be shorted to the Out_POS output signal and the In_NEG input signal will essentially be shorted to the Out_NEG. Data residing on the In_POS and In_NEG input pins will not be clocked into the Jitter Attenuator FIFO via the In_CLK input signal. Further, data will not be clocked out (via the Out_POS and In_NEG_n output pins) upon the Out_CLK_n output signal. This data will simply propagate through the Jitter Attenuator block, just as if it were three wires.

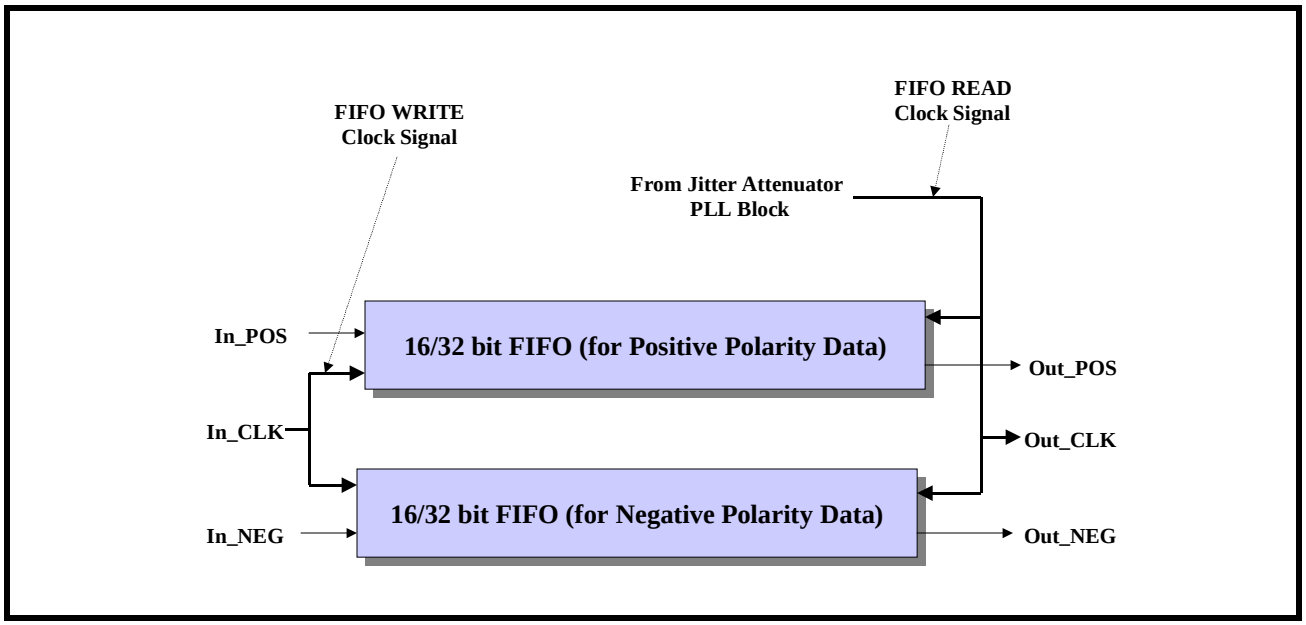
The user can enable or disable the Jitter Attenuator PLL, within the XRT79L71 by executing the procedure presented in Section 4.2.6.2.2.

4.2.6.2.1.2 The Jitter Attenuator FIFO

The Jitter Attenuator Block contains a 2-Channel FIFO. The purpose of this FIFO is to permit the Jitter Attenuator to absorb any instantaneous frequency differences between the In_CLK input signal and the Out_CLK output signal and to mitigate the occurrences of bit-errors.

The Jitter Attenuator FIFO actually consists of 2 FIFO channels, in the sense that one FIFO channel is dedicated for the Positive-Polarity Data (e.g., the In_POS to Out_POS path) and the other FIFO channel is dedicated for the Negative-Polarity Data (e.g., the In_NEG to Out_NEG path). The Physical Architecture of the 2-Channel Jitter Attenuator FIFO is presented below in**Figure 72**.

FIGURE 72. ILLUSTRATION OF THE PHYSICAL ARCHITECTURE OF 2-CHANNEL JITTER ATTENUATOR FIFO ARCHITECTURE WITHIN THE JITTER ATTENUATOR BLOCK



NOTE: *The Logical Architecture of the Jitter Attenuator FIFO is presented in the next section.*

SELECTING THE FIFO SIZE

The user can configure the Jitter Attenuator FIFO to operate with a depth of either 16 or 32 bits. A description of how to configure the FIFO to operate with a 16 or 32 depth and its operation, is presented below in Section 4.2.6.2.2Operating with a Jitter Attenuator FIFO Depth of 16 bits

If the Jitter Attenuator FIFO is configured to operate with a depth of 16 bits, then the following is true.

a. When the XRT79L71 first powers up, or experiences a Hardware RESET, then the location of the FIFO_READ and FIFO_WRITE pointers will be 8 bits (or one-half the FIFO Size) apart from each other.

b. As a consequence, data, which is applied to the Jitter Attenuator block (via the In_POS_n and In_NEG_n input pins) will be written into the FIFO (into a location determined by the FIFO_WRITE pointer). This same data will be read out of the FIFO approximately 8 bit periods later once the

FIFO_READ pointer has incremented around to this particular position within the FIFO. Hence, for 16-bit mode operation, the Jitter Attenuator FIFO imposes a nominal latency of 8 bit periods.

Configuring the Jitter Attenuator FIFO Depth to 16 bits:

The user can configure the Jitter Attenuator FIFO to operate with a depth of 16 bits by executing the procedure presented in Section 4.2.6.2.2.

Operating with a Jitter Attenuator FIFO depth of 32 bits.

If the Jitter Attenuator FIFO is configured to operate with a depth of 32 bits, then the following is true.

a. When the XRT79L71 first powers up, or experiences a Hardware RESET, then the location of the FIFO READ and FIFO WRITE pointers will be 16 bits (or one-half the FIFO size) apart from each other.

b. As a consequence, data, which is applied to the Jitter Attenuator block (via the In_POS and In_NEG input pins) will be written into the FIFO (into a location determined by the FIFO WRITE pointer). This same data will be read out of the FIFO approximately 16 bit periods later once the FIFO_READ pointer has incremented around to this particular position within the FIFO. Hence, for 32-bit mode operation, the Jitter Attenuator FIFO imposes a nominal latency of 16 bit periods.

Configuring the Jitter Attenuator FIFO Depth of 32 bits:

The user can configure the Jitter Attenuator FIFO to operate with a depth of 32 bits by executing the procedure presented in Section 4.2.6.2.2.

WRITING DATA INTO THE FIFO

The Jitter Attenuator block accepts data via the In_POS and In_NEG input pins. Data on these pins are sampled and written into the 2-Channel Jitter Attenuator FIFO upon the appropriate edge of the In_CLK input signal. The exact location (within the FIFO) that this data is written to, depends upon the location of the FIFO_WRITE pointer. Once a given sample of data has been loaded into the FIFO (at the location specified by the FIFO_WRITE pointer), the location of the FIFO_WRITE pointer will then be incremented to the next position within the FIFO, and the process repeats during the next period of In_CLK. It is appropriate to think of the Jitter Attenuator FIFO as a circular-buffer, in the sense that once the FIFO_WRITE pointer has reached the last bit, within the FIFO, it will wrap-around back to the first bit, within the FIFO. This concept is discussed in greater detail, in the FIFO Limit Alarm section of this data sheet.

It is important to note that the writing of data into the FIFO, and the incrementing of the FIFO_WRITE pointer is synchronized to the In_CLK (jittery clock) input signal.

READING DATA FROM THE FIFO

The Jitter Attenuator block, within the XRT79L71, reads out data from the Jitter Attenuator FIFO, and outputs this data via the Out_POS and Out_NEG output pins. Data is output via these pins, upon the appropriate edge of the Out_CLK output signal. The exact location (within the FIFO) that this data is read from, depends upon the location of the FIFO_READ pointer. Once a given sample of data has been extracted from the FIFO (at the location specified by the FIFO_READ pointer), the FIFO_READ pointer will then be incremented to the next position within the FIFO, and the process repeats, during the next period of Out_CLK. As in the case of the FIFO_WRITE pointer, it is appropriate to think of the Jitter Attenuator FIFO as a circular buffer in the sense that once FIFO_READ pointer has reached the last bit, within the FIFO, it will wrap-around back to the first bit, within the FIFO. This concept is discussed in greater detail, in the FIFO Limit Alarm section of this data sheet.

It is important to note that the reading of data from the FIFO, and the incrementing of the FIFO_READ pointer is synchronized to the Out_CLK (smoothed clock) output signal.

THE FIFO LIMIT ALARM

Whenever the XRT79L71 is initially powered-up, or experiences a Hardware RESET, the locations of the FIFO_WRITE and FIFO_READ pointers (within the Jitter Attenuator block) will initially be 8 bit positions (if the Jitter Attenuator FIFO depth is configured to be 16-bits) or 16 bit positions apart from each other (if the Jitter Attenuator FIFO depth is configured to be 32-bits).

In the previous section, we mentioned that the FIFO_WRITE pointer is incremented (within the Jitter Attenuator FIFO) with each period of the In_CLK (e.g., the jittery clock) input signal. Further, we also indicated that the FIFO_READ pointer is incremented (within the Jitter Attenuator FIFO) with each period of the Out_CLK (e.g., the smoothed clock) output signal.

Additionally, we also mentioned that the Jitter Attenuator PLL block accepts a jittery clock signal, via the In_CLK input signal and from this input clock signal, it synthesizes another clock signal (of the exact same frequency, but with considerably less jitter). This synthesized clock signal is output via the Out_CLK output pin.

Therefore, on the average, the frequencies of the In_CLK input signal and the Out_CLK output signal are identical. As a consequence, on the average, the FIFO_READ and FIFO_WRITE pointers will each increment (throughout the Jitter Attenuator FIFO) at the same rate and the two pointers will typically maintain a constant distance from each other. **Figure 73** presents an illustration of the Logical Architecture of the Jitter Attenuator FIFO, within the Jitter Attenuator block. This particular figure illustrates the relationship between the FIFO_WRITE and FIFO_READ pointers.

FIGURE 73. ILLUSTRATION OF THE JITTER ATTENUATOR FIFO AND THE FIFO_WRITE AND FIFO_READ POINTERS.

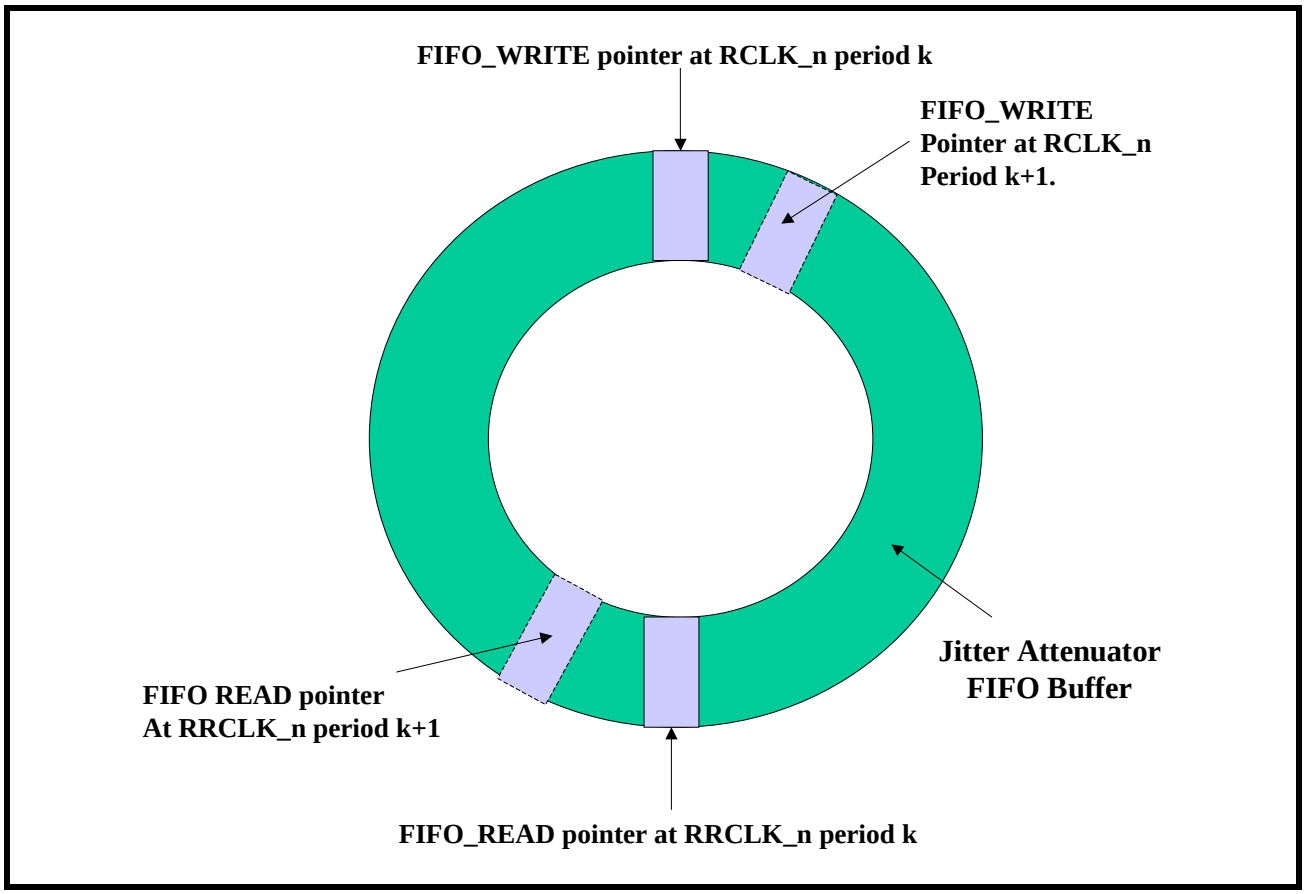


Figure 73 presents an illustration of the behavior of the FIFO_READ and the FIFO_WRITE pointers within the Jitter Attenuator FIFO. For the sake of discussion, this figure indicates that the initial position (e.g., In_CLK period k) of the FIFO_WRITE pointer is at the 12 o'clock position. This figure also indicates that during the very next In_CLK period, that the FIFO_WRITE pointer will have moved (in a clockwise direction) to the next position (which is labeled FIFO_WRITE pointer at In_CLK period k + 1). If the FIFO depth is configured to be 16 bits, then in 16 In_CLK cycles, after power-up or reset, the FIFO_WRITE pointer will have revolved around the circular buffer and will have incremented its location right back to its initial (or the 12 o'clock) position as is shown in Figure 13. Similarly, if the FIFO depth is configured to be 32 bits, then in 32 In_CLK cycles, after

power-up or reset, the FIFO_WRITE pointer will have revolved around the circular buffer and will incremented its location back to its initial (e.g., 12 o'clock) position.

Figure 73 also indicates that the initial position of the FIFO_READ pointer is at the 6 o'clock position. Further, Figure 73 indicates that during the very next Out_CLK period the FIFO_READ pointer will have moved (in a clockwise direction) to the position labeled FIFO_READ pointer at Out_CLK period k + 1.

If the FIFO Depth is configured to be 16 bits, then in 16 Out_CLK cycles, after power-up or reset, the FIFO_READ pointer will have revolved around the circular buffer and have incremented its position right back to the 6 o'clock position) as is shown in Figure 73. Similarly, if the FIFO depth is configured to be 32 bits, then in 32 Out_CLK cycles, after power-up or reset, the FIFO_READ pointer will have revolved around the circular buffer and will have incremented its location back to its initial (e.g., 6 o'clock) position.

Recall that the Out_CLK (smoothed) output clock signal is derived from a PLL that has a very narrow bandwidth. Therefore, the Jitter Attenuator PLL will not be very responsive to instantaneous phase or frequency deviations that occur within the In_CLK (jittery) input clock signal. As the Jitter Attenuator PLL experiences large phase variations (via the In_CLK input pin) it will typically not pass along these large phase changes and variations to the Out_CLK output signal. This phenomenon will result in some modulation in the distance between the FIFO_WRITE and FIFO_READ pointers. If the modulation in the distance (between the FIFO_WRITE and FIFO_READ pointers) is such that the distance between these two pointers approach 0 bits, then bit-errors will occur. This modulation in the distance between the FIFO_WRITE and FIFO_READ pointers and its effect on the integrity of the data is described in some detail below.

The FIFO Under-run Condition

If the jitter within the In_CLK input clock signal is such that, for a very short period of time, the instantaneous frequency of In_CLK deviates, in the negative direction, (e.g., exhibits a lower frequency from that of the Out_CLK output clock signal) then the following events will happen.

- The Jitter Attenuator PLL will continue to generate the Out_CLK output clock signal at virtually a constant rate (e.g., with very little change in frequency). As a consequence, the FIFO_READ pointer will continue to increment and revolve about the circular buffer at largely a fixed rate.
- Because the In_CLK input clock signal is, for a very short period, of a lower frequency, the FIFO_WRITE pointer will now be incremented, around the circular buffer, at a lower rate than that of the FIFO_READ pointer.

If the above-mentioned condition were to exist long enough, the FIFO_READ pointer will eventually revolve around the circular buffer, and catch-up with the FIFO_WRITE pointer. If this happens, then a FIFO Under-run condition will be said to have occurred. In a FIFO Under-run Condition, all new data that has been written into the Jitter Attenuator FIFO has been read out and depleted. At this point, the Jitter Attenuator block is now reading out data which has already been read out, at least once. This phenomenon will cause the XRT79L71 to transmit erroneous data to the remote terminal equipment.

The FIFO Overflow Condition

If the jitter within the In_CLK input clock signal is such that, for a very short period of time, the instantaneous frequency of In_CLK deviates, in the positive direction, (e.g., exhibits a much higher frequency from that of the Out_CLK output clock signal) then the following events will happen.

- The Jitter Attenuator PLL will continue to generate the Out_CLK at virtually a constant rate (e.g., with very little change in frequency). As a consequence, the FIFO_READ pointer will continue to increment and revolve about the circular buffer at largely a fixed rate.
- Because the In_CLK input clock signal is, for a very short period, of a higher frequency, the FIFO_WRITE pointer will now be incremented, around the circular buffer, at a higher rate than that of the FIFO_READ pointer.

If the above-mentioned condition were to exist long enough, the FIFO_WRITE pointer will eventually revolve around the circular buffer, and catch-up with the FIFO_READ pointer. If this happens, then an Overflow condition will be said to have occurred. In an Overflow condition, new data has been written into locations

within the Jitter Attenuator FIFO and overwriting data that the XRT79L71 has yet had an opportunity to read it out. This phenomenon will cause the XRT79L71 to drop or lose data.

Please see Section 4.2.6.2.4 for information on how the Jitter Attenuator block responds to a FIFO Overflow or Underflow condition.

The next few sections describe how to configure and use the Jitter Attenuator block

4.2.6.2.2 Enabling the Jitter Attenuator Block and selecting the Jitter Attenuator FIFO Size

The user will both enable the Jitter Attenuator block and select (and implement) a FIFO size by writing the appropriate values into Bits 0 (Jitter Attenuator PLL/FIFO Operation Mode - Bit 0) and 2 (Jitter Attenuator PLL/ FIFO Operation Mode - Bit 2), as depicted below.

Jitter Attenuator Control Register (Address = 0x1307)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			DFL	Jitter Attenuator FIFO Pointer RESET	Jitter Attenuator PLL/FIFO Operating Mode - Bit 1	Jitter Attenuator in Transmit Path	Jitter Attenuator PLL/FIFO Operating Mode - Bit 0
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	X	0	X

Table 24 presents the relationship between the states of Bits 2 and 0 (within the Jitter Attenuator Control Register) and the (1) Enable/Disable State of the Jitter Attenuator, and (2) the Size of the Jitter Attenuator FIFO.

TABLE 24: THE RELATIONSHIP BETWEEN THE STATES OF BITS 2 AND 0 (WITHIN THE JITTER ATTENUATOR CONTROL REGISTER) AND THE (1) ENABLE/DISABLE STATE OF THE JITTER ATTENUATOR, AND (2) THE SIZE OF THE JITTER ATTENUATOR FIFO

BIT 2	BIT 0	JITTER ATTENUATOR STATE	FIFO SIZE
0	0	Enabled	16 bits
0	1	Disabled	N/A
1	0	Enabled	32 bits
1	1	Disabled	N/A

If the Jitter Attenuator is enabled, then it will perform its Jitter Attenuation processing on any DS3 or E3 signal that passes through it. However, if the Jitter Attenuator is disabled, then the Narrow-band PLL (within the Jitter Attenuator) will be by-passed. In this case, the Jitter Attenuator block will operate in a transparent mode (e.g., the DS3 or E3 signal will pass through the Jitter Attenuator block with no jitter attenuation processing performed on it, at all).

4.2.6.2.3 Configuring the Jitter Attenuator to operate in the Transmit or Receive Path

The user can configure the Jitter Attenuator block to operate in either the Transmit or Receive Direction, by writing the appropriate value into Bit 1 (Jitter Attenuator in Transmit Path), within the Jitter Attenuator Control Register as depicted below.

Jitter Attenuator Control Register (Address = 0x1307)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			DFL	Jitter Attenuator FIFO Pointer RESET	Jitter Attenuator PLL/FIFO Operating Mode - Bit 1	Jitter Attenuator in Transmit Path	Jitter Attenuator PLL/FIFO Operating Mode - Bit 0
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	X	X	X

Setting this bit-field to "0" configures the Jitter Attenuator to operate in the Receive Path. Conversely, setting this bit-field to "1" configures the Jitter Attenuator to operate in the Transmit Path.

4.2.6.2.4 Alarm Conditiions occurring within the Jitter Attenuator Block

As mentioned in Section 4.2.6.2.1.2, the FIFO Under-run and Overflow conditions occur if the distance between the FIFO_READ and FIFO_WRITE pointers reach 0 bits. In either of these conditions an error-condition is said to have occurred. The XRT79L71 contains some circuitry to alert the system that the distance between the FIFO_WRITE and FIFO_READ pointers has fallen to at least two (2) bit-positions. If this condition were to occur, then Jitter Attenuator FIFO is on the verge of experiencing either a FIFO Under-run or Overflow condition. In response to this condition, the XRT79L71 will notify the Line Card circuitry of this phenomenon by declaring a FIFO Limit Alarm condition. The XRT79L71 will indicate that it is declaring a FIFO Limit Alarm condition by doing all of the following.

a. It will set Bit 3 (FIFO Limit Alarm Declared), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

b. It will also generate the Change of FL Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "Low")

b. Setting Bit 3 (Change of FL Condition Interrupt Status) within the LIU Interrupt Status Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Clearing the FIFO Limit Defect Condition

Similarly, once the Jitter Attenuator Block has recovered from the FIFO Limit Alarm condition, then it indicate that it is clearing the FIFO Limit defect condition by doing all of the following.

a. It will set Bit 3 (FIFO Limit Alarm Declared), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

b. It will also generate the Change of FL Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "Low")

b. Setting Bit 3 (Change of FL Condition Interrupt Status) within the LIU Interrupt Status Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

4.2.6.2.5 The Jitter Transfer Characteristic for DS3 Applications

This section presents the Jitter Transfer Characteristics of the XRT79L71, if the Jitter Attenuator PLL has been enabled. As we present the Jitter Transfer Characteristics of the XRT79L71, we will also present some of the various Jitter Transfer Characteristic requirements that the XRT79L71 complies with.

4.2.6.3 The Transmit Control Block

The purpose of the Transmit Control Block is to accept any of the following signals from the on-chip Registers, and route the state of these signals to either the Transmit Pulse Shaping Block and the Line Driver block.

- TxLEV - Transmit Line Build-Out Control
- TAOS - Transmit All OneS Pattern
- TxON - Transmit Driver ON/OFF

The role that TxLEV plays in the Transmit Pulse Shaping block is addressed in Section 4.2.6.4. Similarly, the role that TxON plays is address in Section 4.2.6.5. However, none of these sections address the TAOS signal. As a consequence, TAOS will be addressed in this section.

TAOS - Transmit All OneS

The user can configure the Transmit DS3/E3 LIU Block transmit an un-framed "All Ones" pattern via the TTIP and TRING output pins. This can be accomplished by setting Bit 2 (TAOS), within the LIU Transmit Control Register, to "1" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Moni-tor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	1	0	0

If the TAOS bit-field is set to "1", then the Transmit DS3/E3 LIU Block will overwrite the contents of the outbound DS3 or E3 data-stream (that it receives from the Transmit DS3/E3 Framer block) with an unframed All Ones pattern.

Conversely, if the TAOS bit-field is set to "0", then the Transmit DS3/E3 LIU Block will NOT overwrite the contents of the outbound DS3 or E3 data-stream (that it receives from the Transmit DS3/E3 Framer block) with an unframed All Ones pattern, as it transmit this DS3 or E3 data out onto the line.

4.2.6.4 The Transmit Pulse Shaping Block

For DS3 Applications, the purpose of the Transmit Pulse Shaping block is to permit the user to either enable or disable the Transmit Line Build-Out circuitry within the Transmit DS3/E3 LIU Block. If the Transmit Line Build-Out circuit is enabled, then it will provide some low-pass filtering of the transmit output pulses within the outbound DS3 line signal, which will result in the transmission of some shaped (e.g., not square-wave) pulses onto the DS3 line. In contrast, if the Transmit Line Build-Out circuit is disabled, then it will NOT provide any low-pass filtering of the transmit output pulses. In this case, the Transmit DS3/E3 LIU Block will transmit unshaped (somewhat square-shaped) pulses onto the line.

For DS3 Applications, the Transmit Line Build-Out circuit is used to shape the transmit output pulses within the DS3 line signal such that these pulses with comply with the Isolated Pulse Template requirements per Bellcore GR-499-CORE, when these pulses are measured at the DSX-3 location.

Guidelines for using the Transmit Line Build-Out Circuit

To configure the Transmit DS3/E3 LIU Block within the XRT79L71 such that it will comply with the Isolated Pulse Template requirements per Bellcore GR-499-CORE, the user should use the following guidelines for enabling/disabling the Transmit Line Build-Out circuit.

If the distance between the Transmit Output of the XRT79L71 and the DSX-3 location is less than 225 feet, then the user is advised to enable the Transmit Line Build-Out circuit. In this case, the user can enable the Transmit Line Build-Out circuit by setting Bit 0 (TxLEV) within the LIU Transmit Control Register to "0" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	0	0	0

Figure 74 and Figure 75 each present plots of a captured DS3 pulse versus the Isolated Pulse Template Requirements - per Bellcore GR-499-CORE. Figure 74 presents a Pulse Template Measurement that is taken at 0 feet of cable loss with the TxLEV bit set to "0". Similarly, Figure 75 presents a Pulse Template Measurement that is taken at 225 feet of cable loss with the TxLEV bit set to "0".

FIGURE 74. DS3 PULSE TEMPLATE MEASUREMENT - TAKEN WITH 0 FEET OF CABLE LOSS WITH THE TXLEV BIT SET TO "0"

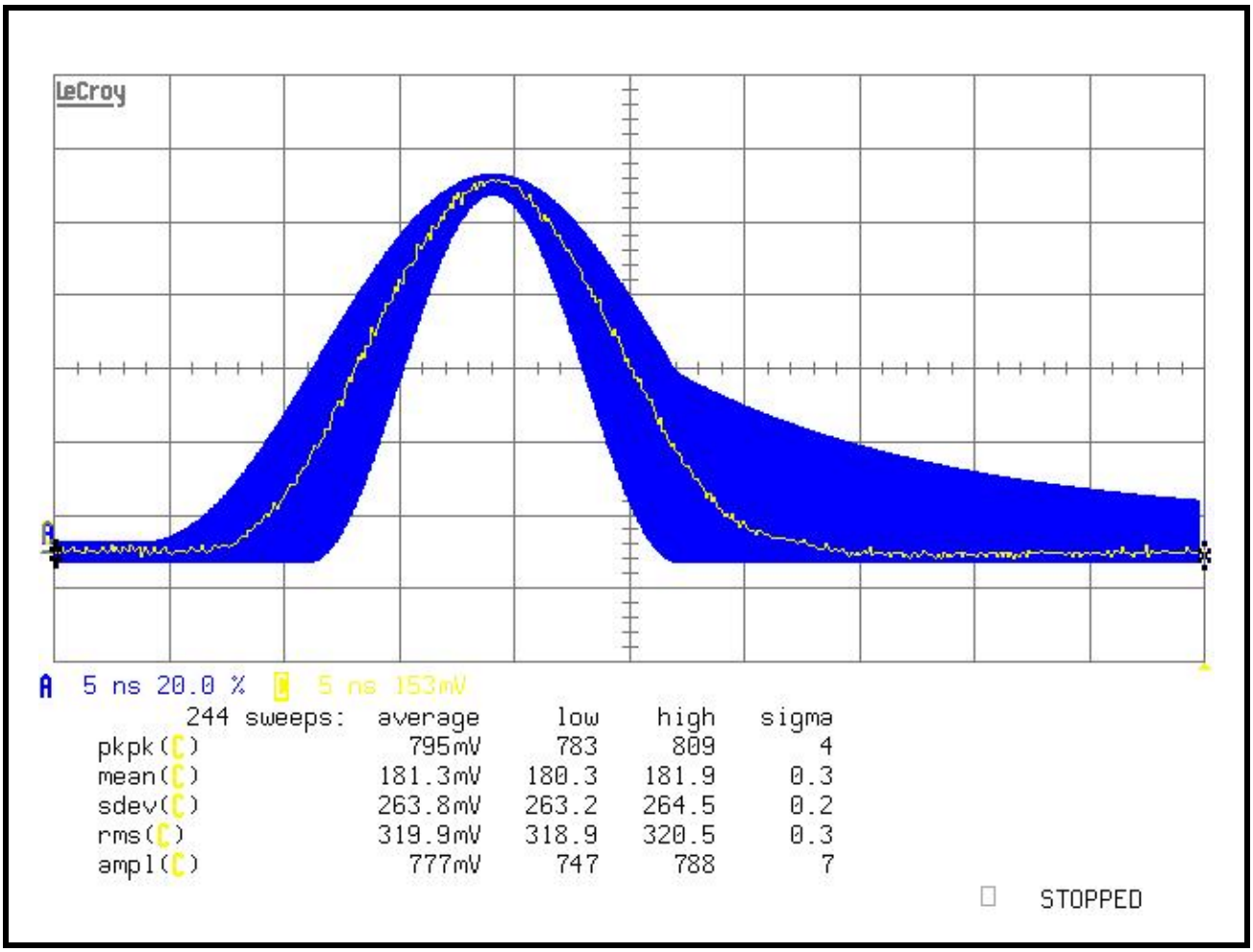
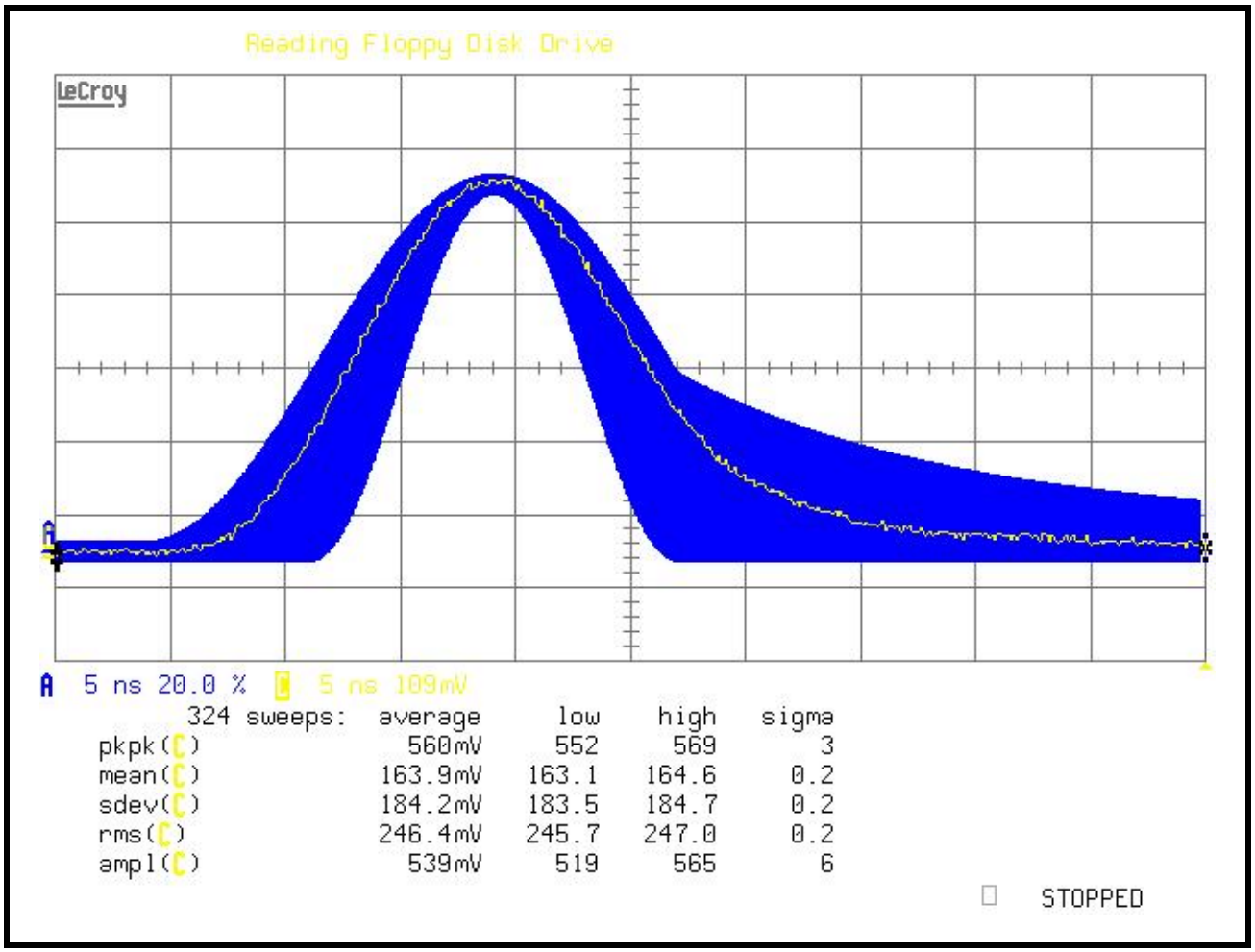


FIGURE 75. DS3 PULSE TEMPLATE MEASUREMENT - TAKEN WITH 225 FEET OF CABLE LOSS WITH THE TXLEV BIT-FIELD SET TO "0"



Conversely, if the distance between the Transmit Output of the XRT79L71 and the DSX-3 location is greater than 225 feet, then the user is advised to disable the Transmit Line Build-Out circuit. In this case, the user can disable the Transmit Line Build-Out circuit by setting Bit 0 (TxLEV) within the LIU Transmit Control Register to "1" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	0	0	1

NOTE: *In this particular setting, the Transmit DS3/E3 LIU Block will generate unshaped (somewhat square-wave) pulses onto the line. The cable loss associated with this long strand of coaxial cable between the Transmit Output of the XRT79L71 and the DSX-3 location will shaped these pulses such that they will comply with the Isolated Pulse Template Requirement for DS3 Applications per Bellcore GR-499-CORE.*

Figure 76 and Figure 77 each presents plots of a captured DS3 pulse versus the Isolated Pulse Template Requirements - per Bellcore GR-499-CORE. Figure 76 presents a Pulse Template Measurements that is taken at 225 feet of cable loss with the TxLEV bit-field set to "1". Similarly, Figure 77 presents a Pulse Template Measurement that is taken at 450 feet of cable loss with the TxLEV bit set to "1".

FIGURE 76. DS3 PULSE TEMPLATE MEASUREMENTS - TAKEN WITH 225 FEET OF CABLE LOSS WITH THE TXLEV BIT-FIELD SET TO "1"

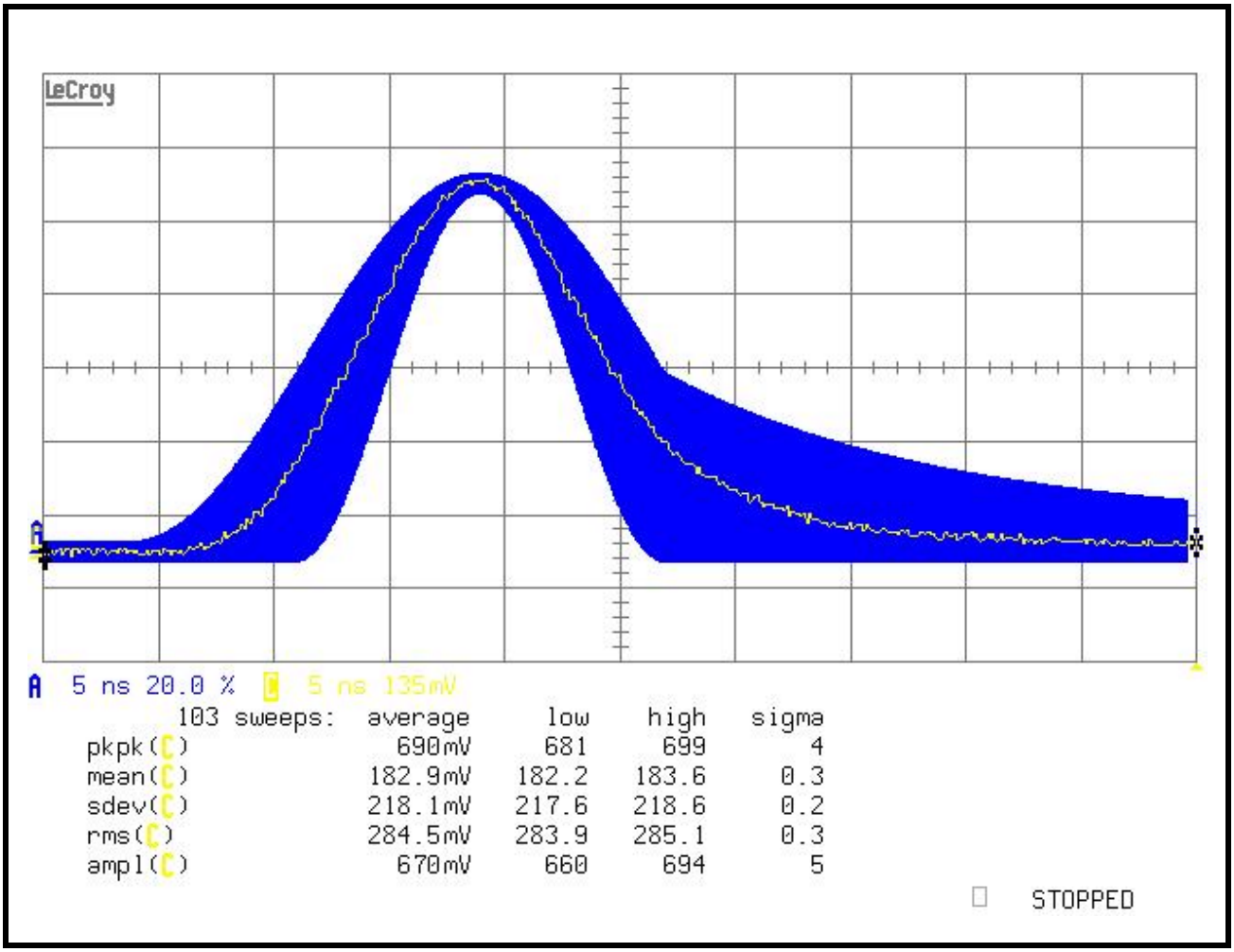
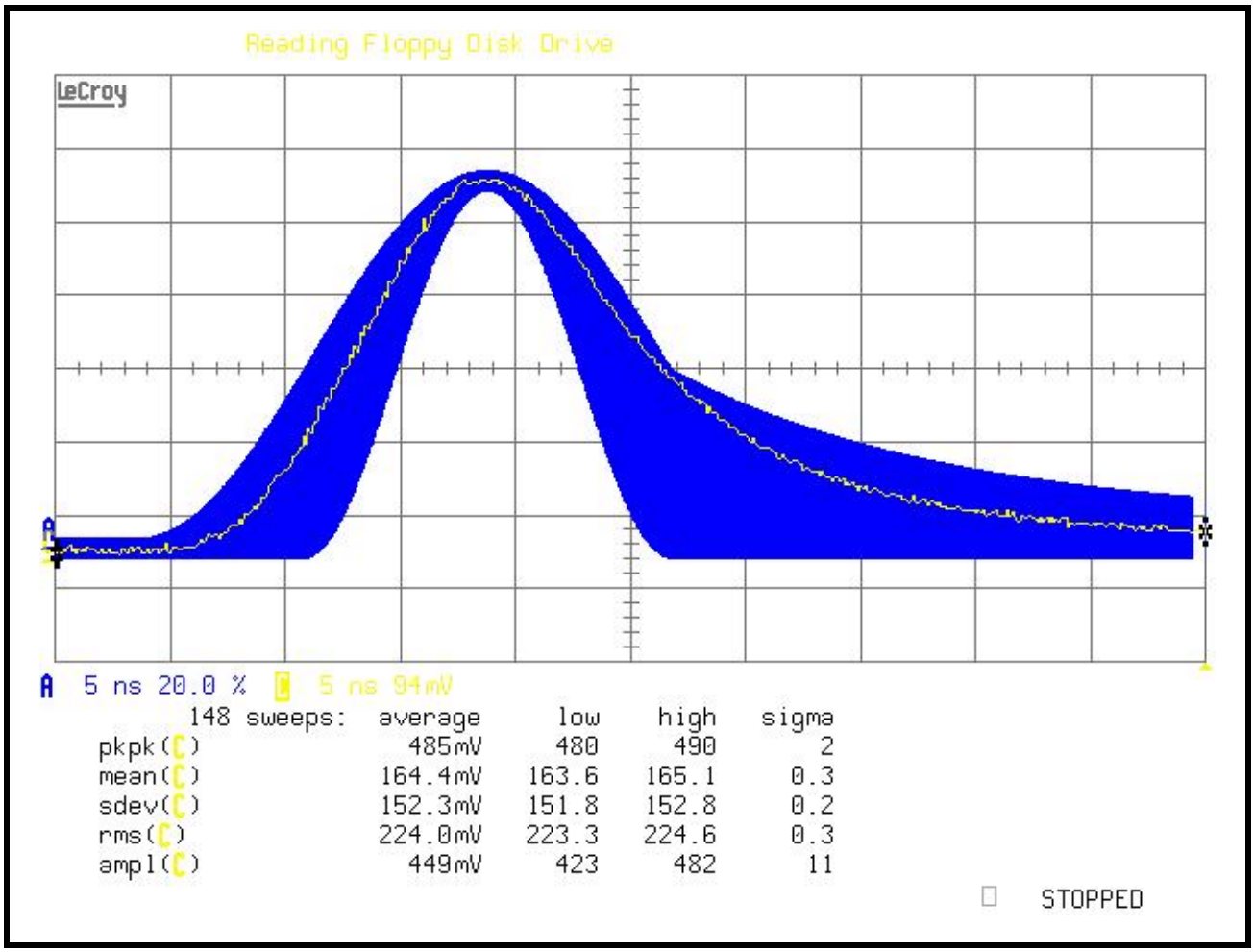


FIGURE 77. DS3 PULSE TEMPLATE MEASUREMENT - TAKEN WITH 450 FEET OF CABLE LOSS WITH THE TXLEV BIT-FIELD SET TO "1"



4.2.6.5 The Transmit Line Driver Block

The Transmit Line Driver block is the very last circuit that an outbound DS3 signal will pass through prior to being output via the TTIP and TRING output pins. The Transmit Line Driver block permits the user to either enable or disable the transmission of a DS3 line signal via the TTIP and TRING output pins.

To enable the Transmit Line Driver block, then the user must make sure that all of the following is true.

1. That Bit 0 (TxON) within the LIU Transmit APS/Redundancy Control Register is set to "1" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

2. That the TxON input pin (Ball R15) is pulled "High".

Conversely, to disable the Transmit Line Driver block, then the user should set Bit 0 (TxON) within the LIU Transmit APS/Redundancy Control Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

NOTES:

1. *If this configuration setting is implemented, then the TTIP and TRING output pins of the XRT79L71 will be tri-stated.*
2. *To control the ON/OFF state of the Transmit Line Driver within the XRT79L71, via software command then the user MUST make sure that the TxON input pin (Ball R15) is pulled to a logic "High".*
3. *To implement the XRT79L71 into a DS3 Redundancy design, then executing a write to this particular register, either enabling or disabling the Transmit Output will be required.*

4.2.6.6 The Transmit Drive Monitor Block

The Transmit Drive Monitor block permits the user to monitor the Transmit Output signal, for continuous bipolar signal activity, and can be used to (perhaps) detect a fault condition, in the Transmit Output line.

Use of the Transmit Drive Monitor block is optional. However, to use this feature, there are two ways that the user can implement this feature.

- Externally, and
- Internally

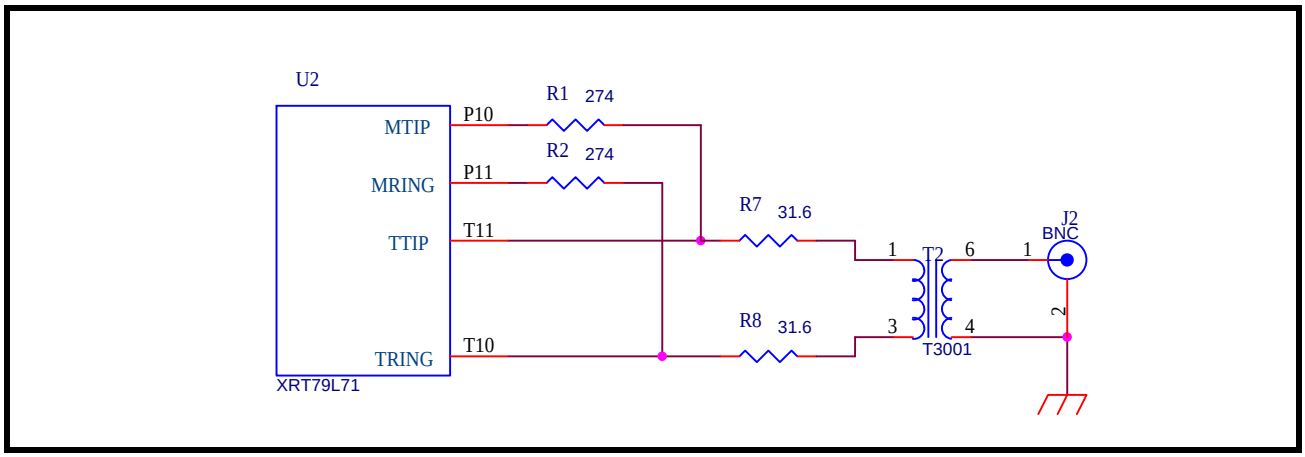
4.2.6.6.1 Implementing the Transmit Drive Monitor via External Means

To implement Transmit Drive Monitoring via External Means, then this means that the Transmit Drive Monitor block will be monitoring (e.g., checking for bipolar activity) within the Transmit Output line signal via the MTIP and MRING input pins. To implement Transmit Drive Monitoring via External Means, the user must execute the following steps.

STEP 1 - Design the Hardware such that (1) the MTIP ball is connected to the TTIP signal through a 274Ω resistor, connected in series, and (2) that the MRING ball is electrically connected to the TRING signal through a 274Ω resistor, connected in series.

These connections are also depicted in the Schematic design below in **Figure 78**,

FIGURE 78. A SCHEMATIC DESIGN, DEPICTING THE REQUIRED CONNECTIONS FOR EXTERNAL TRANSMIT DRIVE MONITORING



STEP 2 - Configure the Transmit Drive Monitor block into the External Mode.

This is accomplished by setting Bit 5 (Internal Transmit Drive Monitor) to "0" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	0	0	0

Once the user has implemented both of these steps, then the Transmit Drive Monitor block will proceed to check the TTIP/TRING lines for bipolar activity via the MTIP and MRING input pins. As long as the Transmit Drive Monitor block detects a regular stream of bipolar pulses via the MTIP/MRING pins, then it will negate the Transmit DMO Condition by continuing to set Bit 0 (Transmit DMO Condition), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

NOTE: *For most applications, the Transmit DMO Condition bit-field being set to "0" is a normal and a desirable condition.*

However, if the Transmit Drive Monitor block ever fails to detect any bipolar pulses (via the MTIP/MRING input pins) for approximately 128 bit periods, then the XRT79L71 will declare the Transmit Drive Monitor defect

condition. Whenever the XRT79L71 declares the Transmit Drive Monitor defect condition, then it will do all of the following.

A. It will set Bit 0 (Transmit DMO Condition), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

B. It will also generate the Change of DMO Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

- **a.** Asserting the Interrupt Request output pin (e.g., by toggling it "Low")
- **b.** Setting Bit 0 (Change of DMO Condition Interrupt Status) within the LIU Interrupt Status Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

Clearing the Transmit DMO Condition

If the Transmit Drive Monitor block (while declaring the Transmit Drive Monitor Defect Condition) detects at least one bipolar pulse via the MTIP/MRING input pins, then the XRT79L71 will clear the Transmit Drive Monitor defect condition. Whenever the XRT79L71 clears the Transmit Drive Monitor defect condition, then it will do all of the following.

A. It will set Bit 0 (Transmit DMO Condition), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

B. It will also generate the Change of DMO Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "Low")

b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the LIU Interrupt Status Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

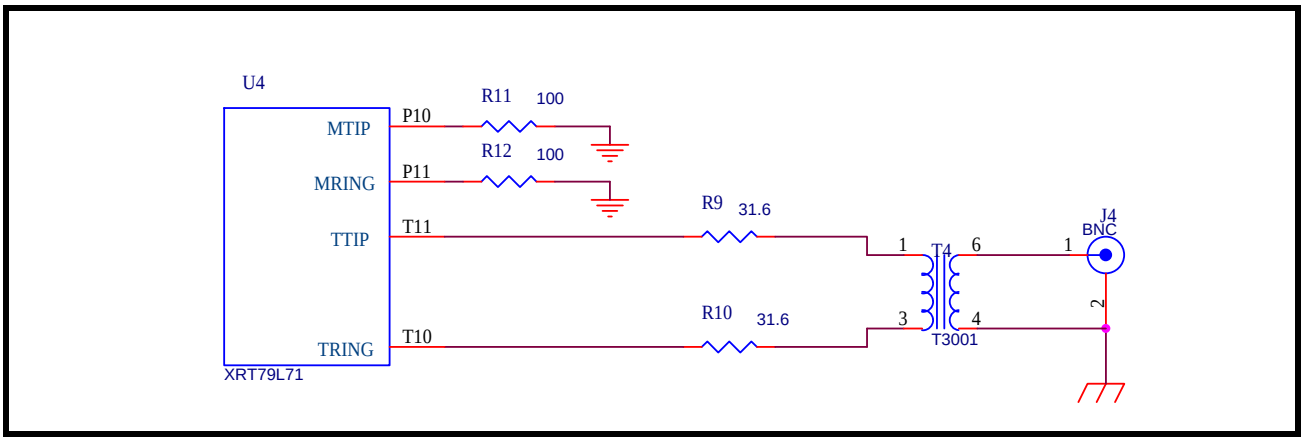
4.2.6.6.2 Implementing the Transmit Drive Monitor via Internal Means

To implement Transmit Drive Monitoring via Internal Means, then this means that the Transmit Drive Monitor block will be monitoring (e.g., checking for bipolar activity) within the Transmit Output line signal, via an internal connection to the TTIP/TRING output pins. To implement Transmit Drive Monitoring via Internal Means, then the user must execute the following steps.

STEP 1 - Design the Hardware such that (1) the MTIP and MRING balls are tied to GND via a 100Ω resistor connected in series.

These connections are also depicted in the Schematic design below in **Figure 79**.

FIGURE 79. A SCHEMATIC DESIGN, DEPICTING THE REQUIRED CONNECTIONS FOR INTERNAL TRANSMIT DRIVE MONITORING



STEP 2 - Configure the Transmit Drive Monitoring block into the Internal Mode.

The user can accomplish this step by setting Bit 5 (Internal Transmit Drive Monitor) to "1" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	1	0	0	1	0	0

NOTE: *To implement Transmit Output Drive Monitoring via Internal Means, then there is NO need for the user to implement the external connections via the MTIP and TTIP and MRING and TRING pins. In this case, the Transmit Drive Monitor circuit will be monitoring (e.g., checking the TTIP/TRING signals for bipolar activity) by direction checking the pads of the TTIP/TRING outputs themselves.*

Once the user has implement this (above-mentioned) step, then the Transmit Drive Monitor block will proceed to internally check the TTIP/TRING lines for bipolar activity. As long as the Transmit Drive Monitor block detects a regular stream of bipolar pulses, then it will negate the Transmit DMO Condition by continuing to set Bit 0 (Transmit DMO Condition), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

NOTE: *For most applications, the Transmit DMO Condition bit-field being set to "0" is a normal and a desirable condition.*

However, if the Transmit Drive Monitor block ever fails to detect any bipolar pulses (via the TTIP/TRING output pads) for approximately 128 bit periods, then the XRT79L71 will declare the Transmit Drive Monitor defect condition. Whenever the XRT79L71 declares the Transmit Drive Monitor defect condition, then it will do all of the following.

A. It will set Bit 0 (Transmit DMO Condition), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

B. It will also generate the Change of DMO Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

- **a.** Asserting the Interrupt Request output pin (e.g., by toggling it "Low")
- **b.** Setting Bit 0 (Change of DMO Condition Interrupt Status) within the LIU Interrupt Status Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	1

Clearing the Transmit DMO Condition

If the Transmit Drive Monitor block (while declaring the Transmit Drive Monitor Defect Condition) detects at least one bipolar pulse via the MTIP/MRING input pins, then the XRT79L71 will clear the Transmit Drive Monitor defect condition. Whenever the XRT79L71 clears the Transmit Drive Monitor defect condition, then it will do all of the following.

A. It will set Bit 0 (Transmit DMON Condition), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

B. It will also generate the Change of DMO Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "Low")

b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the LIU Interrupt Status Register to "1" as depicted below.

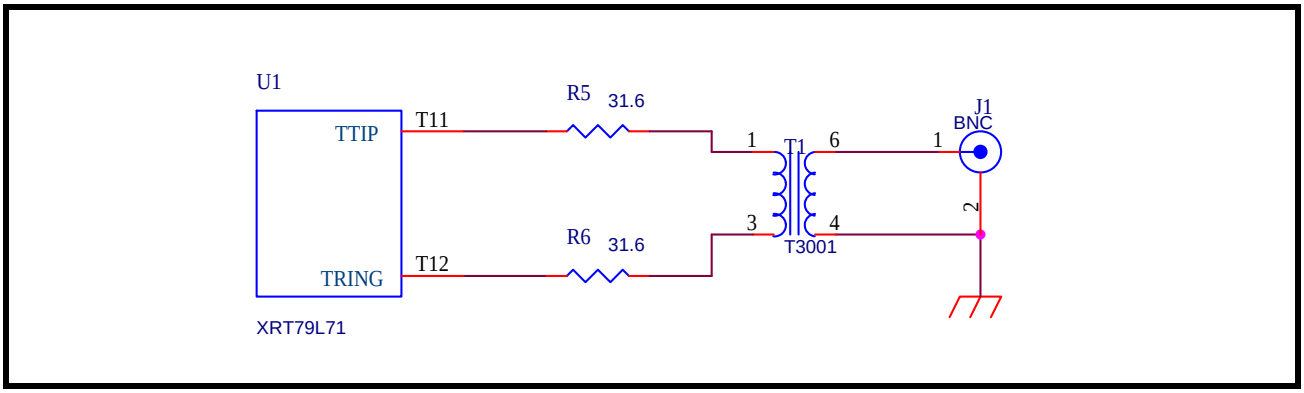
LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

4.2.6.7 Interfacing the Transmit DS3/E3 LIU Block to the Line

Figure 80 presents a schematic design that depicts our recommended approach to interfacing the Transmit DS3/E3 LIU Block to the line.

FIGURE 80. SCHEMATIC DESIGN, DEPICTING HOW TO INTERFACE THE TRANSMIT DS3/E3 LIU BLOCK (OF THE XRT79L71) TO THE LINE



NOTE: *For a more in-depth example on how to interface the XRT79L71 to the outside world, please see Appendix A for the XRT79L71 Evaluation Board Schematic Design.*

4.3 THE RECEIVE DIRECTION

The next several sections will present an in-depth functional description of the all of the blocks that are operating in the Receive Direction, within the XRT79L71, when configured to operate in the Clear-Channel DS3 Framer Mode. Figure 81 presents a functional block diagram of the XRT79L71 Receive Direction circuitry.

FIGURE 81. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE

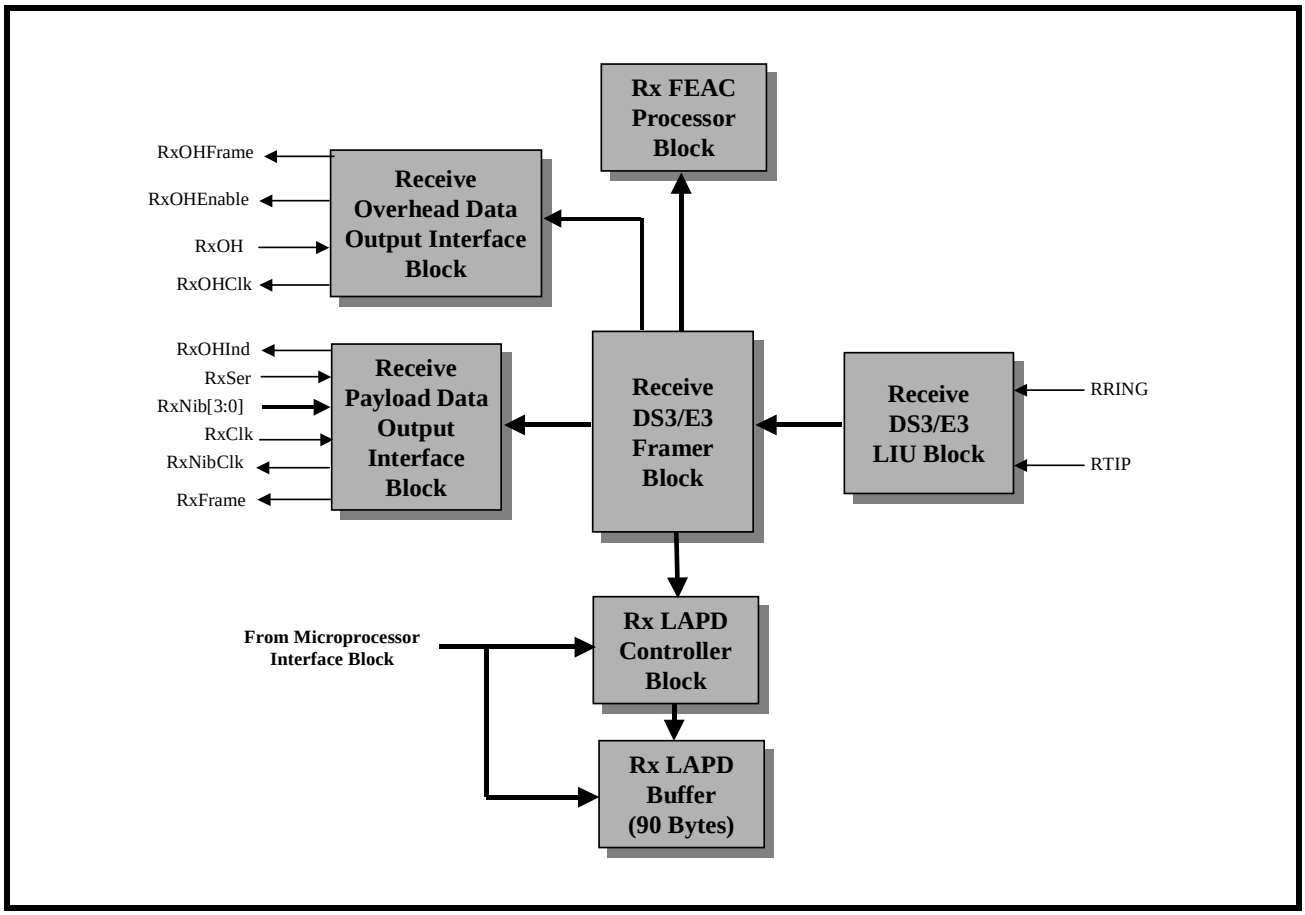


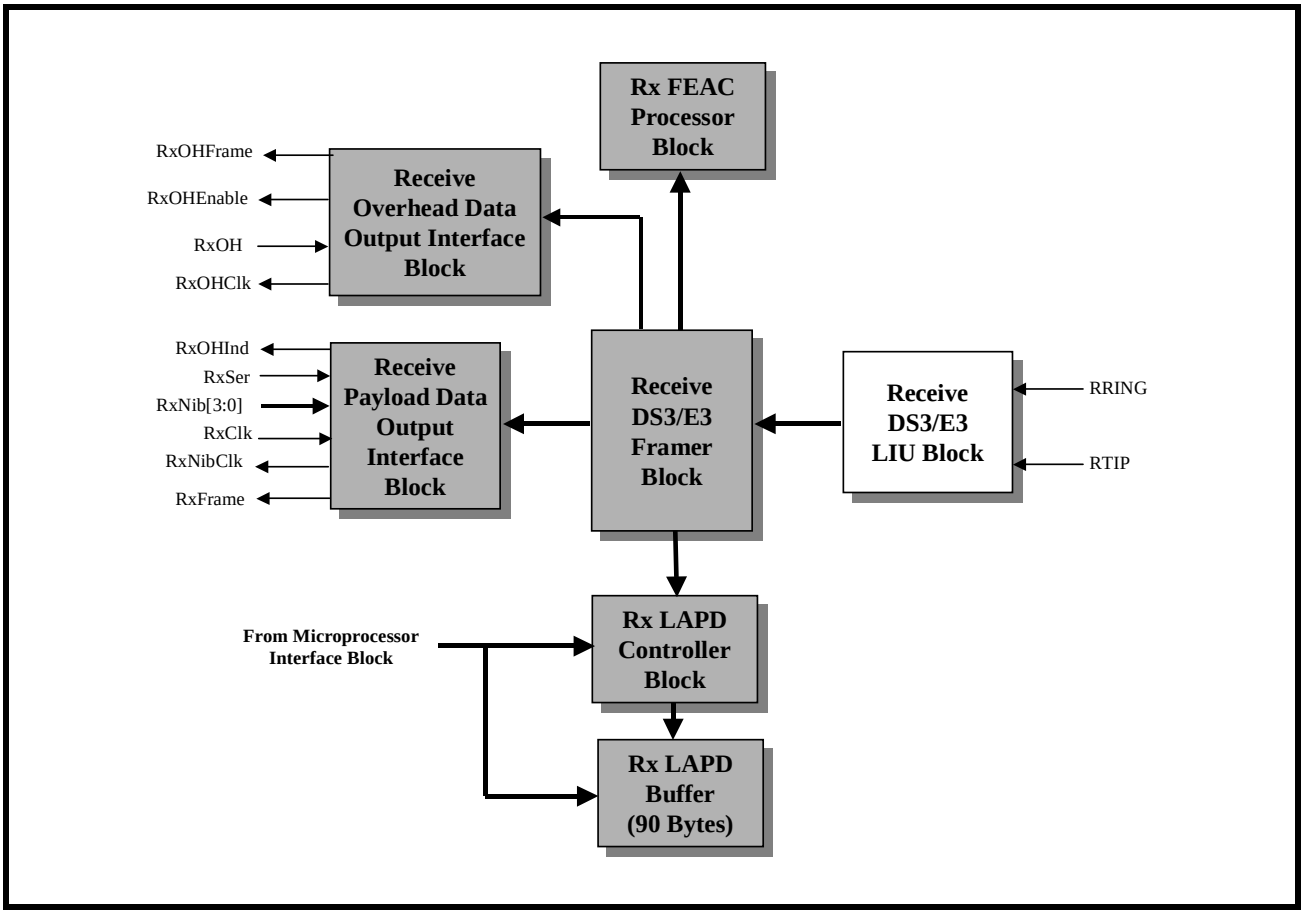
Figure 81 indicates that the Receive Direction circuitry consists of the following functional blocks.

- The Receive DS3 LIU block
- The Receive DS3 Framer block
- The Receive FEAC Processor block
- The Receive LAPD Controller block
- The Receive Payload Data Output Interface block
- The Receive Overhead Data Output Interface block

4.3.1 RECEIVE DS3 LIU BLOCK

The Receive DS3/E3 LIU Block is the very first functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 82 presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive DS3/E3 LIU block highlighted.

FIGURE 82. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE DS3 LIU BLOCK HIGHLIGHTED)



The purpose of the Receive DS3/E3 LIU Block is to accept an DS3/E3 line signal and to perform all of the following operations on this data.

- To be able to receive a distorted DSX-3 signal, that has been attenuated by at least 450 feet of cable loss, along with an additional 6dB of flat or resistive loss in an error-free manner.
- To perform Clock and Data Recovery on this incoming DS3 line signal.
- To declare and clear the LOS (Loss of Signal) defect condition
- To declare and clear the LOL (Loss of Lock) defect condition
- To decode this incoming DS3 line signal from the B3ZS line code, back into a binary data-stream, prior to routing this signal to the Receive DS3/E3 Framer block.
- To be able to comply with the Category I and II Jitter Tolerance Requirements per Bellcore GR-499-CORE

This particular section will describe the functionality and configuration options of the Receive DS3/E3 LIU Block for DS3 applications. The functionality and configuration options of the Receive DS3/E3 LIU Block for E3 applications will be discussed in Section 5.3.1.

Figure 83 presents a more detailed illustration of the Receive DS3/E3 LIU Block within the XRT79L71.

FIGURE 83. ILLUSTRATION OF THE RECEIVE DS3/E3 LIU BLOCK WITHIN THE XRT79L71

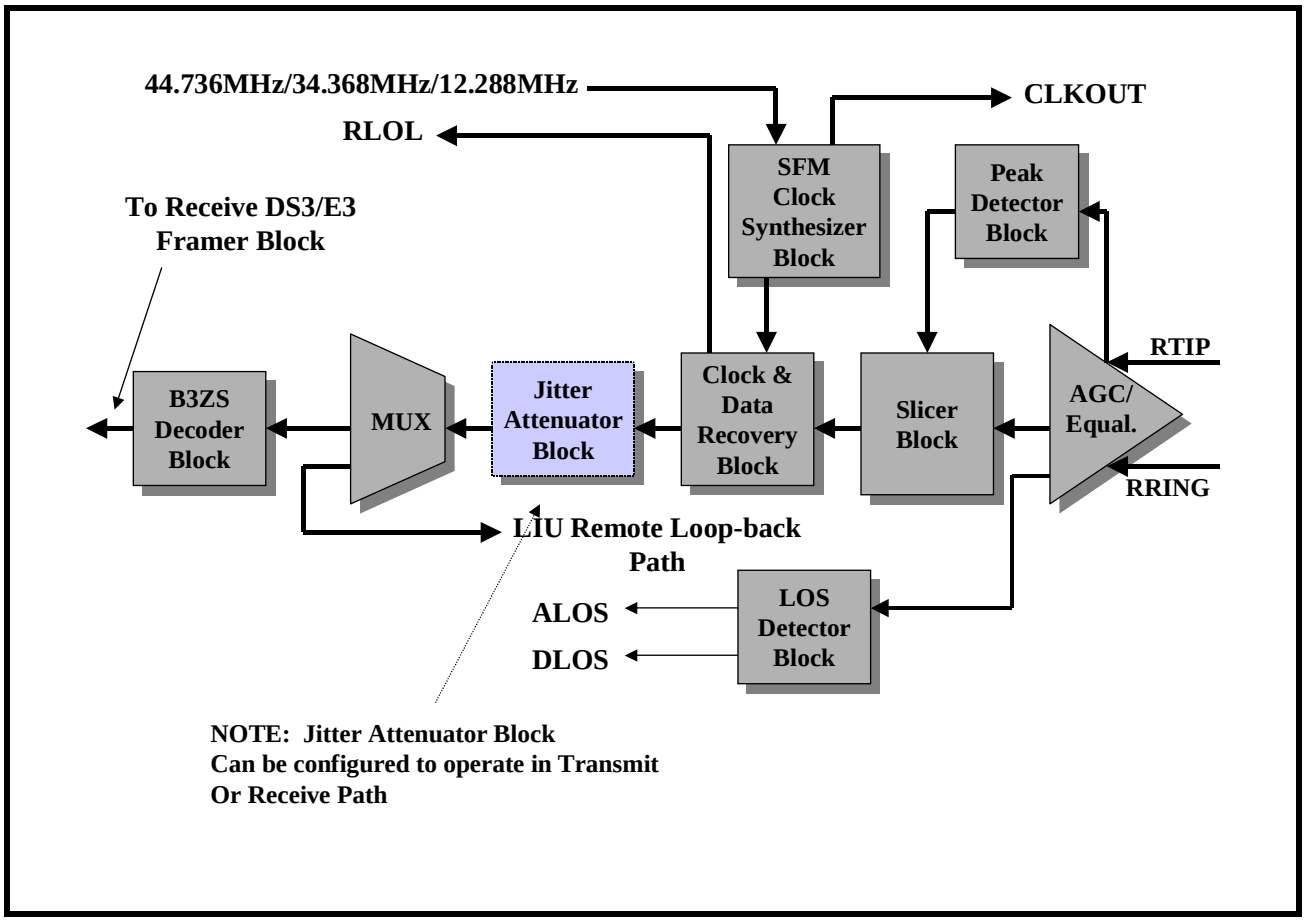


Figure 83 indicates that the Receive DS3/E3 LIU Block consists of the following functional blocks.

- The AGC (Automatic Gain Control) and Equalizer Block
- The Peak Detector Block
- The Slicer Block
- The SFM (Single Frequency Mode) Clock Synthesizer Block
- The Clock and Data Recovery Block
- The Jitter Attenuator Block
- The LOS Detector Block
- The B3ZS Decoder Block

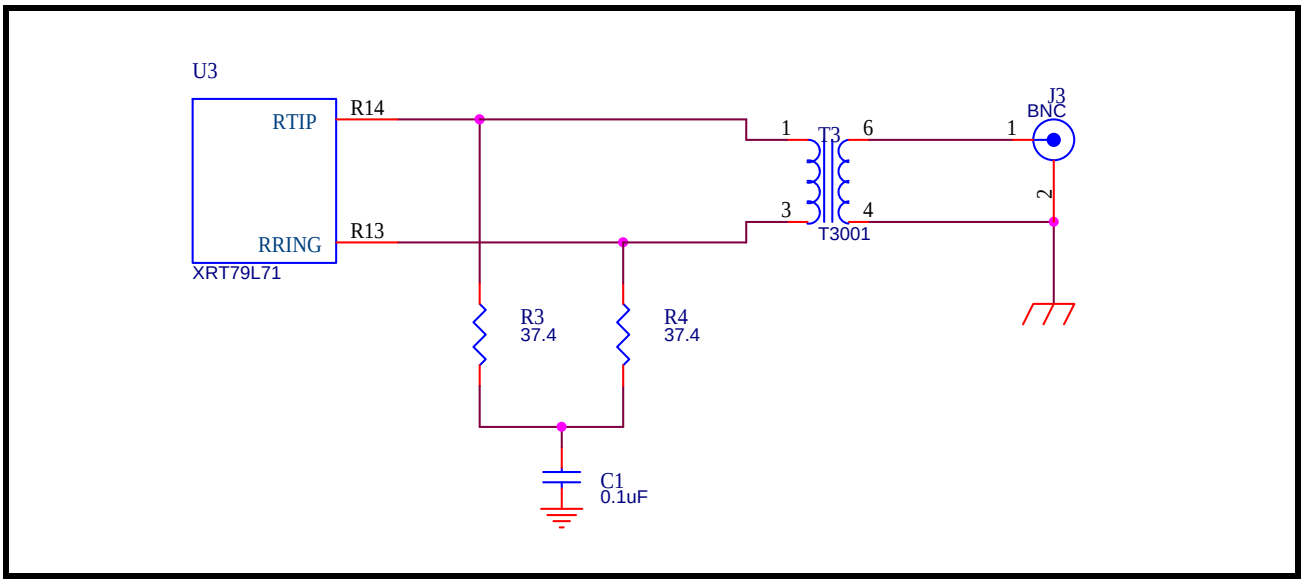
Each of these functional blocks is discussed in considerable detail in the sections below.

4.3.1.1 Interfacing the Receive DS3/E3 LIU Block to the Line

Prior to walking through and discussing each of the various sub-blocks within the Receive DS3/E3 LIU Block, we will first present our recommendation on how to interface the Receive DS3/E3 LIU Block to the line.

Figure 84 presents a schematic design, depicting our recommended approach to interfacing the Receive DS3/E3 LIU Block to the line.

FIGURE 84. SCHEMATIC DESIGN, DEPICTING HOW TO INTERFACE THE RECEIVE DS3/E3 LIU BLOCK (OF THE XRT79L71) TO THE LINE



NOTE: *For a more in-depth example on how to interface the XRT79L71 to the outside world, please see Appendix A for the XRT79L71 Evaluation Board Schematic Design.*

4.3.1.2 The Automatic Gain Control Block

The AGC (or Automatic Gain Control) Block is the very first sub-lock to receive either a DS3 or E3 signal from RTIP/RRING input pins. For DS3 Applications, the purpose of the AGC block is two-fold.

- To compensate for any flat-loss that the incoming line signal may have experienced as it travels from the source (e.g., remote) to the destination (e.g., local) terminal.
- To function as a part of the ALOS (Analog LOS) Detector

Each of these roles/purposes is briefly discussed below.

4.3.1.2.1 Compensating for Flat Loss

As the name of this sub-block implies, the AGC (Automatic Gain Control) block functions by automatically adjusting its ampification gain (of the incoming line signal) in order to insure that this amplitude of the incoming line signal (after being amplified by the AGC block) is within a certain desirable range for optimal internal signal processing within the remainder of the Receive DS3/E3 LIU Block circuitry.

Therefore, if the incoming DS3 or E3 line signal is of an amplitude that is less than this desired amplitude, then the AGC block will increase the amplitude of the incoming DS3/E3 signal by providing the appropriate amount of Gain to this signal.

If the amplitude of the incoming DS3 or E3 signal were to suddenly increase, then the AGC block will seek to make sure that the amplitude of this internal signal remains within the desirable range by lowering its gain to the appropriate level. Conversely, if the amplitude of the incoming DS3 or E3 signal were to suddenly decrease, then the AGC block will seek to make sure that the amplitude of this internal signal remains within the desirable range by increasing its gain to the appropriate level.

4.3.1.2.2 Functioning as a Part of the ALOS Detector

If the amplitude of the incoming DS3 signal were to drop to a very low level, then the AGC block will respond by increasing its gain (in order to drive the internal signal level back to within the desirable range). However, if the AGC block increases its gain beyond a certain amount, then the Receive DS3/E3 LIU Block will conclude that the amplitude of the incoming DS3 or E3 signal is simply too low and that it should declare the LOS (Loss of Signal) defect condition.

For more information the Analog LOS Detector, and the LOS Defect Declaration criteria, please see Section 4.3.1.6.2.

4.3.1.2.3 Configuration Options associated with the AGC Block - Operating the Receive DS3/E3 LIU Block in the Receive Monitor Mode

For some test equipment applications, it is desirable for a given piece of equipment to be able to receive a Monitor signal. In this case, we define a Monitor signal as a DSX-3 signal that has been attenuated by 20dB of flat loss.

In order to configure the Receive DS3/E3 LIU Block to be capable of properly receiving (and monitoring) a Monitor signal, the user must configure it to operate in the Receive Monitor Mode. The user can configure the XRT79L71 to operate in the Receive Monitor Mode by setting Bit 1 (Receive Monitor Mode Enable), within the LIU Receive Control Register, to "1" as depicted below.

LIU Receive Control Register (Address = 0x1305)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Disable DLOS Detector	Disable ALOS Detec-tor	Unused	LOSMUT Enable	Receive Monitor Mode Enable	Receive Equalizer Enable
R/O	R/O	R/W	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	1	1

4.3.1.3 The Receive Equalizer Block

As a given pulse within a DS3 or E3 line rate signal travels from its source to the destination terminal via coaxial cable, it will experience a frequency-dependent loss, in which the high-frequency portions of this signal are more greatly attenuated than are the lower-frequency components of the signal. The result of this frequency-dependent loss is manifested by a change in shape of a given pulse within this DS3 or E3 line signal. More specifically, one will typically note that (as a given pulse travels along the communication medium) fast rising and falling edges give way to slower rising and falling edge. Pulses that originally of the square-wave shape (at the output of the source terminal) become less square and more rounded in shape as they travel along the coaxial cable. If the DS3 data-stream travels a sufficiently long distance, then an entity that is responsible for properly receiving a given incoming DS3 signal, will have trouble receiving this particular signal, due to phenomenon such as ISI (Inter-Symbol Interference).

The purpose of the Receive Equalizer block is to compensate for this frequency-dependent attenuation that occurs within a DS3 or E3 signal, traveling via coaxial cable. In essence, the Receive Equalizer block accomplishes this by applying higher gain to higher frequency signals, than it does for lower frequency signals.

The Register Set, within the XRT79L71, permits the user to either enable or disable the Receive Equalizer, within the Receive DS3/E3 LIU Block. In general, we strongly recommend that the user ALWAYS ENABLE the Receive Equalizer block for all applications.

The user can enable the Receive Equalizer block by setting Bit 0 (Receive Equalizer Enable), within the LIU Receive Control Register, to "1" as depicted below.

LIU Receive Control Register (Address = 0x1305)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Disable DLOS Detector	Disable ALOS Detector	Unused	LOSMUT Enable	Receive Monitor Mode Enable	Receive Equalizer Enable
R/O	R/O	R/W	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	1

NOTE: *By default, the Receive Equalizer block will (upon power up) be disabled. Therefore, the user MUST include enabling the Receive Equalizer block as a part of the start-up/configuration procedure.*

4.3.1.4 The Clock and Data Recovery Block

The purpose of the Clock and Data Recovery block (within the Receive DS3/E3 LIU Block) is two-fold.

- To acquire and maintain phase-lock with the incoming DS3 or E3 line signal.
- To insure that downstream circuitry, (such as the Receive DS3/E3 Framer block) is always provided with a line-rate clock signal (to use as it timing reference).

Upon power-up, the Clock and Data Recovery block will attempt to acquire phase-lock with the incoming DS3 or E3 line signal. Once the Clock and Data Recovery block has acquired lock with the incoming DS3 or E3 signal, then it will indicate this fact by doing all of the following.

- It will set Bit 2 (Receive LOL Defect Declared), within the LIU Alarm StatusRegister, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Unused	Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the Change of LOL (Loss of Lock) Defect Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by (a) asserting the Interrupt Request output pin (by toggling it "Low"), and (b) by setting Bit 2 (Change of LOL Condition Interrupt), within the LIU Interrupt Status Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

For the duration that the Clock and Data Recovery block is maintaining lock with the incoming DS3 line signal, then all of the following will be true.

- The Clock and Data Recovery block will be synthesizing a 44.736MHz recovered clock signal that is derived from the incoming DS3 line signal.
- The Clock and Data Recovery block will route this 44.736MHz clock signal to all down-stream circuitry (e.g., the Receive DS3/E3 Framer block, etc.). Each of these down-stream blocks will use this 44.736MHz clock signal as their timing source.
- The Clock and Data Recovery block will be continuously comparing the frequency of its Recovered Clock signal with a Reference Clock signal that it receives from the SFM Synthesizer Block. As long as the differences between these two frequencies is less than 0.5% (or 5000ppm), then the Clock and Data Recovery block will continue to operate in this normal mode.

NOTE: *Information on the SFM Synthesizer block can be found in Section 4.3.1.5The LOL (Loss of Lock) Defect Declaration Criteria*

As mentioned above, the Clock and Data Recovery block will be continuously monitoring the frequency of its recovered clock signal with a Reference Clock signal that it receives from the SFM Synthesizer block. As long as the differences between these two frequencies is less than 0.5% (or 5000ppm), then the Clock and Data Recovery block will continue to operate in this normal mode.

However, if the difference between these two frequencies exceeds 0.5% (or 5000ppm), then the Clock and Data Recovery block will declare the LOL (Loss of Lock) Defect Condition. Whenever the Clock and Data Recovery block declares the Loss of Lock Defect Condition, then it will cease its attempt to acquire and maintain phase-lock with the incoming DS3 line signal. Instead, the Clock and Data Recovery will now lock onto the reference clock signal from the SFM Synthesizer block. The Clock and Data Recovery will make this transition, in order to guarantee that all down-stream circuitry (e.g., the Receive DS3/E3 Framer block) will be provided with a proper line-rate clock signal.

The XRT79L71 will inform the user that it is declaring the Loss of Lock Condition by doing all of the following.

- It will set Bit 2 (Receive LOL Defect Declared), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Unused	Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	0	0

- It will generate the Change of LOL (Loss of Lock) Defect Condition Interrupt. The XRT79L71 will indicate that it is generating this interrupt by (a) asserting the Interrupt Request output pin (by toggling it "Low"), and (b) by setting Bit 2 (Change of LOL Condition Interrupt), within the LIU Interrupt Status Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

For the duration that the Clock and Data Recovery block is declaring the LOL Defect Condition then all of the following will be true.

- The Clock and Data Recovery block will be synthesizing a 44.736MHz clock signal that is derived from the Reference clock signal (which originates from the SFM Synthesizer block).
- The Clock and Data Recovery block will route this 44.736MHz clock signal to all down-stream circuitry (e.g., the Receive DS3/E3 Framer block, etc.). Each of these down-stream blocks will use this 44.736MHz clock signal as their timing source.

The Clock and Data Recovery Block and its LOL Defect Declaration Behavior during Signal Present and No Signal Present Conditions

Once the Clock and Data Recovery Block has declared the LOL Defect Condition, then any subsequent behaviour of the Clock and Data Recovery Block, depends upon whether the AGC Block (within the Receive DS3/E3 LIU Block) determines that some sort of signal energy is present (at the RTIP/RRING input pins) or not.

If No Signal is present at the RTIP/RRING Input Pins

If the incoming DS3 line signal were to be removed, such that there is absolutely no signal energy being applied to the RTIP/RRING input pins, then all of the following will happen.

- The Receive DS3/E3 LIU Block will declare the LOS Defect Condition (please see Section 4.3.1.6 for more information on how the Receive DS3/E3 LIU Block declares the LOS Defect Condition).

- The Clock and Data Recovery block (now having no incoming DS3 line signal to lock onto) will begin to drift towards its VCO center frequency. As the Clock and Data Recovery block begins to do this, at some point, the Recovered Clock frequency (of the Clock and Data Recovery) will drift to beyond the point where it differs from the Reference Clock signal (being supplied by the SFM Synthesizer block) by 0.5% (or 5000ppm). Once the Recovered Clock signal differs from the Reference Clock signal by 5000ppm, or more, then it will declare the LOL Defect Condition.
- Once the Clock and Data Recovery block has declared the LOL Defect condition, it will now lock onto the Reference Clock signal (from the SFM Synthesizer block). In this case, the Recovered clock signal (from the Clock and Data Recovery block) will be derived from the Reference Clock signal (from the SFM Synthesizer block).

For the duration that the AGC Block is in the High-Gain Mode (which indicates that no signal energy is presented at the RTIP/RRING input pins), then the Clock and Data Recovery block will remain in the above-mentioned state. In other words, it will continue to declare the LOL Defect Condition. It will also continue to synthesize and route a 44.736MHz clock signal (to the down-stream circuitry, such as the Receive DS3/E3 Framer block) based upon the Reference Clock signal from the SFM Synthesizer block.

If A Signal is present at the RTIP/RRING Input Pins

If there is an impairment within the incoming DS3 line signal, such that signal energy was still present at the RTIP/RRING input pins, however Recovered Clock frequency (from the Clock and Data Recovery) differs from the Reference Clock Frequency by more than 5000ppm, then all of the following will happen.

- The Receive DS3/E3 LIU Block may (or may not) declare the LOS Defect condition (please see Section 4.3.1.6 for more information on how the Receive DS3/E3 LIU Block declares the LOS Defect Condition). This condition depends upon the amplitude of the signal being applied to the RTIP/RRING input pins.
- The Clock and Data Recovery block (now locking onto a signal that is more than 5000ppm off in frequency, from the Reference Clock signal (being supplied by the SFM Synthesizer block). As a consequence, the Clock and Data Recovery will declare the LOL Defect Condition.
- Once the Clock and Data Recovery block has declared the LOL Defect condition, it will now lock onto the Reference Clock signal (from the SFM Synthesizer block). In this case, the Recovered clock signal (from the Clock and Data Recovery block) will be derived from the Reference Clock signal (from the SFM Synthesizer block).

In this case, the AGC Block will not be in the High-Gain Mode (which indicates that there is some signal energy present at the RTIP/RRING input pins). Whenever this is the case, then the Clock and Data Recovery block will constantly be cycling through the following states.

STATE 1: The Clock and Data Recovery is locked onto a line signal that is more than 5000ppm off in frequency from the Reference Clock signal (that is supplied by the SFM Synthesizer block). As a consequence, the Clock and Data Recovery declares the LOL Defect Condition.

STATE 2: Now that the Clock and Data Recovery is declaring the LOL Defect condition, it (for the time-being) will cease to attempt to lock onto the incoming line signal (via the RTIP/RRING input pins), and will (instead) lock onto the Reference Clock signal (that originates from the SFM Synthesizer block). In this case, the difference between Recovered Clock and the Reference Clock is no longer be greater than 5000ppm. It will now be 0ppm (because the Recovered Clock is now being synthesized from the Reference Clock). As a consequence, the Clock and Data Recovery will now clear the LOL Defect condition.

STATE 3: Now that the Clock and Data Recovery has cleared the LOL Defect condition, it will now attempt to acquire lock with the incoming DS3 line signal. As the Clock and Data Recovery does this, it will continue to make the comparison between the frequency of its Recovered Clock and the frequency of the Reference Clock.

4.3.1.5 The SFM (Single-Frequency Mode) Synthesizer Block

The purpose of the SFM Synthesizer block is to provide the Clock and Data Recovery with a proper Reference clock signal that is of the frequency 44.736MHz (for DS3 applications) or 34.368MHz (for E3 applications).

The Clock and Data Recovery block will use this Reference Clock signal in order to determine whether or not it should declare the LOL (Loss of Lock) Defect Condition.

The SFM Synthesizer block can be configured to operate in one of the following two modes.

- The SFM (Single Frequency) Mode, or
- The Multiplexer Mode

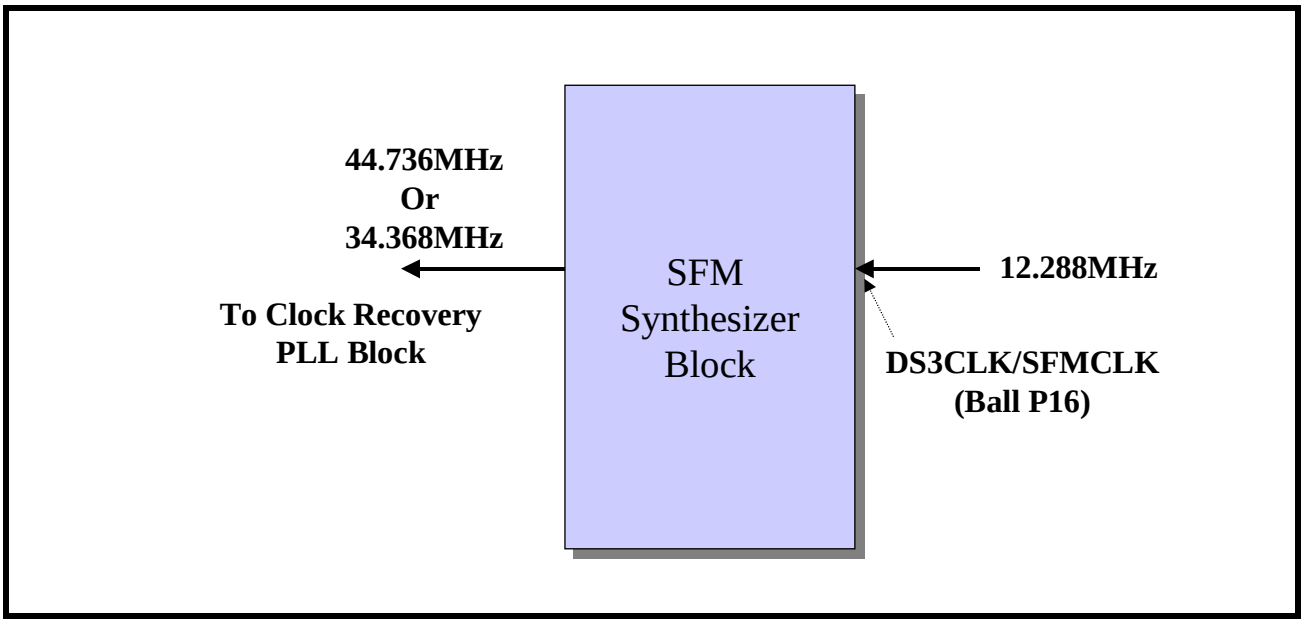
Each of these modes of operation will be discussed in detail below.

4.3.1.5.1 Operating the SFM Synthesizer Block in the SFM (Single-Frequency) Mode

To configure the SFM Synthesizer block to operate in the SFM Mode, then the user is expected to provide a 12.288MHz clock signal to the DS3CLK/SFMCLK input pin (Ball P16). The SFM Synthesizer block will then accept this 12.288MHz clock signal, and it will synthesize either a 44.736MHz clock signal (if the XRT79L71 has been configured to operate in the DS3 Mode), or it will synthesize a 34.368MHz clock signal (if the XRT79L71 has been configured to operate in the E3 Mode). The Clock and Data Recovery will, in turn, use this synthesized clock signal as its frequency reference, in order to determine whether or not it should declare the LOL defect condition.

This mode is referred to as the Single-Frequency Mode, because the user only needs to supply a single clock frequency (12.288MHz, in this case) to the DS3CLK/SFMCLK input pin, in order to permit the SFM Synthesizer block to fully support all operational requirements of the XRT79L71. Figure 85 presents a simple illustration that depicts how the SFM Synthesizer Block functions whenever it has been configured to operate in the SFM Mode.

FIGURE 85. A SIMPLE ILLUSTRATION THAT DEPICTS HOW THE SFM SYNTHESIZER BLOCK FUNCTIONS WHENEVER IT HAS BEEN CONFIGURED TO OPERATE IN THE SFM MODE



Configuring the SFM Synthesizer Block to operate in the SFM (Single-Frequency) Mode

The user can configure the SFM Synthesizer Block to operate in the SFM (Single-Frequency) Mode by executing the following steps.

STEP 1 - Apply a 12.288MHz clock signal to the DS3CLK/SFMCLK Input pin (Ball P16)

STEP 2 - Tie the E3CLK input pin (Ball M16) to GND.

STEP 3 - Set Bit 5 (SFM Enable) within the LIU Channel Control Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	1	0	0	0	0	0

Using the 44.736MHz/34.368MHz Synthesized Clock signal (from the SFM Synthesizer Block) as an External Clock Signal

As mentioned earlier, if the SFM Synthesizer block has been configured to operate in the SFM (Single-Frequency) Mode, then it will (1) accept a 12.288MHz clock signal (via the DS3CLK/SFMCLK input pin), and it will use this 12.288MHz clock signal to (2) synthesize either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) and route this signal to the Clock and Data Recovery block.

In some applications there may be a desire to externally use the 44.736MHz or 34.368MHz clock signal that is synthesized by the SFM Synthesizer block elsewhere in the user's board design. If this is the case, then the XRT79L71 can support this requirement.

The XRT79L71 includes a pin (or ball) that is called CLKOUT (Ball K16). This output pin can be configured to output the 44.736MHz/34.368MHz clock signal that is synthesized by the SFM Synthesizer block.

The user can invoke this feature by setting Bit 6 (SFM Clock Out Enable), within the LIU Channel Control Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	1	1	0	0	0	0	0

Once the user sets this bit-field to "1", then the output driver (associated with the CLKOUT pin) will become active, and either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) will be output via this pin.

NOTE: *If this CLKOUT feature is invoked, the user must be aware that this clock signal is ultimately derived from the 12.288MHz clock signal (being applied to the DS3CLK/SFMCLK input pin) and is NOT the Recovered clock signal from the Clock and Data Recovery. block*

4.3.1.5.2 Operating the SFM Synthesizer Block in the Multiplexer Mode

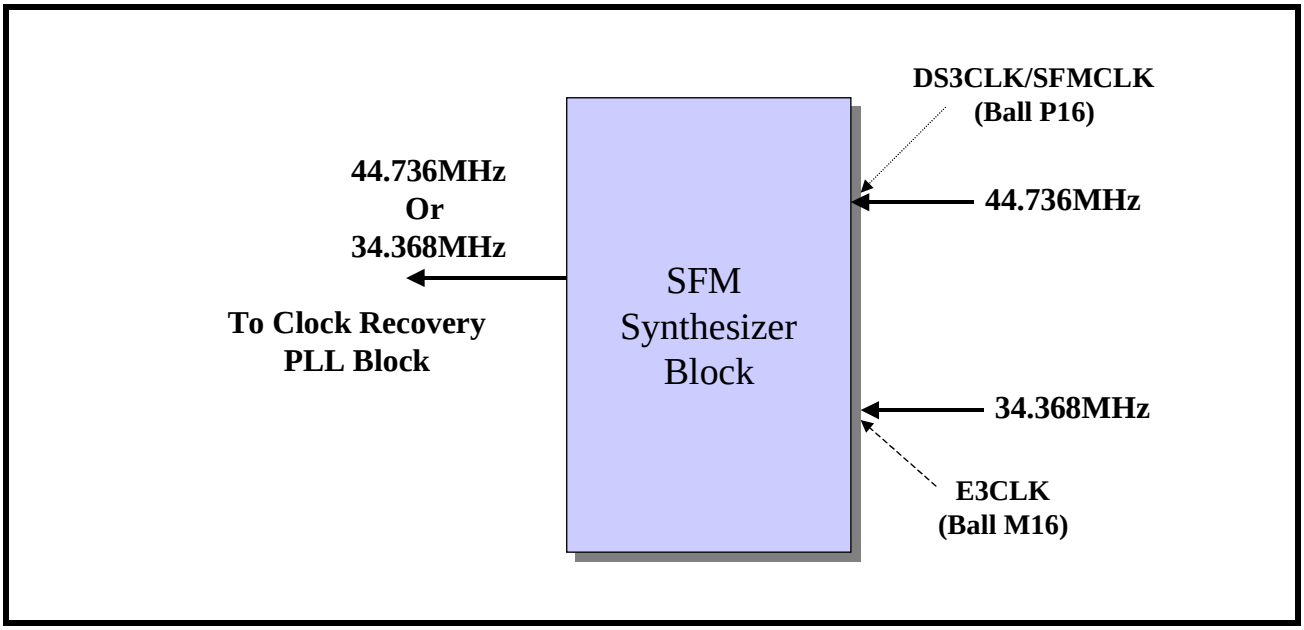
To configure the SFM Synthesizer block to operate in the SFM Mode, provide any one of the following signals to the following input pins.

- A 44.736MHz clock signal to the DS3CLK/SFMCLK input pin (Ball P16), or/and
- A 34.368MHz clock signal to the E3CLK input pin (Ball M16)

The SFM Synthesizer block will then multiplex and route the appropriate clock signal (e.g., the 44.736MHz or the 34.368MHz) to the Clock and Data Recovery, depending upon whether the XRT79L71 has been configured to operate in the DS3 or the E3 Mode.

This mode is referred to as the Multiplexer Mode, because the SFM Synthesizer block can accept up to both a 44.736MHz and a 34.368MHz via their respective inputs, and it will only output one of these clock signals (to the Clock and Data Recovery block) depending upon which mode (e.g., either DS3 or E3) that the XRT79L71 has been configured to operate in. Figure 86 presents a simple illustration that depicts how the SFM Synthesizer Block functions whenever it has been configured to operate in the Multiplexer Mode.

FIGURE 86. A SIMPLE ILLUSTRATION THAT DEPICTS HOW THE SFM SYTHESIZER BLOCK FUNCTIONS WHENEVER IT HAS BEEN CONFIGURED TO OPERATE IN THE MULTIPLEXER MODE



4.3.1.5.2.1 *Configuring the SFM Synthesizer Block to operate in the Multiplexer Mode*

The user can configure the SFM Synthesizer Block to operate in the Multiplexer Mode by executing the following steps.

STEP 1 - Apply a 44.736MHz clock signal (if available) to the DS3CLK/SFMCLK input pin (Ball P16), and apply a 34.368MHz clock signal (if available) to the E3CLK input pin (Ball M16).

STEP 2 - Set Bit 5 (SFM Enable) within the LIU Channel Control Register to "0" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	0	0	0	0	0	0

Using the 44.736MHz/34.368MHz Synthesized Clock signal (from the SFM Synthesizer Block) as an External Clock Signal

As mentioned earlier, if the SFM Synthesizer block has been configured to operate in the Multiplexer Mode, then it will (1) accept a 44.736MHz clock signal (via the DS3CLK/SFMCLK input pin), and a 34.368MHz clock signal (via the E3CLK input pin) and it (2) will output ether one of these clock signals to the Clock and Data Recovery block (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode).

In some applications there may be a desire to externally use the 44.736MHz or 34.368MHz clock signal that is synthesized by the SFM Synthesizer block elsewhere in the user's board design. If this is the case, then the XRT79L71 can support this requirement.

The XRT79L71 includes a pin (or ball) that is called CLKOUT (Ball K16). This output pin can be configured to output the 44.736MHz/34.368MHz clock signal that is synthesized by the SFM Synthesizer block.

The user can invoke this feature by setting Bit 6 (SFM Clock Out Enable), within the LIU Channel Control Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	1	0	0	0	0	0	0

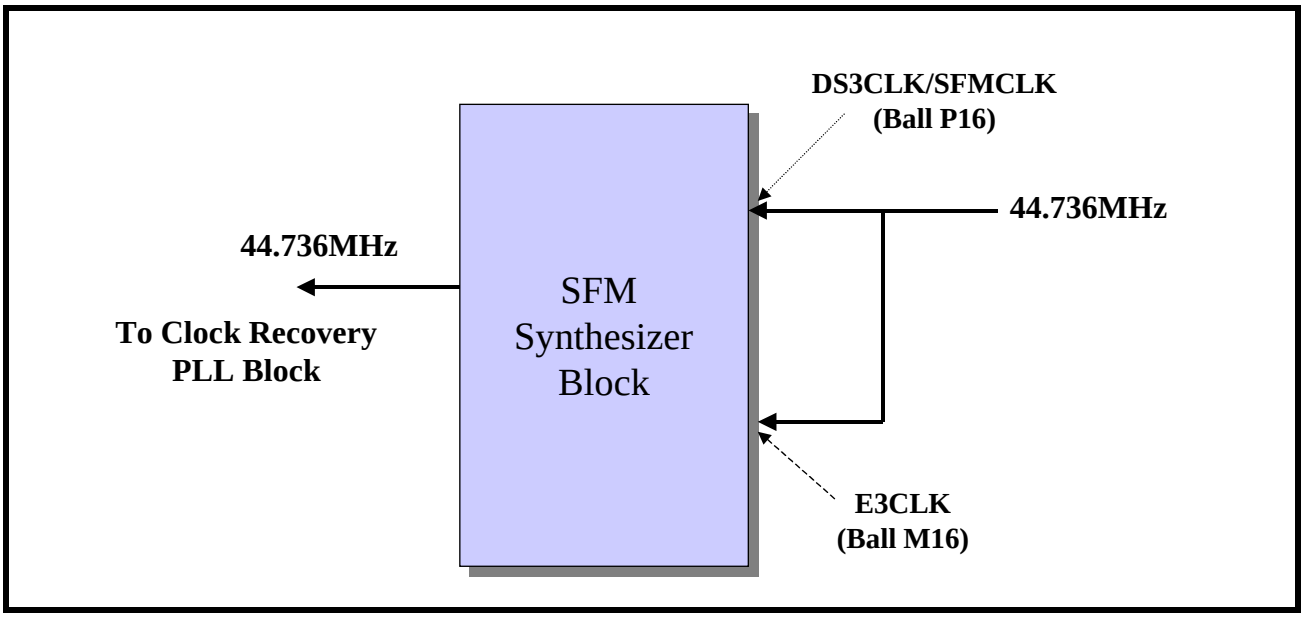
Once the user sets this bit-field to "1", then the output driver (associated with the CLKOUT pin) will become active, and either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) will be output via this pin.

4.3.1.5.2.2 Approach to supporting DS3 Applications only, if the SFM Synthesizer Block is configured to operate in the Multiplexer Mode

To configure the XRT79L71 to operate in only the DS3 Mode and if the SFM Synthesizer block is configured to operate in the Multiplexer Mode, then a 44.736MHz clock signal should be applied to both the DS3CLK/SFMCLK input pin (Ball P16) and the E3CLK input pin (Ball M16). This recommendation is also illustrated below in **Figure 87**.

NOTE: *The reason for this recommendation is that (by default) the XRT79L71 will be configured to operate in the E3, ITU-T G.751 framing format. In summary, all of this means that upon power-up (or following a hardware RESET), if no DS3 line signal is present at the RTIP/RRING input pins, then the Clock and Data Recovery block will initially lock on the clock signal being applied to the E3CLK input pin. By applying the 44.736MHz clock signal to the E3CLK input pin as well, the user will insure that the Clock and Data Recovery block (when initially declaring the LOL defect condition - due to no incoming line signal being present at the RTIP/RRING input pins) will be provided with a clock signal of the correct frequency upon power up.*

FIGURE 87. ILLUSTRATION OF RECOMMENDATIONS FOR THE DS3CLK/SFMCLK AND E3CLK INPUT PINS, IF THE SFM SYNTHESIZER BLOCK IS CONFIGURED TO OPERATE IN THE MULTIPLEXER MODE, ONLY TO SUPPORT DS3 MODE OPERATION.



4.3.1.6 The LOS Declaration and Clearance Criteria for DS3 Applications

The Receive DS3/E3 LIU Block consists of two different types of LOS (Loss of Signal) Detectors.

- An Analog LOS Detector, and
- A Digital LOS Detector

The Receive DS3/E3 LIU Block will declare an LOS defect condition, anytime either one of these LOS Detectors is declaring the LOS defect condition. Therefore, the overall LOS State of the Receive DS3/E3 LIU Block is simply the Wired-OR of the LOS States of both the Analog and Digital LOS Detectors.

Each of these LOS detectors will be discussed in detail below.

4.3.1.6.1 The Digital LOS Detector

The Digital LOS Detector functions by checking for the occurrence of pulses (which are derived from the incoming DS3 line signal) from the Slicer Block. The LOS Defect Declaration and Clearance criteria for the Digital LOS Detector are described below.

The LOS Defect Declaration Criteria

The Digital LOS Detector (within the Receive DS3/E3 LIU Block) will declare the LOS Defect condition anytime it detects an absence of DS3 Pulses for 160 consecutive bit-periods. The Receive DS3/E3 LIU Block will indicate (to the outside world) that the Digital LOS Detector is declaring the LOS defect condition, by doing all of the following.

- It will set Bits 5 (Digital LOS Defect Declared) and 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Unused	Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	1	0	0	0	1	0

- It will generate the Change of LOS Condition Interrupt.

NOTE: *The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the Interrupt Request output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the LIU Interrupt Status Register to "1" as depicted below.*

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

The LOS Defect Clearance Criteria

Once the Digital LOS Detector (within the Receive DS3/E3 LIU Block) declares the LOS Defect condition it will proceed to search the incoming DS3 data-stream for DS3 pulses. In short, the Digital LOS Detector will clear the LOS Defect Condition, if it detects at least a 33% pulse density within the DS3 data-stream.

However, more specifically, the Digital LOS Defect will only clear the LOS Defect condition if it detects at least 10 pulses within each of 5 consecutive 32-bit period blocks (thereby resulting in a pulse density of 33%).

The Receive DS3/E3 LIU Block will indicate (to the outside world) that the Digital LOS Detector is clearing the LOS defect condition, by doing all of the following.

- It will set Bit 5 (Digital LOS Defect Declared), within the LIU Alarm Status Register to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	1	0	0	0	X	0

NOTE: *The state of Bit 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block) within the LIU Alarm Status Register (at this point) will depend upon the state of Bit 4 (Analog LOS Defect Declared).*

- It MAY generate the Change of LOS Condition Interrupt.

NOTES:

1. *The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the Interrupt Request output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the LIU Interrupt Status Register to "1" as depicted below.*

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2. *The XRT79L71 will only generate this Change of LOS Condition Interrupt if the Digital LOS Detector is also NOT currently declaring the LOS defect condition.*

Disabling the Digital LOS Detector

By default, both the Analog and Digital LOS detectors will be enabled. However, if (for some reason) the user wishes to disable the Digital LOS Detector, then the user can accomplish this by setting Bit 5 (Disable DLOS Detector), within the LIU Receive Control Register to "1" as depicted below.

LIU Receive Control Register (Address = 0x1305)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Disable DLOS Detector	Disable ALOS Detec-tor	Unused	LOSMUT Enable	Receive Monitor Mode Enable	Receive Equalizer Enable
R/O	R/O	R/W	R/W	R/O	R/W	R/W	R/W
0	0	1	0	0	0	0	1

4.3.1.6.2 The Analog LOS Detector

The LOS Defect Declaration and Clearance criteria for the Analog LOS Detectors are described below.

The Functional Operation of the Analog LOS Detector

The Receive DS3/E3 LIU Block consists of a Peak Detector and an AGC (Automatic Gain Control) Amplifier. These two blocks function as a significant portion of the Analog LOS Detector. The purpose of the Peak Detector is to determine and reflect a running average of the peak amplitude of the incoming DS3 line signal. The purpose of the AGC block is to amplify the incoming DS3 line signal by an appropriate factor, such that this Received line signal will be represented by a signal (inside the Receiver circuitry) which contains an amplitude that ranges in between some pre-determined MIN and MAX value (in order to permit proper operation of the chip).

The AGC block uses the output of the Peak Detector in order to determine the amplification factor that it should use on the incoming DS3 line signal.

During an LOS condition, the amplitude of the Receive line signal will drop to a very low voltage level. At this point, the Peak Detector will reflect this low signal amplitude in its output to the AGC block. The AGC block will then proceed to increase its gain in order to amplify the incoming DS3 line signal such that the internal signal will have an amplitude between the some MIN and MAX voltage. Now, because the amplitude of the incoming DS3 line signal is very low (as one would expect during an LOS condition), the AGC block is going to increase its gain significantly. Once the gain of the AGC block exceeds a certain value, then the Analog LOS Detector is going to make the presumption that there is no line signal (at the receive input of the chip) and that it is experiencing an LOS defect condition. As a consequence, the entire Receive DS3/E3 LIU Block will, in turn, be declaring the LOS defect condition.

NOTE: *It is the gain of the AGC block that is driving this declaration of the LOS defect condition.*

At some later time, the Receive Line signal will eventually be restored. Whenever this occurs, then the amplitude of this line signal will increase. At this point, the Peak Detector will reflect this increase in the incoming line signal amplitude, in its output to the AGC block. The AGC block will then proceed to reduce its gain in order to amplify the incoming line signal by the appropriate amount. Once the gain of the AGC block drops below a certain value, then the Analog LOS Detector is going to make the presumption that there is a line signal (at the RTIP/RRING input pins of the chip) and that it is no longer experiencing an LOS defect condition. At this time, the Analog LOS defect will then clear the LOS defect condition.

The LOS Defect Declaration Criteria

The Analog LOS Detector (within the Receive DS3/E3 LIU Block) will declare the LOS Defect condition anytime the amplitude of the incoming DS3 line signal is determined to be less than _ (when measured across the RTIP and RRING input pins). The Receive DS3/E3 LIU block will indicate (to the outside world) that the Analog LOS Detector is declaring the LOS defect condition, by doing all of the following.

- It will set Bits 4 (Analog LOS Defect Declared) and 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	1	0	0	1	0

- It will generate the Change of LOS Condition Interrupt.

NOTE: *The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the Interrupt Request output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the LIU Interrupt Status Register to "1" as depicted below.*

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

The LOS Defect Clearance Criteria

Once the Analog LOS Detector is currently declaring the LOS Defect condition, it will only clear the LOS Defect condition, whenever it determines that the amplitude of the incoming DS3 line signal is greater than _ (when measured across the RTIP and RRING input pins).

The Receive DS3/E3 LIU Block will indicate (to the outside world) that the Analog LOS Detector is clearing the LOS defect condition, by doing all of the following.

- It will set Bit 5 (Digital LOS Defect Declared), within the LIU Alarm Status Register to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Unused	Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	1	0	0	0	X	0

NOTE: *The state of Bit 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block) within the LIU Alarm Status Register (at this point) will depend upon the state of Bit 4 (Analog LOS Defect Declared).*

- It MAY generate the Change of LOS Condition Interrupt.

NOTES:

1. *The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the Interrupt Request output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the LIU Interrupt Status Register to "1" as depicted below.*

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

2. *The XRT79L71 will only generate this Change of LOS Condition Interrupt if the Digital LOS Detector is also NOT currently declaring the LOS defect condition.*

4.3.1.7 Jitter Attenuator Block

Please see Section 4.2.6.2 for a description of the Jitter Attenuator Block

4.3.1.8 The B3ZS Decoder Block

The purpose of the B3ZS Decoder block is to decode the inbound DS3 traffic from the B3ZS Line Code, into a digital data-stream. In the case of the XRT79L71, the B3ZS Decoder block will always be enabled and the user has no ability to disable the B3ZS Decoder block.

4.3.1.9 Performance Characteristics of the Receive DS3 LIU Block

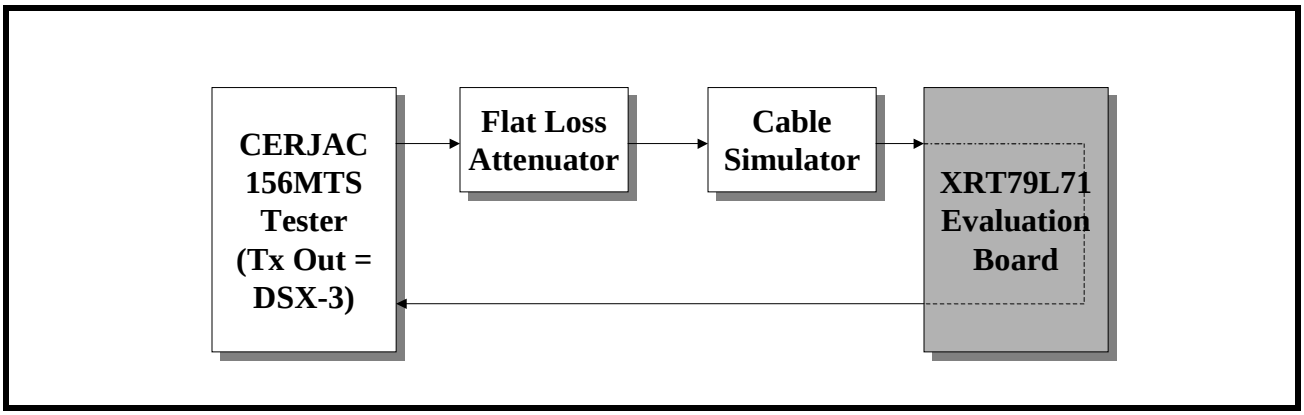
These next few sections will present the performance characteristics of the Receive DS3/E3 LIU Block, within the XRT79L71. In particular these sections will address the following parameters for DS3 Applications.

- Receive Sensitivity
- Interference Margin
- Jitter Tolerance

4.3.1.9.1 Receive Sensitivity Capability of the Receive DS3 LIU Block

For DS3 Applications, the Receive DS3 LIU Block MUST be capable of receiving a DSX-3 signal that has been attenuated by anywhere from 0 to 450 feet of cable loss, and at least 6dB of flat loss, in a un-erred manner. **Table 25** summarizes the Receive Sensitivity of the Receive DS3/E3 LIU Block, within the XRT79L71.

FIGURE 88. ILLUSTRATION OF TEST SET-UP TO PERFORM THE RECEIVE SENSITIVITY LOW-LEVEL TEST



Test Approach

The CERJAC 156MTS tester was configured to generate ATM cells and to map these ATM cells into a DS3 data stream, which was then to be output as a bipolar line signal. The Transmit Output of the CERJAC Tester was set to DSX-3. Hence, the Tester will generate (within its output line signal) pulses that comply with the DSX-3 Isolated Pulse Template requirement per Bellcore GR-499-CORE.

The DS3 line signal was then routed to the HP355C VHF (Flat Loss) Attenuator. The purpose of this flat-loss attenuator was to reduce the amplitude of the pulses, within the Transmit Output DS3 line signal, to 360mVpk (the minimum allowable pulse amplitude at the DSX-3 Cross-connect locations).

After the DS3 line signal has been reduced (in amplitude) by the appropriate amount of flat-loss, it was then routed to the ME-1005 75W Coaxial Cable Simulator (from Mountain Engineering).

The Cable Simulator was configured to insert the user-selected amount of shaped (or cable) loss into this DS3 line signal.

After this DS3 line signal has been subjected to an appropriate amount of flat-loss and cable loss, it was then routed to the Receive Input of the XRT79L71 Evaluation Board. The XRT79L71 Evaluation Board was configured to operate in the UTOPIA Loop-back Mode. Therefore, the XRT79L71 Evaluation Board would handle the attenuated DS3 line signals in the following manner.

- The Receive DS3/E3 LIU Block (within the XRT79L71) would receive this DS3 line signal and perform Clock and Data Recovery on this line signal.
- This Recovered Clock and Data would then be routed to the Receive DS3/E3 Framer block, the Receive ATM Cell Processor and the Receive UTOPIA Interface block. The UTOPIA FPGA would then read out the contents of these ATM cells and it would then write these ATM cells back into the Transmit UTOPIA Interface block.
- The Transmit Section of the XRT79L71 would then accept these ATM cells and it would (once again) map these ATM cells back into a DS3 data-stream. The XRT79L71 would output a DS3 line signal, which would be routed back into the Receive Input of the CERJAC 156MTS Tester.

The CERJAC 156MTS Tester can determine whether or not bit-errors have occurred in the loop-back path (through the Flat-Loss Attenuators, the Cable Simulator, and the XRT79L71 Evaluation Board) by checking for DS3 alarms, occurrences of HEC Byte errors (within ATM cells) and/or pattern sync errors within the ATM cell payload, etc.

Test Results

The Receive Sensitivity Test Results are summarized below in **Table 25**.

TABLE 25: RECEIVE SENSITIVITY TEST RESULTS (DS3 APPLICATIONS)

TEST NUMBER	POWER SUPPLY VOLTAGE	RECEIVE SENSITIVITY	COMMENTS
1	3.15V	DSX-3 + 1215 feet of Cable Loss	
2	3.30V	DSX-3 + 1215 feet of Cable Loss	
3	3.45V	DSX-3 + 1215 feet of Cable Loss	

4.3.1.9.2 Interference Margin Capability of the Receive DS3 LIU Block

The purpose of an Interference Margin Test (or specification) is to verify that the Receive DS3/E3 LIU Block (within the XRT79L71) can tolerate a certain amount of inband noise and still properly receive a DS3 or E3 signal.

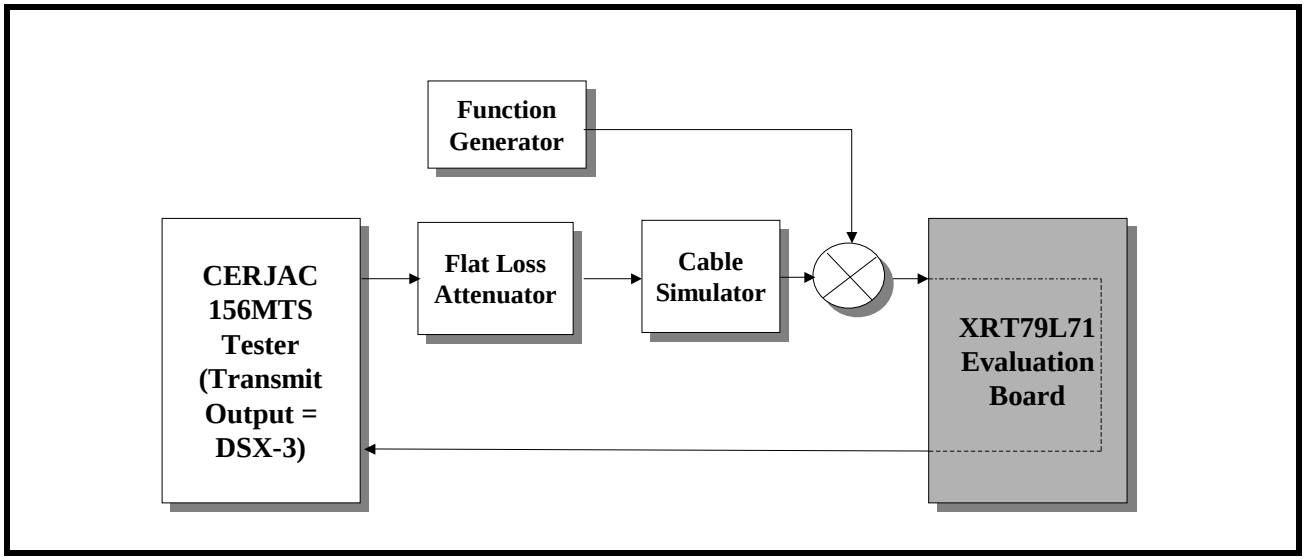
Although Interference Margin is not a specific requirement for DS3 applications, numerous customers do inquire about the Interference Margin capabilities of our DS3 LIU Products. As a consequence, we have specifically designed the Receive DS3/E3 LIU Block such that it will have sufficient Interference Margin in order to meet our Customer's requirements.

This particular section documents the Interference Margin capability of the Receive DS3/E3 LIU Block, for DS3 applications.

Test Approach

The basic test configuration used for Interference Margin measurements is illustrated below in **Figure 89**.

FIGURE 89. ILLUSTRATION OF TEST SET-UP USED TO TEST THE XRT79L71 FOR INTERFERENCE MARGIN



The CERJAC Tester was configured to internally generate a DS3 signal, which was filled with an un-framed 2^23-1 PRBS pattern. The Transmit Output of the CERJAC 156MTS Tester was set to DSX-3. Hence, the Tester will generate (within its output line signal) pulses that comply with the DSX-3 Isolated Pulse Template requirements per Bellcore GR-499-CORE.

The DS3 line signal was then routed to the ME-1005 75W Coaxial Cable Simulator (from Mountain Engineering). The Cable Simulator could either be configured to 0 or 450 feet of shaped loss into the DS3/ STS-1 line signal.

Next, this attenuated signal will be routed to one of the inputs of a summing network. The output of the function-generator (the output frequency of which is set at half the bit-rate) will be applied to the other input.

This signal represents the Interfering Tone or Noise in this test. The summing network will add the Noise signal to the distorted signal, and will output this composite signal to the Receive Input of the XRT79L71 Evaluation Board.

NOTE: *For DS3 testing, the function generator was configured to generate a sinewave that has a frequency of 22.368MHz.*

Test Results

Table 26 presents the Interference Margin Test results for DS3 Applications.

TABLE 26: INTERFERENCE MARGIN TEST RESULTS FOR DS3 APPLICATIONS

TEST NUMBER	CABLE LENGTH (IN RECEIVE DIRECTION)	INTERFERENCE MARGIN TEST RESULTS
1	DSX-3 + 0 feet	16dB
2	DSX-3 + 225 feet	15dB
3	DSX-3 + 450 feet	15dB

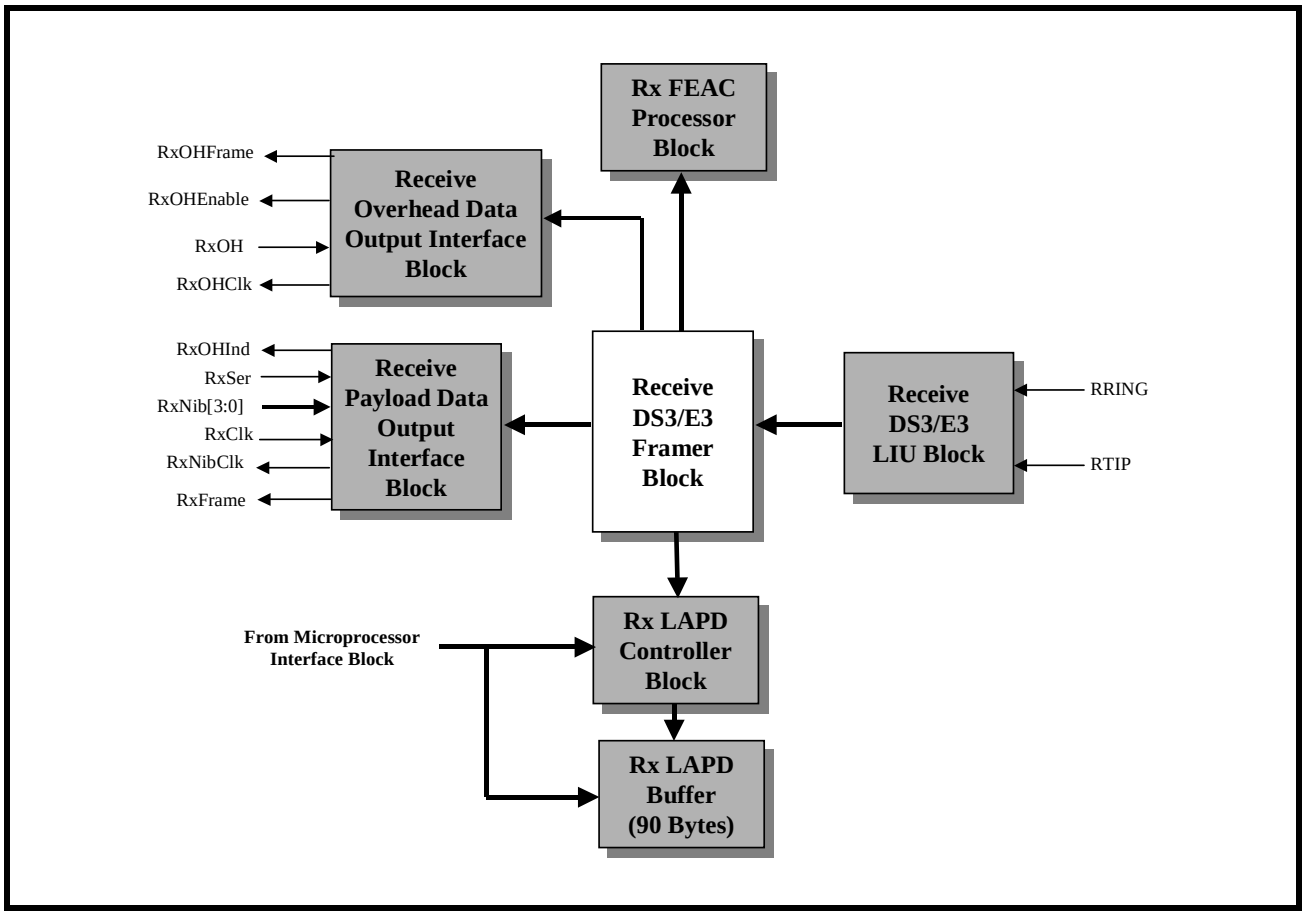
4.3.1.9.3 Jitter Tolerance Capability of the Receive DS3 LIU Block

4.3.1.10 Receive DS3/E3 LIU Block Interrupts

4.3.2 RECEIVE DS3 FRAMER BLOCK

The Receive DS3/E3 Framer Block is the second functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 90 presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive DS3/E3 Framer block highlighted.

FIGURE 90. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE DS3/E3 FRAMER BLOCK HIGHLIGHTED)



The purpose of the Receive DS3 Framer block within the XRT79L71 is to accomplish the following.

- To acquire and maintain frame synchronization with the incoming DS3 data stream.
- To detect and declare all of the following defect/error conditions.
  - **a.** LOS (Loss of Signal)
  - **b.** AIS (Alarm Indication Signal)
  - **c.** OOF (Out-of Frame)
  - **d.** FERF (Far-End Receive Failure) or Yellow Alarm
  - **e.** FEBE (Far-End Block Error) Events
  - **f.** P-bit Errors
  - **g.** CP-bit Errors

Each of these functions is described in detail below.

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 frame, or

- **The Frame Maintenance Mode:**

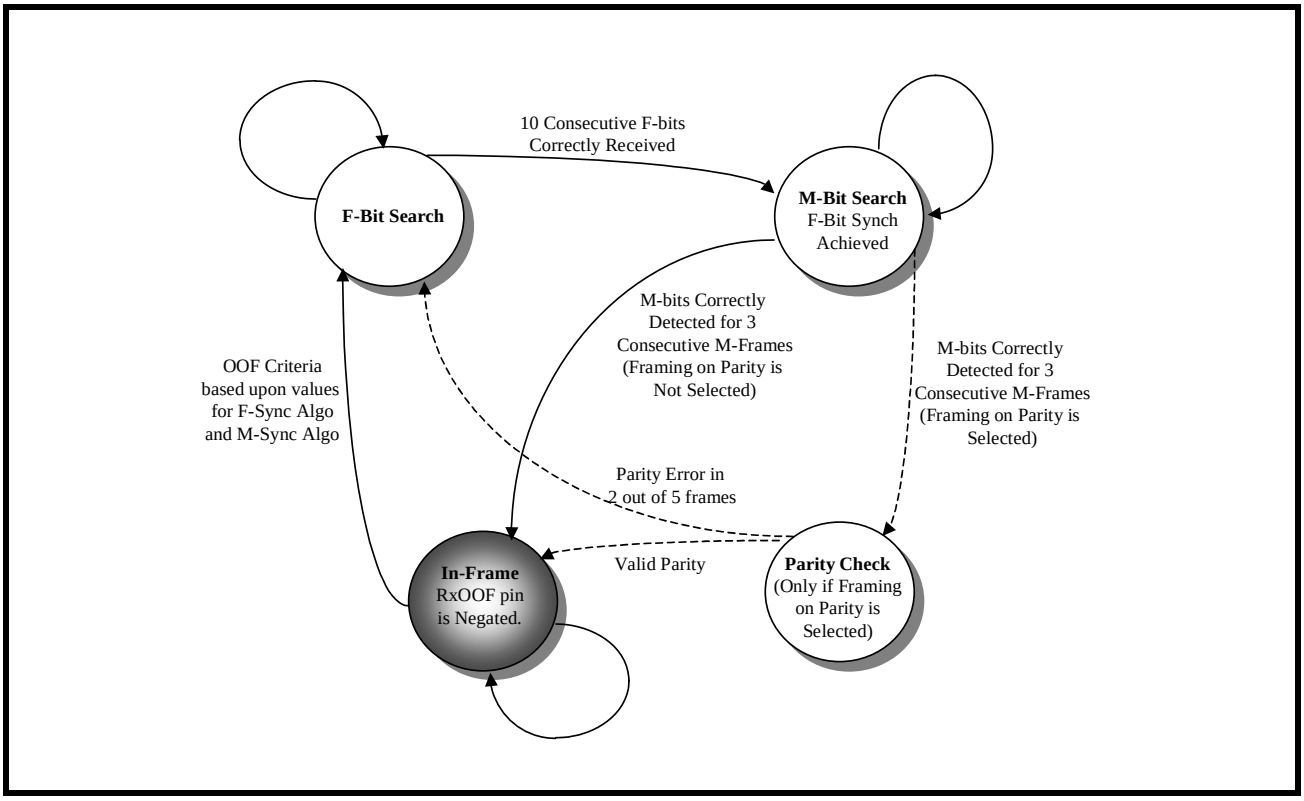
In this mode, the Receive DS3 Framer block is trying to maintain frame synchronization with the incoming DS3 Frames.

The operation of the Receive DS3 Framer block, in both the Frame Acquisition and the Frame Maintenance Modes will be discussed, in considerable detail below.

4.3.2.1 THE FRAME-ACQUISITION MODE

The operation of the Receive DS3 Framer block, while in the Frame Acquisition mode, is best understood by reviewing the Receive DS3 Framer block's Frame Acquisition/Maintenance Algorithm state machine diagram, presented below in Figure 91.

FIGURE 91. THE STATE MACHINE DIAGRAM FOR THE RECEIVE DS3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM



Processing through the Frame Acquisition Modes

The Receive DS3 Framer block will be performing Frame Acquisition operation while it is operating in the following states per the DS3 Frame Acquisition/Maintenance algorithm State Machine diagram, as depicted in Figure 91.

- F-bit Search
- M-bit Search
- Parity Check (optional)

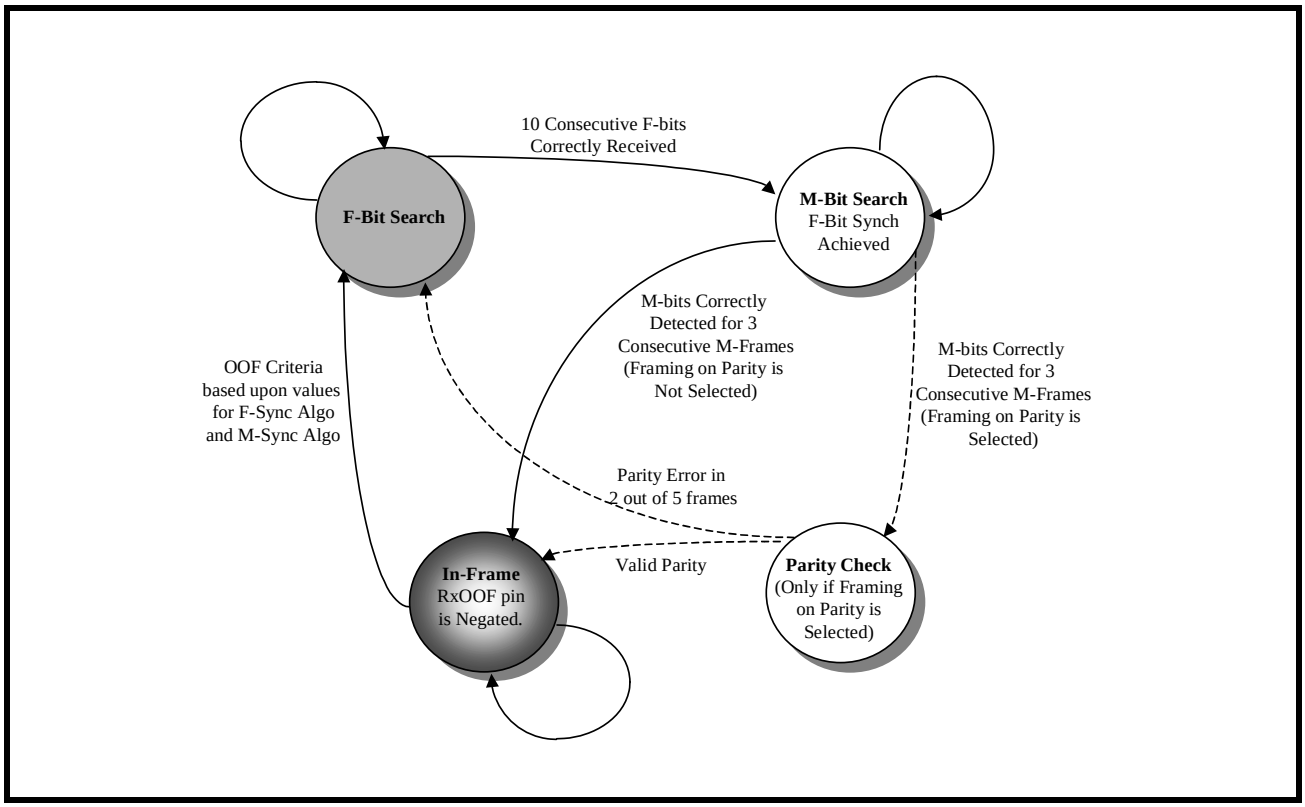
Once the Receive DS3 Framer block enters the In-Frame state (per Figure 91), then it will begin Frame Maintenance operation. The Receive DS3 Framer block's operation in each of the Frame Acquisition states is presented below.

The F-bit Search State

When the XRT79L71 is first powered up, or exits a Hardware RESET event the Receive DS3 Framer block will be operating in the frame-acquisition mode. At this point, the very first thing that the Receive DS3 Framer

block will do is to begin to look for valid DS3 frames by first searching for the F-bits. 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. In order to clearly convey the mode that the Receive DS3 Framer block is currently operating in, **Figure 91** has been repeated below, with the F-Bit Search state shaded.

FIGURE 92. THE STATE MACHINE DIAGRAM FOR THE RECEIVE DS3 FRAMER BLOCK'S FRAME ACQUISTION/MAINTENANCE ALGORITHM (WITH THE F-BIT SEARCH STATE SHADED)



Recall from the discussion in Section 4.1, 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 or 16 (depending upon user configuration) consecutive F-bits are properly detected. At this point, the Receive DS3 Framer block will declare the F-Bit Sync condition. After the F-Bit Sync condition has been declared, the Receive DS3 Framer block will then transition to the M-Bit Search state within the DS3 Frame Acquisition/ Maintenance algorithm (per **Figure 92**).

Configuration Options associated with the F-Bit Search State

The Receive DS3 Framer block circuitry contains the following two sets of configuration options that permit the user to select their criteria for declaration of F-Bit Sync. The details associated with each of these configuration options are presented below.

F-bit Sync is Declared after 10 or 16 consecutive correct F-bits

By default, when the Receive DS3 Framer block within the XRT79L71 is operating in the F-Bit Search state within the DS3 Frame Acquisition/Maintenance state machine, it will declare F-bit Sync and move on to the M-Bit Search state upon the successful detection of 10 consecutive F-bits. The XRT79L71 Command Register set permits the user to modify the F-bit Sync declaration criteria such that the Receive DS3 Framer block will declare F-bit Sync after the successful detection of 16 F-bits.

The user can accomplish this by setting Bit 1 (F-Algorithm), within the Receive DS3 Sync Detect Register to "1" (as illustrated below).

Receive DS3 Sync Detect Register (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						F Algorithm	One and Only
R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	1	0

Conversely, setting Bit 1 (F-Algorithm) within the Receive DS3 Sync Detect Register to "0" configures the Receive DS3 Framer block to declare F-bit Sync after the successful detection of 10 consecutive F-bits.

Handling of Mimicking F-bits

It was mentioned earlier that the Receiver DS3 Framer block, when operating in the Frame Acquisition Mode, will use five (5) Framing Bit Searching circuits in parallel. As each of these Framing Bit Searching circuits parses through the incoming DS3 data-stream, they are each searching for a sequence of bits that exhibit a certain pattern and spacing from each other. At any given time, it may be possible that more than one Framing Bit Searching circuit has honed in on more than one viable F-bit candidate. Whenever this occurs, it may be possible for the Receive DS3 Framer block to become fooled by data bits mimicking the F-bits. This phenomenon can result in the Receive DS3 Framer block proceeding along the DS3 Frame Acquisition/ Maintenance State Machine, while being locked onto to mimicking data, and not to the actual F-bits. At some point, the Receive DS3 Framer block will recognize that it is not locked onto the actual F-bits and that it will have to return to the F-Bit Search state. However, all of the time that the Receive DS3 Framer block spent, proceeding along with the M-Bit Search state will have been wasted and will increase the overall Frame Acquisition time.

The Command Register set, within the XRT79L71, permits the user to minimize the potential of the Framing Bit Searching circuitry being fooled by mimicking data, by setting Bit 0 (One and Only), within the Receive DS3 Sync Detect Register to "1", as illustrated below.

Receive DS3 Sync Detect Register (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						F Algorithm	One and Only
R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	X	01

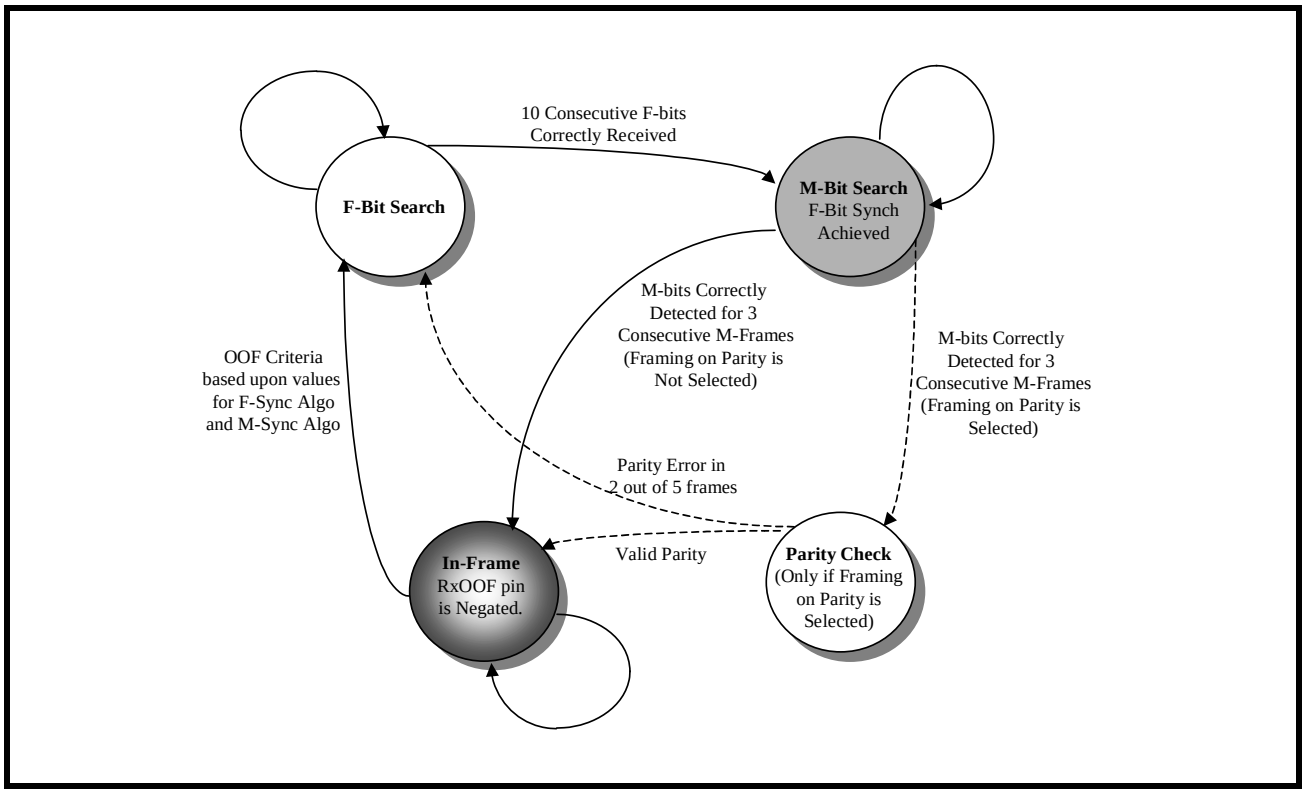
If this configuration option is implemented, then the Receive DS3 Framer block will only declare F-Bit Sync when all of the following criteria are met.

- A given Framing Bit Searching circuit has successfully located 10 or 16 consecutive, correct F-bits.
- None of the remaining four (4) Framing Bit Searching circuits have any viable candidates for F-bits. One and Only one viable candidate exists for all of the five Framing Bit Searching circuits.

The M-bit Search State

When the Receive DS3 Framer block reaches the M-Bit Search state, it will begin searching for valid M-bits within the incoming DS3 data-stream. In order to clearly convey the mode that the Receive DS3 Framer block is currently operating in, **Figure 91** has been repeated below, with the M-Bit Search state shaded.

FIGURE 93. THE STATE MACHINE DIAGRAM FOR THE RECEIVE DS3 FRAMER BLOCK'S FRAME ACQUISTION/MAINTENANCE ALGORITHM (WITH THE M-BIT SEARCH STATE SHADED)



Recall from the discussion in Section 4.1 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-bit Sync 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 either transition into the Parity-Check or directly into the In-Frame state depending upon user configuration. Each of these remaining states within the Frame Acquisition/Maintenance Algorithim is described below.

The Parity (P-Bit) Check State (Optional)

The P-Bit Check state is an optional state within the Frame Acquisition/Maintenance Algorithm. In other words, the Receive DS3 Framer block will only transition into, and operate in this state if it is configured to do so by the user. If configured accordingly, the Receive DS3 Framer block will transition into this state after it has declared M-Bit Sync. Once the Receive DS3 Framer block transitions into this state, it will check for correct P-bits within the incoming DS3 data-stream. If the Receive DS3 Framer block detects correct P-bits within the incoming DS3 data-stream then it will transition into the In-Frame state. On the other hand, if the Receive DS3 Framer block detects erred P-bits within the incoming DS3 data-stream, then it will conclude that it has not properly acquired DS3 framing and it will transition back into the F-Bit Search state.

Configuring the Receive DS3 Framer Block to support the Parity-Check State

The user can configure the Receive DS3 Framer block to operate such that 'valid parity' (P-bits) must also be detected before the Receive DS3 Framer block can declare itself In Frame. The user can select this configuration by setting Bit 2 (Framing with Valid P-Bits) within the Receive DS3 Configuration and Status Register to "1", as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	1	0	1	0	0

The Table 27 relates the contents of this bit field to the framing acquisition criteria.

TABLE 27: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 2 (FRAMING WITH VALID P-BITS) WITHIN THE RX DS3 CONFIGURATION AND STATUS REGISTER, AND THE RESULTING FRAMING ACQUISITION CRITERIA

FRAMING WITH VALID P-BITS (BIT 2)	FRAMING ACQUISITION CRITERIA
0	The In-frame is declared after F-bit synchronization (10 or 16 F-bit matches) followed by M-bit synchronization (M-bit matches for 3 DS3 M-frames) ***NOTE:*** *In this case, the P-bits are not checked as part of the In-Frame declaration criteria.*
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 or P-bit errors in 2 or more out of 5 frames starts a frame search

The In-Frame State

If the Receive DS3 Framer block either declares the M-Bit Sync condition or successfully passes through the Parity-Check state, then it 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 Defect Condition interrupt to the µC/µP.
- Set Bit 4 (Rx OOF Defect Declared) within the Receive DS3 Configuration and Status Register to "0" as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	1	0	0

Once the Receive DS3 Framer block is in the In-Frame condition, normal data recovery and processing of the DS3 data stream begins. The maximum average reframing time of the Receive DS3/E3 Framer block is less than 1.5 ms.

4.3.2.2 THE FRAME-MAINTENANCE MODE - THE OOF/LOF DEFECT DECLARATION CRITERIA

When the Receive DS3 Framer block is operating in the In-Frame state (per Figure 91), 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. In general, as long as the F- and M-bits are present and at their proper locations with a

small number of errors, the Receive DS3 Framer block will continue to declare itself being in the In-Frame condition, and will continue to operate in the Frame Maintenance Mode. However, if the Receive DS3 Framer block begins to detect a large number of certain types of errors within the incoming DS3 data-stream then it will exit the In-Frame state and will then declare the OOF Defect condition. In short, the Receive DS3 Framer block will declare the OOF defect condition when any one of the following events occur.

- When 3 or 6 F-bits (depending upon user setting), within the last 16 received F-bits are incorrect.
- When M-bit errors are detected within 3 out of four (4) consecutive incoming DS3 frames (optional)
- When P-bit errors are detected within 2 out of five (5) consecutive incoming DS3 frames (optional).

Each of these events that can possibly cause the Receive DS3 Framer block to declare the OOF defect condition will be discussed in some detail below. Additionally, any configuration options that permit the user to select or modify the OOF Defect Declaration criteria will be discussed below.

THE DS3 OOF DEFECT DECLARATION CRITERIA FOR F-BIT ERRORS

While the Receive DS3 Framer block is operating in the Frame Maintenance mode, it will declare an Out-of-Frame (OOF) defect condition anytime it determines that 3 or 6 F-bits (depending upon user selection) out of 16 consecutive F-bits are in error. The user makes this selection for the OOF Defect Declaration criteria by writing the appropriate value into bit 1 (F-Sync Algo) within the Receive DS3 Configuration and Status Register, as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	1	0	1	0	0

The following table relates the contents of this bit-field to the OOF Defect Declaration criteria for the F-bits.

TABLE 28: THE RELATIONSHIP BETWEEN THE CONTENTS OF BIT 1 (F-SYNC ALGO) WITHIN THE RECEIVE DS3 CONFIGURATION AND STATUS REGISTER, AND THE RESULTING F-BIT OOF DECLARATION CRITERIA FOR 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 determined to be in error.
1	OOF is declared when 3 out of 16 consecutive F-bits are determined to be in error.

NOTE: *Once the Receive DS3 Framer block has declared the OOF Defect condition, it will transition back to the F-Bit Search state within the DS3 Frame Acquisition/Maintenance algorithm (per* **Figure 91***).*

THE DS3 OOF DEFECT DECLARATION CRITERIA FOR M-BIT ERRORS

In addition to selecting the OOF Defect Declaration criteria for the F-bits, the user has the following two options for configuring the OOF Defect Declaration criteria based upon M-bits.

1. M-bit errors do not result in the declaration of the OOF/LOF defect condition, or
2. The LOF/OOF defect condition will be declared if M-bit errors are detected within any three (3) out of four (4) consecutive DS3 frames.

The user can select between these two options by writing the appropriate value into Bit 0 (M-Sync Algo) within the Receive DS3 Configuration and Status Register as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	1	0	1	0	0

The following table relates the contents of this Bit Field to the M-Bit Error criteria for Declaration of the OOF defect condition for the M-bits.

Table 29: The Relationship between the contents of Bit 0 (M-Sync Algo) within the Receive DS3 Configuration and Status Register, and the resulting M-Bit OOF Declaration Criteria for the Receive DS3 Framer Block

MSync Algo	OOF Declaration Criteria
0	M-Bit Errors do not result in the declaration of the OOF defect condition.
1	The LOF/OOF defect condition will be declared if M-bit errors are detected within any 3 out of 4 consecutive DS3 frames.

THE DS3 OOF DEFECT DECLARATION CRITERIA FOR P-BIT ERRORS (OPTIONAL)

As mentioned earlier, the XRT79L71 offers the Framing with Valid P-Bits option, which configures the Receive DS3 Framer block to verify correct P-bits prior to transitioning into the In-Frame state. This particular configuration setting also effects the OOF Defect Declaration criteria. More specifically, this same setting will configure the Receive DS3 Framer block to also declare the OOF Defect Condition if a P-bit error is detected in 2 of the last 5 M-frames.

Whenever the Receive DS3 Framer block declares the OOF defect condition after being in the In-Frame State the following will happen.

- It will transition back into the F-Bit Search state within the Frame Acquisition/Maintenance Algorithm
- Bit 4 (RxOOF Defect Declared) within the Receive DS3 Configuration and Status Register will be set to "1" as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	1	0	1	0	0

- The Receive DS3 Framer block will also issue a Change in the OOF Defect Condition interrupt request, anytime there is a change in the OOF defect condition. The Receive DS3/E3 Framer block will indicate that it is generating this interrupt by doing all of the following.

a. Toggling the INT* output pin of the XRT79L71 "Low".

b. By asserting Bit 1 (Change of OOF Defect Condition Interrupt Status) within the Receive DS3 Interrupt Status Register, as depicted below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

Forcing a Reframe via Software Command

The Receive DS3/E3 Framer block permits the user to command a reframe procedure with the Receive DS3 Framer block via software command. The user can accomplish this by inducing a "0" to "1" transition in Bit 0 (Reframe), within the I/O Control Register, as depicted below. Once the user executes this step then the Receive DS3 Framer block will be forced into the Frame Acquisition Mode, or more specifically, in the F-Bit Search State per **Figure 93**, and will begin its search for valid F-Bits. The XRT79L71 will also respond to this command by setting the OOF Defect Declared bit-field to "1" and generating the Change in the OOF Defect Condition interrupt.

I/O Control Register (Address = 0x1F01)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	Unused				Reframe
R/W	R/O	R/W	R/O	R/O	R/O	R/O	R/W
1	0	1	0	1	0	0	0 -> 1

NOTE: *After the user has implemented the "0" to "1" transition within Bit 0 (Reframe), they should also go back and induce a "1" to "0" transition within this bit-field.*

4.3.2.3 DECLARING AND CLEARING THE LOS DEFECT CONDITION

The Receive DS3 Framer block has the responsibility for declaring and clearing the LOS (Loss of Signal) defect condition within the incoming DS3 data-stream, as described below.

4.3.2.3.1 Declaring the LOS Defect Condition

The Receive DS3 Framer block will declare the Loss of Signal (LOS) defect condition when it detects at least 180 consecutive "0s" within the incoming DS3 data-stream or if the Receive DS3/E3 LIU Block declares the LOS defect condition. The Receive DS3 Framer block will indicate that it is declaring the LOS defect condition by:

- Setting Bit 6 (LOS Defect Declared) within the Receive DS3 Configuration and Status Register to "1", as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	LOS Defect Declared	RxIdle Condition Detected	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	1	0	1	0	0	0	0

- The Receive DS3 Framer block will also generate the Change of LOS Defect Condition interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 6 (Change of LOS Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of DS3 Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

NOTE: *The LOS Defect Declaration Criteria for the Receive DS3/E3 LIU Block will be discussed in Section 4.3.1.6.*

4.3.2.3.2 Clearing the LOS Defect Condition

The Receive DS3 Framer block will clear the LOS defect condition when both of the following conditions are met.

a. When at least 60 out of 180 consecutive received bits within the incoming DS3 data-stream are "1s"

b. When the Receive DS3/E3 LIU Block clears the LOS defect condition.

The Receive DS3 Framer block will indicate that it is clearing the LOS defect condition by

- Setting Bit 6 (LOS Defect Declared) within the Receive DS3 Configuration and Status Register to "0", as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AIS Defect Declared	LOS Defect Declared	Idle Condition Detected	OOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The Receive DS3 Framer block will also generate the Change of LOS Defect Condition interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 6 (Change of LOS Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

NOTE: *The LOS Clearance criteria for Receive DS3/E3 LIU Block will be discussed in Section 4.3.1.6.*

Configuration Options for the LOS Declaration/Clearance Criteria

The Receive DS3/E3 Framer block within the XRT79L71 permits the user to change the LOS Declaration criteria such that the LOS defect condition is declared only if the Receive DS3/E3 LIU Interface declares the LOS defect condition. If this configuration selection is implemented, then the internally-generated LOS criteria of 180 consecutive "0s" will be disabled. The user can accomplish this configuration selection by writing a "0" to bit 3 (Internal LOS Enable) within the Operating Mode Register, as depicted below.

Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	1

4.3.2.3.3 The Relationship between the LOS Defect condition being declared or cleared in the Receive DS3/E3 LIU Block and in the Receive DS3/E3 Framer Block

4.3.2.4 DECLARING AND CLEARING THE AIS DEFECT CONDITION

The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the AIS (Alarm Indication Signal) defect, as described below.

The AIS Patterns that are supported by the Receive DS3/E3 Framer block

The Receive DS3 Framer block can be configured to declare the AIS defect condition, in response to detecting either of the following types of AIS patterns.

- The Bellcore GR-499-CORE standard AIS signal.
- The Relaxed Bellcore GR-499-CORE standard AIS signal
- The Unframed, All Ones AIS signal

The steps to configuring the Receive DS3/E3 Framer block into each of these modes will be presented below.

Declaring the AIS Defect condition, in response to only the Bellcore GR-499-CORE standard AIS signal

The Receive DS3 Framer block can be configured such that it will identify and declare the AIS defect 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

Configuring the Receive DS3/E3 Framer block to only declare the AIS defect condition whenever it detects the Bellcore GR-499-CORE Compliant AIS pattern

To configure the Receive DS3/E3 Framer block to declare the AIS defect condition whenever it detects the Bellcore GR-499-CORE Compliant AIS pattern thent make sure that Bits 6 (DS3 AIS Non Stuck Stuff) and 7 (DS3 AIS Unframed All Ones) within the Receive DS3 Pattern Register are set to "0" as depicted below.

Receive DS3 Pattern Register (Address = 0x112F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DS3 AIS-Unframed All Ones	DS3 AIS -Non Stuck Stuff	Unused	Receive LOS Pattern	Receive DS3 Idle Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	0	0

Declaring the AIS Defect condition in response to the Relaxed Bellcore GR-499-CORE Standard AIS signal

The Receive DS3 Framer block can be configured such that it will identify and declare the AIS defect condition if it detects all of the following conditions in the incoming DS3 data-stream.

- Valid M-bits, F-bits and P-bits
- X-bits are set to "1"
- The Payload portion of the DS3 Frame exhibits a repeating "1010…" pattern
- C-bits are NOT necessarily set to "0"

Configuring the Receive DS3/E3 Framer block to only declare the AIS defect condition whenever it detects the Relaxed Bellcore GR-499-CORE Compliant AIS pattern

To configure the Receive DS3/E3 Framer block to declare the AIS defect condition whenever it detects the Relaxsed Bellcore GR-499-CORE Compliant AIS pattern then make sure that Bit 6 (DS3 AIS Non Stuck Stuff) and Bit 7 (DS3 AIS Unframed All Ones), within the Receive DS3 Pattern Register are set to "1" and "0" respectively, as depicted below.

Receive DS3 Pattern Register (Address = 0x112F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DS3 AIS -Unframed All Ones	DS3 AIS -Non Stuck Stuff	Unused	Receive LOS Pattern	Receive DS3 Idle Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	1	0	0	1	1	0	0

Declaring the AIS Defect condition in response to the Unframed, "All Ones" DS3 AIS signal

The Receive DS3 Framer block can be configured such that it will identify and declare the AIS defect condition it is detects an Unframed "All Ones" pattern in the incoming DS3 data-stream

Configuring the Receive DS3/E3 Framer block to only declare the AIS defect condition whenever it detects the Unframed, All Ones Pattern

To configure the Receive DS3/E3 Framer block to declare the AIS defect condition whenever it detects the Unframed, All Ones pattern then make sure that Bit 7 (DS3 AIS - Unframed All Ones) within the Receive DS3 Pattern Register is set to "1" as depicted below.

Receive DS3 Pattern Register (Address = 0x112F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DS3 AIS - Unframed All Ones	DS3 AIS -Non Stuck Stuff	Unused	Receive LOS Pattern	Receive DS3 Idle Pattern[3:0]			
R/W	R/W	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	1	0	0

The Receive DS3 Framer block contains, within its circuitry, an Up/Down Counter that supports the declaration and clearance of the AIS defect condition. The contents within this counter is initially of the value 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 decremented, or kept at zero value, whenever the Receive DS3 Framer block detects a non-AIS type M-frame. The Receive DS3 Framer block will declare the AIS Defect Condition if this counter reaches the value of 63 M-frames or greater. Explained another way, the AIS defect 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 reaches the number 63, then the Receive DS3 Framer block will do all of the following:

- It will set Bit 7 (AIS Defect Declared) within the Receive DS3 Configuration and Status Register to "1", as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AIS Defect Declared	LOS Defect Declared	Idle Condition Detected	OOF Defect Detected	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
1	0	0	1	0	0	0	0

- The Receive DS3 Framer block will also generate a Change in the AIS Defect Condition interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 5 (Change of AIS Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register to "1", as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

Clearing the AIS Defect Condition

The Receive DS3 Framer block will clear the AIS defect condition when the following expression is true.

NAIS - NVALID $\geq$ 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 Defect Condition.

The Receive DS3 Framer block will indicate that it is clearing the AIS defect by:

- Setting Bit 7 (AIS Defect Declared) within the Receive DS3 Configuration and Status Register to "0", as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AISDefect Declared	LOS Defect Declared	Idle Condition Detected	OOF Defect Declared	Unused	Framing with Valid P-Bits	F-SyncAlgo	M-SyncAlgo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The Receive DS3 Framer block will also generate the Change in AIS Defect Condition interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 5 (Change of AIS Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

4.3.2.5 DECLARING AND CLEARING THE DS3 IDLE PATTERN

The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the DS3 Idle condition indicator, as described below.

4.3.2.5.1 Declaring the Idle Condition

The Receive DS3 Framer block will identify and declare an Idle Condition if it receives a sufficient number of M-Frames within the DS3 data stream 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 contains, within its circuitry, an Up/Down Counter that supports the assertion and clearance of the Idle Condition. The counter begins with the value of 0x00 upon power up or reset. The counter is incremented anytime 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. The Receive DS3 Framer block will declare an Idle Condition if this counter reaches the value of 63 M-frames or greater. 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 Idle Patterns

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:

- Set Bit 5 (RxIdle of the Receive DS3 Configuration and Status Register, as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	1	0	0	0	0	0

- The Receive DS3 Framer block will also generate a Change in Idle Status interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 4 (Idle Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address =0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of DS3 Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

4.3.2.5.2 Clearing the Idle Condition

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". The Receive DS3 Framer block will indicate that it is clearing the Idle Condition by:

- Setting Bit 5 of the (RxIdle Condition Declared) within the Receive DS3 Configuration and Status Register to "0" as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxAIS Defect Declared	RxLOS Defect Declared	RxIdle Condition Declared	RxOOF Defect Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The Receive DS3 Framer block will also generate the Change in Idle Condition interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 4 (Change of Idle Condition Interrupt Status), within the Receive DS3 Interrupt Status Register to "1", as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

4.3.2.6 DECLARING AND CLEARING THE FERF INDICATOR

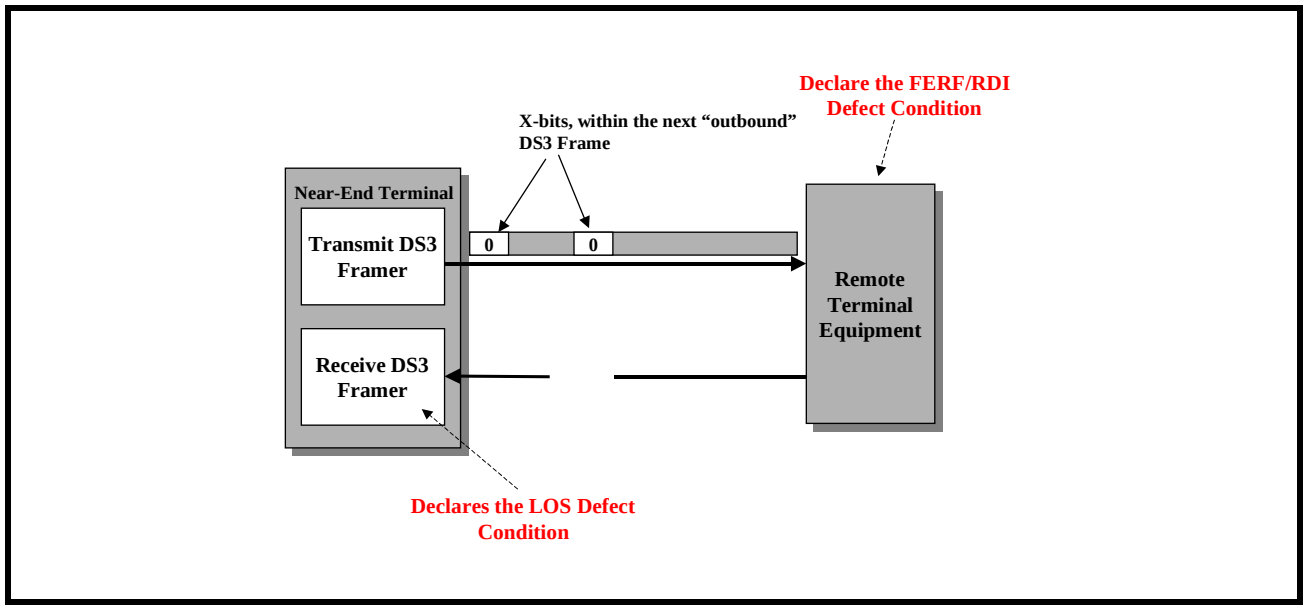
The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the FERF indicator, as described below.

4.3.2.6.1 Declaring the FERF (Far-End Receive Failure) Defect condition

The Receive DS3 Framer block will declare the FERF (Far-End-Receive Failure) or the DS3 RDI (Remote Defect Indicator) defect condition, if it starts to receive DS3 frames with all of its X-bits set to "0".

Recall, that back in Section 4.2.5.4, we described how one can configure the Transmit DS3/E3 Framer block to automatically transmit the FERF indicator to the remote terminal, anytime and for the duration that the Near-End Corresponding Receive DS3/E3 Framer Block declares either the LOS, LOF/OOF or AIS defect condition. **Figure 94** recaps some of this discussion by presenting a figure that depicts the Transmit DS3/E3 Framer block automatically transmitting the FERF indicator to the remote terminal equipment by setting the X-bits within each outbound DS3 frame, to "0" because the Near-End Corresponding Receive DS3 Framer block was declaring the LOS condition.

FIGURE 94. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



In Section 4.2.5.4, we described how a given Terminal will generate the FERF/RDI indicator. In this Section, we will describe how a given Terminal that contains the XRT79L71 will respond to receiving the FERF/RDI indicator from the remote terminal equipment.

When the Receive DS3 Framer block declares the FERF defect condition in the incoming DS3 frames, then it will then do the following.

- It will assert Bit 4 (FERF Defect Declared) within the Receive DS3 Status Register, to "1", as depicted below.

Receive DS3 Status Register (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FERF Defect Declared	RxAIC	RxFEBE[2:0]		
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	1	0	0	0	0

This bit-field will remain asserted for the duration that the Receive DS3/E3 Framer block declares the FERF defect condition.

- The Receive DS3 Framer block will also generate a Change in FERF Defect Condition Status interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 3 (Change of FERF Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1", as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

4.3.2.6.2 Clearing the FERF Defect Condition

The Receive DS3 Framer block will clear the FERF Defect Condition if it ceases to receive DS3 frames, with all of their X bits set to "0". The Receive DS3 Framer block will indicate that it is clearing the FERF Defect Condition by:

- Setting Bit 4 (FERF Defect Declared) within the Receive DS3 Status Register to "0" as depicted below.

Receive DS3 Status Register (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FERF Defect Declared	RxAIC	RxFEBE[2:0]		
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

The Receive DS3 Framer block will also generate the Change in FERF Defect Condition interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 3 (Change of FERF Defect Condition Interrupt Status), within the Receive DS3 Interrupt Status Register to "1", as illustrated below.

Receive DS3 Interrupt Status Register (Address =0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

4.3.2.7 DETECTING P-BIT ERRORS

The Receive DS3/E3 Framer block has the responsibility for detecting and flagging the occurrences of P-bit Errors, as described below.

Processing at the Remote Terminal Equipment

As the remote terminal is generating and transmit the incoming DS3 data-stream to the local terminal equipment, it will compute the even parity of an entire DS3 frame. The results of this parity calculation will be inserted into the two P-bit fields, within the very next outbound DS3 data stream. The purpose of these P-bits

is to support Performance Monitoring and Error Detection within the DS3 data-stream, as it is transported from one terminal equipment to another.

Processing at the Local Terminal Equipment

The Receive DS3 Framer block will compute and verify the P-bits within each DS3 frame that it receives. If the Receive DS3 Framer block determines that the P-bits within a given DS3 frame are erred, then it will do the following.

- Generate the Detection of P-bit Error interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 0 (Detection of P-Bit Error Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

- It will increment the PMON P-bit/Parity Error Count Register once for each DS3 frame that is determined to have erred P-bits. The PMON P-bit/Parity Error Count Register is located at Address = 0x1154 and 0x1155. The bit-format for each of these registers is presented below.

PMON Parity/P-Bit Error Count Register - MSB (Address = 0x1154)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Parity/P-Bit Error Count Register - LSB (Address = 0x1155)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these Performance Monitor Register, please see Section 2.5.*

- It will also increment the One Second - P-bit/Parity Error Count - Accumulator Register once for each DS3 frame, with erred P-bits. detected. The One Second - P-Bit/Parity Error Count -Accumulator Register is located at Address = 0x1170 and 0x1171. The bit-format for this 16-bit register is presented below.

One Second - Parity Error Accumulator Register - MSB (Address = 0x1170)

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

One Second - Parity Error Accumulator Register - LSB (Address = 0x1171)

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

4.3.2.8 DETECTING CP-BIT ERRORS

The Receive DS3/E3 Framer block has the responsibility for detecting and flagging the occurrences of CP-bit Errors, as described below.

Processing at the Remote Terminal Equipment

As the remote terminal is generating and transmit the incoming DS3 data-stream to the local terminal equipment, it will compute the even parity of an entire DS3 frame. The results of this parity calculation will be inserted into the three CP-bit fields, within the very next outbound DS3 data stream. The purpose of these CP-bits is to support Path Performance Monitoring and Error Detection of the DS3 data-stream.

Processing at the Local Terminal Equipment

The Receive DS3 Framer block will compute and verify the CP-bits within each DS3 frame that it receives. If the Receive DS3 Framer block determines that the CP-bits within a given DS3 frame are erred, then it will do the following.

- Generate the Detection of CP-bit Error interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 7 (Detection of CP-Bit Error Interrupt Status), within the Receive DS3 Interrupt Status Register, to "1" as illustrated below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
1	0	0	0	0	0	0	0

- It will increment the PMON CP-bit/Parity Error Count Register once for each DS3 frame that is determined to have a CP-bit error. The PMON CP-bit/Parity Error Count Register is located at Address 0x1158 and 0x1159. The bit-format for these registers is presented below.

PMON CP-Bit Error Count Register - MSB (Address = 0x1158)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_CP-Bit_Error_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON CP-Bit Error Count Register - LSB (Address = 0x1159)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_CP-Bit_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these "Performance Monitor" Registers please see Section 2.5*

- It will also increment the One Second - CP-bit/Parity Error Count - Accumulator Register once for each DS3 frame that is determined to have a CP-bit error. The "One Second - CP-Bit/Parity Error Count -Accumulator Register is located at Address 0x1172 and 0x1173.

One Second - CP Bit Error Accumulator Register - MSB (Address = 0x1172)

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

One Second - CP Bit Error Accumulator Register - LSB (Address = 0x1173)

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

NOTE: *The Near-End Transmit DS3/E3 Framer block, within this particular XRT79L71, will automatically be configured to set the three FEBE bit-fields to some value other than [1, 1, 1] in order to indicate an erred condition for each time in which the Receive DS3 Framer block receives a DS3 frame with erred CP-bits. Please see Section 4.2.5.6 for more information on this transmission of the FEBE indicator.*

4.3.2.9 DETECTING CHANGES IN THE AIC BIT

As mentioned earlier in this Data Sheet, the AIC (Application Identification Channel) Bit resides within the third C-bit, within F-Frame # 1 (as depicted in **Figure 14**). The purpose of the AIC bit within the DS3 data-stream is to permit a given Terminal Equipment to determine, though not conclusively, if it receiving a DS3 signal is of the M13/M23 framing format or is of the C-bit Parity framing format.

For C-bit Parity Applications, a given Transmitting DS3 Terminal Equipment will set the AIC bit-field to "1". However, for Channelized M13/M23 applications, the AIC bit-position within a given DS3 frame is simply a C-bit that is either set to "0" or "1" in order to denote stuff opportunities that either were or were not taken whenever this particular DS3 signal was created by multiplexing the seven (7) lower tributary DS2 signals.

Therefore, if a given DS3 Terminal Equipment receives a DS3 signal, in which the AIC bit is set to "0", then the Terminal Equipment can definitely conclude that it is receiving a DS3 signal that is of the M13/M23 Framing format. However, if a given DS3 Terminal Equipment receives a DS3 signal in which the AIC bit is always set to "1" then the Terminal Equipment can largely assume, but not definitely conclude that it is receiving a DS3 signal that is of the C-bit Parity Framing format.

Determining the value of the AIC Bit, within a given DS3 data-stream

The user can determine the value of the AIC bit, within a given incoming DS3 data-stream by reading out the state of Bit 3 (RxAIC) within the Receive DS3 Status Register, as depicted below.

Receive DS3 Status Register (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FERF Defect Declared	RxAIC	RxFEBE[2:0]		
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	X	0	0	0

If RxAIC is set to "0" then the user can conclude that the Receive DS3 Framer block is currently receiving a DS3 signal that is of the M13/M23 Framing format. Conversely, if RxAIC is set to "1", then the user can conclude that the Receive DS3 Framer block is mostly likely receiving a DS3 signal that is of the C-bit Parity Framing format.

Tracking Changes within the AIC Bit

If the Receive DS3 Framer block were to suddenly receive a DS3 data-stream that now has a different AIC value, and if this change in AIC value persists for at least two DS3 frame periods, then the following will happen.

- The Receive DS3 Framer block will update the contents of Bit 3 (RxAIC) within the Receive DS3 Status Register.

Receive DS3 Status Register (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FERF Defect Declared	RxAIC	RxFEBE[2:0]		
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	X	0	0	0

- The Receive DS3 Framer block will generate the Change of AIC State Interrupt. The Receive DS3 Framer block will indicate that it is generating this interrupt by asserting the Interrupt Request output pin by toggling it "Low" and by setting Bit 2 (Change of AIC State Interrupt Status) within the Receive DS3 Interrupt Status Register, to "1" as depicted below.

Receive DS3 Interrupt Status Register (Address = 0x1113)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Detection of CP Bit Error Interrupt Status	Change of LOS Defect Condition Interrupt Status	Change of AIS Defect Condition Interrupt Status	Change of Idle Condition Interrupt Status	Change of FERF Defect Condition Interrupt Status	Change of AIC State Interrupt Status	Change of OOF Defect Condition Interrupt Status	Detection of P-Bit Error Interrupt Status
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

4.3.2.10 DETECTING FEBE (FAR-END BLOCK ERROR) EVENTS

The Receive DS3/E3 Framer block has the responsibility for detecting and tallying the number of times that it receives disturbed FEBE (Far-End-Block Error) indicators from the remote terminal equipment as described below.

Each DS3 frame consists of three FEBE bit-fields. The remote terminal equipment which is generating the incoming DS3 data-stream will set the FEBE bit-fields to values that indicate whether or not the remote terminal is experiencing any F, M or CP-bit errors. If the remote terminal is currently not experiencing any errors, then it will set the all of the FEBE bit-fields, within each outbound DS3 data stream to "1". Hence, the FEBE value for an un-erred condition is "1, 1, 1".

Conversely, if the remote terminal is detecting an F, M or CP-bit errors, then it will proceed to set the FEBE bit-fields within each outbound DS3 data stream to some value other than "1, 1, 1". The important thing to note is that the FEBE value is a reflection of the receive condition of the remote terminal equipment, not the local terminal equipment.

NOTES:

1. *The remote terminal equipment will set the FEBE bit-fields to some value other than [1, 1, 1], within a given outbound DS3 frame, each time the Near-End Corresponding Receive DS3 Framer block has received a DS3 frame with either a CP-bit or Framing bit error.*
2. *The FEBE values within the incoming DS3 data-stream is a reflection of the receive conditions of the remote terminal equipment, not the local terminal equipment.*

If the Receive DS3 Framer block receives any DS3 frames that contain FEBE bits, with values other than "1, 1, 1", then it will increment the PMON FEBE Event Count Register. The Receive DS3 Framer block will increment this register once, for each DS3 frame that it receives, in which the FEBE values are not set to "1, 1, 1". The bit-format and address locations of these registers are presented below.

PMON FEBE Event Count Register - MSB (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FEBE Event Count Register - LSB (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these "Performance Monitor" Registers please see Section 2.5.*

4.3.2.11 DETECTING FRAMING BIT ERRORS

The Receive DS3/E3 Framer block has the responsibility for detecting and flagging the occurrences of Framing (e.g., F or M) bit errors as described below.

While the Receive DS3 Framer block is operating in the Frame Maintenance Mode, it will continue to check for valid F and M bits within the incoming DS3 data-stream. If the Receive DS3 Framer block detects any errors in the F or M bits, then it will do the following.

- Increment the PMON Framing Bit/Byte Error Count Register once for each DS3 frame that is determined to contain an erred F or M bits. The PMON Framing Bit/Byte Error Count Register is located at Address = 0x1152/0x1153. The bit-format of these registers is presented below.

PMON Framing Bit/Byte Error Count Register - MSB (Address = 0x1152)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Upper_Byte[7:0]							
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 = 0x1153)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

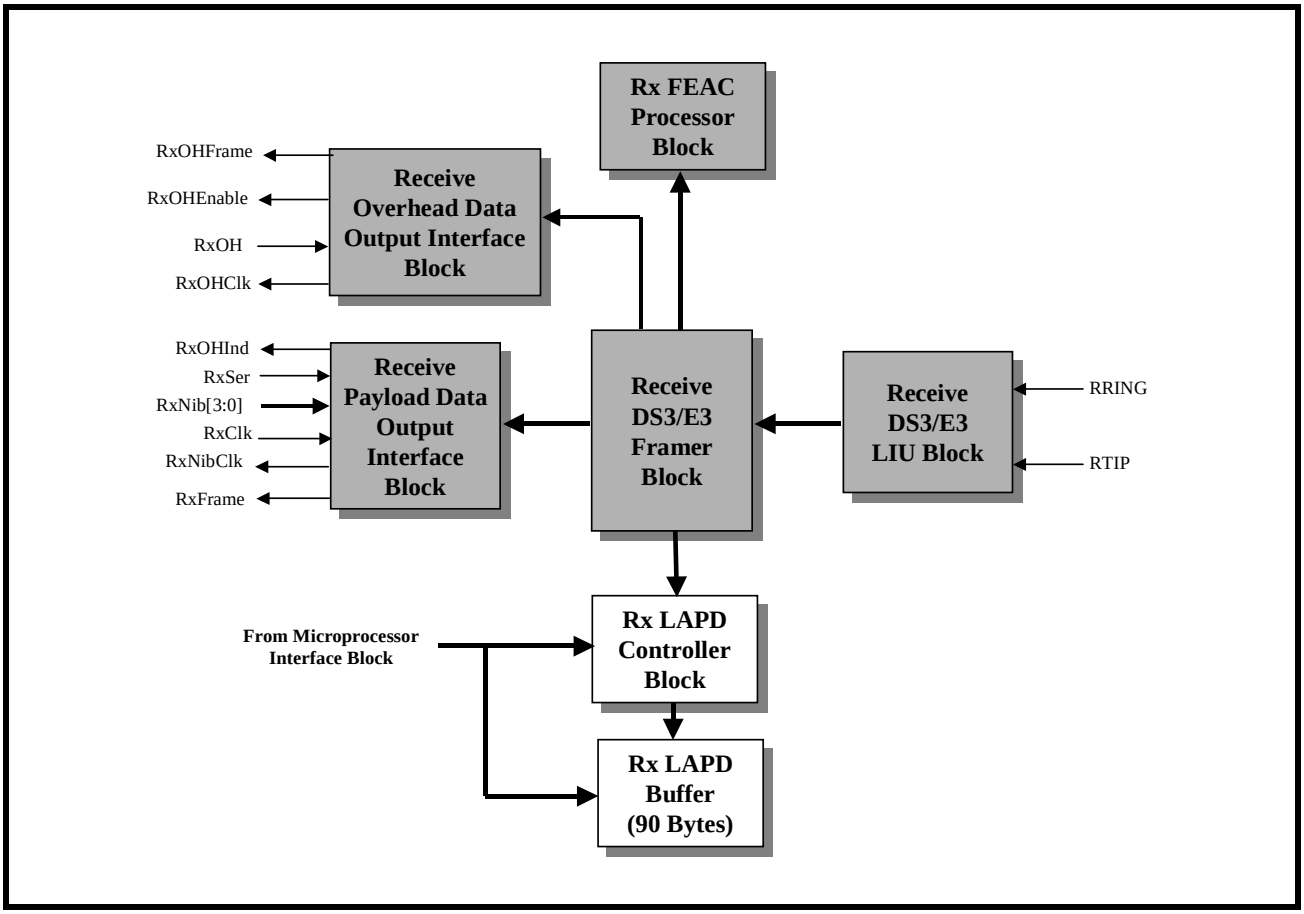
NOTE: *For instructions on how to read out these "Performance Monitor" Registers please see Section 2.5.*

4.3.2.12 Receive DS3/E3 Framer Block Interrupts

4.3.3 RECEIVE LAPD CONTROLLER BLOCK

The Receive LAPD Controller Block is the four functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 95** presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive LAPD Controller block highlighted.

FIGURE 95. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE LAPD CONTROLLER BLOCK HIGHLIGHTED)



The Receive LAPD Controller block consists of the following sections.

- The Receive LAPD Message Buffer
- The Receive LAPD Controller

The Receive LAPD Message Buffer

The purpose of the Receive LAPD Message Buffer is to store the contents of LAPD/PMDL Messages that have been received by the Receive LAPD Controller block. The Receive LAPD Message Buffer will serve as a temporary storage location of these incoming LAPD/PMDL Messages until they can be read out by the Microprocessor. the Receive LAPD Message Buffer is actually a 90 byte FIFO that is located at Address Location 0x11C0 within the XRT79L71 address space.

The Receive LAPD Controller

The Receive LAPD Controller permits the user to receive path maintenance data link (PMDL) message from the remote terminal equipment via the inbound DS3 frames. In this case, the PMDL Message is extracted out of the 3 DL bit-fields of F-Frame # 5 within each incoming DS3 M-frame. The on-chip Receive LAPD Controller is capable of receiving both standard and non-standard PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the Receive LAPD Message buffer), to store the contents of newly received PMDL Messages. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in **Figure 96**.

FIGURE 96. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		
76 OR 82 OR ANY-SIZE BYTES OF INFORMATION (PAYLOAD)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

Where, for standard Bellcore GR-499-CORE applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the Receive LAPD Controller is not currently receiving a LAPD Message, it will instead be receiving a continuous stream of Flag Sequence bytes via the DL bits within each inbound DS3 frame.

SAPI - Service Access Point Identifier

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for standard Bellcore GR-499-CORE applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For standard Bellcore GR-499-CORE applications the user must use the Control byte the value 0x03. Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The Information Payload is the 76 bytes, 82 bytes or any number of bytes of data (e.g., the PMDL Message) that the user has written into the on-chip Receive LAPD Message buffer which is located at Address 0x11C0.

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, $x^{16} + x^{12} + x^5 + 1$. Afterwards, this FCS value is inserted into the two-octet FCS value position, within the LAPD Message frame. The Receive LAPD Controller block will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see Section 4.2.4 on how the Transmit LAPD Controller block computes and inserts the FCS values into its outbound LAPD Message frames.

Operation of the Receive LAPD Controller

As mentioned earlier, the Receive LAPD Controller permits the user to receive either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

The procedure for receiving these types of LAPD Messages is presented below.

4.3.3.1 Receiving Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) configure the Receive LAPD Controller block to extract out the contents of the incoming PMDL Messages from the incoming DS3 data-stream, and (2) to properly read out the contents of a newly received message which is being stored in the Receive LAPD Message buffer, by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the C-Bit Parity Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "1" and that Bit 2 (Frame Format) is set to "0" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	1	X	0	X	0	X	X

STEP 2 - Enable the Receive LAPD Controller

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive DS3 LAPD Control Register to "1" as depicted below.

Receive DS3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	0	0

NOTES:

1. *For normal operation, it is imperative that the user also make sure that Bit 7 (Receive LAPD Any) within this register is set to "0".*
2. *Once the user executes the above-mentioned step, then the Receive LAPD Controller will begin to extract out the contents of any incoming LAPD/PMDL Message that is being transported via the DL bits within the incoming DS3 data-stream. In most cases, the Receive LAPD Controller block will simply begin to receive the Flag Sequence octet which is originating from the remote terminal.*

STEP 3 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming DS3 data-stream, then it will assert Bit 0 (Flag Present) within the Receive DS3 LAPD Status Register, as depicted below.

Receive DS3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 4 - Enable the Receive LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the Receive LAPD Interrupt is actually a three-step process.

STEP 4a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 4b - Enable the Receive DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the Receive DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 4c - Enable the Receive LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the Receive DS3 LAPD Control Register, as depicted below.

Receive DS3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 5 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 6 - Service the Receive LAPD Interrupt

Please see Section 4.3.3.3 on how to service the Receive LAPD Interrupt.

STEP 7 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive DS3 LAPD Status Register as depicted below.

Receive DS3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an un-erred manner (e.g., there are no FCS errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an erred manner.

NOTE: The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the RxFCS Error bit-field, prior to processing a given message.

STEP 8 - Identify the Type and Size of Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of Bits 4 and 5 (RxLAPDType[1:0]) within the Receive DS3 LAPD Status Register, as depicted below.

Receive DS3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	0	1	0

Table 30 presents the relationship between the contents within Bits 4 and 5 (RxLAPDType[1:0]) and the corresponding LAPD Message that is residing within the Receive LAPD Message Buffer.

TABLE 30: THE RELATIONSHIP BETWEEN THE CONTENTS WITHIN BITS 4 AND 5 (RXLAPDTYPE[1:0]) AND THE TYPE OF LAPD/PMDL MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER

RXLAPDTYPE[1:0]	TYPE OF LAPD/PMDL MESSAGE RESIDING IN THE RECEIVE LAPD MESSAGE BUFFER	SIZE OF MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER (INFORMATION PAYLOAD/TOTAL MESSAGE SIZE)
00	CL Path Identification Message	76 Bytes/82 Bytes
01	Idle Signal Identification	76 Bytes/82 Bytes
10	Test Signal Identification	76 Bytes/82 Bytes
11	ITU-T Path Identification	82 Bytes/88 Bytes

Table 30 indicates that if the value within the RxLAPDType[1:0] bit-fields are set to [1, 1], then the size of total message residing within the Receive LAPD Message Buffer is 88 bytes, and that the size of the Information Payload within these 88 bytes of data is 82 bytes. Likewise, this table also indicates that if the value within the RxLAPDType[1:0] bit-fields are set to some value other than [1, 1], then the size of the total message residing within the Receive LAPD Message Buffer is 82 bytes, and that the size of the Information Payload within these 82 bytes of data is 76 bytes.

The Relationship between the Total LAPD/PMDL Message Size and Information Payload Size

The Relationship between the size of the total LAPD/PMDL Message, and the size of the Information Payload, is best explained by presenting the byte-format of the LAPD Message below in Figure 97.

FIGURE 97. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		

FIGURE 97. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)
76 OR 82 OR ANY-SIZE BYTES OF INFORMATION (PAYLOAD)
FCS - MSB
FCS - LSB
Flag Sequence (8-bits)

Figure 97 indicates that the complete LAPD/PMDL Message will consist of an information payload of either 76 or 82 bytes, along with a total of six overhead bytes consisting of the Flag Sequence, SAPI, TEI, Control and the two FCS bytes.

NOTE: *When receiving and processing the Standard 76 or 82-byte type of LAPD Messages, then the data residing within the Receive LAPD Message will be exactly of the byte-format, as presented above in* **Figure 97**.

Why determining the Size of the Incoming LAPD Message is important

If the Microprocessor has learned based upon reading out the values of the RxLAPDType[1:0] bit-fields that the size of the total LAPD/PMDL Message is 88 bytes, then the Microprocessor knows when it comes time to read out the contents of the Message, residing within the Receive LAPD Message Buffer that it can read out and process the contents of 88-bytes out of 90 bytes within this Buffer. Conversely, if the Microprocessor has learned that the size of the total LAPD/PMDL Message is only 82 bytes, then the Microprocessor must know that it can only read out and process the first 82 bytes of data within the Receive LAPD Message Buffer. The last eight bytes within the Buffer are simply junk bytes and have no value.

STEP 9 - Read out the contents of the Receive LAPD Message Buffer

The instructions that follow were written with the assumption that the user only wishes to extract out the Information Payload bytes, from the complete LAPD Message that is residing within the Receive LAPD Message Buffer. Whenever the user wishes to read out the contents of newly received PMDL/LAPD Messages from the Receive LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that will be presented below.

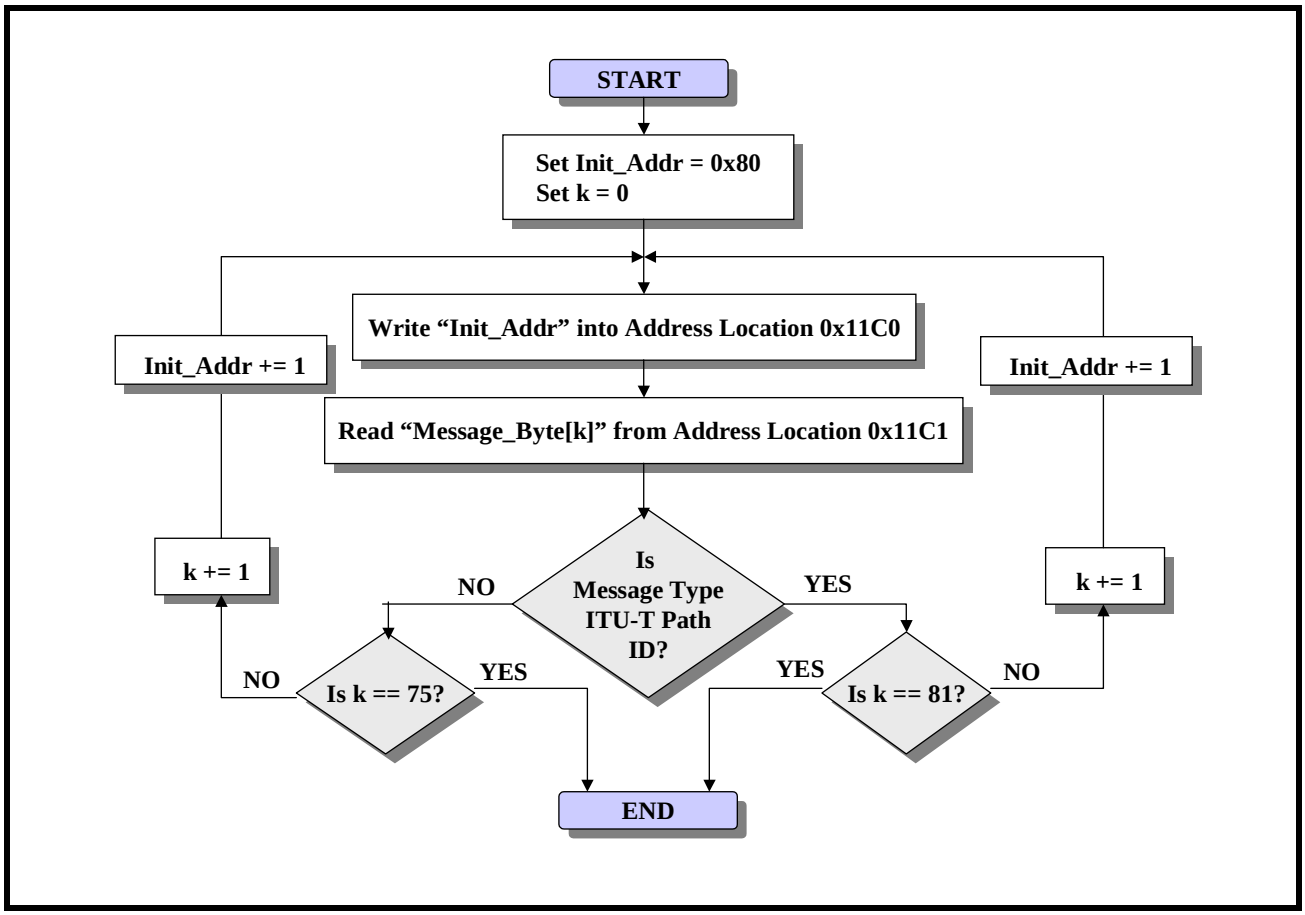
In order to begin the process of reading out the contents of the incoming PMDL Message, then the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Receive LAPD Message Buffer. The approach that one should use to access the Transmit LAPD Message Buffer is presented in Section 4.2.4.1.

The exact approach that one should use, when reading out the contents of the newly received PMDL Message from the Receive LAPD Message buffer is presented in the flow-chart below.

FIGURE 98. FLOW-CHART DEPICTING AN APPROACH THAT CAN BE USED FOR READING OUT THE CONTENTS OF A NEWLY RECEIVED LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER



NOTE: About **Figure 98***:*

The answer to Is Message Type ITU-T Path? Decision Diamond within the flow-chart was obtained during STEP 8 (see above).

STEP 10 - Remove Header and FCS Bytes from this newly received and read out PMDL/LAPD Message

As mentioned above, the byte of the data, residing within the Receive LAPD Message buffer, will be as is presented within **Figure 98**. Therefore, if the user is only interested in processing the Information Payload portion of this LAPD/PMDL Message, then the user must remove these additional bytes from this block of data, before further processing.

4.3.3.2 Receiving Non-Standard Variable Length (e.g., up to 82 bytes) LAPD Messages)

The user can (1) enable the Receive LAPD Controller, and (2) read out a non-Standard incoming PMDL/LAPD message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the C-bit Parity Framing Format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "1" and that Bit 2 (Frame Format) is set to "0" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	Timing Reference Select [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	1	X	0	X	0	X	X

STEP 2 - Enable the Receive LAPD Controller

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive DS3 LAPD Control Register to "1" as depicted below.

Receive DS3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	0	0

STEP 3 - Configure the Receive LAPD Controller to receive a non-standard size LAPD Message

This is accomplished by setting Bit 7 (RxLAPD Any) within the Receive DS3 LAPD Control Register, to "1", as depicted below.

Receive DS3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
1	0	0	0	0	1	0	0

STEP 4 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming DS3 data-stream, then it will assert Bit 0 (Flag Present) within the Receive DS3 LAPD Status Register, as depicted below.

Receive DS3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 5 - Enable the Receive LAPD Interrupt (Optional).

This step is optional. However, if this step is executed the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the Receive LAPD Interrupt is actually a three-step process.

STEP 5a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 5b - Enable the Receive DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/ E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the Receive DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 5c - Enable the Receive LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the Receive DS3 LAPD Control Register, as depicted below.

Receive DS3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 6 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 7 - Service the Receive LAPD Interrupt

Please see Section 4.3.3.3 on how to service the Receive LAPD Interrupt.

STEP 8 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive DS3 LAPD Status Register as depicted below.

Receive DS3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an un-erred manner (e.g., there are no FCS errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an erred manner.

NOTE: *The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the RxFCS Error bit-field, prior to processing a given message.*

STEP 9 - Determine the Size of the Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of the Receive LAPD Byte Count Register, as depicted below.

Receive LAPD Byte Count Register (Address = 0x1184)

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

NOTES:

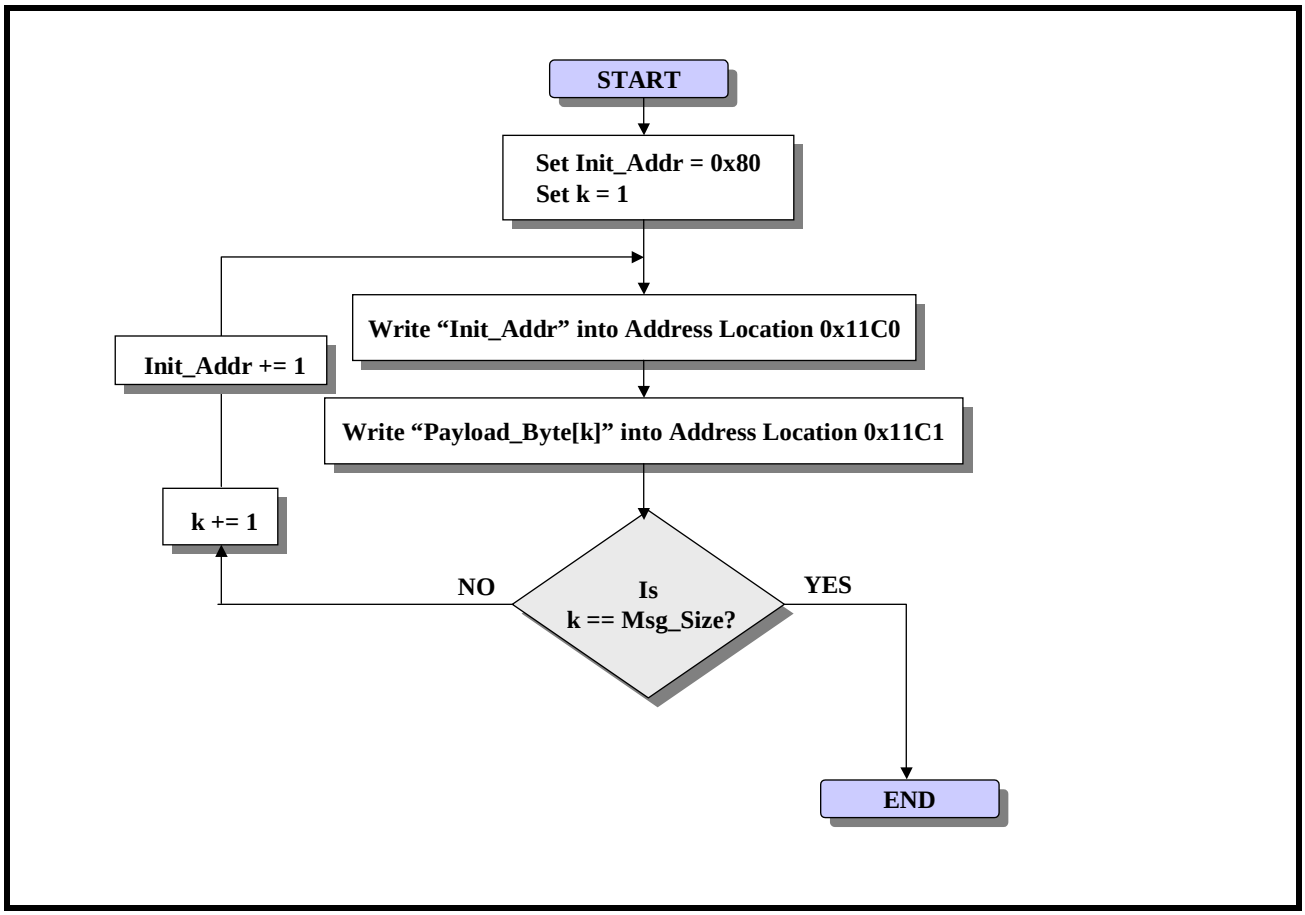
1. *The Receive LAPD Byte Count Register will have the size of the newly received LAPD/PMDL Message, in terms of bytes.*

2. *This register is only active if the Receive LAPD Controller is receiving a non-standard LAPD/PMDL Message.*

STEP 10 - Read out the contents of the Receive LAPD Message Buffer

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 99. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO READING OUT THE CONTENTS OF THE NEWLY RECEIVE LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER



NOTE: *About* **Figure 99***:*

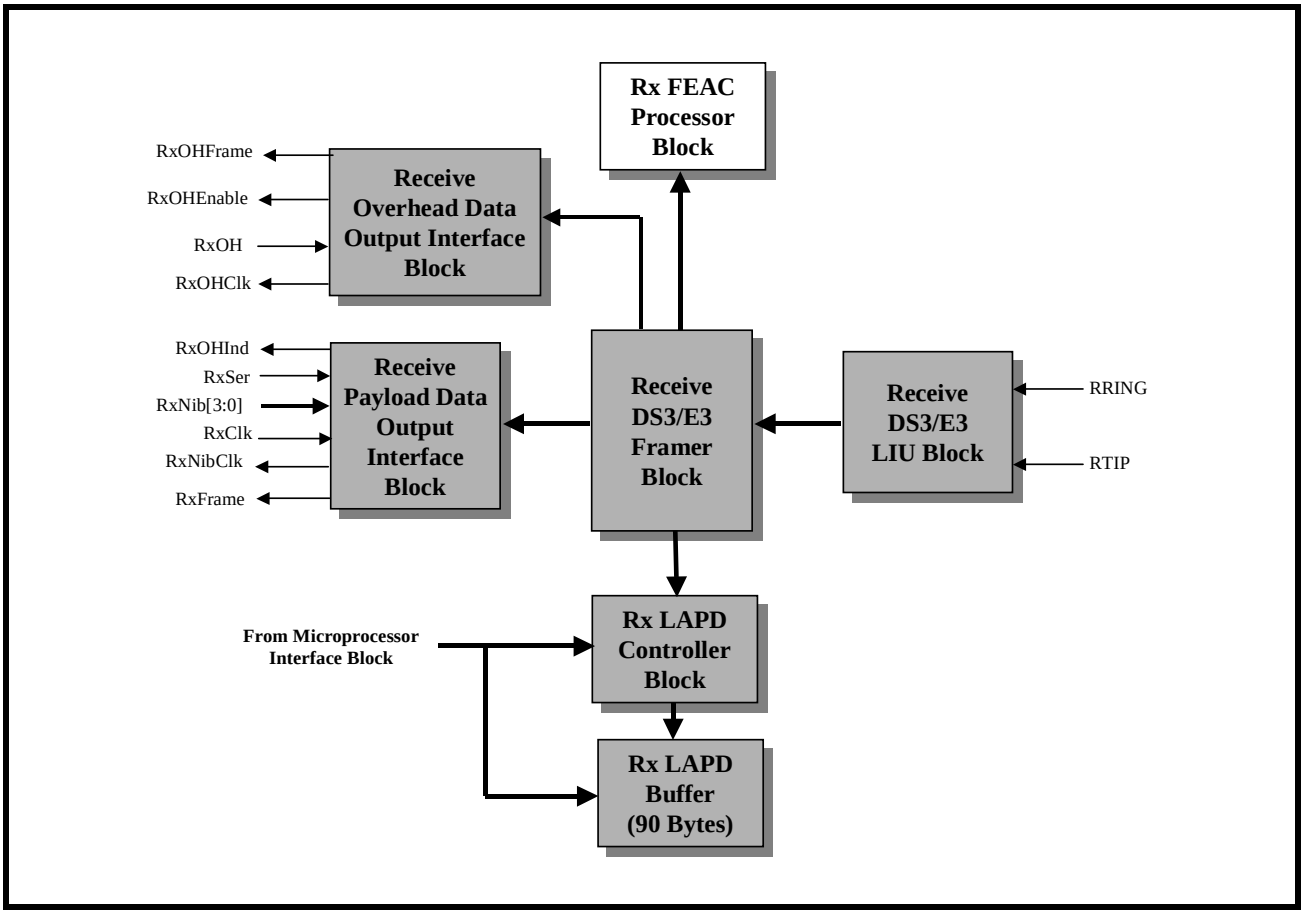
The answer to the Is k=Msg_Size? Decision Diamond within the flow-chart was obtained during STEP 9 (see above).

4.3.3.3 Receive LAPD Controller Block Interrupts

4.3.4 RECEIVE FEAC CONTROLLER BLOCK

The Receive FEAC Controller Block is the third functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 100** presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive DS3/E3 Framer block highlighted.

FIGURE 100. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE FEAC CONTROLLER BLOCK HIGHLIGHTED)



If the Receive DS3/E3 Framer block is operating in the DS3, C-bit Parity Mode, then the FEAC bit-field within the DS3 Frame can be used to receive FEAC (Far End Alarm and Control) messages (See **Figure 101**). 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:

FIGURE 101. THE BIT-FORMAT OF THE FEAC MESSAGE

	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 (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 x can represent 64 distinct messages, of which 43 have been defined in the standards.

The Receive FEAC Controller block frames and validates the incoming FEAC data from the remote Transmit FEAC Controller block via the received FEAC channel. Additionally, the Receive FEAC Controller block will write the Received FEAC code words into an 8 bit Receive DS3 FEAC register. Framing is performed by looking for two "0s" spaced 6 bits apart preceded by 8 "1s". The Receive FEAC Controller block contains two registers that support FEAC Message Reception.

- Receive DS3 FEAC Register (Address = 0x1116)

- Receive DS3 FEAC Interrupt Enable/Status Register (Indirect Address = 0x1117)

The Receive FEAC Controller Block generates an interrupt upon validation and removal of the incoming FEAC Code words.

4.3.4.1 Operation of the Receive DS3 FEAC Controller Block

The Receive FEAC Controller block will validate or remove FEAC code words that it receives from the remote Transmit FEAC Controller block. The FEAC Code Validation and Removal functions are described below.

FEAC Code Validation

When the remote Transmit DS3 Framer block wishes to send a FEAC message to the local Receive DS3 Framer block, the remote Transmit DS3 Framer block will transmit this 16 bit message, repeatedly for a total of 10 times. The Receive FEAC Controller block will frame to this incoming FEAC Code Message, and will attempt to validate this message. Once the Receive FEAC Controller block 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 Controller will then inform the µP/µC of this Receive FEAC validation event by generating a Rx FEAC Valid interrupt and asserting the FEAC Valid and the Receive FEAC Valid Interrupt Status Bits in the Receive DS3 Interrupt Enable/Status Register, as depicted below. The Bit Format of the Receive DS3 FEAC Register is presented below.

Receive DS3 FEAC Interrupt Enable/Status Register (Address = 0x1117)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Interrupt Enable	RxFEAC Valid Interrupt Status
R/O	R/O	R/O	R/O	R/W	RUR	R/W	RUR
0	0	0	1	0	0	1	1

Receive DS3 FEAC Register (Address = 0x1116)

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

The purpose of generating an interrupt to the µP/µC, upon FEAC Code Word Validation is to inform the µP/µC that the Receive FEAC Controller block has a newly received FEAC message that needs to be read. The µC/µP should then read-in this FEAC code word from the Receive DS33 FEAC Register (Address = 0x1116).

FEAC Code Removal

After the 10th transmission of a given FEAC code word, the remote Transmit DS3 Framer may start to transmit a different FEAC code word. When the Receive FEAC Controller detects this occurrence, it must Remove the FEAC codeword that is presently residing in the Rx DS3 FEAC Register. The Receive FEAC Controller Block 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 Controller Block will inform the µP/µC of this removal event by generating a Rx FEAC Removal interrupt, and asserting the RxFEAC Remove Interrupt Status bit in the Receive DS3 Interrupt Enable/Status Register, as depicted below.

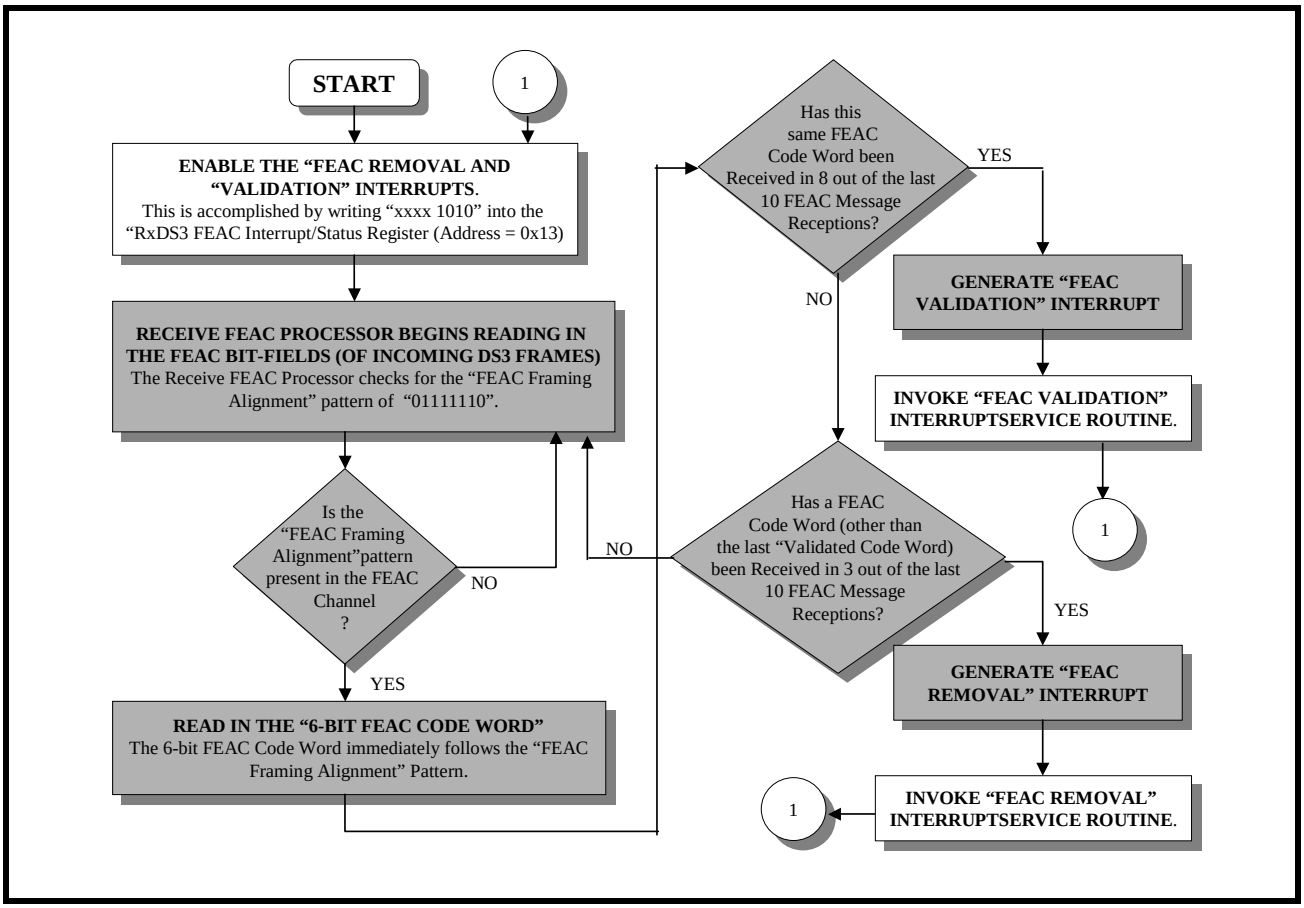
Receive DS3 FEAC Interrupt Enable/Status Register (Address = 0x1117)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			FEAC Valid	RxFEAC Remove Interrupt Enable	RxFEAC Remove Interrupt Status	RxFEAC Valid Inter-rupt Enable	RxFEAC Valid Inter-rupt Status
R/O	R/O	R/O	R/O	R/W	RUR	R/W	RUR
0	0	0	0	1	1	1	0

Additionally, the Receive FEAC Controller block will also denote the removal event by setting the FEAC Valid bit-field (Bit 4), within the Receive 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 4.3.4.2. **Figure 102** presents a flow diagram depicting how the Receive FEAC Controller Block functions.

FIGURE 102. FLOW DIAGRAM DEPICTING HOW THE RECEIVE FEAC CONTROLLER BLOCK FUNCTIONS

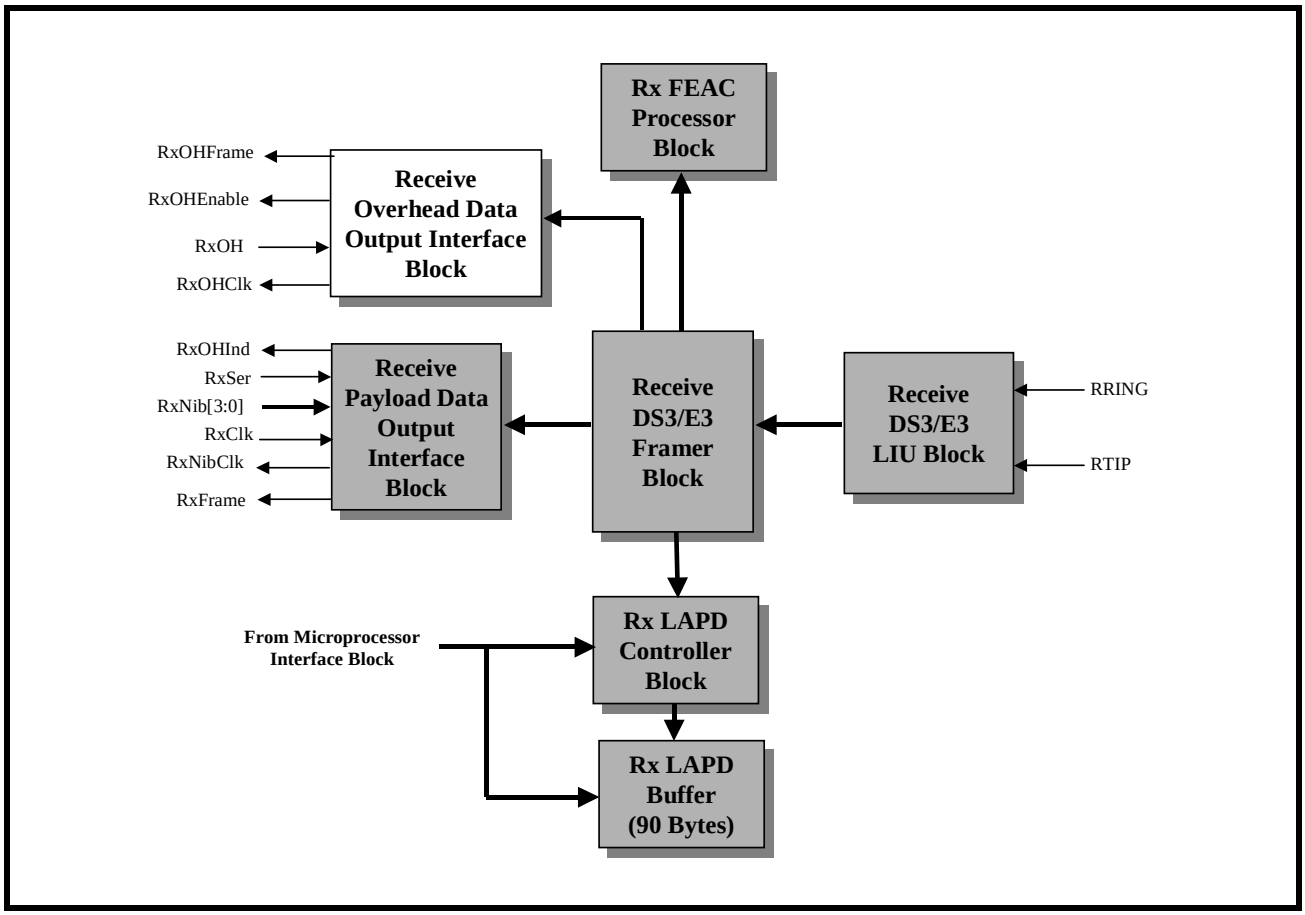


4.3.4.2 Receive FEAC Controller Block Interrupts

4.3.5 RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

The Receive Overhead Data Output Interface block is the fifth functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 103** presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive Overhead Data Output Interface block highlighted.

FIGURE 103. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK HIGHLIGHTED)



Some Background Information

In order to fully understand the role of the Receive Overhead Data Output Interface, some background information needs to be discussed first.

As mentioned in Section 4.1, the DS3 frame consists of 4760 bits. Of these bits, 4704 bits are payload bits and the remaining 56 bits are overhead bits. The XRT79L71 has been designed to handle and process both the payload type and overhead type of bits for each DS3 frame. Within the XRT79L71, the Receive Payload Data Output Interface Block (which is discussed in considerable detail in Section 4.3.6) has been designed to output the payload data that has been extracted from the incoming DS3 data-stream to the System-Side Terminal Equipment. 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 XRT79L71 does not offer the user a means to shut off this transmission of overhead data. However, the Receive Overhead Data Output Interface block does provide the user with the appropriate output signals for an external Data-Link Layer or System-Side Terminal Equipment to sample and process these overhead bits

In order to accomplish this, the Receive Overhead Data Output Interface block has numerous output pins. **Table 31** presents a list and a brief definition of each of these pins.

TABLE 31: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
RxOH	C7	O	**Receive Overhead Data Output Interface block - Data Output pin:** The method to sample this output pin depends upon whether using Method 1 (see Section 4.3.5.1) or Method 2 (see Section 4.3.5.2) is used to extract data from the Receive Overhead Data Output Interface block, as described below and in the following sections. **If Method 1 is used:** The XRT79L71 outputs the overhead bits, within the incoming DS3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will output a given bit, upon the falling edge of RxOHClk. Hence, the System-Side Terminal Equipment should be designed or configured to sample the data, at this pin, upon the rising edge of RxOHClk. **If Method 2 is used:** The XRT79L71 outputs the overhead bits, within the incoming DS3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will assert the RxOHEnable output pin for one RxClk period whenever the data, residing on the RxOH output pin has become stable and is safe for sampling. In this case, the user should design or configure the System-Side Terminal Equipment to sample and latch the RxOH data upon the falling edge of RxClk coincident to whenever the RxOHEnable output pin is sampled "High". The XRT79L71 will always output the DS3 overhead bits via this output pin. There are no external input pins or register bits settings available that will disable this output pin.
RxOHClk	A7	O	**Receive Overhead Data Output Interface block - Clock Output pin:** This particular output pin is only used if Method 1 is employed to extract overhead data from the Receive Overhead Data Output Interface port (see Section 4.3.5.1 below). The XRT79L71 will output the overhead bits within the incoming DS3 data-stream, via the RxOH output pin, upon the falling edge of this particular clock signal. As a consequence, the System-Side Terminal Equipment should be designed or configured to sample the data, at this pin, upon the rising edge of RxOHClk. ***NOTE:*** *This output clock signal is always active.*

TABLE 31: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
RxOHFrame	A8	O	**Receive Overhead Data Output Interface block - Start of Frame Indicator Output pin:** The exact approach to used to sample this output pin depends upon whether Method 1 (see Section 4.3.5.1) or Method 2 (see Section 4.3.5.2) is used to extract data from the Receive Overhead Data Output Interface block, as described below and in the following sections. **If Method 1 is used:** The XRT79L71 will pulse this output pin "High" for one period of RxOHClk coincident to whenever the Receive Overhead Data Output Interface block outputs the very first overhead bit of the most recently received DS3 frame. This output pin will be "Low" at all other times. The Receive Overhead Data Output Interface block will update this output pin, upon the falling edge of RxOHClk. Hence, the System-Side Terminal Equipment should be designed or configured to sample the data, at this pin, upon the rising edge of RxOHClk. **If Method 2 is used:** The Receive Overhead Data Output Interface block will assert the RxOHEnable output pin for one RxClk period whenever the data, residing on the RxOH output pin has become stable and is safe for sampling. In this case, the user should design or configure the System-Side Terminal Equipment to sample and latch both the RxOH and the RxOHFrame output pins upon the falling edge of RxClk coincident to whenever the RxOHEnable output pin is sampled "High".
RxOHEnable	B7	O	**Receive Overhead Data - Enable Output Pin:** This particular output pin is only used if Method 2 is employed, (see Section 4.3.5.2 below). The Receive Overhead Data Output Interface will assert this signal (e.g., pulse it "High") for one RxClk period, coincident to whenever it is safe for the System-Side Terminal Equipment to sample the overhead bit that is being output via the RxOH output pin. This output pin will remain "Low" at all other times. If Method 2 is used, then design or configure the System-Side Terminal Equipment such that it will sample and latch the RxOH output from the XRT79L71, upon the falling edge of RxClk, coincident to whenever the RxOHEnable output pin is sampled "High".

The Two Methods of Extracting out Overhead Data from the Receive Overhead Data Output Interface block

There are two methods that can be used to extract the overhead data from the Receive Overhead Data Output Interface block. One Method is referred to as Method 1 or the RxOHClK Method, and the other method is referred to as Method 2 or the RxClk/RxOHEnable Method. Each of these methods is described in considerable detail below.

4.3.5.1 Operating the Receive Overhead Data Output Interface Block using Method 1 - The RxOHClk Method

This particular method is referred to as the RxOHClk method for the following reasons.

a. The System-Side Terminal Equipment will use the RxOHClk clock output signal from the Receive Overhead Data Output Interface block to sample and latch the overhead data via the RxOH output pin.

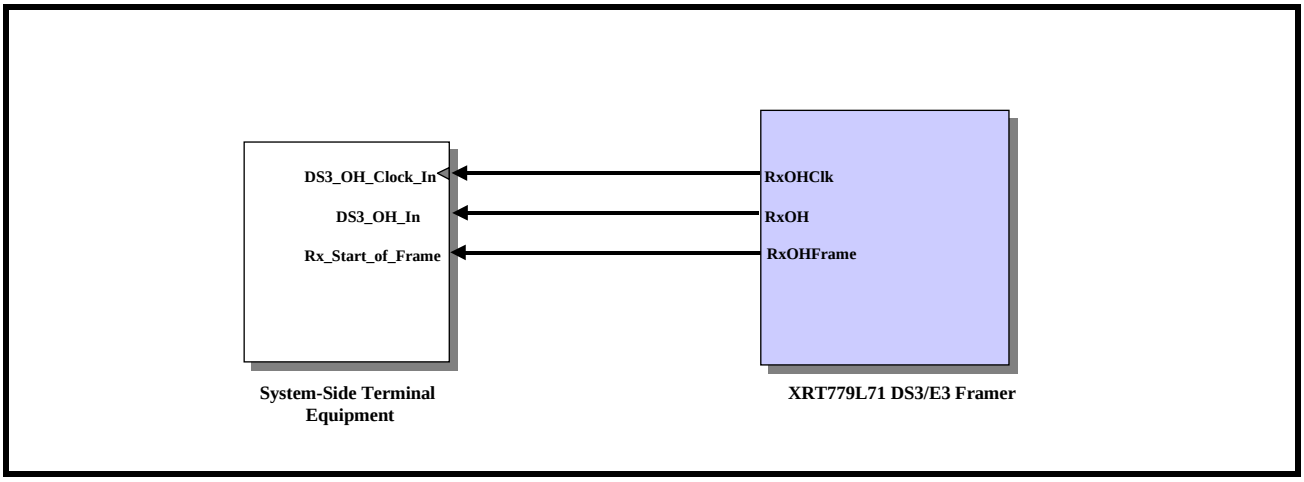
b. The Receive Overhead Data Output Interface block will update the data via the RxOH output pin upon the falling edge of the RxOHClk clock output signal.

If Method 1 is used, the System-Side Terminal Equipment will need to interface to the following Receive Overhead Data Output Interface pins.

- RxOH
- RxOHClk
- RxOHFrame

To use Method 1 the user must design their system such that the System-Side Terminal Equipment will be interfaced to the Receive Overhead Data Output Interface block in the manner as presented below in Figure 104.

FIGURE 104. ILLUSTRATION ON HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK WHEN USING METHOD 1



Method 1 Operation of the Receive Overhead Data Output Interface Block

To operate the Receive Overhead Data Output Interface block, design/configure the System-Side Terminal Equipment to continuously execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the state of the RxOHFrame output pin from the XRT79L71 upon the rising edge of the RxOHClk clock signal which is also output from the XRT79L71. Whenever the System-Side Terminal Equipment samples the RxOHFrame output pin "High", then it will know that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit within the most recently received DS3 frame via the RxOH output pin.

TASK # 2: As the System-Side Terminal Equipment' samples the RxOHFrame output signal, it must also keep track of the number of rising edges within the RxOHClk signal that have occurred since the last time RxOHFrame was sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bit is being output via the RxOH output pin, at any given RxOHClk period. When the System-Side Terminal Equipment knows which overhead bit is being output via the RxOH output pin during a particular RxOHClk clock period, then it can make sense of these overhead bits and process these overhead bits as appropriate.

Table 32 relates the number of rising clock edges, within the RxOHClk output signal, since the RxOHFrame output signal was sampled "High" to the DS3 Overhead Bit being output via the RxOH output pin of the Receive Overhead Data Output Interface block, for Method 1.

TABLE 32: THE RELATIONSHIP BETWEEN 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 PROCESSED BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN RXOHCLK, SINCE RXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
0 (RxOHClk Clock Edge is coincident with the RxOHFrame signal being sampled "High")	X Bit # 1
1	F1
2	AIC (C11)
3	F0
4	NA (C12)
5	F0
6	FEAC (C13)
7	F1
8	X Bit # 2
9	F1
10	UDL Bit # 1 (C21)
11	F0
12	UDL Bit # 2 (C22)
13	F0
14	UDL Bit # 3 (C23)
15	F1
16	P
17	F1
18	CP Bit # 1 (C31)
19	F0
20	CP Bit # 2 (C32)
21	F0
22	CP Bit # 3 (C33)
23	F1
24	P
25	F1
26	FEBE # 1 (C41)
27	F0
28	FEBE # 2 (C42)
29	F0

TABLE 32: THE RELATIONSHIP BETWEEN 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 PROCESSED BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN RXOHCLK, SINCE RXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
30	FEBE # 3 (C43)
31	F1
32	M0
33	F1
34	DL Bit # 1 (C51)
35	F0
36	DL Bit # 2 (C52)
37	F0
38	DL Bit # 3 (C53)
39	F1
40	M1
41	F1
42	UDL Bit # 4 (C61)
43	F0
44	UDL Bit # 5 (C62)
45	F0
46	UDL Bit # 6 (C63)
47	F1
48	M0
49	F1
50	UDL Bit # 7 (C71)
51	F0
52	UDL Bit # 8 (C72)
53	F0
54	UDL Bit # 9 (C73)
55	F1

4.3.5.2 Operating the Receive Overhead Data Input Interface block using Method 2 - The RxClk/ RxOHEnable Method

This particular method is referred to as the RxClk/RxOHEnable method for the following reasons.

a. The System-Side Terminal Equipment will use the RxOHEnable output pin from the Receive Overhead Data Output Interface block to keep track of which overhead bit is being processed by the Receive Overhead Data Output Interface via the RxOH output pin as any given time.

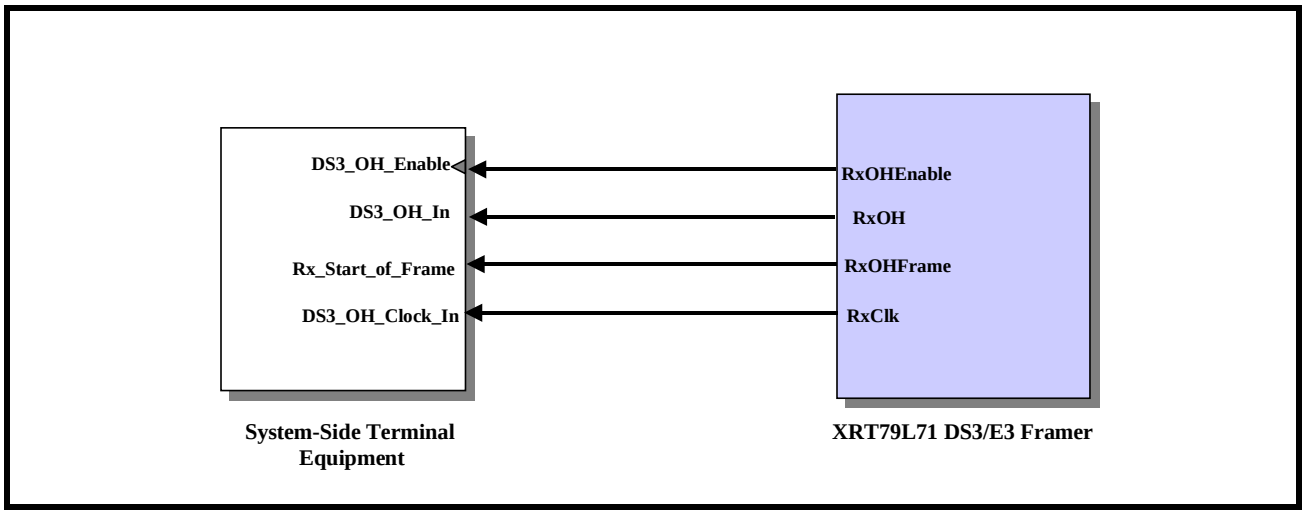
b. The System-Side Terminal Equipment must use the falling edge of RxClk in order to sample and latch the data residing on the RxOH output pin.

If the Method 2 is to be used then the System-Side Terminal Equipment will need to interface to the following Receive Overhead Data Output Interface pins.

- RxOH
- RxOHFrame
- RxOHEnable
- RxClk

To use Method 2 then the user must design their system such that the System-Side Terminal Equipment will be interfaced to the Receive Overhead Data Output Interface block, in a manner as presented below in Figure 105.

FIGURE 105. ILLUSTRATION ON HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK WHEN USING METHOD 2



Method 2 Operation of the Receive Overhead Data Output Interface Block

To operate the Receive Overhead Data Output Interface block per Method 2, design/configure the System-Side Terminal Equipment to continuously execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the states of the RxOHFrame, RxOH and the RxOHEnable output pins from the XRT79L71 upon the falling edge of RxClk clock output signal. The XRT79L71 will pulse the RxOHEnable output pin "High" for one RxClk period, coincident to whenever it is safe to sample the states of the RxOH and the RxOHFrame output pins. If the System-Side Terminal Equipment samples both the RxOHEnable and the RxOHFrame output pin "High", then it knows that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit within the most recently received DS3 frame via the RxOH output pin.

TASK # 2: As the System-Side Terminal Equipment samples the RxOHEnable and RxOHFrame output signals, it must also keep track of the number of times that the RxOHEnable output pin has been sampled "High" since the last time that both the RxOHEnable and RxOHFrame output pins have been sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bits are being output via the RxOH output at any time. This will permit the System-Side Terminal Equipment to properly handle these overhead bits as appropriate.

Table 33 relates the number of RxOHEnable output pulses that have occurred since both the RxOHFrame and the RxOHEnable output pins were sampled "High", to the DS3 Overhead Bit that is being output via the RxOH

output pin. The user can use this table as a guide for extracting the appropriate overhead bits, via the Receive Overhead Data Output Interface block, whenever Method 2 is used.

TABLE 33: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE RXOHENABLE SIGNAL, SINCE THE RXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS BEING OUTPUT (VIA THE RXOH OUTPUT PIN) BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF PULSES IN RXOHENABLE, SINCE RXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN
0 (RxOHEnable and RxOHFrame are sampled "High" simul-taneously)	X Bit # 1
1	F1
2	AIC (C11)
3	F0
4	NA (C12)
5	F0
6	FEAC (C13)
7	F1
8	X Bit # 2
9	F1
10	UDL Bit # 1 (C21)
11	F0
12	UDL Bit # 2 (C22)
13	F0
14	UDL Bit # 3 (C23)
15	F1
16	P
17	F1
18	CP Bit # 1 (C31)
19	F0
20	CP Bit # 2 (C32)
21	F0
22	CP Bit # 3 (C33)
23	F1
24	P
25	F1
26	FEBE # 1 (C41)
27	F0

TABLE 33: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE RXOHENABLE SIGNAL, SINCE THE RXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE DS3 OVERHEAD BIT THAT IS BEING OUTPUT (VIA THE RXOH OUTPUT PIN) BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

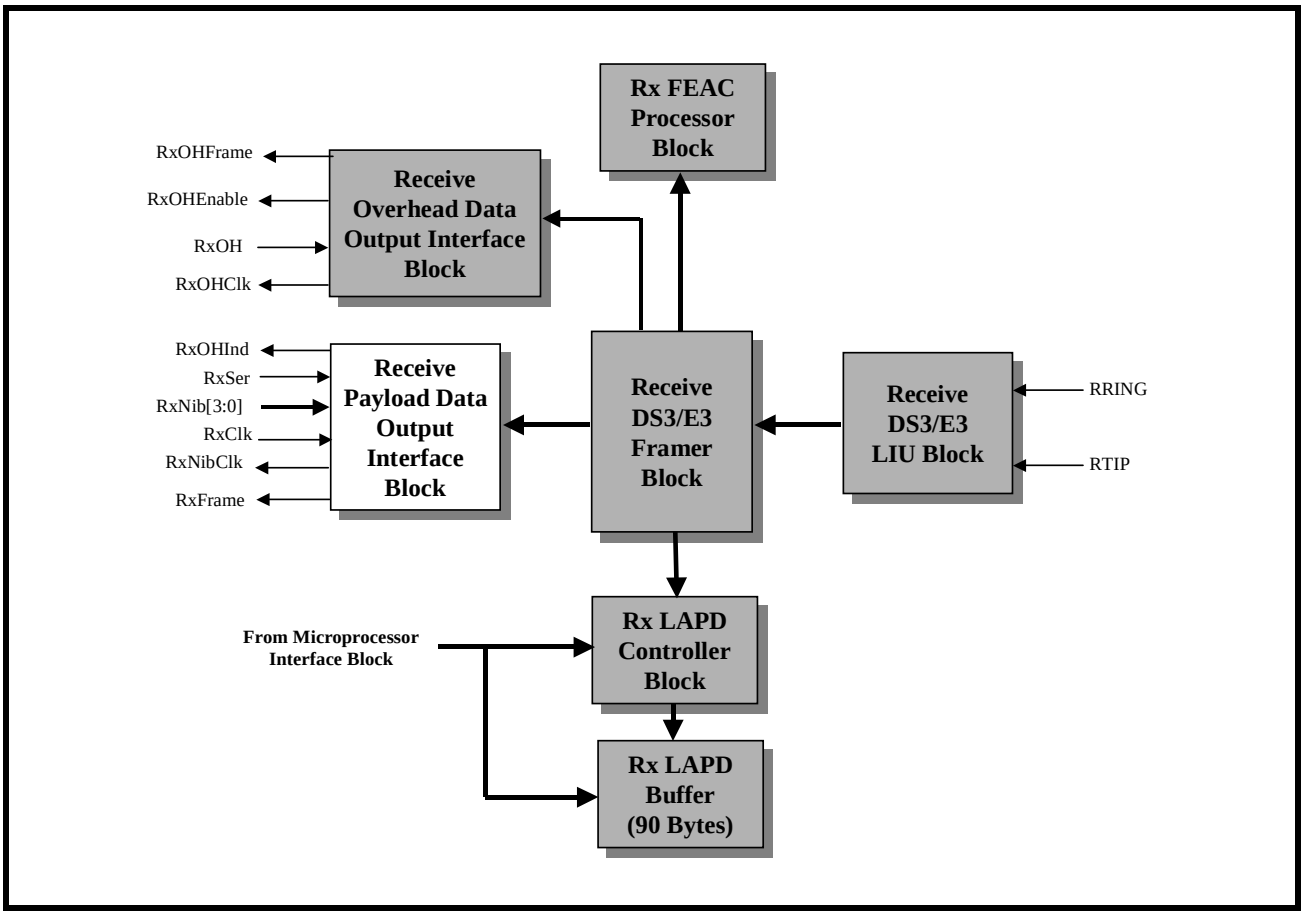
NUMBER OF PULSES IN RXOHENABLE, SINCE RXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT THAT IS BEING OUTPUT VIA THE RXOH OUTPUT PIN
28	FEBE # 2 (C42)
29	F0
30	FEBE # 3 (C43)
31	F1
32	M0
33	F1
34	DL Bit # 1 (C51)
35	F0
36	DL Bit # 2 (C52)
37	F0
38	DL Bit # 3 (C53)
39	F1
40	M1
41	F1
42	UDL Bit # 4 (C61)
43	F0
44	UDL Bit # 5 (C62)
45	F0
46	UDL Bit # 6 (C63)
47	F1
48	M0
49	F1
50	UDL Bit # 7 (C71)
51	F0
52	UDL Bit # 8 (C72)
53	F0
54	UDL Bit # 9 (C73)
55	F1

4.3.6 RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

The Receive Payload Data Output Interface block is the sixth and final functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 106** presents

an illustration of the Receive Direction circuitry whenever the XT79L71 has been configured to operate in the DS3 Clear-Channel Framer Mode, with the Receive Payload Data Output Interface block highlighted.

FIGURE 106. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE DS3 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK HIGHLIGHTED)



The purpose of the Receive Payload Data Output Interface block is to output payload data that has been extracted from the incoming DS3 data-stream that has been received and processed by the Receive Direction circuitry within the XRT79L71.

In order to accomplish this, the Receive Payload Data Output Interface block has numerous output pins. **Table 34** presents a list and a brief definition of each of these pins.

TABLE 34: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
RxSer	A5	O	**Receive Serial Payload Data Output pin:** If the Receive Payload Data Output Interface block is operated in the Serial Mode, then the XRT79L71 will output the payload data that has been extracted from the incoming DS3 data-stream, via this output pin. The XRT79L71 will update the data on this pin upon the rising edge of the RxCLK output clock signal. The user is advised to design or configure the System-Side Terminal Equipment such that it will sample the data that is output via this output pin, upon the falling edge of RxCLK. ***NOTES:*** 1. *This signal is only active if the NibIntf input pin is pulled "Low".* 2. *In reality, for Serial Mode operation, the entire incoming DS3 data-stream (payload bits and overhead bits) will be output via the RxSer output pin. The user will need to use the RxOHInd/RxGapClk signals in order to distinguish the payload bits from the overhead bits as they are output via the RxSer output pin.*
RxNib[3:0]	B4 A4 D6 C5	O	**Receive Nibble-Parallel Payload Data Output Pin:** If the Receive Payload Data Output Interface block is operated in the Nibble-Parallel Mode, then the XRT79L71 will output the payload data that has been extracted from the incoming DS3 data-stream, via these output pins, in a Nibble-Parallel manner. The XRT79L71 will update the data via these four output pins upon the falling edge of the RxCLK output signal. The user is advised to design or configure the System-Side Terminal Equipment such that it will sample this data upon the rising edge of RxCLK. ***NOTES:*** 1. *These pins are only active if the NibIntf input pin is pulled "High".* 2. *In contrast to Serial Mode operation, whenever the XRT79L71 has been configured to operate in both the DS3 and the Nibble-Parallel Mode, then only the payload data which has been extracted out of the incoming DS3 data-stream will be output via these output pins.*

TABLE 34: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
RxCLK/ RxNibClk	A6	O	**Receive Payload Data Output Interface -Clock Output Pin:** The exact behavior of this signal depends upon whether the XRT79L71 has been configured to operate in the Serial Mode or in the Nibble-Parallel Mode, as described below. **Serial Mode Operation - RxCLK** If the Receive Payload Data Output Interface block has been configured to operate in the Serial Mode, then this signal will be a 44.736MHz clock output signal. The Receive Payload Data Output Interface block will update the data via the RxSer output pin upon the rising edge of this clock signal. For Serial Mode operation, the user is advised to design or configure the System-Side Terminal Equipment to sample the data on the RxSer output pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation - RxNibClk** If the Receive Payload Data Output Interface block has been configured to operate in the Nibble-Parallel Mode, then the XRT79L71 will pulse this output pin 1176 times for each inbound DS3 frame. The Receive Payload Data Output Interface block will update the data via the RxNib[3:0] output pins upon the falling edge of this clock signal. For Nibble-Parallel Mode operation, the user is advised to design or configure the System-Side Terminal Equipment to sample the data on the RxNib[3:0] output pins, upon the rising edge of this clock signal. ***NOTE:*** *This output clock signal is ultimately derived from the Recovered Clock signal via the Receive DS3/E3 LIU Block.*

TABLE 34: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL #	TYPE	DESCRIPTION
RxOHInd/RxGapClk	C6	O	**Receive Overhead Bit Indicator Output/Receive Gap-Clock Output:** The exact function of this output pin depends upon whether the XRT79L71 has been configured to operate in the Gapped-Clock Mode or Not. **Non-Gapped Clock Mode - RxOHInd:** This output pin will pulse "High", for one RxClk period, coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the RxSer output pin. This output pin will be held "Low" at all other times. The purpose of this output pin is to alert the System-Side Terminal Equipment that the current bit (e.g., the one that is currently residing on the RxSer output pin) is an overhead bit and should not be processed by the System-Side Terminal Equipment. The XRT79L71 will update output signal upon the rising edge of RxClk. Therefore, the user is advised to design or configure the System-Side Terminal Equipment to sample this signal along with the data on the RxSer output pin on the falling edge of the RxClk signal.N ***NOTE:*** *For DS3 Applications, this output pin is only active in the RxOHInd role if the Receive Payload Data Output Interface block has been configured to operate in the Serial Mode. This output pin will be held "Low" at all times.* **Gapped Clock Mode - RxGapClk:** If the XRT79L71 has been configured to operate in the Gapped-Clock Mode, then this particular output pin will function as a payload bit output clock signal. In other words, in this mode, the Receive Payload Output Interface block will only generate a clock pulse via this output pin coincident to whenever it outputs a payload bit via the RxSer output pin. The Receive Payload Data Output Interface block will NOT generate a clock edge via this output pin coincident to whenever it outputs an overhead bit via the RxSer output pin. As a consequence, there will be gaps within this particular clock output signal, hence the name Gapped Clock Mode. If the XRT79L71 is configured to operate in the Gapped Clock Mode, then they must design or configure the System-Side Terminal Equipment to sample and latch the RxSer data upon the falling edge of the RxOHInd/RxGapClk clock signal.
RxFrame	B6	O	**Receive Payload Data Output Interface - Receive Start of Frame Output Indicator:** The exact behavior of this output pin depends upon whether the XRT79L71 has been configured to operate in the Serial or in the Nibble-Parallel Mode, as described below. **Serial Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "High" for one RxCLK period coincident to whenever it outputs the very first bit of a new DS3 frame via the RxSer output pin. This output pin will remain "Low" at all other times. **Nibble-Parallel Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "High" for one RxCLK or Nibble-Period coincident to whenever it outputs the very first nibble of a new DS3 frame via the RxNib[3:0] output pins. This output pin will remain "Low" at all other times.

Operation of the Receive Payload Data Output Interface Block

The Receive Payload Data Output Interface block permits the user to configure it to operate in either of the following modes.

- The Serial Mode
- The Nibble-Parallel Mode.

4.3.6.1 Serial Mode Operation of the Receive Payload Data Output Interface

If the Receive Payload Data Output Interface Block is configured to operate in the Serial Mode then further configuriguration of the Receive Payload Data Output Interface block can be made to operate in either the "Non-Gapped Clock" Mode or in the "Gapped Clock" Mode.

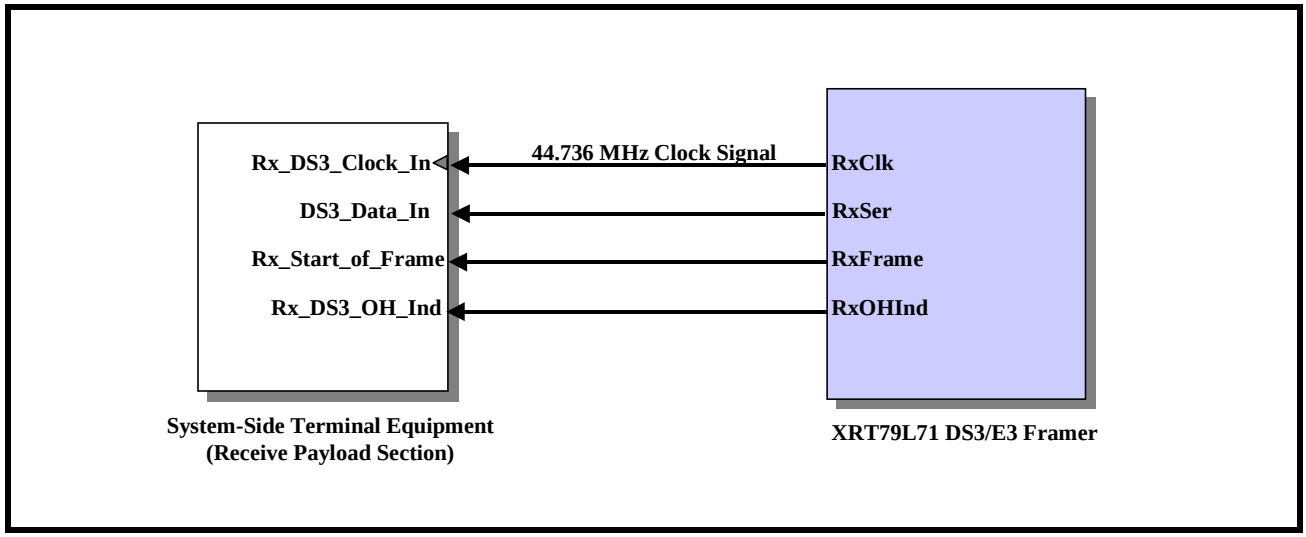
4.3.6.1.1 Operating the Receive Payload Data Output Interface in the "Non-Gapped Clock" Mode

If the Receive Payload Data Output Interface block is configured to operate in the "Serial/Non-Gapped-Clock" Mode, then all of the following is true.

- The XRT79L71 will output the entire contents of the incoming DS3 data-stream consisting of both payload and overhead bits via the RxSer output pin, upon the rising edge of the RxCLK signal which is a 44.736MHz clock signal.
- The user will need to rely on the RxOHInd/RxGapClk output pin in order to distinguish a payload bit from an overhead bit, within the data that is output via the RxSer output pin.
- The XRT79L71 will pulse the RxFrame output pin "High" for one RxCLK period, coincident to whenever it outputs the very first bit within a given DS3 frame via the RxSer output pin. The RxFrame output pin will be held "Low" at all other times.

Figure 107 presents an illustration of how to interface the System-Side Terminal Equipment to the Receive Payload Data Output Interface block of the XRT79L71, for Serial Mode Operation.

FIGURE 107. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT79L71 FOR SERIAL MODE OPERATION



Whenever the XRT79L71 has been configured to operate in this mode, then the Receive Payload Data Output Interface block will function as the source of the 44.736MHz clock signal via the RxCLK output signal. This clock signal is used as the System-Side Terminal Equipment clock source by both the Receive Payload Data Output Interface block of the XRT79L71 and the System-Side Terminal Equipment device or circuitry.

The Receive Payload Data Output Interface block will serially output the entire contents of the incoming DS3 data-stream via the RxSer output pin. As mentioned earlier, the Receive Payload Data Output Interface block

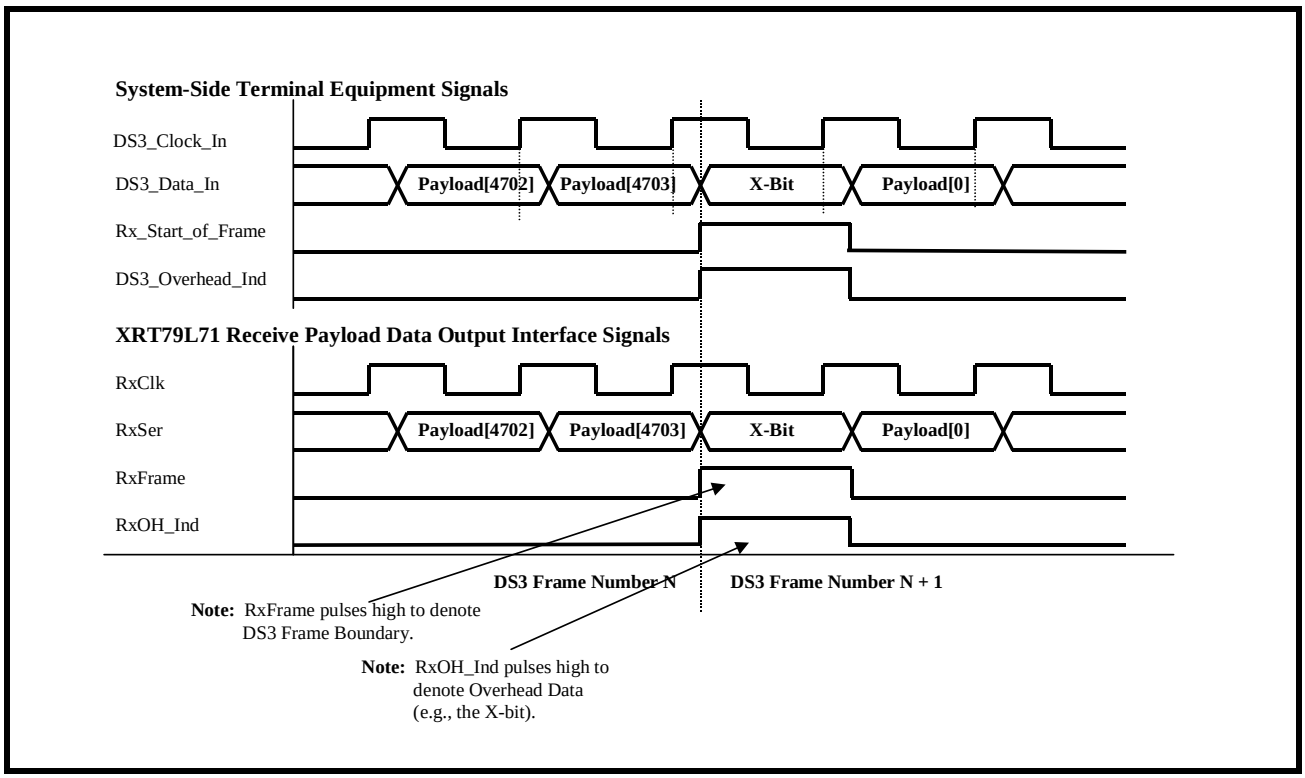
will output this data upon the rising edge of the RxCLK signal. As a consequence, the user is advised to design or configure the System-Side Terminal Equipment circuitry to sample and latch this data via the DS3_Data_In input pin upon the falling edge of RxCLK (Rx_DS3_Clock_In), as depicted below in Figure 107.

The Receive Payload Data Output Interface block within the XRT79L71 will indicate that it is processing the very first bit of a given DS3 frame by pulsing the RxFrame output pin "High" for one bit-period. The RxFrame output pin will be held "Low" at all other times.

Finally, the Receive Payload Data Output Interface block within the XRT79L71 permits the System-Side Terminal Equipment to identify a given bit that is being output via the RxSer output pin as either an overhead or a payload bit by pulsing the RxOH_Ind output pin "High" for one bit-period coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the RxSer output pin. Conversely, the Receive Payload Data Output Interface block will hold the RxOH_Ind output pin "Low" coincident to whenever the Receive Payload Data Output Interface block outputs a payload bit via the RxSer output pin.

Figure 108 presents an illustration of the System-Side Terminal Equipment/Receive Payload Data Output Interface signal for Serial Mode Operation.

FIGURE 108. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR SERIAL MODE OPERATION



Configuring the XRT79L71 to operate in Serial Mode.

The user can configure the XRT79L71 to operate in the Serial Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in Figure 107.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

NOTE: *This step also configures the Transmit Payload Data Input Interface block to operate in the Serial Mode.*

Operating the Receive Payload Data Output Interface in the Non-Gapped Clock Mode

If the Receive Payload Data Output Interface block within the XRT79L71 has been configured to operate in the Serial Mode, then we have recommended that the user design or configure their System-Side Terminal Equipment to do the following, when receiving/accepting DS3 data via the RxSer output pin.

- Check the state of the RxOH_Ind output pin from the XRT79L71 upon the falling edge of the RxClk signal.
- Perform either of the following actions, depending upon the sampled state of the RxOH_Ind output pin, as described below.

If RxOH_Ind is sampled "Low"

Then the System-Side Terminal Equipment should accept this particular data bit that is being sampled from the RxSer output pin and treat it as a payload bit.

If RxOH_Ind is sampled "High"

Then the System-Side Terminal Equipment should NOT accept this particular data bit that is being sampled from the RxSer output pin and should definitely NOT treat it as a payload bit.

In this particular approach, the user must gate the acceptance of a particular data bit being sampled via the RxSer output pin based upon the corresponding sampled state of the RxOH_Ind output pin. While implementing such a design into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Receive Payload Data Output Interface block to operate in the Gapped-Clock Mode.

4.3.6.1.2 Operating the Receive Payload Data Output Interface block in the in the Gapped-Clock Mode

As mentioned above, in order to simplify the task of interfacing the Receive Payload Data Output Interface block to certain devices, the XRT79L71 permits the user to configure the Transmit Payload Data Input Interface and the Receive Payload Data Output Interface blocks to operate in the "Gapped-Clock" Mode. If the Receive Payload Data Output Interface block is configured to operate in the Gapped-Clock Mode, then the role of the RxOH_Ind output pin will change from being the Overhead Indicator output pin, to now being a payload data clock output pin. In other words, If the Receive Payload Data Output Interface block is configured to operate in the Gapped-Clock Mode, then it the Receive Payload Data Output Interface block will only generate a clock pulse via the RxOH_Ind output pin coincident to whenever it outputs a payload bit via the RxSer output pin. Whenever the Receive Payload Data Output Interface block outputs an overhead bit via the RxSer output pin then it will NOT generate a clock pulse via the RxOH_Ind output pin. This action will result in the Receive Payload Data Output Interface block generating a gapped clock signal via the RxOH_Ind output pin, hence the term, Gapped-Clock Mode.

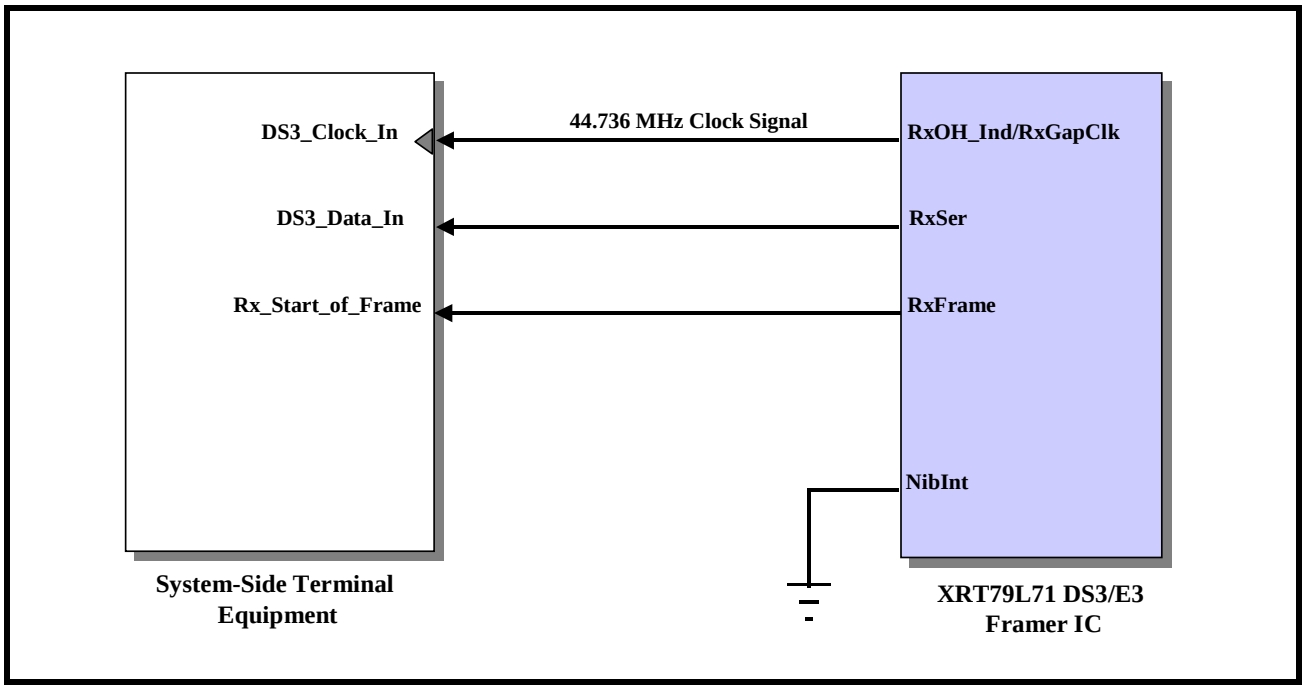
If the Receive Payload Data Output Interface block is configured to operate in the Gapped-Clock Mode, then the System-Side Terminal Equipment will be expected to sample the data that is being output via the RxSer output pin upon the rising edge of RxClk. In this case, there is no need to check the state of the certain output pin, then gate the acceptance/treatment of the next bit output via the RxSer output pin based upon the state of this particular output pin. The System-Side Terminal Equipment only needs to sample the RxSer output signal upon the rising edge of this particular Gapped-Clock signal.

Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode

To configure the Receive Payload Interface block to operate in the Gapped-Clock Mode, do all of the following.

STEP 1 - Interface the System-Side Terminal Equipment to the Receive Payload Data Input Interface block, in a manner as indicated below.

FIGURE 109. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE RECEIVE PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR GAPPED-CLOCK MODE OPERATIONS



STEP 2 - Set Bit 6 (Receive Gapped Clock Mode Enable), within the Framer Test Register to "1" as depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	1	0	0	0	0	0	0

Figure 110 presents the resulting behavior of the Receive Payload Data Output Interface signals, if it has been configured to operate in the Gapped-Clock Mode.

Figure 110. An Illustration of the Behavior of the Receive Payload Data Output Interface Block signals, whenever it has been configured to operate in the "Gapped-Clock" Mode

4.3.6.2 Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface

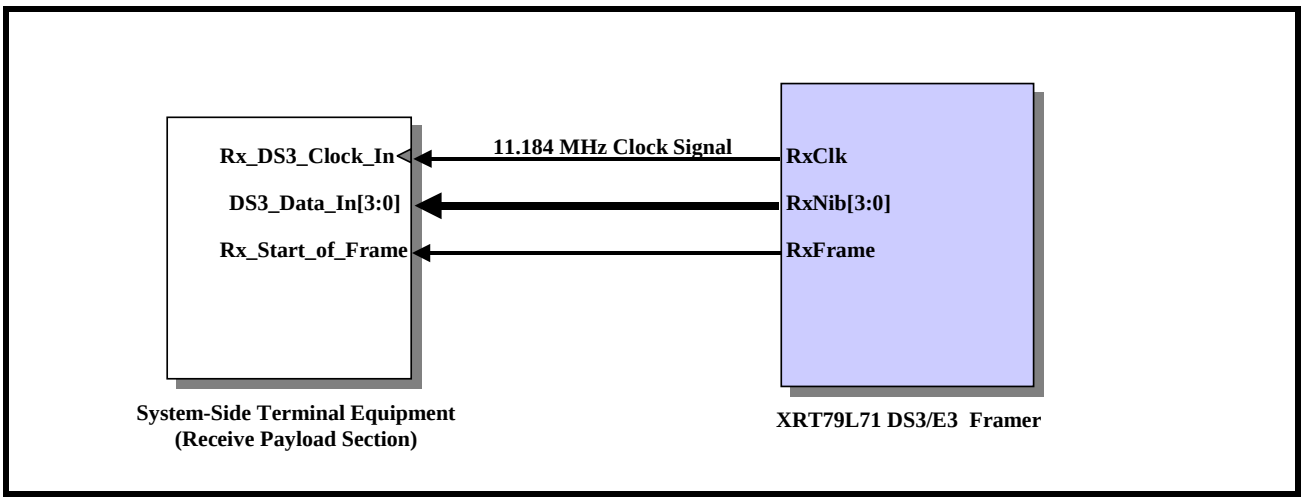
If the XRT79L71 is configured to operate in the Nibble-Parallel Mode, then all of the following is true.

- The XRT79L71 will only output the contents of the payload bits within the incoming DS3 data-stream via the RxNib[3:0] output pins, upon the falling edge of the RxCLK signal.

- The XRT79L71 will pulse the RxFrame output pin "High" for one RxCLK or Nibble Period coincident to whenever it outputs the very first nibble within a given DS3 frame via the RxNib[3:0] output pins. The RxFrame output pin will be held "Low" at all other times.

Figure 111 presents an illustration of how to interface the System-Side Terminal Equipment to the Receive Payload Data Output Interface block of the XRT79L71 for Nibble-Parallel Mode Operation.

FIGURE 111. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK OF THE XRT79L71 FOR NIBBLE-PARALLEL MODE OPERATION



Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface Block

Whenever the XRT79L71 has been configured to operate in the Nibble-Parallel Mode, then the Receive Payload Data Output Interface block will function as the source of a Nibble Clock signal via the RxCLK output signal.

The Receive Payload Data Output Interface block will output all of the payload data that has been extracted out to the incoming DS3 data-stream in a Nibble-Parallel Mode via the RxNib[3:0] output pins. In contrast to whenever the Receive Payload Data Output Interface block is configured to operate in the Serial Mode, no overhead bits will be output via the RxNib[3:0] output pins, whenever the Receive Payload Data Output Interface block has been configured to operate in the Nibble-Parallel Mode. As mentioned earlier, the Receive Payload Data Output Interface block will output this data upon the falling edge of the RxCLK signal. As a consequence, the user is advised to design or configure the System-Side Terminal Equipment circuitry to sample and latch this data via the DS3_Data_In[3:0] input pins upon the rising edge of RxCLK (Rx_DS3_Clock_In), as depicted below in Figure 112.

The Receive Payload Data Output Interface block within the XRT79L71 will indicate that it is processing the very first payload nibble of a given DS3 frame by pulsing the RxFrame output pin "High" for one nibble-period. The RxFrame output pin will be held "Low" at all other times.

Finally, since (for DS3 Applications) the Receive Payload Data Output Interface block within the XRT79L71 does not process nor output any overhead bits, then there is no need for it to distinguish the payload bits from the overheads bits, as this data is output via the RxNib[3:0] output pins. As a consequence, whenever the XRT79L71 is configured to operate in both the DS3 and the Nibble-Parallel Mode, the RxOH_Ind output pin will ALWAYS be held "Low".

The Frequency of the RxClk Signal for DS3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Receive Payload Data Input Interface has been configured to operate in the Nibble-Parallel Mode, it will NOT process the DS3 overhead bits. Only DS3 payload data is processed through the Receive Payload Data Input Interface (e.g., via the RxNib[3:0] input pins). As a consequence, the frequency of the RxClk signal for Nibble-Parallel Mode applications will NOT simply be 44.736MHz/4 or 11.184MHz.

If we were to look at this issue another way, we would recall that each DS3 frame consists of 4760 bits. Of these bits, 56 are overhead bits and the remaining 4704 bits are payload bits. This means that there are 4704 bits/4 = 1176 nibbles of payload bits within each DS3 frame.

The frame repetition rate (for DS3) is 9.398kHz. Therefore, if one performs the following multiplication:

{9398 Frames/sec **X** 1176 Nibble/Frame} = 11.052MHz (for the Average Frequency of the RxClk output signal).

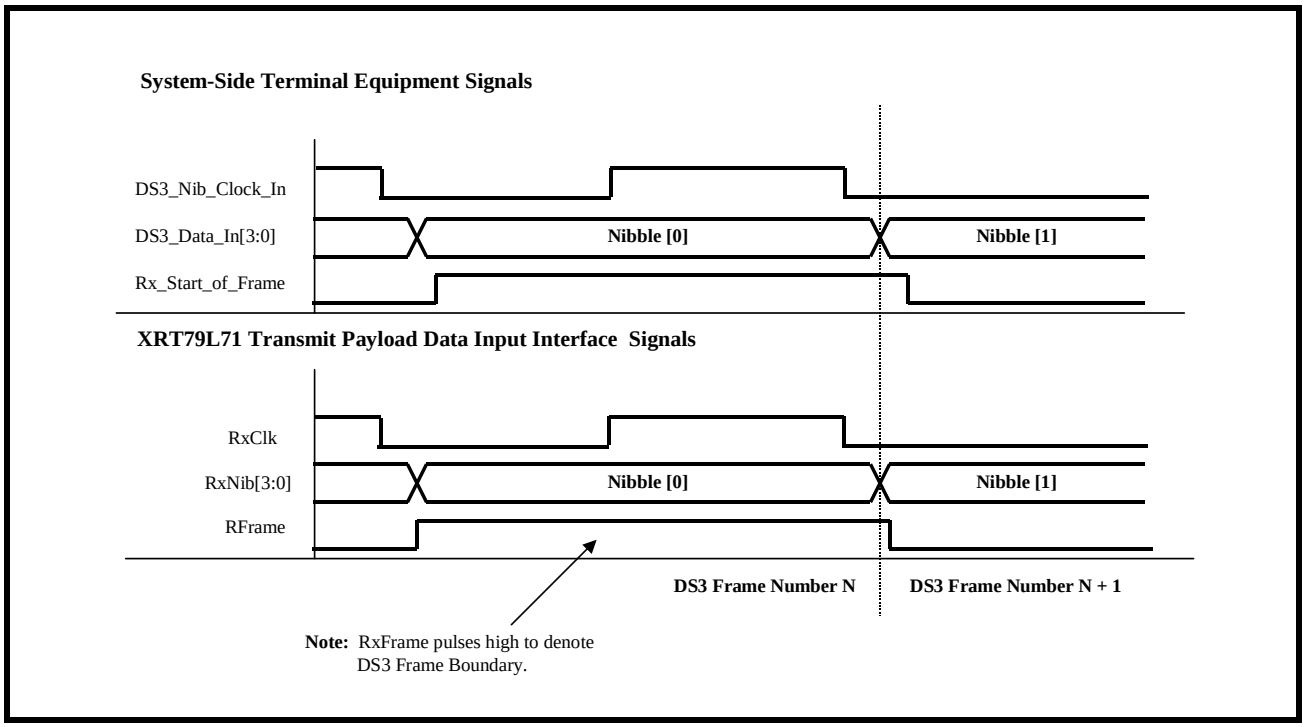
How the 1176 Clock Edges within the RxClk output signal are distributed throughout a DS3 frame.

In general, for 1120 RxClk periods, the instantaneous frequency of the RxClk output clock signal will be 11.184MHz (e.g., each of these clock periods will corresponding to exactly 4 DS3 bit-periods). However, for the remaining 56 of these RxClk periods, the periods of these clock signals will be lengthened to five (5) of DS3 bit-periods.

For this reason, in DS3/Nibble-Parallel Mode applications, if the RxClk signal was monitored with a scope, a considerable amount of jitter could be seen within this particular clock signal.

Figure 112 presents an illustration of the behavior of System-Side Terminal Equipment/Receive Payload Data Output Interface signals for Nibble-Parallel Mode Operation.

FIGURE 112. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR NIBBLE-PARALLEL MODE OPERATION



Configuring the XRT79L71 to operate in the Nibble-Parallel Mode

The user can configure the XRT79L71 to operate in the Nibble-Parallel Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in Figure 112.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

NOTE: *This step also configures the Transmit Payload Data Input Interface block to operate in the Nibble-Parallel Mode.*

5.0 ARCHITECTURAL/FUNCTIONAL DESCRIPTION OF THE XRT 79L71 - E3, ITU-T G.751 MODE OPERATION

This particular section discusses Clear-Channel Framer over E3, ITU-T G.751 operation of the XRT79L71. Prior to discussing the architecture and the role of the DS3/E3 Framer blocks, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Mode, it is imperative to discuss the E3, ITU-T G.751 Frame structure.

5.1 DESCRIPTION OF THE E3, ITU-T G.751 FRAME STRUCTURE AND THE OVERHEAD BITS

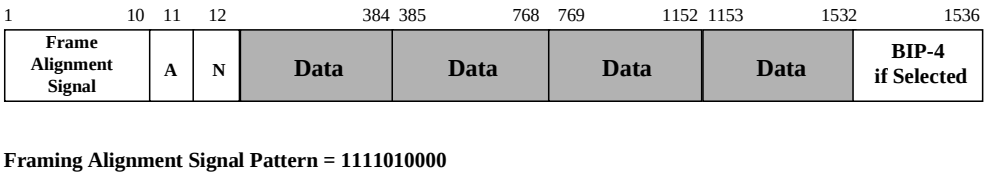
The E3, ITU-T G.751 frame contains 1536 bits, of which 12 bits are overhead bits and the remaining 1524 bits are payload bits.

Each E3 frame consists of the following twelve (12) overhead bits.

- A 10-bit FAS (Framing Alignment Signal) pattern. This pattern is assigned the constant pattern of [1, 1, 1, 1, 0, 1, 0, 0, 0, 0], and is used by the Receive E3 Framer block to acquire and maintain Frame Synchronization with the incoming E3 frame.
- 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 22,375 frames per second, thereby resulting in the standard E3 bit rate of 34.368Mbps. Figure 113 presents an illustration of the E3, ITU-T G.751 Frame Format.

FIGURE 113. ILLUSTRATION OF THE E3, ITU-T G.751 FRAMING FORMAT



Configuring the XRT79L71 to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode

The user can configure the XRT79L71 to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode by executing the following three steps.

STEP 1- Configure the XRT79L71 to operate in the Clear-Channel Framer/HDLC Controller Mode

This can be accomplished by setting Bit 0 (Configuration Control) within the Operation Control Register - Byte 3 to "1" as depicted below.

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

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Configura-tion Control
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	1

STEP 2 - Configure the XRT79L71 to operate in the Clear-Channel Framer Mode

The user can accomplish this by setting Bit 6 (HDLC Controller Enable), within the Payload HDLC Control Register to "0" as depicted below.

Payload HDLC Control Register, (Address = 0x1182)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer By-Pass	HDLC Controller Enable	HDLC CRC-32	Unused	HDLC Loop-back	Unused		
R/W	R/W	R/W	R/O	R/W	R/O	R/O	R/O
0	0	0	0	0	0	0	0

STEP 3 - Configure the XRT79L71 to operate in the E3, ITU-T G.751 Framing Format

The user can configure the XRT79L71 to operate in the E3, ITU-T G.751 Framing format by writing the appropriate data into Bits 6 (IsDS3) and 2 (Frame Format) within the Framer Operating Mode Register (Address = 0x1100), as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimReference Select [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

The following table lists the relationship between the value of these bit-fields and the resulting framing formats for the XRT79L71.

TABLE 35: THE RELATIONSHIP BETWEEN THE CONTENTS OF BITS 2 (FRAME FORMAT) AND 6 (ISDS3) WITHIN THE FRAMER OPERATING MODE REGISTER, AND THE RESULTING FRAMING FORMAT

BIT 6 (ISDS3)	BIT 2 (FRAME FORMAT)	RESULTING FRAMING FORMAT OF XRT79L71
0	0	E3, ITU-T G.751
0	1	E3, ITU-T G.832
1	0	DS3, C-bit Parity
1	1	DS3, M13/M23

The XRT79L71 will be operating in the E3, ITU-T G.751 Framing Format, upon power-up or hardware reset.

NOTE: *This bit setting configures the framing format for both the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block.*

Definition of the E3, ITU-T G.751 Overhead Bits

In general, the E3 ITU-T G.751 Overhead bits serve the following three purposes.

1. To support Frame Synchronization between the Near-End and remote E3 Terminals.
2. To (optionally) provide parity bits in order to facilitate performance monitoring and error detection within the E3 data-stream.

3. To support the transmission of Alarms, Status and Data Link Information between the Near-End and the remote E3 Terminals.

The E3 Overhead bits supporting each of these purposes are further defined below.

Frame Synchronization Bits - The Frame Alignment Signal (FAS) Bits

Each E3, ITU-T G.751 Frame contains a total of 10 bits that support frame synchronization. These 10 bits are referred to as the FAS (or Framing Alignment Signal). A given Transmitting E3 Terminal Equipemnt will set the FAS bits to the value "[1, 1, 1, 1, 0, 1, 0, 0, 0, 0]". The Receiving E3 Terminal Equipment will use these FAS bits in order to acquire and maintain frame synchronization with the incoming E3 data-stream. For more information on how the Receive DS3/E3 Framer block uses these bit-fields, please see **SEE"THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DECLARATION CRITERIA" ON PAGE 379.**.

Performance Monitoring Bits - The BIP-4 Bits

Each E3 frame can be configured to carry a BIP-4 trailer (e.g., the last four bits of each given E3 frame). These BIP-4 bits carries the BIP-4 (Bit Interleaved Parity - 4) value of the previous E3 frame for performance monitoring. As a Transmitting E3 Terminal assembles an E3 frame (prior to transmitting this E3 signal to the remote terminal equipment) it will (optionally) compute the Bit-Interleave-Parity - 4 (BIP-4) over this entire outbound E3 frame. The Transmitting E3 Terminal will then insert the resulting BIP-4 value into the BIP-4 nibble position within the very next outbound E3 frame.

As a Receiving E3 Terminal receives a given incoming E3 frame, it will (if configured accordingly) locally compute its value for the BIP-4 nibble. Afterwards, this Receiving E3 Terminal will compare its locally-computed BIP-4 value with the value of the BIP-4 Nibble within the very next incoming E3 frame. If these two BIP-4 values match, then the Receiving E3 Terminal will presume that the first (of these two) incoming E3 frames received in an error-free manner. If these two BIP-4 values DO NOT match, then the Receiving E3 Terminal will presume that the first (of these two) incoming E3 frames was received in an erred manner.

For information on how the Receive DS3/E3 Framer block handles the BIP-4 nibble within the incoming E3 data-stream, please see **SEE"DETECTING BIP-4 NIBBLE ERRORS" ON PAGE 388.**.

Alarm and Signaling-Related Overhead Bits

The E3, ITU-T G.751 frames includes two bits that are used to support the handling of alarms/defects and signaling information. Each of these bit-fields is defined below.

The A (Alarm) Bit

According to the ITU-T G.751 Specification, the A is suppose to function as the FERF/RDI (Far-End Receive Failure/Remote Defect Indicator) bit. The XRT79L71 also has provisions to permit the user to configure the "A" bit to function as the FEBE/REI (Far-End Block Error/Remote Error Indicator) bit.

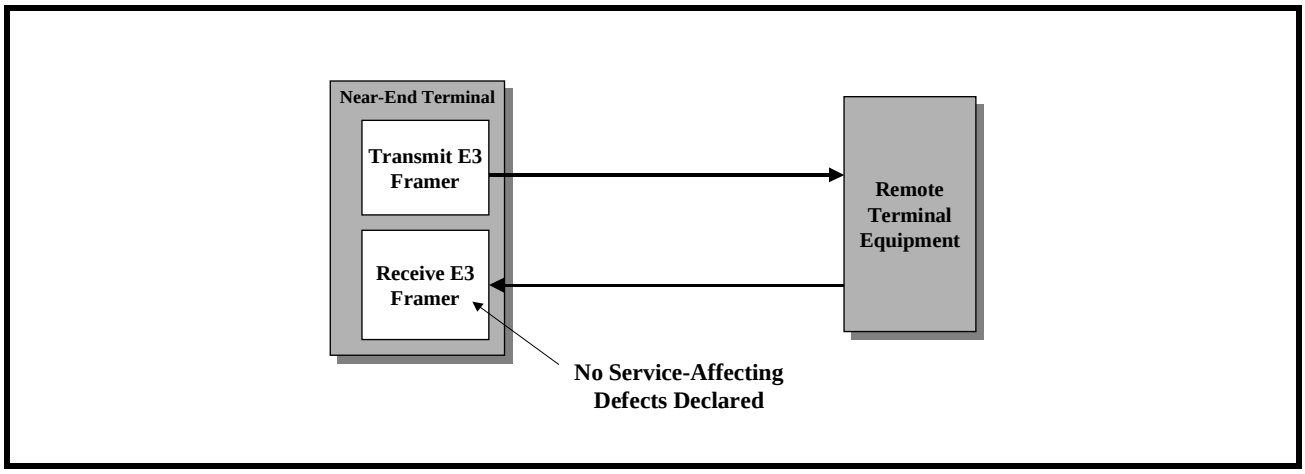
Each of these possible uses of the "A" bit are listed below.

Using the A Bit as the FERF/RDI Bit

According to the ITU-T G.751 Specification, the "A" bit carries the FERF (Far-End Receive Failure)/RDI (Remote Derfect Indicator) bit within the E3, ITU-T G.751 frame. The purpose of the FERF/RDI bit is to permit a given E3 Terminal Equipment to transmit a FERF/RDI indicator to the remote terminal equipment. Whenever a Near-End E3 Terminal equipment declares a servicing-affecting defect condition (e.g., the LOS or the LOF/OOF defect condition) within its incoming E3 data-stream, then it will inform the remote terminal equipment (e.g., the source of this defective E3 signal) of this fact by transmitting the FERF/RDI indicator back out to the Remote Terminal Equipment via the outbound (returning) E3 signal. This Near-End E3 Terminal will indicate the FERF/RDI condition by setting the "A" bit-field, within each outbound E3 frame to "1" for the duration that the Receive-related defect condition exists. Conversely, this Near-End E3 Terminal will indicate that it is NOT transmitting the FERF/RDI indicator by setting the "A" bit-field to "0" for the duration that no Receive-related/Service-affecting defect conditions exists. This concept of transmitting the FERF/RDI indicator in response to certain defect conditions in reinforced in **Figure 114** through **Figure 117**, below.

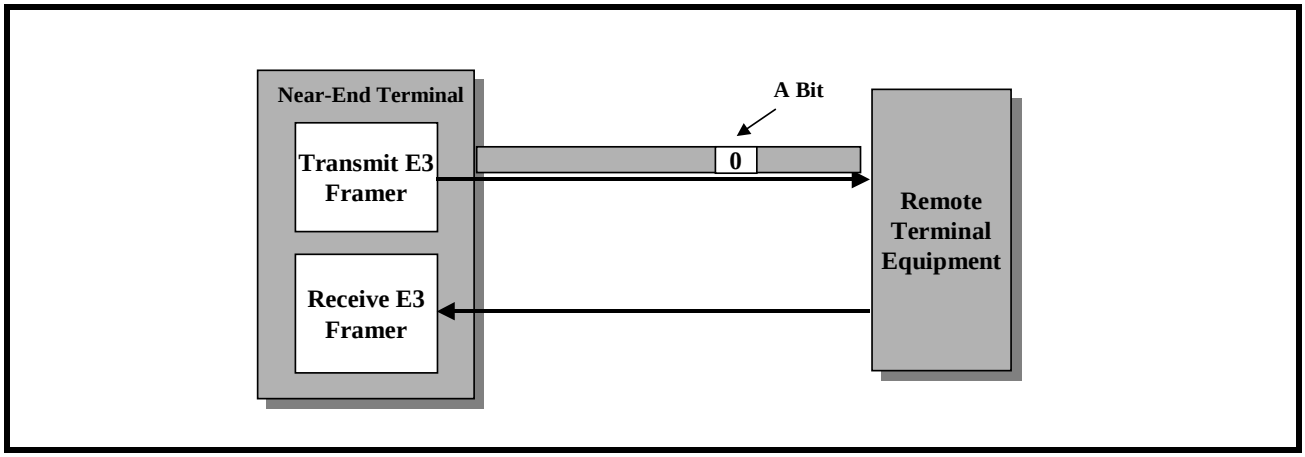
Figure 114 presents an illustration of a given Near-End Terminal that is exchanging E3 data with a remote terminal, in an un-erred manner.

FIGURE 114. A SIMPLE ILLLUSTRATION OF A NEAR-END TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



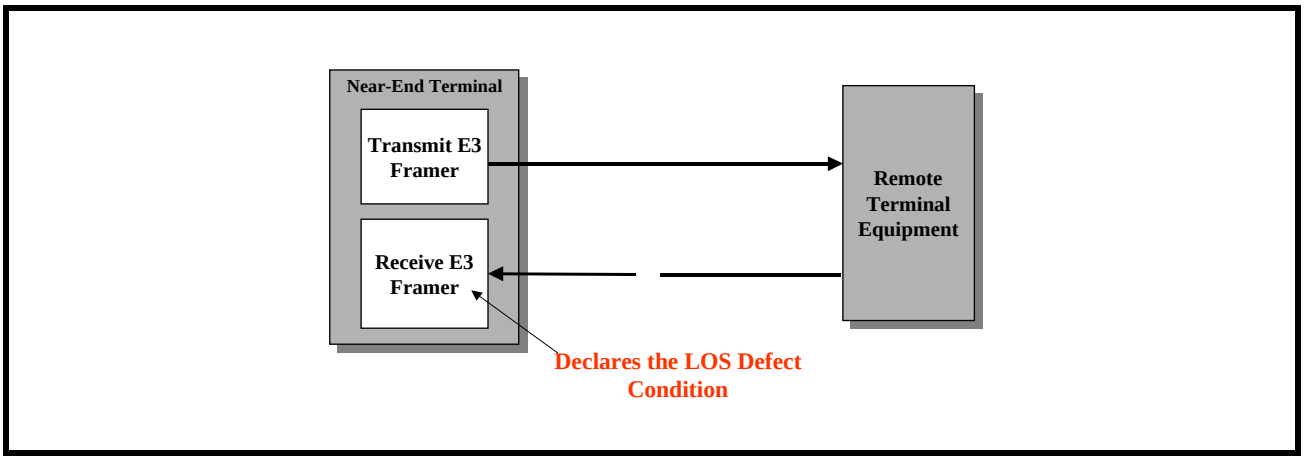
In response to this un-erred condition, the Transmit E3 Framer (within the Near-End Terminal) will respond by setting the FERF/RDI bit-field within each of its outbound E3 frames to "0", in order to denote an un-erred condition. **Figure 115** presents an illustration of the Transmit E3 Framer sending an un-erred indication to the Remote Terminal Equipment.

FIGURE 115. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



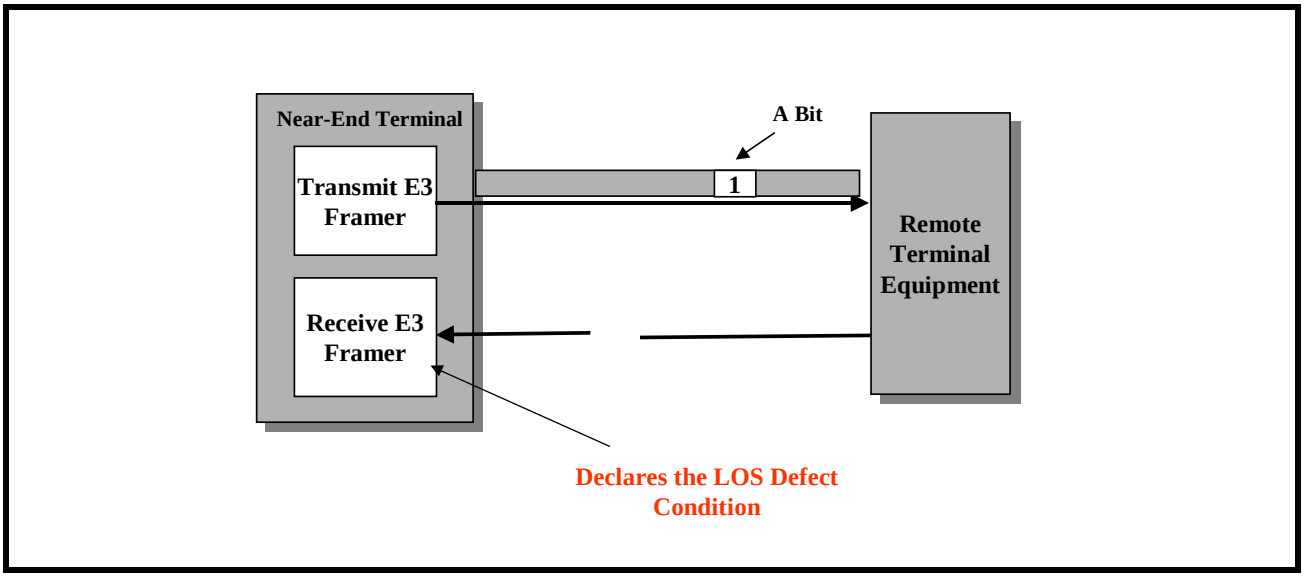
Next, **Figure 116** presents an illustration of a given Near-End Terminal that is declaring the LOS defect with its incoming E3 signal.

FIGURE 116. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DECLARING THE LOS DEFECT CONDITION WITHIN ITS INCOMING E3 SIGNAL



In response to this erred condition, the Transmit E3 Framer (within the Near-End Teriminal) will respond by setting the "A" bit, within each of its outbound E3 frames to "1" in order to denote a FERF (Far-End Receive Failure) or RDI (Remote Defect Indicator) condition. **Figure 117** presents an illustration of the Transmit E3 Framer sending the FERF/RDI indicator to the Remote Terminal Equipment.

FIGURE 117. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block (within the XRT79L71) handle the FERF/RDI indication will be described in **SEE"TRANSMITTING THE FERF/ RDI INDICATOR" ON PAGE 323.** and **SEE"SETTING THE TRANSMIT E3 FRAMER BLOCK TIMING REFERENCE" ON PAGE 329.**, respectively.

Using the "A" Bit as the FEBE/REI Bit

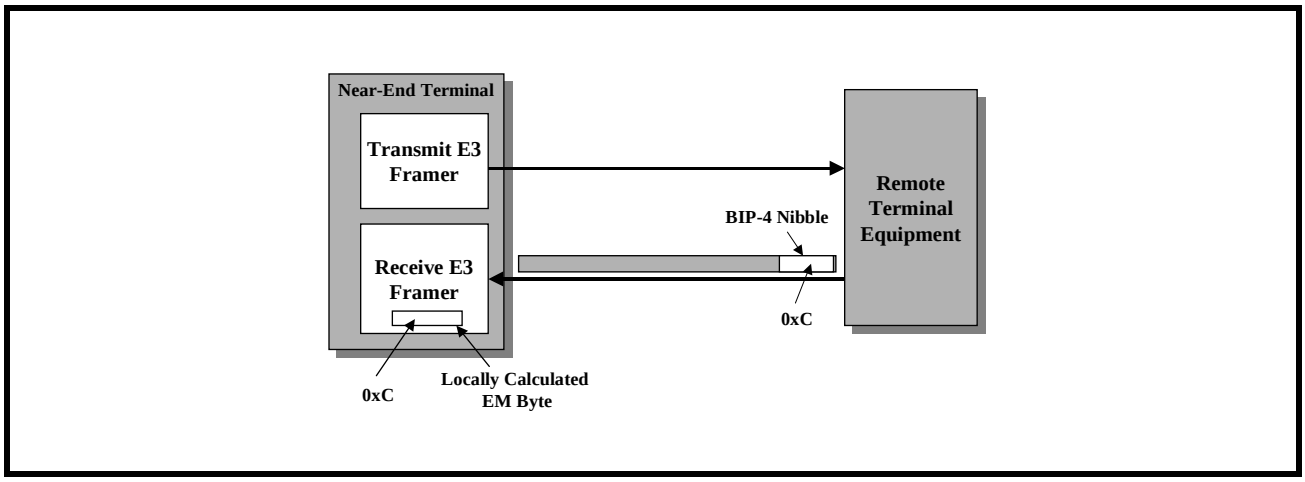
The XRT79L71 permits the user to use the "A" bit as a FEBE/REI indicator, provided that the user has also enabled BIP-4 verification within the E3 frames that it processes. If this feature is invoked, then (in lieu of carrying the FERF/RDI indicator) the "A" bit will carry the FEBE/REI indicator. In this case, if the Near-End XRT79L71 detects BIP-4 errors within the incoming (received) E3 data-stream, then it will inform the remote terminal of this fact by setting the "A" bit-field (within the outgoing, or return E3 frame) to the value "1" in order

to indicate an erred condition. The Transmit Section of the XRT79L71 will set the "A" bit-field to "0" in order to denote an un-erred condition.

This concept of transmitting the FEBE/REI indicator (via the "A" bit) in response to the detection of BIP-4 errors is reinforced in Figure 118 through Figure 121, below.

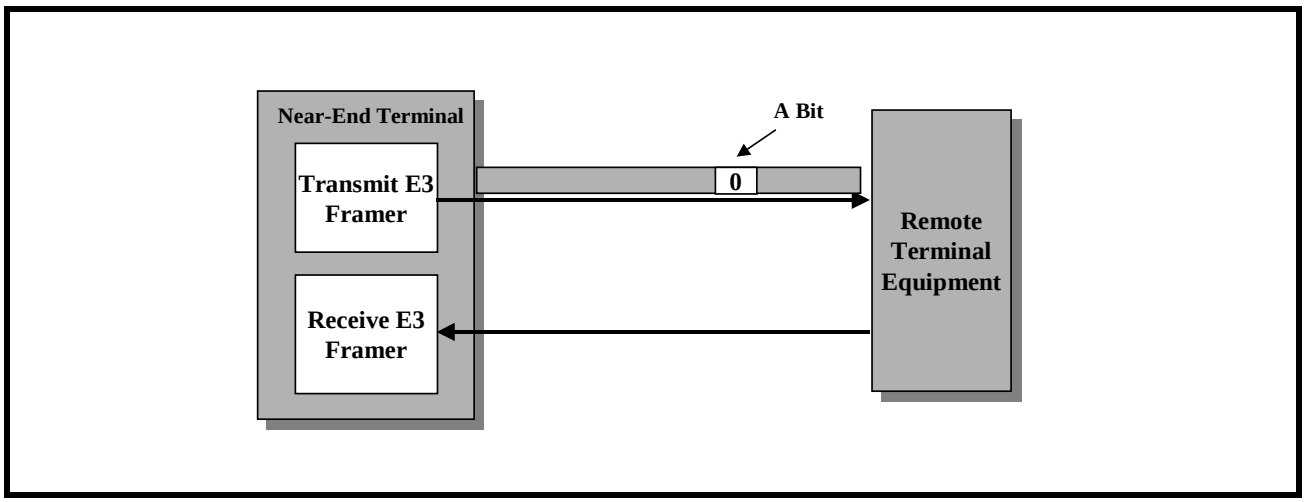
Figure 118 presents an illustration of a given Near-End Terminal that is exchanging E3 data with the remote terminal, in an un-erred manner. More specifically, the Receive E3 Framer block has locally computed a BIP-4 value of "0xC" (that was computed over a given incoming E3 frame). Figure 119 indicates that the BIP-4 nibble value (which resides within the very next incoming E3 frame) is of the value "0xC". As a consequence, there is no BIP-4 nibble error being detected at this time.

FIGURE 118. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



In response to this un-erred condition, the Transmit E3 Framer (within the Near-End Terminal) will respond by setting the "A" bit-field "0". Figure 119 presents an illustration of the Transmit E3 Framer sending this un-erred indication to the Remote Terminal Equipment.

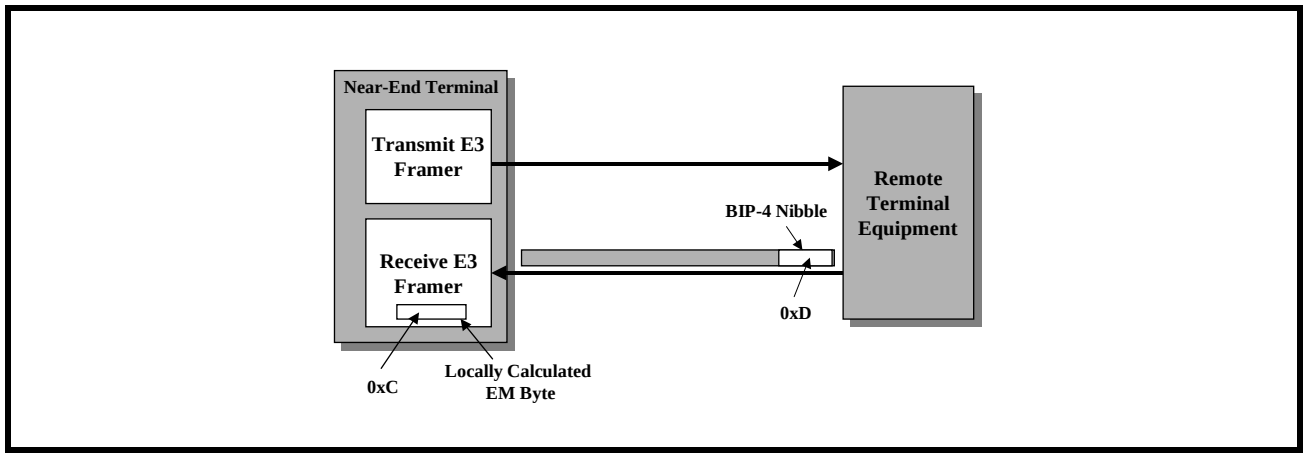
FIGURE 119. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



Next, Figure 120 presents an illustration of a given Near-End Terminal that is detecting a BIP-4 nibble error within its incoming E3 signal. More specifically, the Receive E3 Framer block has locally computed a BIP-4 nibble of "0xC" (that was computed over a given incoming E3 frame). Figure 120 indicates that the BIP-4

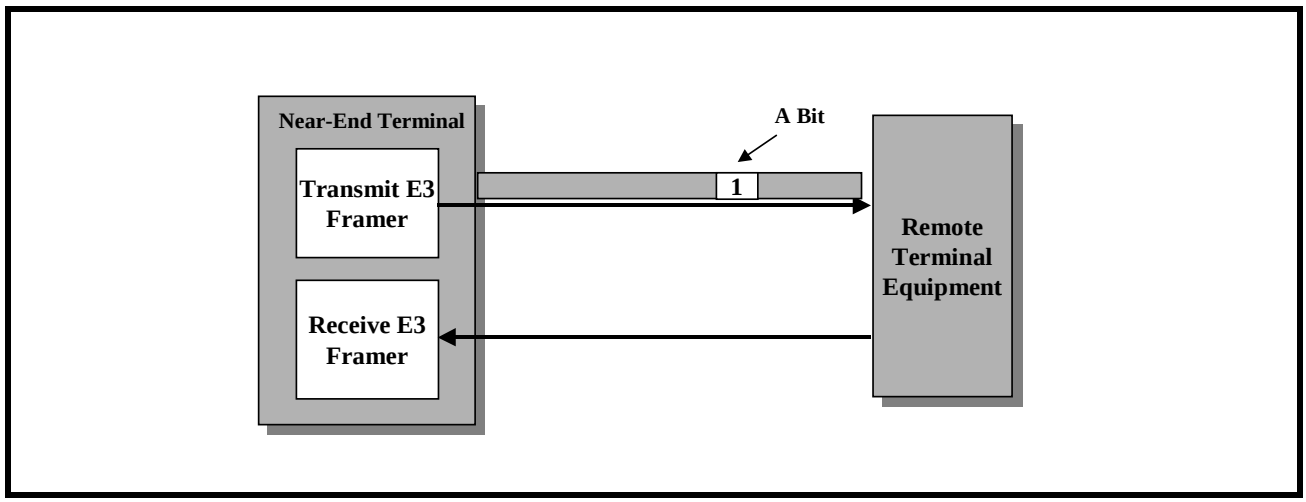
nibble value (which resides within the very next incoming E3 frame) is of the value "0xD". As a consequence, a BIP-4 nibble error is being detected at this time.

FIGURE 120. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DETECTING BIP-4 NIBBLE ERRORS WITHIN ITS INCOMING E3 SIGNAL



In response to this erred condition, the Transmit E3 Framer (within the Near-End Terminal) will respond by setting the "A" bit, within the very next outbond E3 frame to the value "1" in order to denote a FEBE/REI event. More specifically, the Transmit E3 Framer (within the Near-End Terminal) will transmit an E3 frame (with the "A" bit set to "1") each time the corresponding Receive E3 Framer (also within the Near-End Terminal) detects a BIP-4 nibble error within an E3 frame. **Figure 121** presents an illustration of the Transmit E3 Framer sending the FEBE/REI indicator to the Remote Terminal Equipment.

FIGURE 121. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FEBE/REI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block (within the XRT79L71) handles the "A" bit (when configured to function as the FEBE/REI indicator bit) can be found in **SEE"TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR" ON PAGE 324.** and **SEE"DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS" ON PAGE 390.**.

The "N" (National) Bit

The "N" bit is referred to (in the ITU-T G.751 Specification) as the National Bit. In the case of the XRT79L71, it permits the user to perform any of the following tasks with the "N" bit.

a. To (via software control) set it to any value.

b. To configure the "N" bit to transport the LAPD/PMDL Message

Information on how to control and monitor the state of the "N" bit (within a given E3 data-stream) can be found in SEE"INTERFACING THE TRANSMIT DS3/E3 LIU BLOCK TO THE LINE" ON PAGE 352. and SEE"MONITORING THE STATE OF THE N-BIT IN THE INCOMING E3 DATA STREAM" ON PAGE 392., respectively.

Information on how to configure the XRT79L71 to transport LAPD/PMDL Messages via the "N" bit, within a given E3 data-stream can be found in SEE"TRANSMIT LAPD CONTROLLER BLOCK" ON PAGE 303. andSEE"THE RECEIVE LAPD CONTROLLER BLOCK" ON PAGE 392..

5.2 THE TRANSMIT DIRECTION - E3, ITU-T G.751 CLEAR-CHANNEL FRAMER APPLICATIONS

Now that the basics of the E3, ITU-T G.751 frame structure have been discussed, the next several sections present an in-depth functional description of all of the blocks that are operating in the Transmit Direction, within the XRT79L71, when configured to operate in the Clear-Channel E3 Framer Mode. Figure 122 presents a functional block diagram of the Transmit Direction circuitry within the XRT79L71.

FIGURE 122. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE

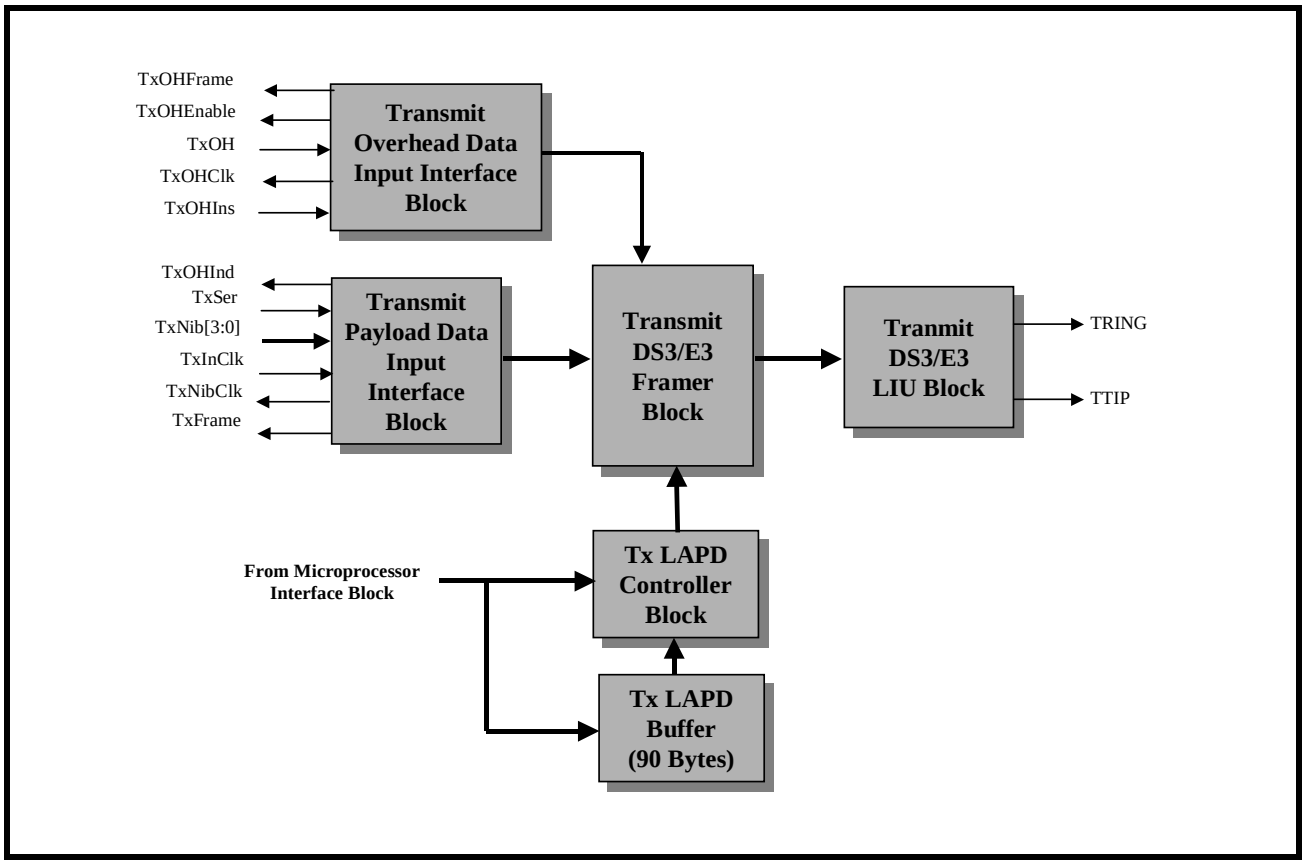


Figure 122 indicates that the Transmit Direction circuitry consists of the following functional blocks.

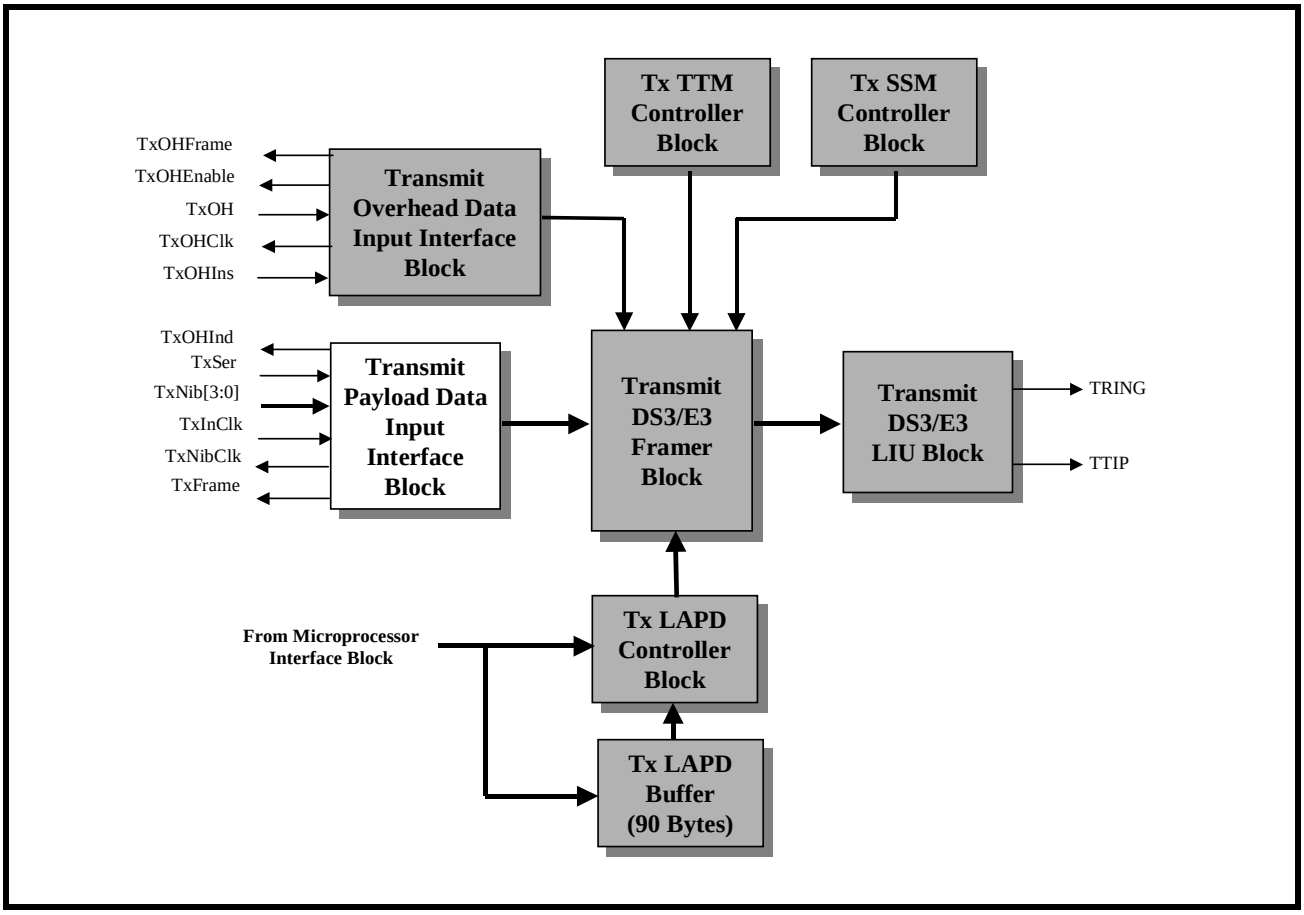
- The Transmit Payload Data Input Interface block
- The Transmit Overhead Data Input Interface block
- The Transmit LAPD Controller block
- The Transmit E3 Framer block

- The Transmit E3 LIU block

5.2.1 TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

The Transmit Payload Data Input Interface block is the very first functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. **Figure 123** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, Clear-Channel Framer Mode, with the Transmit Payload Data Input Interface block highlighted.

FIGURE 123. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK HIGHLIGHTED)



The purpose of the Transmit Payload Data Input Interface block is to accept payload data from system-side or up-stream source and to pass this payload data along to the Transmit E3 Framer block that will ultimately map this payload data into the payload bytes within each outbound E3 frame.

In order to accomplish this, the Transmit Payload Data Input Interface block has numerous input and output pins. **Table 36** presents a list and a brief description of each of these pins.

TABLE 36: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/ BALL #	TYPE	DESCRIPTION
TxSer	C9	I	**Transmit Serial Payload Data Input Pin:** If the Transmit Payload Data Input Interface block is operated in the Serial Mode, then the System-Side equipment is expected to apply the payload data that is to be transported via the outbound E3 data-stream to this input pin, in a serial manner. The XRT79L71 samples the data that is on this input pin upon the rising edge of either the RxOutClk for loop-timing applications or the TxInClk signal for local-timing applications. ***NOTE:*** *This signal is only active if the NibIntf input pin is pulled "Low".*
TxNib[3:0]	B8 C8 D8 A9	I	**Transmit Nibble-Parallel Payload Data Input Pins:** If the Transmit Payload Data Input Interface block is operated in the Nibble-Parallel Mode, then the System-Side equipment is expected to apply the payload data that is to be transported via the outbound E3 data-stream to these input pins, in a nibble-parallel manner. The XRT79L71 samples the data that is placed on these input pins upon the third rising edge of TxInClk, following a given rising edge of the TxNibClk output pin. ***NOTE:*** *These signals are only active if the NibIntf input pin is pulled "High".*
TxNibFrame	A10	O	**Transmit End of Frame Output Indicator - Nibble Mode:** The Transmit Section of the XRT79L71 pulses this output pin "High" for one nibble period whenever the Transmit Payload Data Input Interface block is processing the very last nibble within a given E3 frame. The purpose of this output pin is to alert the System-Side Terminal Equipment that is needs to begin transmission of a new E3 frame to the Transmit Payload Data Input Interface of the XRT79L71. ***NOTE:*** *This output pin is only active if the XRT79L71 has been configured to operate in the Nibble-Parallel Mode.*
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the Transmit Circuitry within the XRT79L71. Additionally, if the XRT79L71 has been configured to operate in both the Serial and Local-Timing Mode, then the XRT79L71 will sample the data, residing on the TxSer input pin, upon the rising edge of this input clock signal.
TxNibClk	D9	O	**Transmit Nibble-Mode Clock Output Signal:** If the XRT79L71 has been configured to operate in the Nibble-Parallel Mode, then the XRT79L71 will derive this output clock signal from either the TxInClk or the LIU Recovered Clock signal. ***NOTE:*** *The frequency of this clock output signal is one-fourth of the TxInClk or the RxOutClk signals.* The user is advised to update the Nibble-Parallel data via the TxNib[3:0] output pins, upon the rising edge of this clock output signal. The XRT79L71 will sample the TxNib[3:0] input pins upon the third rising edge of the TxInClk clock input signal, following a rising edge in this particular signal.

TABLE 36: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/ BALL #	TYPE	DESCRIPTION
TxOHInd/ TxGapClk	B9	O	**Transmit Overhead Bit Indicator Output/Transmit Gap-Clock Output:** The exact function of this output pin depends upon whether the XRT79L71 has been configured to operate in the Gapped-Clock Mode or Not. **Non-Gapped Clock Mode - TxOHInd:** This output pin will pulse "High" one bit period prior to the time that the Transmit Section of the XRT79L71 is processing an Overhead bit. This output pin will be held "Low" at all other times. The purpose of this output pin is to warn the System-Side Terminal Equipment that during the very next bit-period, the XRT79L71 is going to be processing an Overhead bits and will be, during this very next bit-period, ignoring any data that is applied to the TxSer input pin. **Gapped-Clock Mode - TxGapClk:** If the XRT79L71 has been configured to operate in the Gapped-Clock Mode, then this particular output pin will function as a demand output clock signal. In this case, the System-Side Terminal Equipment will be expected to update the data on the TxSer input pin, upon the rising edge of this particular output signal. The XRT79L71 will sample and latch the TxSer data, upon the falling edge of the TxGapClk signal. ***NOTE:*** *In the Gapped-Clock Mode, the XRT79L71 will only generate a clock edge via this output pin whenever the Transmit Payload Data Input Interface is processing payload data. The XRT79L71 will NOT generate a clock edge via this output pin whenever the Transmit Payload Data Input Interface is processing an overhead bit.*
TxFrame	B10	O	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT79L71 pulses this output pin "High" for one bit-period coincident to whenever the Transmit Payload Data Input Interface block is processing the last bit of a given E3 frame. The purpose of this output pin is to alert the System-Side Terminal Equipment that it needs to begin transmission of a new E3 frame to the Transmit Payload Data Input Interface block of the XRT79L71 (e.g., to permit the XRT79L71 to maintain Transmit E3 Framing alignment control over the System-Side Terminal Equipment).

TABLE 36: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/ BALL #	TYPE	DESCRIPTION
TxFrameRef	A11	I	**Transmit E3 Frame Reference Input:** The XRT79L71 permits the user to configure this input pin to function as the Transmit E3 Frame Generation Reference Input. If this particular configuration option is invoked, the TransmitE3 Framer block within the XRT79L71 will initiate E3 frame generation anytime it detects a rising edge of this input pin. ***NOTE:*** *To implement this configuration option, it is imperative that this particular input signal is synchronous with the TxInClk input signal. Failure to do so will result in the transmission of erred E3 frames to the remote terminal equipment.*
RxOutClk	B5	O	**Loop-Timing Reference Clock Output Pin:** If the XRT79L71 is configured to operate in the Loop-Timing Mode, then the Transmit Section of the XRT79L71 will be configured to use the LIU Recovered Clock signal as its timing source. In this case, the XRT79L71 will output a 34.368MHz clock signal via this particular output pin. In this configuration, the TxInClk signal will be inactive and will NOT be used to sample and latch the data on the TxSer input pin. In this case, the XRT79L71 will now be configured to sample the TxSer input pin upon the rising edge of the RxOutClk signal. ***NOTE:*** *This output pin will always be active, in the sense that it will always generate a 34.368MHz clock signal even when the XRT79L71 is NOT configured to operate in the Loop-Timing Mode. However, the XRT79L71 will only use this particular clock signal to sample and latch the data on the TxSer input pin whenever the XRT79L71 has been configured to operate in the Loop-Timing Mode.*

Operation of the Transmit Payload Data Input Interface Block

The Transmit Payload Data Input Interface block permits the user to configure it to operate in the following combination of modes.

- The Serial or Nibble-Parallel Interface Mode
- The Loop-Timing or Local-Timing Mode

If the XRT79L71 is configured to operate in the Local-Timing Mode, then there are two additional sub-options.

- The Frame-Master or Frame-Slave Mode

With these three sets of configuration options, the Transmit Payload Data Input Interface block can be configured to operate in any one of the following six (6) Modes.

- Mode 1 - Serial/Loop-Timing Mode
- Mode 2 - Serial/Local-Timing/Frame Slave Mode
- Mode 3 - Serial/Local-Timing/Frame Master Mode
- Mode 4 - Nibble-Parallel/Loop-Timing Mode
- Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode
- Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode

TABLE 37: A SUMMARY OF THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" MODES

MODE	NIBBLE-PARALLEL/ SERIAL MODE	SOURCE OF "SYSTEM-SIDE TERMINAL EQUIPMENT" CLOCK	FRAMING ALIGNMENT TIMING SOURCE
1	Serial	**Loop-Timing Mode:** The XRT79L71 will output a 34.368MHz clock signal via the "RxOutClk" output pin. This clock signal is ultimately derived from Recovered Line Clock (from Receive DS3/E3 LIU Block).	Asynchronous upon Power up.
2	Serial	**Local-Timing Mode:** The user is expected to apply a 34.368MHz clock signal to the TxInClk Input pin	TxFrameRef Input
3	Serial	**Local-Timing Mode:** Therefore, the user is expected to apply a 34.368MHz clock signal to the TxInClk Input pin	Asynchronous upon Power up
4	Nibble-Parallel	**Loop-Timing Mode:** Ther XRT79L71 will output an 8.592MHz "Nibble-Clock" signal (via the "TxNibClk" output). This clock signal is ultimately derived from the Recovered Line Clock (from the Receive DS3/E3 LIU Block).	Asynchronous upon Power up
5	Nibble-Parallel	**Local-Timing Mode:** Ther user is expected to apply a 34.368MHz clock to the TxInClk input pin. The XRT 79L71 will use the TxInClk signal to derive the 8.592MHz clock signal (which is output via the "TxNibClk" output pin).	TxFrameRef Input pin
6	Nibble-Parallel	**Local-Timing Mode:** The user is expected to apply a 34.368MHz clock signal to the TxInClk input pin. The XRT 79L71 will use the TxInClk signal to derive the 8.592MHz clock signal (which is output via the "TxNibClk" output pin).	Asynchronous upon Power up

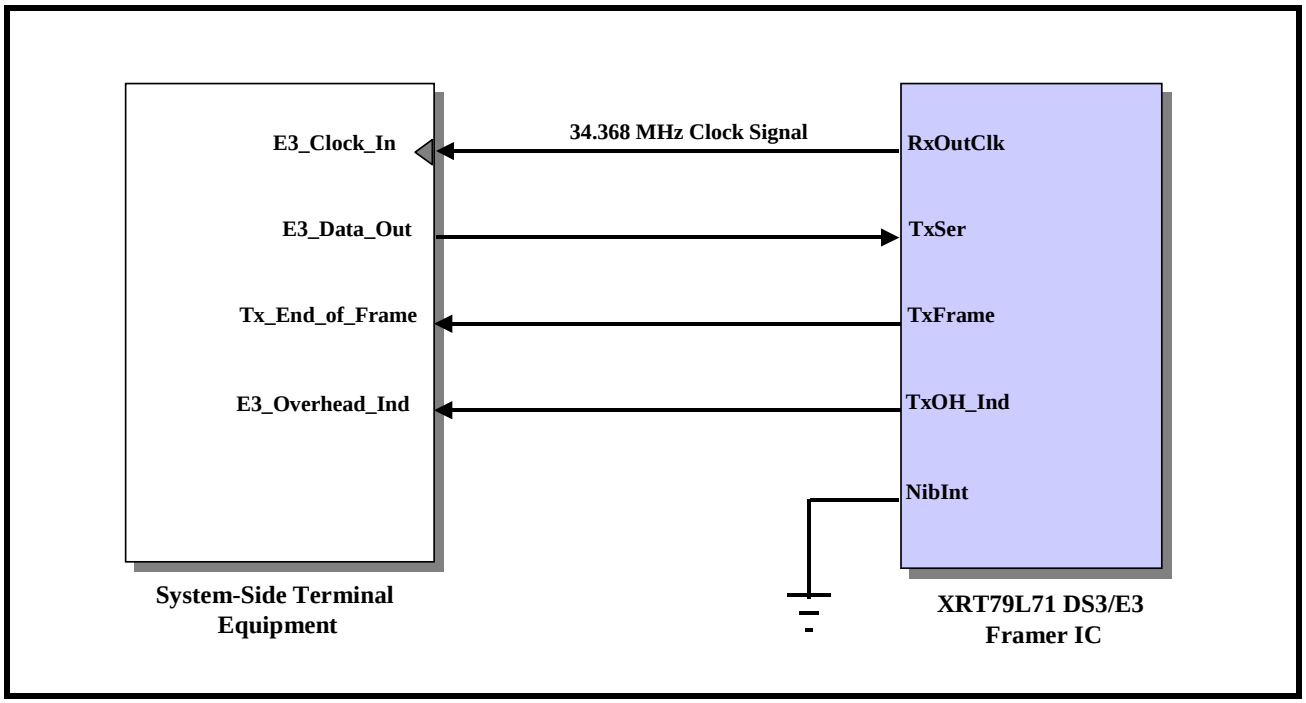
5.2.1.1 Mode 1 - Serial/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 1 then all of the following is true.

- The XRT79L71 will be configured to operate in the Loop-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal from the Receive E3 LIU Block as its timing source.
- Since the XRT79L71 is configured to operate in the Serial-Mode, it will sample and latch the data, being applied to the TxSer input pin upon the rising edge of the RxOutClk output signal.
- The XRT79L71 will pulse the TxFrame output pin "High" for one bit-period coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given E3 frame.

Figure 124 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 1 operation.

FIGURE 124. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR MODE 1 (SERIAL/LOOP-TIMING) OPERATION



Mode 1 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of the 34.368MHz clock signal via the RxOutClk output signal. This clock signal is used as the System-Side Terminal Equipment clock source by both the Transmit Payload Data Input Interface block of the XRT79L71 and the System-Side Terminal Equipment device or circuitry.

NOTE: *The "RxOutClk" signal is a buffered version of the "Recovered Line Clock" signal, from the Receive DS3/E3 LIU Block).*

The System-Side Terminal Equipment should serially output the payload data that is to be transported via the outbound E3 data-stream via its E3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out output pin upon the rising edge of the 34.368MHz clock signal at its E3_Clock_In input pin as depicted below in **Figure 125**.

The XRT79L71 will latch the contents of the TxSer input pin, upon the rising edge of the RxOutClk signal. The XRT79L71 will indicate that it is processing the very last bit of a given E3 frame by pulsing its TxFrame output pin "High" for one bit-period. The TxFrame output pin will have held "Low" at all other times. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_Start_of_Frame input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the E3_Data_Out output or the TxSer input pin.

Finally, the Transmit Payload Data Input Interface block within the XRT79L71 will indicate that it is about to process an overhead bit by pulsing the TxOH_Ind output pin "High" for one period prior to its processing. In **Figure 124**, the TxOH_Ind output pin of the XRT79L71 is connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment circuitry. Whenever the E3_Overhead_Ind input pin is pulsed "High" the System-Side Terminal Equipment is expected to NOT transmit an E3 payload bit upon the very next rising edge of E3_Clock_In. Instead, the System-Side Terminal Equipment is expected to delay its transmission of the very next payload bit by one RxOutClk clock period.

NOTES:

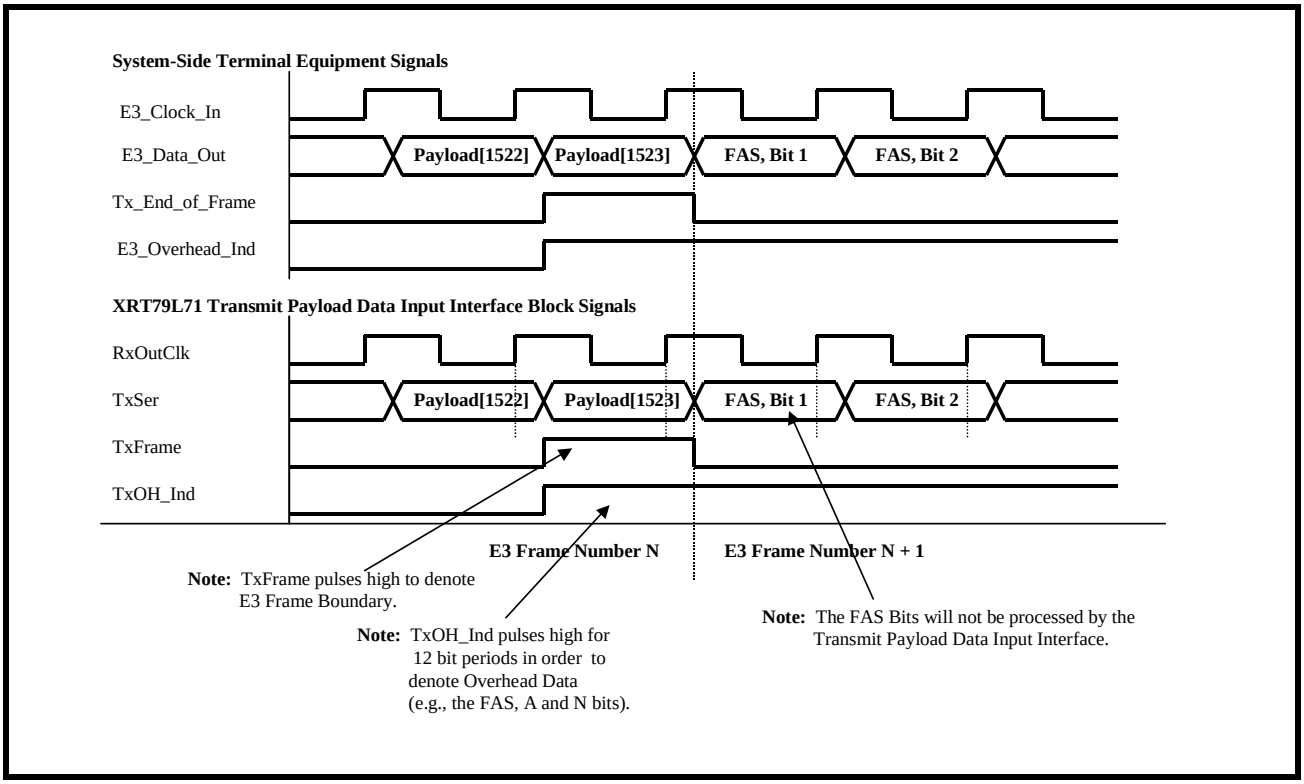
1. *Since the E3, ITU-T G.751 Frame consists of twelve consecutive overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High" it will do so for 12 consecutive bit-periods when processing the FAS, A and N*

bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of RxOutClk, and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".

2. *2.If toperating the "Transmit Payload Data Input Interface" block in the "Gapped-Clock" Mode, refer to* ***SEE"OPERATING THE TRANSMIT PAYLOAD DATA INPUT INTERFACE IN THE GAPPED CLOCK MODE" ON PAGE 288.***

Figure 125 presents an illustration of the Behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 1 Operation.

FIGURE 125. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 1 (SERIAL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 1 (Serial/Loop-Timing)

The user can configure the XRT79L71 to operate in Mode 1 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above inFigure 125.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

NOTE: *This step also configures the Receive Payload Data Output Interface to operate in the Serial Mode.*

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 0] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

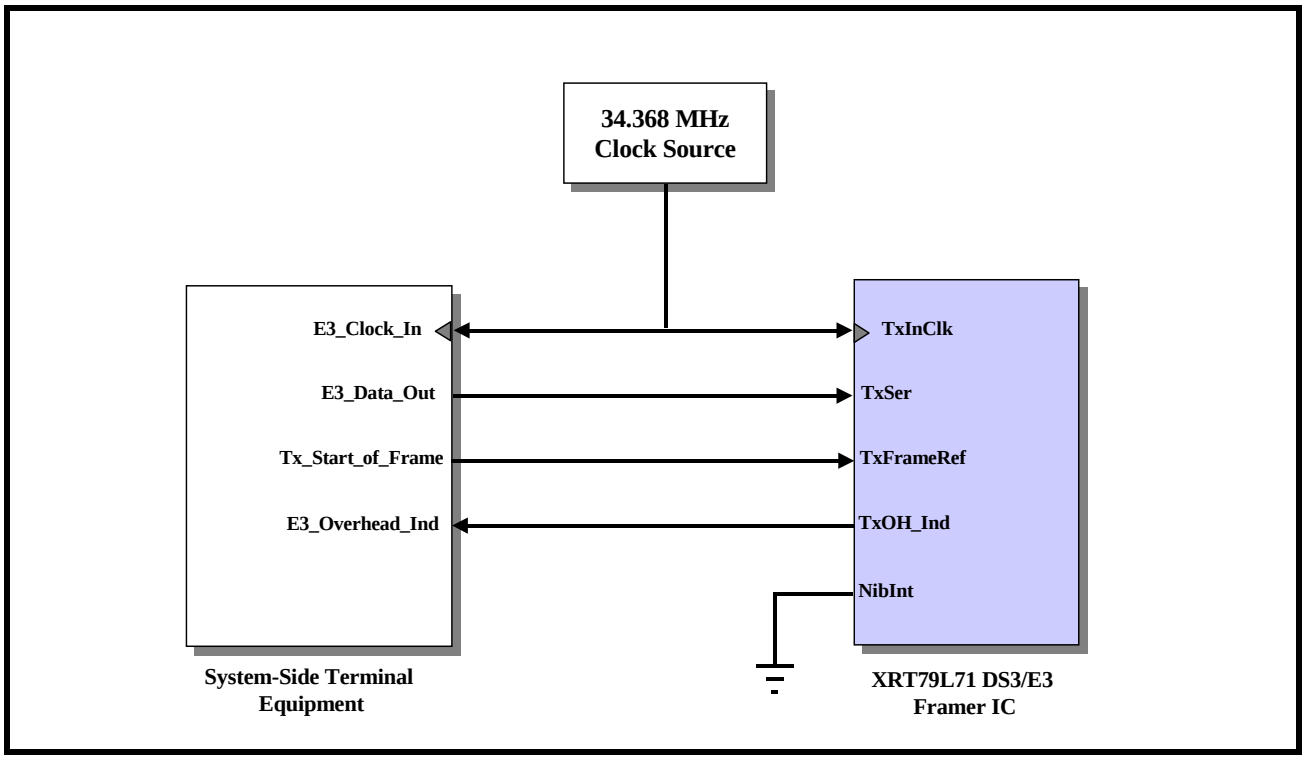
5.2.1.2 Mode 2 - Serial/Local-Timing/Frame Slave Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 2 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- The Transmit Section of the XRT79L71 will initiate the generate and transmission of a new E3 frame anytime it detects a rising edge at the TxFrameRef input pin.
- The XRT79L71 will still pulse the TxFrame output pin coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given E3 frame.

Figure 126 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 2 operation.

FIGURE 126. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 2 (SERIAL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Mode 2 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 34.368MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More specifically, this 34.368MHz clock signal will be applied to both the E3_Clock_In input of the System-Side Terminal Equipment and the TxInClk input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data that is to be transported via the outbound E3 data-stream via its E3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out output pin upon the rising edge of the 34.368MHz clock signal at its E3_Clock_In input pin as depicted below in **Figure 127**.

The XRT79L71 will latch the contents of the TxSer input pin, upon the rising edge of the TxInClk signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a Framing Reference signal to the XRT79L71 by pulsing its TxFrameRef input pin "High" for one bit-period, coincident with the first bit a new outbound E3 frame being applied to the TxSer input pin. Once the XRT79L71 detects the rising edge of the input at its TxFrameRef input pin, it will begin to generate and transmit a new E3 frame.

NOTES:

1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of Frame Generation and is referred to as the Frame Master. Since the XRT79L71 does not control or dictate the instant that it will generate a new E3 frame, but is dictated by the System-Side Terminal Equipment it is referred to as the Frame Slave.*
2. *It the XRT79L71 is configured to operate in the Mode 2 then it is imperative that the Tx_Start_of_Frame or TxFrameRef signal is synchronized to the TxInClk input clock signal. If the user fails to insure that the TxFrameRef input signal is synchronized to the "TxInClk" input clock signal, then the XRT 79L71 will transmit erred E3 data to the remote terminal equipment.*

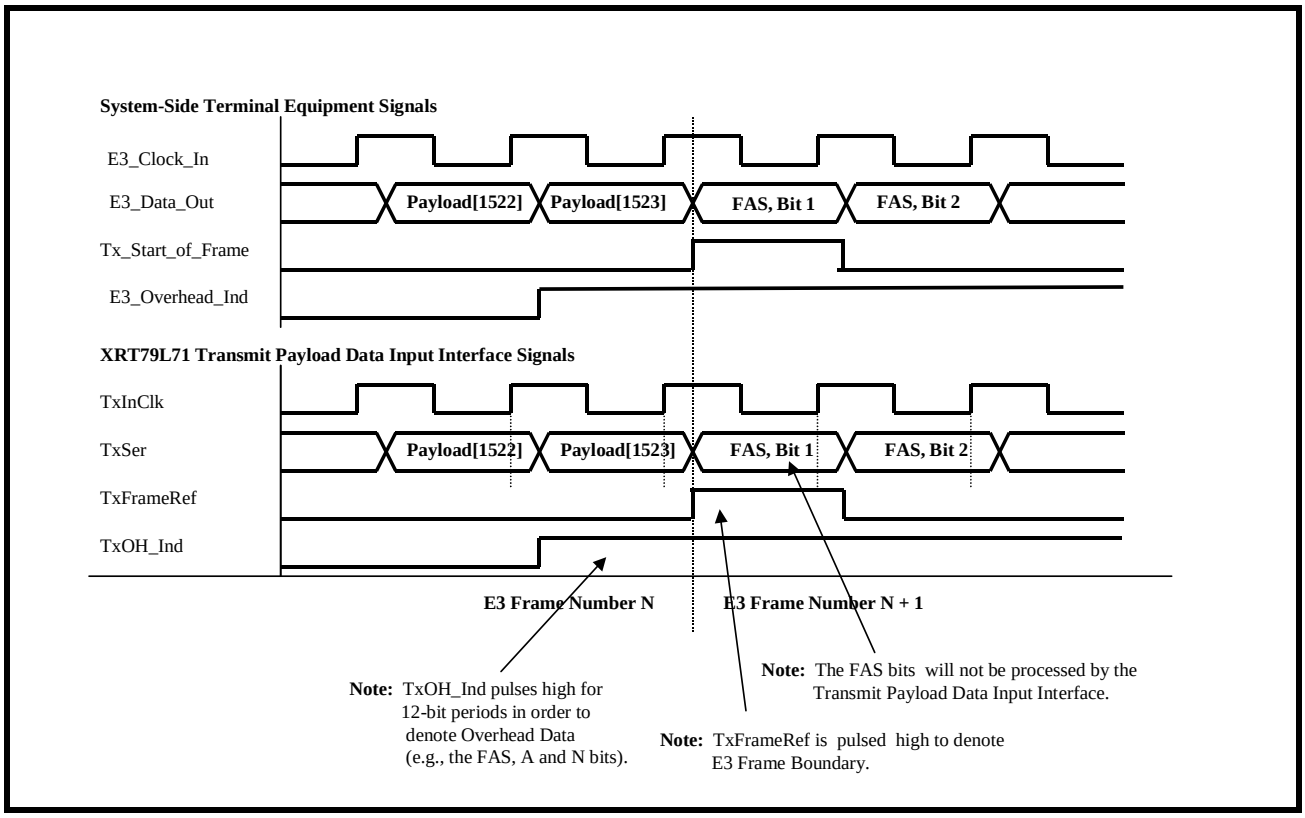
Finally, the XRT79L71 pulses its TxOHInd output pin "High" one bit-period prior to its processing a given overhead bit within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment, whenever the XRT79L71 pulses its TxOH_Ind output pin "High", it will also drive the E3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay transmission of the very next E3 payload bit by one TxInClk clock period.

NOTES:

1. *Since the E3, ITU-T G.832 Frame consists of 12 consecutive overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High" it will do so for 16 consecutive bit-periods when processing the FAS, A and N bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of TxInClk and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".*
2. *2.If configuring the Transmit Payload Data Input Interface block in the "Gapped Clock" Mode, trefer to Section 5.2.1.7*

Figure 127 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 2 Operation.

FIGURE 127. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 2 (SERIAL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 2 (Serial/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 2 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above inFigure 127.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the NibIntf input pin to a logic "Low".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 1] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	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

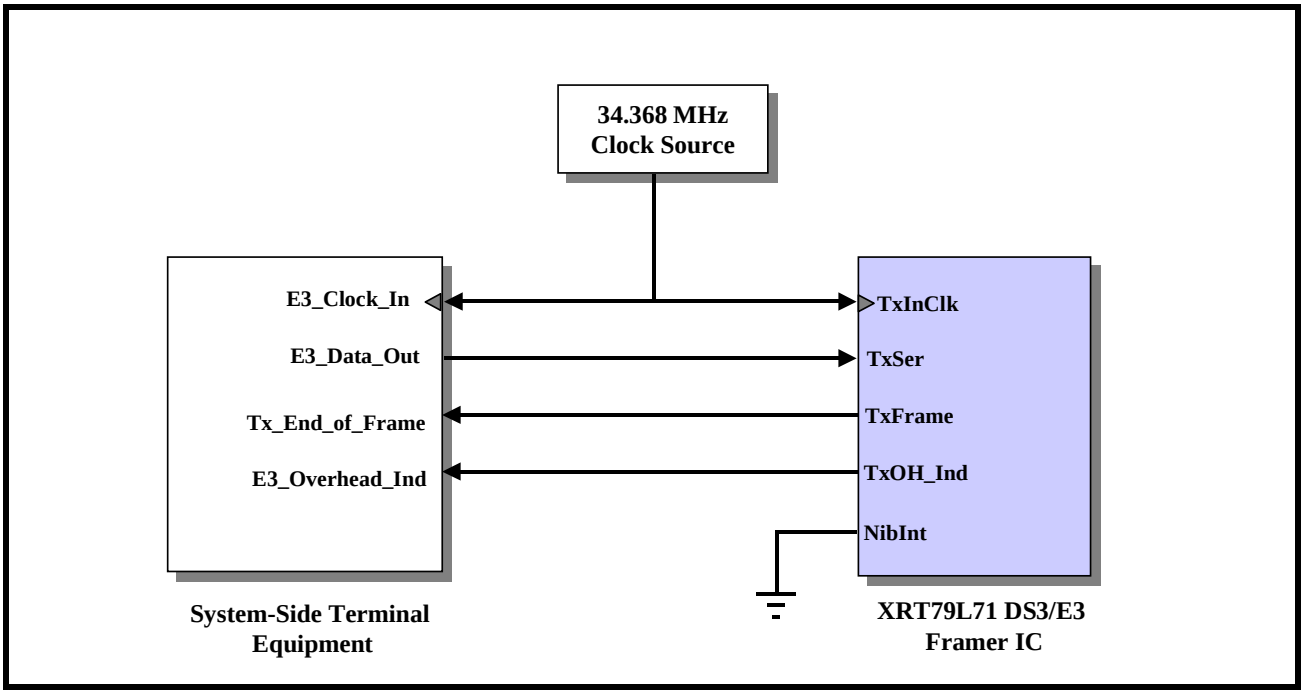
5.2.1.3 Mode 3 - Serial/Local-Timing/Frame Master Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in Mode 3, then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- Since the XRT79L71 is configured to operate in the Serial-Mode, it will sample and latch the data, being applied to the TxSer input pin upon the rising edge of the TxInClk input signal.
- The XRT79L71 will still pulse the TxFrame output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given E3 frame.

Figure 128 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 3 operation.

FIGURE 128. AN ILLUSTRATION AS TO HOW SHOULD INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Mode 3 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 34.368MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More specifically, this 34.368MHz clock signal will be applied to both the E3_Clock_In input of the System-Side Terminal Equipment and the TxInClk input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data that is to be transported via the outbound E3 data-stream via its E3_Data_Out output pin. The user is advised to design the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out output pin upon the rising edge of the 34.386MHz clock signal at its E3_Clock_In input pin as depicted below in Figure 129.

The XRT79L71 will latch the contents of the TxSer input pin upon the rising edge of the TxInClk signal.

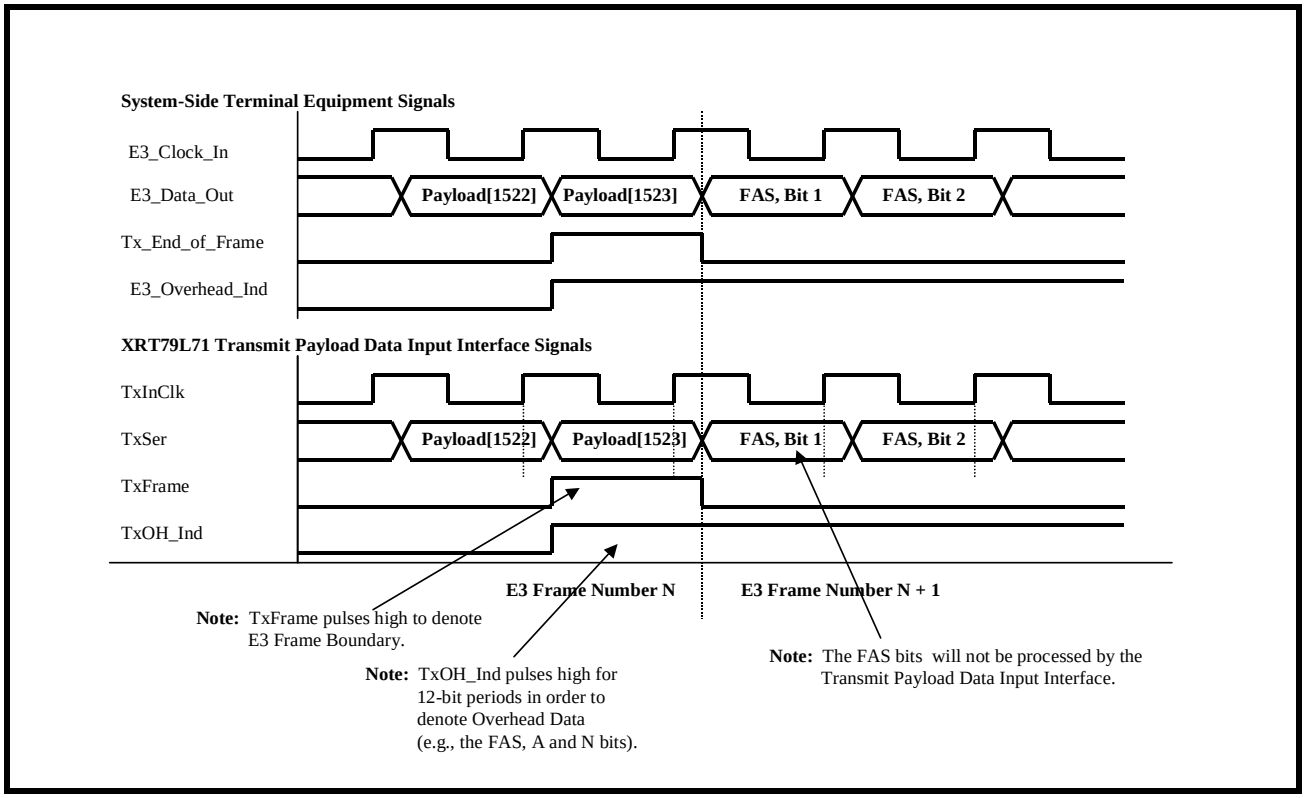
Finally, the XRT79L71 pulses its TxOH_Ind output pin "High" one bit-period prior to it processing a given overhead bit within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment, whenever the XRT79L71 pulses its TxOH_Ind output pin "High", it will also drive the E3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay transmission of the very next E3 payload bit by one TxInClk clock period.

NOTES:

1. *Since the E3, ITU-T G.751 Frame consists of 12 consecutive overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High" it will do so for 12 consecutive bit-periods when processing the FAS, A and and N bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of TxInClk and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".*
2. *In this particular mode, the Transmit Direction circuitry (within the XRT 79L71) is dictating the instant that it will initiate the generation of a new E3 frame; and is referred to as the "Frame Master". As a consequence, this particular mode is referred to as the "Frame Master" Mode.*
3. *In contrast to "Mode 2" operation, if the XRT 79L71 is configured to operate in "Mode 3", then it is NOT imperative that the "TxFrameRef" input be synchronized to the "TxInClk" input clock signal. In this case, we recommend that the user tie the "TxFrameRef" input pin to GND.*
4. *If operating the Transmit Payload Data Input Interface block in the "Gapped-Clock" Mode, trefer to* ***SEE"OPERATING THE TRANSMIT PAYLOAD DATA INPUT INTERFACE IN THE GAPPED CLOCK MODE" ON PAGE 288.****.*

Figure 129 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 3 Operation.

FIGURE 129. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 3 (Serial/Local-Timing/Frame-Master Mode)

The user can configure the XRT79L71 to operate in Mode 3 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 129.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode.

This can be accomplished by setting the NibIntf input pin to a logic "Low".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [1, X] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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

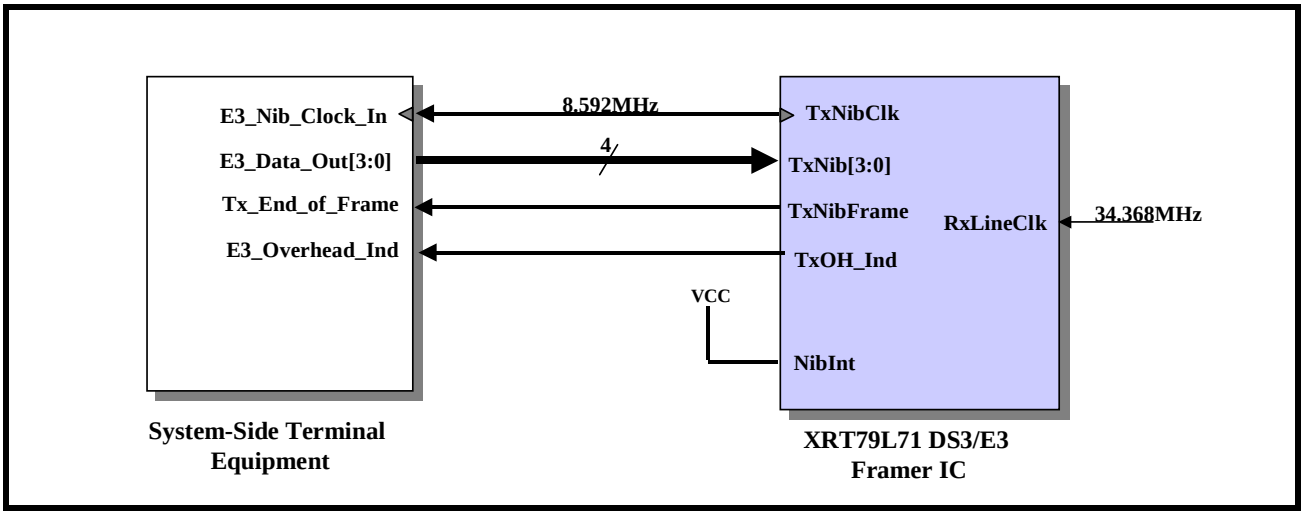
5.2.1.4 *Mode 4 - Nibble-Parallel/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block*

If the XRT79L71 is configured to operate in Mode 4 then all of the following is true.

- The XRT79L71 will be configured to operate in the Loop-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal from the Receive E3 LIU Block as its timing source.
- In this mode, the XRT79L71 will use the LIU Recovered Clock signal to derive the TxNibClk signal.
- For E3 Applications, the TxNibClk frequency is exactly one-fourth of the LIU Recovered Clock signal or 8.592MHz.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pins upon the third rising edge of the RxOutClk output clock signal, following a given rising edge of the TxNibClk output clock signal.
- The XRT79L71 will pulse the TxNibFrame output pin "High" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble of a given E3 frame.

Figure 130 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 4 operation.

FIGURE 130. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMING) OPERATION



Mode 4 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of both the 34.368MHz clock signal via the RxOutClk output signal and a Nibble Clock signal via the TxNibClk output signal.

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound E3 data-stream in a Nibble-Parallel manner via its E3_Data_Out[3:0] output pins. The user is advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its E3_Nib_Clock_In input pin as depicted below in **Figure 131**.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the RxOutClk signal following a given rising edge in the TxNibClk signal. The XRT79L71 will indicate that it is processing the very last nibble of a given E3 frame by pulsing its TxNibFrame output pin "High" for one nibble-period. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_Start_of_Frame input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the E3_Data_Out[3:0] output or TxNib[3:0] input pins.

The Transmit Payload Data Input Interface block's handling of E3 Overhead bits when configured to operate in the Nibble-Parallel Mode

In contrast to the DS3 Framing formats which are bit-oriented framing formats, the E3, ITU-T G.751 framing format can be viewed as a nibble-oriented framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface within the XRT79L71 will be processing an E3 overhead nibble, and the TxOH_Ind output pin in this case DOES have meaning. In Mode 4 Operation, the XRT79L71 will pulse its TxOH_Ind output pin "High" one nibble-period prior to the instant that it will process a given Overhead nibble within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment, whenever the XRT79L71 pulses its TxOH_Ind output pin "High", it will also drive the E3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay the transmission of the very next E3 payload nibble by one TxNibClk clock period.

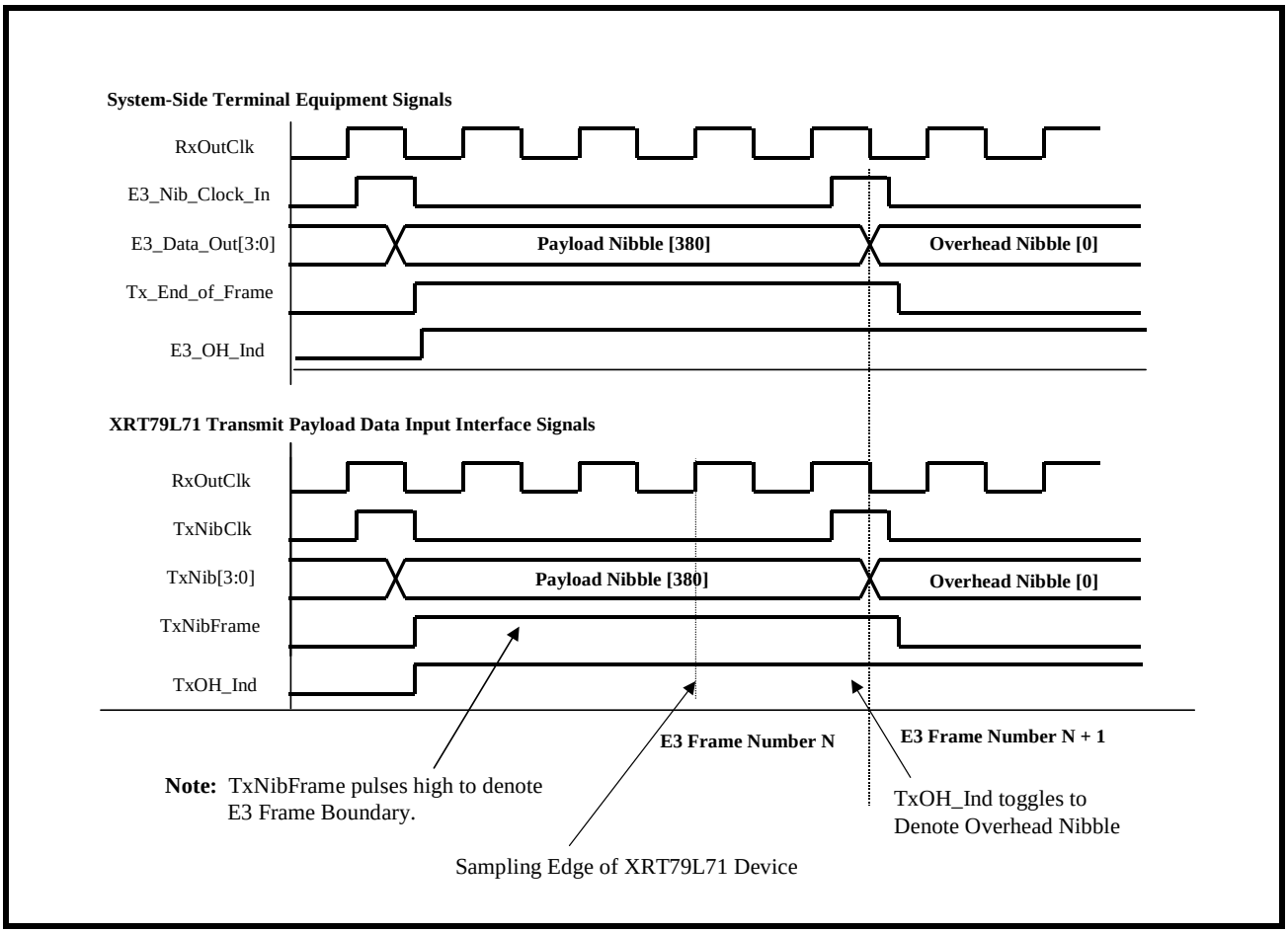
NOTE: *Since the E3, ITU-T G.751 Frame consists of 12 consecutive overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High", it will do so for three (3) consecutive nibble-periods when processing the FAS, A and N bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of TxNibClk and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the RxOutClk signal.

Figure 131 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 4 Operation.

FIGURE 131. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 4 (Nibble-Parallel/Loop-Timing) Mode

The user can configure the XRT79L71 to operate in Mode 4 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 131.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 0] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	0	0

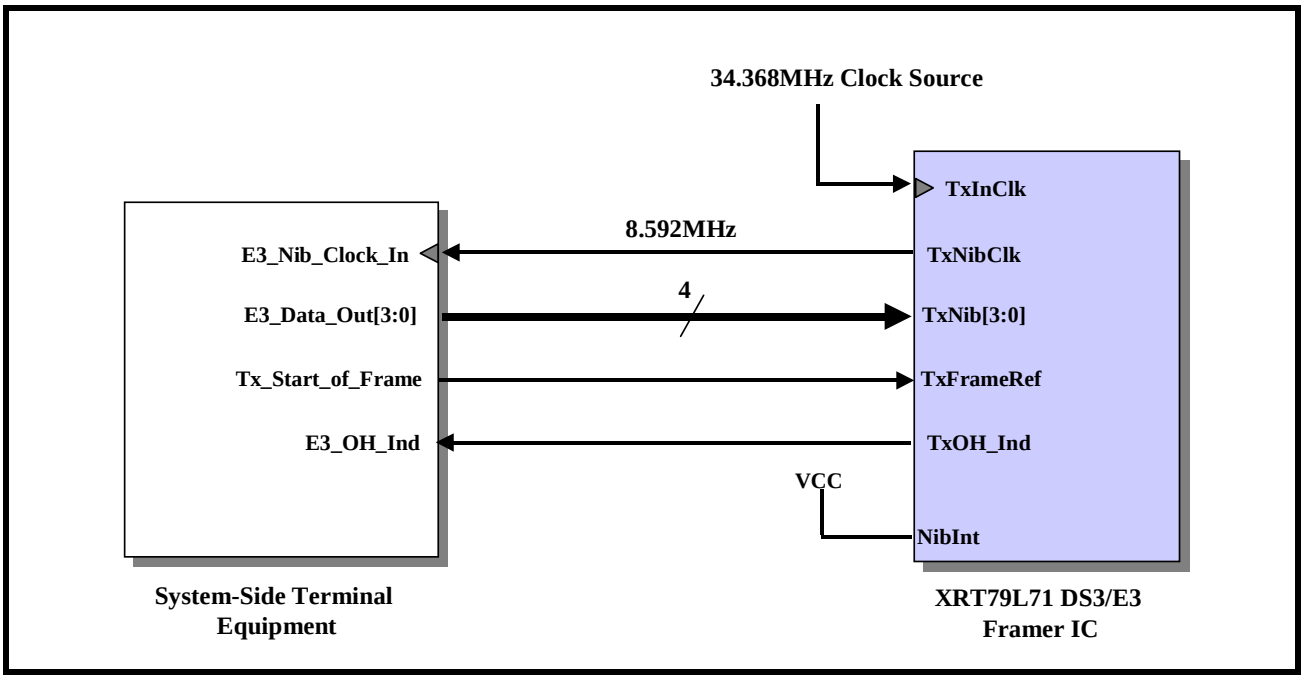
5.2.1.5 *Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode Operation for the Transmit Payload Data Input Interface Block*

If the XRT79L71 is configured to operate in Mode 5 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal has its timing source.
- In this mode, the XRT79L71 will use the TxInClk signal to derive the TxNibClk signal.
- For E3 Applications, the TxNibClk frequency is exactly one-fourth of the TxInClk clock input signal or 8.592MHz.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pins upon the third rising edge of the TxInClk input clock signal, following a given rising edge of the TxNibClk output clock signal.
- The Transmit Section of the XRT79L71 will initiate the generate and transmission of a new E3 frame anytime it detects a rising edge in the TxFrameRef input pin.
- The XRT79L71 will pulse the TxNibFrame output pin "High" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble within a given E3 frame.

Figure 132 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 5 operation.

FIGURE 132. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Mode 5 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble Clock signal via the TxNibClk output signal.

NOTE: *For "Mode 5" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound E3 data-stream in a Nibble-Parallel manner via its E3_Data_Out[3:0] output pins. The user is advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its E3_Nib_Clock_In input pin as depicted below in **Figure 133**.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the TxInClk signal following a given rising edge in the TxNibClk signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a Framing Reference signal to the XRT79L71 by pulsing its TxFrameRef input pin "High" for one nibble-period, coincident with the first nibble of a new outbound E3 frame being applied to the TxNib[3:0] input pins. Once the XRT79L71 detects the rising edge of the input at its TxFrameRef input pin, it will begin to generate and transmit a new E3 frame.

NOTES:

1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of Frame Generation and is referred to as the Frame Master. Since the XRT79L71 does not control nor dictate the instant that it will generate a new E3 frame, but is driven by the System-Side Terminal Equipment, it is referred to as the Frame Slave.*
2. *If the XRT79L71 is configured to operate in Mode 5, then it is imperative that the Tx_Start_of_Frame or TxFrameRef signal is synchronized to the TxInClk input clock signal. Failure to do this will result in the transmission of erred E3 data to the remote terminal equipment.*

The Transmit Payload Data Input Interface block's handling of E3 Overhead Bytes when configured to operate in the Nibble-Parallel Mode

In contrast to the DS3 Framing formats which are bit-oriented framing formats, the E3, ITU-T G.751 framing format can be viewed as being a nibble-oriented framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface within the XRT79L71 will be processing an E3 overhead

nibble, and the TxOH_Ind output pin, in this case DOES have meaning. In Mode 5 Operation, the XRT79L71 will pulse its TxOH_Ind output pin "High" one nibble-period prior to the instant that it will process a given Overhead nibble within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment, whenever the XRT79L91 device pulses its TxOH_Ind output pin "High", it will also drive the E3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay the transmission of the very next E3 payload nibble by one TxNibClk clock period.

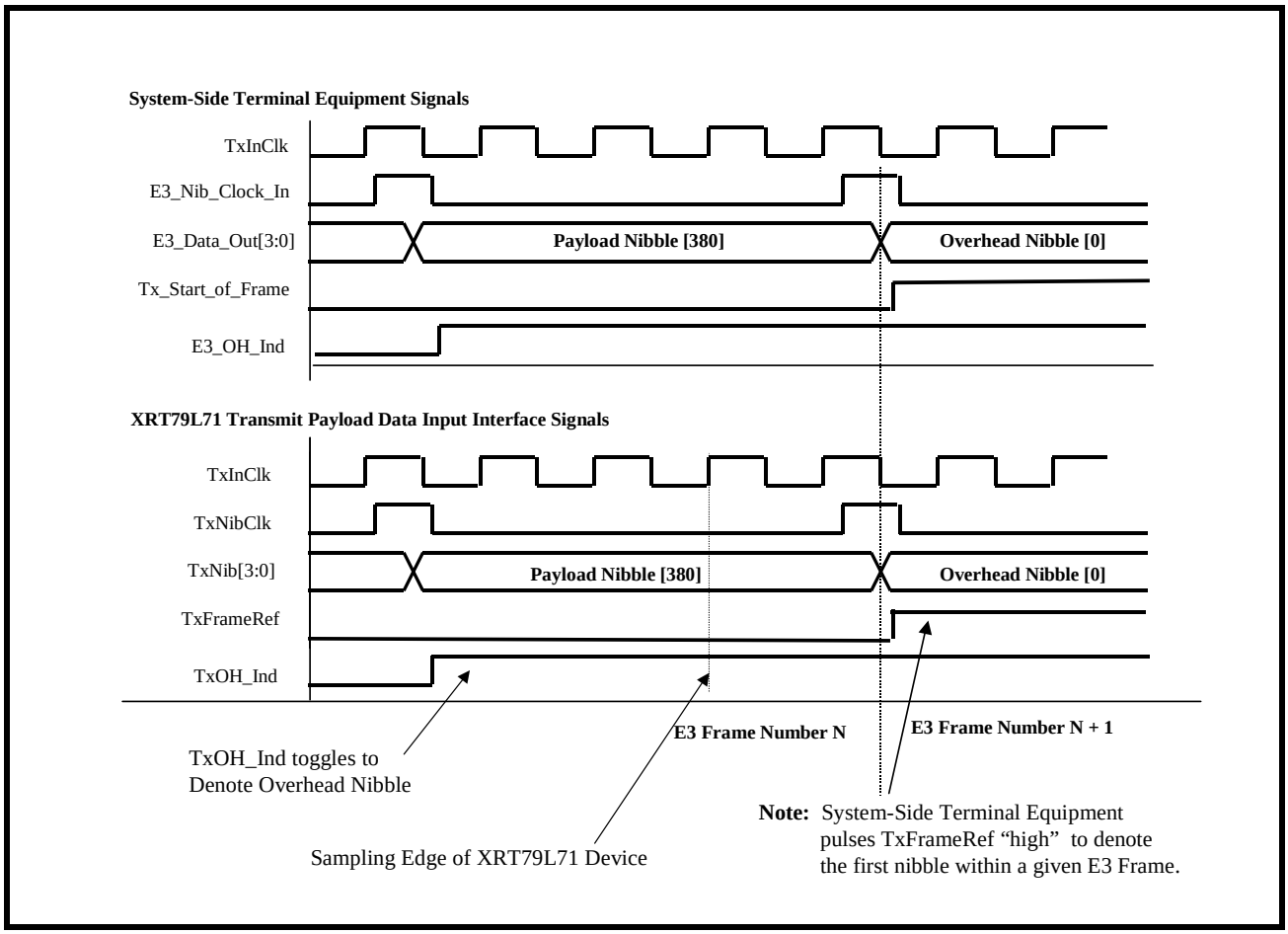
NOTE: *Since the E3, ITU-T G.751 Frame consists of 12 overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High" it will do so for three (3) consecutive nibble-periods when processing the FAS, A and N bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of TxNibClk and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the TxInClk signal.

Figure 133 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 5 Operation.

FIGURE 133. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 5 (Nibble-Parallel/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 5 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 133.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [0, 1] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	0	1

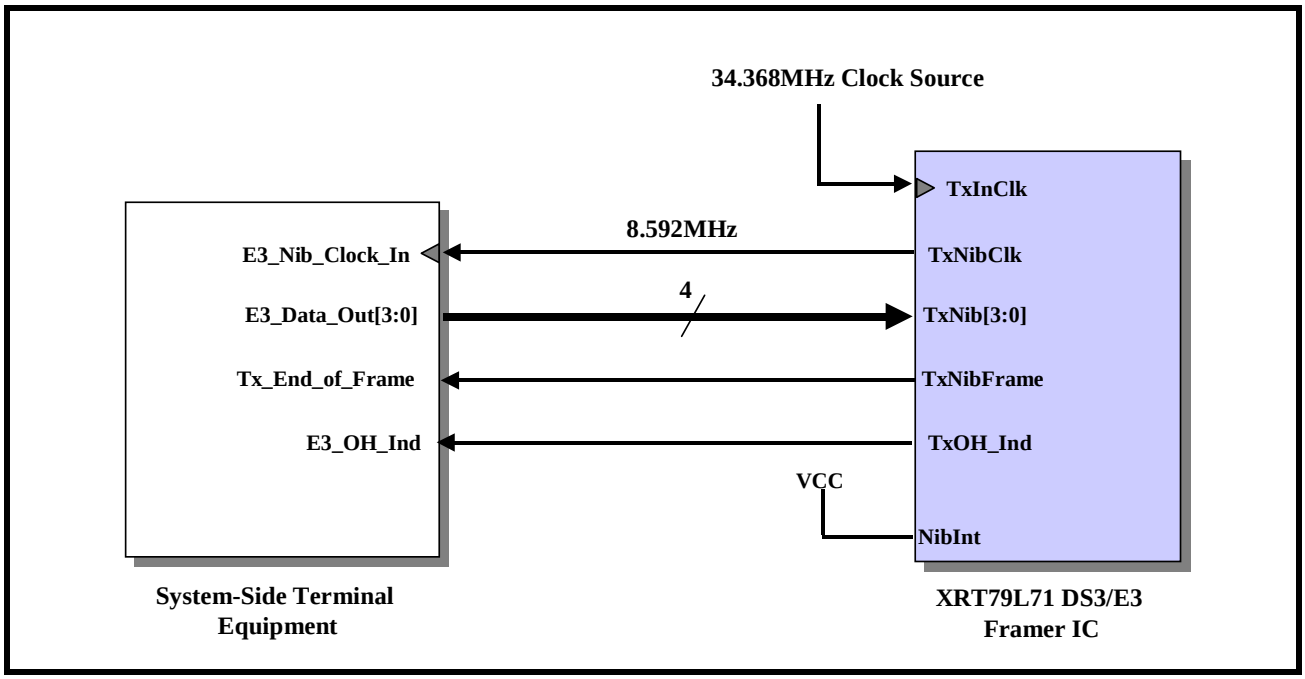
5.2.1.6 *Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode Operation for the Transmit Payload Data Input Interface Block*

If the XRT79L71 is configured to operate in Mode 6 then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the TxInClk input signal as its timing source.
- In this mode, the XRT79L71 will use the TxInClk signal to derive the TxNibClk signal.
- For E3 Applications, the TxNibClk frequency is exactly one-fourth of the TxInClk clock signal or 8.592MHz.
- Since the XRT79L71 is configured to operate in the Nibble-Parallel Mode, it will sample and latch the data, being applied to the TxNib[3:0] input pin upon the third rising edge of TxInClk input clock signal, following a given rising edge of the TxNibClk output clock signal.
- The XRT79L71 will pulse the TxNibFrame output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given E3 frame.

Figure 134 presents an illustration of how to Interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block of the XRT79L71 for Mode 6 operation.

FIGURE 134. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Mode 6 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble-Clock signal via the TxNibClk output signal.

NOTE: *For "Mode 6" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The System-Side Terminal Equipment should output the payload data that is to be transported via the outbound E3 data-stream in a Nibble-Parallel manner via its E3_Data_Out[3:0] output pins. The user is advised to design or configure the System-Side Terminal Equipment circuitry such that it will update the data via the E3_Data_Out[3:0] output pins upon the rising edge of the TxNibClk clock signal at its E3_Nib_Clock_In input pins as depicted below in **Figure 135**.

The XRT79L71 will latch the contents of the TxNib[3:0] input pins, upon the third rising edge of the TxInClk signal following a given rising edge in the TxNibClk signal. The XRT79L71 will indicate that it is processing the very last nibble of a given E3 frame by pulsing its TxNibFrame output pin "High" for one nibble-period. Whenever the System-Side Terminal Equipment detects this pulse at its Tx_End_of_Frame input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the E3_Data_Out[3:0] output or TxNib[3:0] input pins.

NOTES:

1. *In this particular mode, the Transmit Direction circuitry (within the XRT 79L71) dictates the instant that it will generate a new E3 frame; and is (therefore) referred to as the "Frame Master". As a consequence, this particular mode is referred to as the "Frame Master" Mode.*
2. *In contrast to "Mode 5" operation, if the XRT 79L71 is configured to operate in "Mode 6", then it is NOT imperative that the "TxFrameRef" input be synchronized to the "TxInClk" clock signal. In this case, we recommend that the user tie "TxFrameRef" to GND.*

The Transmit Payload Data Input Interface block's handling of E3 Overhead Bytes when configured to operate in the Nibble-Parallel Mode

In contrast to the DS3 Framing formats, which are bit-oriented framing formats, the E3, ITU-T G.751 framing format can be viewed as a nibble-oriented framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface within the XRT79L71 will be processing an E3 overhead nibble, and the

TxOH_Ind output pin, in this case DOES have meaning. In Mode 6 Operation, the XRT79L71 will pulse its TxOH_Ind output pin "High" one nibble-period prior to the instant that it will process a given Overhead nibble within the outbound E3 frame. Since the TxOH_Ind output pin of the XRT79L71 is electrically connected to the E3_Overhead_Ind input pin of the System-Side Terminal Equipment, whenever the XRT79L91 device pulses its TxOH_Ind output pin "High", it will also drive the E3_Overhead_Ind input pin of the System-Side Terminal Equipment "High". Whenever the System-Side Terminal Equipment detects this pin toggling "High" it should delay the transmission of the very next E3 payload nibble by one TxNibClk clock period.

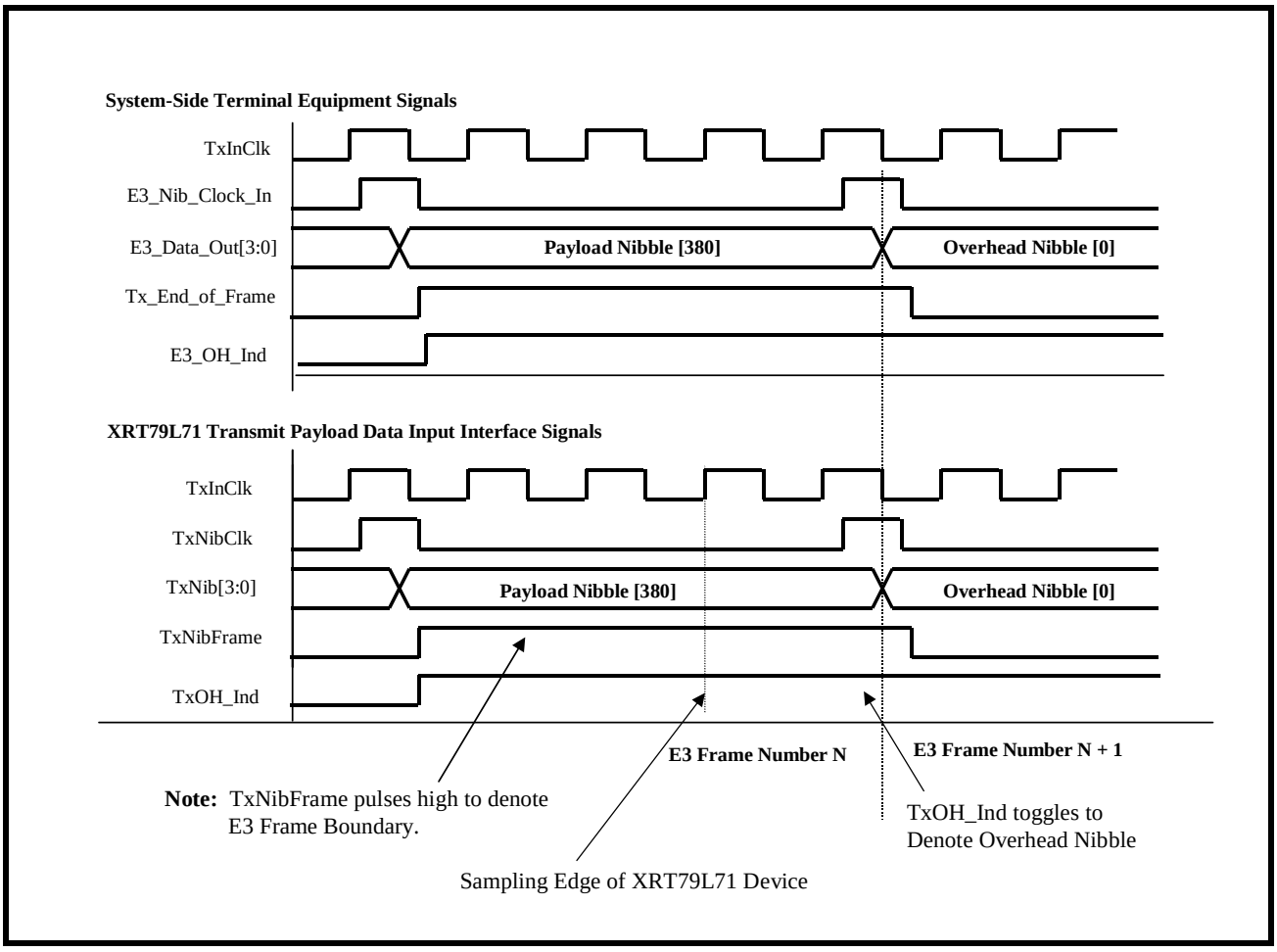
NOTE: *Since the E3, ITU-T G.751 Frame consists of 12 overhead bits, whenever the TxOH_Ind output pin of the XRT79L71 pulses "High" it will do so for three (3) consecutive nibble-periods when processing the FAS, A and N bits. Therefore, for the E3, ITU-T G.751 framing format, whenever the System-Side Terminal Equipment detects the TxOH_Ind output pin being pulled "High", it is expected to (1) continuously sample the state of the TxOH_Ind output pin with each rising edge of TxNibClk and (2) to NOT transmit an E3 payload bit to the Transmit Payload Data Input Interface block until it samples the TxOH_Ind output pin toggling "Low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the TxInClk signal.

Figure 135 presents an illustration of the behavior of the System-Side Terminal Equipment/Transmit Payload Data Input Interface signals for Mode 6 Operation.

FIGURE 135. AN ILLUSTRATION OF THE BEHAVIOR OF THE SYSTEM-SIDE TERMINAL EQUIPMENT SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 6 (Nibble-Parallel/Local-Timing/Frame Master Mode)

The user can configure the XRT79L71 to operate in Mode 6 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 135.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the NibIntf input pin to a logic "High".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to [1, X] as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	1	X

5.2.1.7 Operating the Transmit Payload Data Input Interface in the Gapped Clock Mode

In contrast to the DS3 Mode, if the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Mode, then the Transmit Payload Data Input Interface block can be configured to operate in either the Gapped-Clock or the Non-Gapped Clock Mode as described below.

5.2.1.7.1 Operating the Transmit Payload Data Input Interface in the Serial/Gapped-Clock Mode

This particular section discusses both the Non-Gapped Clock and the Gapped-Clock Modes of operation of the Transmit Payload Data Input Interface block.

If the Transmit Payload Data Input Interface block has been configured to operate in the Non-Gapped Clock Mode

If the Transmit Payload Data Input Interface (within the XRT79L71) has been configured to operate in any one of the Serial Modes (e.g., Modes 1 through 3, as described in **SEE"MODE 1 - SERIAL/LOOP-TIMING MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 271.**, **SEE"MODE 2 - SERIAL/LOCAL-TIMING/FRAME SLAVE MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 274.** and **SEE"MODE 3 - SERIAL/LOCAL-TIMING/FRAME MASTER MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 276.**), then we have recommended that the user design or configure their System-Side Terminal Equipment to do the following, when supplying payload data to the TxSer input pin.

- Check the state of the TxOH_Ind output pin (from the XRT79L71) upon the falling edge of either the TxInClk or the RxOutClk signal.
- Perform either of the following actions, depending upon the sampled state of the TxOH_Ind output pin, as described below.

If TxOH_Ind is sampled "Low"

Then the System-Side Terminal Equipment should proceed to place the very next payload bit on the TxSer input pin upon the very next rising edge of either the TxInClk or RxOutClk signal.

If TxOH_Ind is sampled "High"

Then the System-Side Terminal Equipment should NOT proceed to place the very next payload bit on the TxSer input pin, upon the very next rising edge of either the TxInClk or RxOutClk signal. In this case, the

System-Side Terminal Equipment should hold (or NOT advance the very next payload bit) to the TxSer input pin (of the XRT79L71) until it samples the TxOH_Ind output "Low" once again.

In this particular approach, the user must design in the appropriate State Machine circuitry within the System-Side Terminal Equipment in order to properly respond to the state of the TxOH_Ind output pin, while providing the payload data to the Transmit Payload Data Input Interface. While designing such a State Machine into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode.

If the Transmit Payload Data Input Interface block has been configured to operate in the Gapped-Clock Mode

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the role of the TxOH_Ind output pin will change from being the Overhead Indicator output pin, to now being a demand-clock output pin. In other words, If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then it (the Transmit Payload Data Input Interface block) will generate a clock pulse (via the TxOH_Ind output pin) if and only if it is ready to accept and process a payload bit. If the Transmit Payload Data Input Interface block is about to process an overhead bit, then it will not generate a clock pulse via the TxOH_Ind output pin. This action will result in the Transmit Payload Data Input Interface block generating a gapped clock signal via the TxOH_Ind output pin (hence the term Gapped-Clock Mode).

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the System-Side Terminal Equipment will be expected to update the data on the TxSer input pin (of the XRT79L71) upon the rising edge of the TxOH_Ind output signal. The XRT79L71 will sample and latch the TxSer data, upon the falling edge of the TxOH_Ind output signal. In this case, there is no need to check the state of a certain output pin, and then gate the placement of the next payload bit (on the TxSer input pin) with the sampled state of this particular signal. The System-Side Terminal Equipment only needs to respond to the rising edge of this particular Gapped-Clock signal.

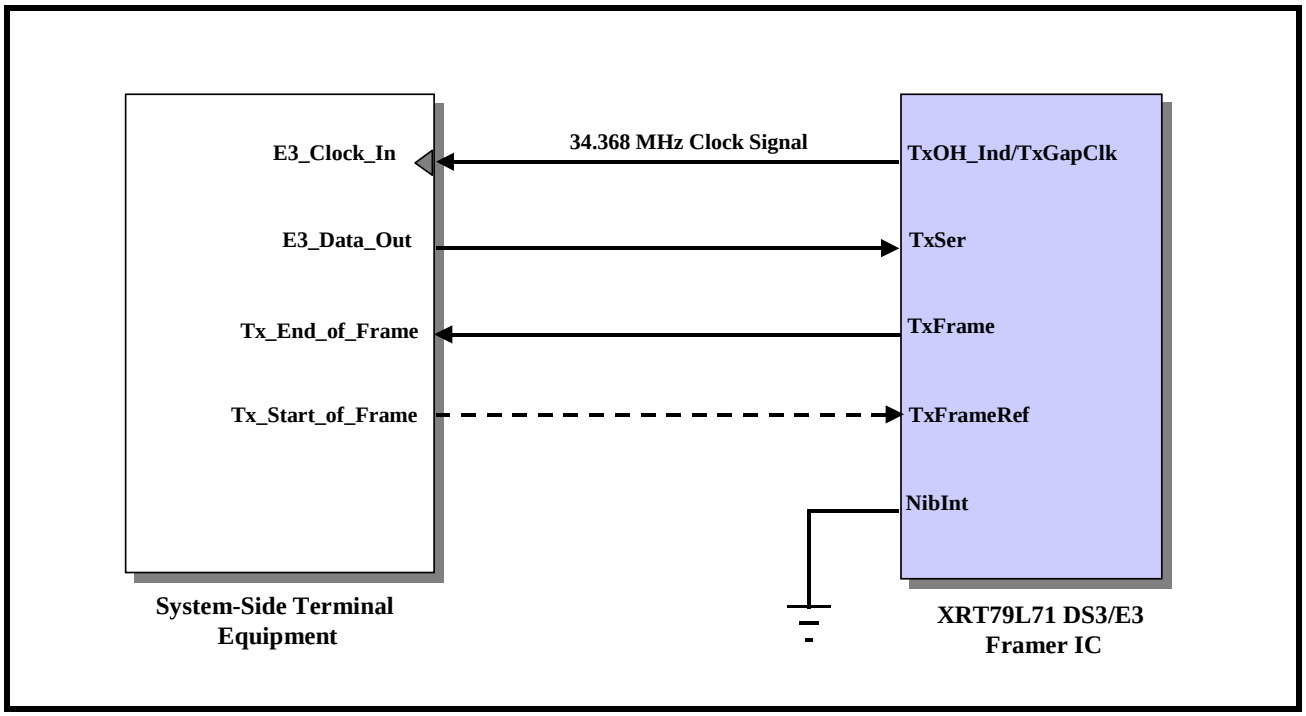
If the XRT79L71 has been configured to operate in the Loop-Timing Mode (e.g., Mode 1), then this Gapped-Clock signal (from the TxOH_Ind output pin) will be derived from the LIU Recovered Clock signal (from the Receive DS3 LIU Block). Similarly, if the XRT79L71 has been configured to operate in the Local-Timing Mode, then this Gapped-Clock signal will be derived form the TxInClk input signal.

Configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode

To configure the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode, do all of the following.

STEP 1 - the user should interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block, in a manner as indicated below in Figure 136.

FIGURE 136. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK (OF THE XRT79L71) FOR GAPPED-CLOCK MODE OPERATIONS



STEP 2 - Set Bit 5 (TxGapped Clock Mode Enable), within the Framer Test Register to "1" as depicted below.

Framer Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	0	1	0	0	0	0	0

NOTE: *This setting only configures the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode. Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode is discussed separately in* ***SEE"OPERATING THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK HAS BEEN CONFIGURED TO OPERATE IN THE "GAPPED-CLOCK" MODE" ON PAGE 418.***.

5.2.1.7.2 Operating the Transmit Payload Data Input Interface in the Nibble-Parallel/Gapped-Clock Mode

This particular section discusses both the Non-Gapped Clock and the Gapped-Clock Modes of operation of the Transmit Payload Data Input Interface block.

If the Transmit Payload Data Input Interface block has been configured to operate in the Non-Gapped Clock Mode

If the Transmit Payload Data Input Interface within the XRT79L71 has been configured to operate in any one of the Nibble-Parallel Modes (e.g., Modes 4 through 6, as described in **SEE"MODE 4 - NIBBLE-PARALLEL/ LOOP-TIMING MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK"**

ON PAGE 279., SEE"MODE 5 - NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE MODE OPERATION FOR THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 282. and SEE"MODE 6 - NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER MODE OPERATION FOR THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 285.), then we have recommended that the user design or configure their System-Side Terminal Equipment to do the following, when supplying payload data to the TxNib[3:0] input pins.

- Check the state of the TxOH_Ind output pin (from the XRT79L71) upon the falling edge of the TxNibClk signal.
- Perform either of the following actions, depending upon the sampled state of the TxOH_Ind output pin, as described below.

If TxOH_Ind is sampled "Low"

Then the System-Side Terminal Equipment should proceed to place the very next payload nibble on the TxNib[3:0] input pins upon the very next rising edge of either the TxNibClk signal.

If TxOH_Ind is sampled "High"

Then the System-Side Terminal Equipment should NOT proceed to place the very next payload nibble on the TxNib[3:0] input pins, upon the very next rising edge of either the TxNibClk signal. In this case, the System-Side Terminal Equipment should hold (or NOT advance the very next payload nibble) to the TxNib[3:0] input pin of the XRT79L71 until it samples the TxOH_Ind output "Low" once again.

In this particular approach, the user must design in the appropriate State Machine circuitry within the System-Side Terminal Equipment in order to properly respond to the state of the TxOH_Ind output pin, while providing the payload data to the Transmit Payload Data Input Interface. While designing such a State Machine into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode.

If the Transmit Payload Data Input Interface block has been configured to operate in the Gapped-Clock Mode

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the role of the TxOH_Ind output pin will change from being the Overhead Indicator output pin to now being a demand-clock output pin. In other words, If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the Transmit Payload Data Input Interface block will generate a clock pulse via the TxOH_Ind output pin, if and only if it is ready to accept and process a payload nibblet. If the Transmit Payload Data Input Interface block is about to process an overhead bit, then it will not generate a clock pulse via the TxOH_Ind output pin. This action will result in the Transmit Payload Data Input Interface block generating a gapped clock signal via the TxOH_Ind output pin, hence the term Gapped-Clock Mode.

If the Transmit Payload Data Input Interface block is configured to operate in the Gapped-Clock Mode, then the System-Side Terminal Equipment will be expected to update the data on the TxNib[3:0] input pin of the XRT79L71 upon the rising edge of the TxOH_Ind output signal. The XRT79L71 will sample and latch the TxNib[3:0] data, upon the falling edge of the TxOH_Ind output signal. In this case, there is no need to check the state of a certain output pin and then gate the placement of the next payload nibble (on the TxNib[3:0] input pins) with the sampled state of this particular signal. The System-Side Terminal Equipment only needs to respond to the rising edge of this particular Gapped-Clock signal.

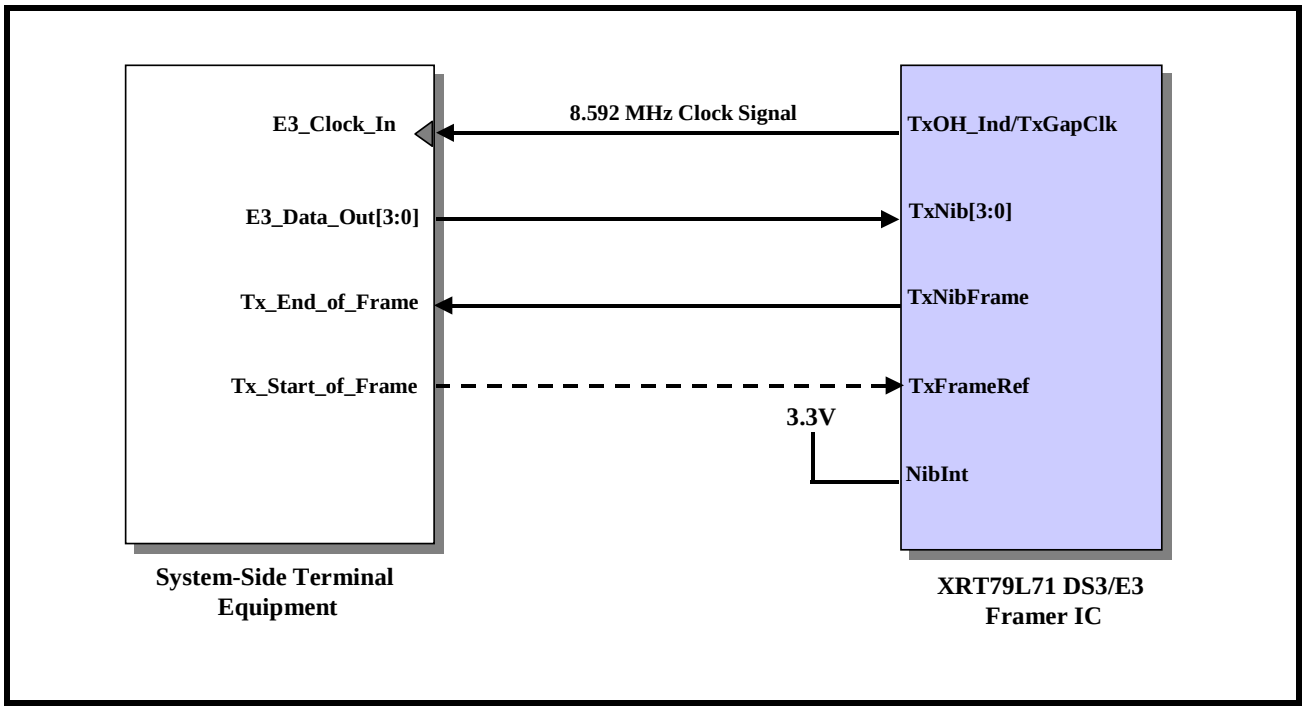
If the XRT79L71 has been configured to operate in the Loop-Timing Mode (e.g., Mode 1), then this Gapped-Clock signal from the TxOH_Ind output pin will be derived from the LIU Recovered Clock signal from the Receive E3 LIU Block. Similarly, if the XRT79L71 has been configured to operate in the Local-Timing Mode, then this Gapped-Clock signal will be derived form the TxInClk input signal.

Configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode

To configure the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode, then the do all of the following.

STEP 1 - The user should interface the System-Side Terminal Equipment to the Transmit Payload Data Input Interface block, in a manner as indicated below in Figure 137.

FIGURE 137. AN ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK OF THE XRT79L71 FOR NIBBLE-PARALLEL GAPPED-CLOCK MODE OPERATIONS



STEP 2 - Set Bit 5 (TxGapped Clock Mode Enable), within the Framer Test Register to "1" as depicted below.

Framer Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	0	1	0	0	0	0	0

NOTE: *This setting only configures the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode. Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode is discussed separately in Section 5.3.5.2.*

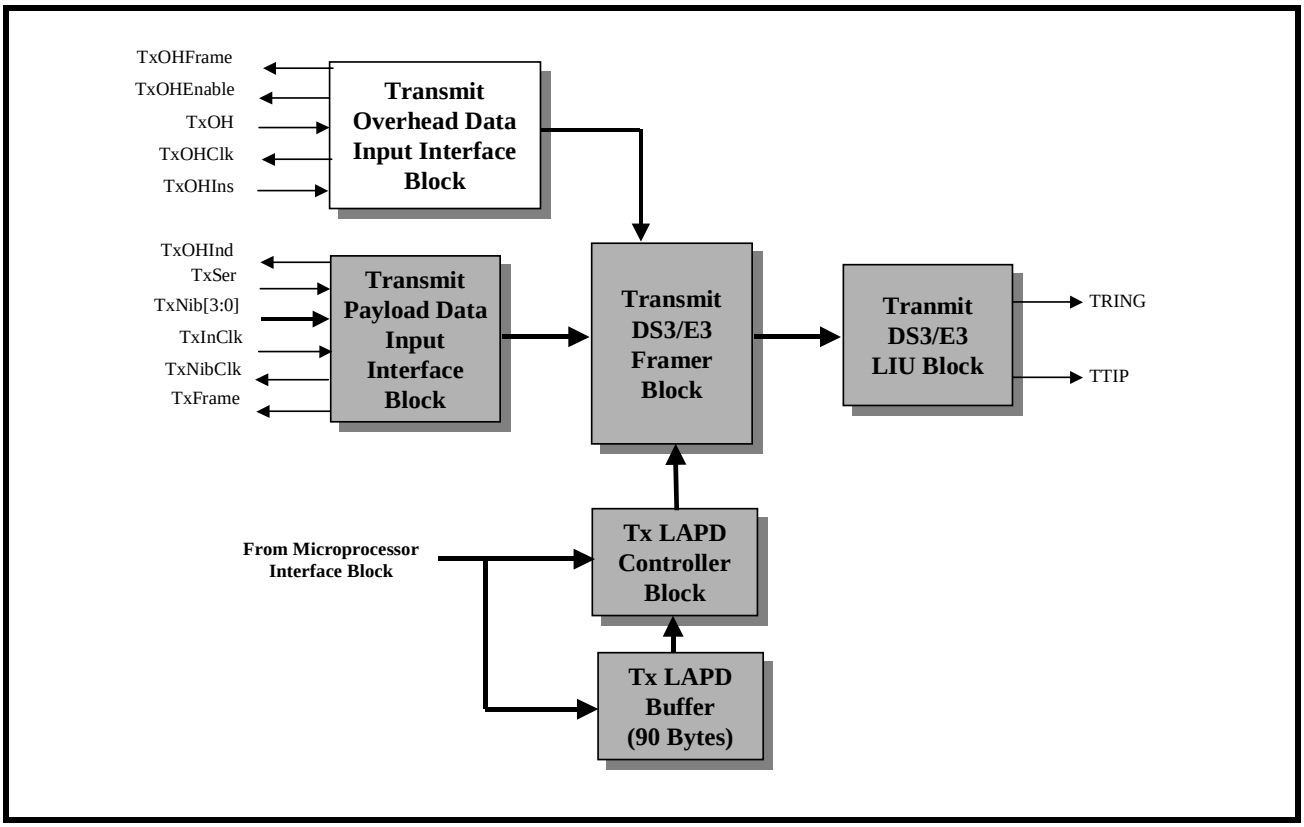
5.2.1.8 Accepting and Inserting E3 Overhead Bytes via the Transmit Payload Data Input Interface Block

5.2.2 TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

The Transmit Overhead Data Input Interface block is the second functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3 Clear-Channel Framer Applications. Figure 220 presents an

illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3 Clear-Channel Framer Mode, with the Transmit Overhead Data Input Interface block highlighted.

FIGURE 138. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK HIGHLIGHTED)



Some Background Information

In order to fully understand the role of the Transmit Overhead Data Input Interface, some background information needs to be discussed first.

As mentioned in SEE"DESCRIPTION OF THE E3, ITU-T G.751 FRAME STRUCTURE AND THE OVERHEAD BITS" ON PAGE 259., the E3, ITU-T G.751 frame consists of 1536 bits. Of these bits, 1524 bits are payload bits and the remaining 12 bits are overhead bits. The XRT79L71 has been designed to handle and process both the payload type and overhead bytes of bits/bytes for each E3 frame. Within the XRT79L71, the Transmit Payload Data Input Interface Block (which was discussed in considerable detail in SEE"TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 267.) has been designed to accept the payload data from the "System-Side" Terminal Equipment. Likewise, the Transmit Overhead Data Input Interface block has been designed to handle and process the overhead bits.

Accepting and Inserting E3 Overhead Bits via the Transmit Overhead Data Input Interface

By default, the Transmit E3 Framer block will be configured to internally generate and insert all of the overhead bytes within its outbound E3 data-stream. More specifically, the Transmit E3 Framer block will internally generate the E3 overhead bytes by doing all of the following, as presented below in Table 38.

TABLE 38: HOW THE TRANSMIT E3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS/BYTES - E3, ITU-T G.751 APPLICATIONS

BIT NAME	BIT/BYTE DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT E3 FRAMER BLOCK
FAS Bits[9:0]	Framing Alignment Bits	These 10 bits are set to the value [1111010000]
A	Alarm Bit	The Transmit E3 Framer block will handle the "A" bit, as described below. **If BIP-4 Insertion/Verification is enabled:** The Transmit E3 Framer block will automatically set the "A" bit to "1" anytime the corresponding Receive E3 Framer block detects a BIP-4 error within an incoming E3 frame. Conversely, the Transmit E3 Framer block will automatically set the "A" bit to "0" anytime the corresponding Receive E3 Framer block DOES NOT detect a BIP-4 error within an incoming E3 frame. **If BIP-4 Insertion/Verification is NOT enabled:** The Transmit E3 Framer block will set the "A" bit to the appropriate value as dictated by Software Control.
N	National Bit	The Transmit E3 Framer will handle the "N" bit, as described below. **If the Transmit LAPD Controller block is enabled:** The "N" bits will be used to transport the contents of the outbound LAPD/PMDL Messages from the local Terminal to the remote Terminal. **If the Transmit LAPD Controller block is DISABLED:** The Transmit E3 Framer block will set the "N" bits to the appropriate value as dictated by Software Control.
BIP-4[3:0]	BIP-4 Bits	The Transmit E3 Framer block will handle the BIP-4 bits, as described below. **If BIP-4 Insertion/Verification is enabled:** The Transmit E3 Framer block will compute the BIP-4 value over a given outbound E3 frame. The resulting BIP-4 value will be inserted into the BIP-4 bit positions within the very next outbound E3 frame. **If BIP-4 Insertion/Verification is disabled:** The BIP-4[3:0] bits will be used to carry payload bits that have been accepted via the Transmit Payload Data Input Interface block.

However, the Transmit Section of the XRT79L71 can also be configured to externally accept values (via a certain input port) and to insert this data into certain user specified overhead bits, within the outbound E3 data-stream. The XRT79L71 permits the user to implement this overhead E3 data-stream. The XRT79L71 permits the user to implement this overhead bit insertion by either of the following methods.

- By configuring the Transmit Section of the XRT79L71 to accept E3 overhead data via the Transmit Payload Data Input Interface block (as was discussed in **SEE"ACCEPTING AND INSERTING E3 OVERHEAD BYTES VIA THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 292.**).
- By configuring the Transmit Section of the XRT79L71 to accept E3 overhead data via the Transmit Overhead Data Input Interface (This approach will be discussed below).

The purpose of the Transmit Overhead Data Input Interface block is to accept overhead data from some system-side or up-stream source and to overwrite the contents of the overhead bits, within the outbound E3 data-stream with these particular overhead bit values.

In order to accomplish this, the Transmit Overhead Data Input Interface block has numerous input and output pins. **Table 39** presents a list and a brief definition of each of these pins.

TABLE 39: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxOHEnable	A12	O	**Transmit Overhead Data - Enable Output Pin:** The Transmit Overhead Data Input Interface will assert this signal (e.g., pulse it "High") for one TxInClk period, just prior to the instant that it (the Transmit Overhead Data Input Interface block) is processing an overhead bit. ***NOTE:*** *This output pin will remain "Low" at all other times.*
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input Pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the Transmit Circuitry within the XRT79L71.
TxOHFrame	B11	O	**Transmit Overhead Input Interface Enable Input Pin:** This output pin pulses "High" whenever the Transmit Overhead Data Input Port is processing the last bit within a given E3 frame.
TxOHIns	D10	I	**Transmit Overhead Data - Insert Enable Input Pin:** Asserting this input pin (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the System-Side Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data on the TxOH input pin, upon the falling edge of either the TxInClk or the TxOHClk clock signals. Setting this input pin "Low" configures the Transmit Overhead Data Input Interface block to NOT sample (e.g., ignore) the data on the TxOH input pin upon the falling edge of either the TxInClk or the TxOHClk clock signals. ***NOTE:*** *If the System-Side Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface block (e.g., if the System-Side Terminal Equipment asserts the TxOHIns input pin at a time whenever one of the non-insertable overhead bits are being processed), then this particular insertion effort will be ignored.*
TxOH	C11		**Transmit Overhead Data Input Interface - Overhead Data Input Pin:** The Transmit Overhead Data Input Interface block accepts the overhead data via this input pin and inserts this data into the appropriate overhead bit-position within the very next outbound E3 frame. If the "TxOHIns" input pin is pulled "high", then the Transmit Overhead Data Input Interface block will sample the data, residing on this input pin, upon either the rising edge of TxInClk or the falling edge of TxOHClk (depending upon the Insertion Method used). If the "TxOHIns" input pin is pulled "low", then the Transmit Overhead Data Input Interface block will NOT sample the data, residing on this input pin, upon the rising edge of "TxInClk" nor on the falling edge of "TxOHClk".
TxOHClk	B12	O	**Transmit Overhead Clock Output:** This output pin functions as the Transmit Overhead Data Input Interface clock signal. If the Transmit Overhead Data Input Interface block is enabled by asserting the "TxOHIns" input pin, then the Transmit Overhead Data Input Interface block will sample and latch the data (residing on the "TxOH" input pin) upon the falling edge of this signal.

The Two Methods of Inserting Overhead Data Into the Transmit Overhead Data Input Interface Block

There are two methods that can be used to insert the overhead data into the Transmit Overhead Data Input Interface Block. One Method is referred to as Method 1 or the TxOHClk Method, and the other method is

referred to as Method 2 or the TxInClk Method. Each of these methods is described in considerable detail below.

5.2.2.1 *Operating the Transmit Overhead Data Input Interface block using Method 1 - The TxOHClk Method*

This particular method is referred to as the TxOHClk method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHClk clock output signal (from the Transmit Overhead Data Input Interface block) to clock overhead data onto the TxOH input pin.

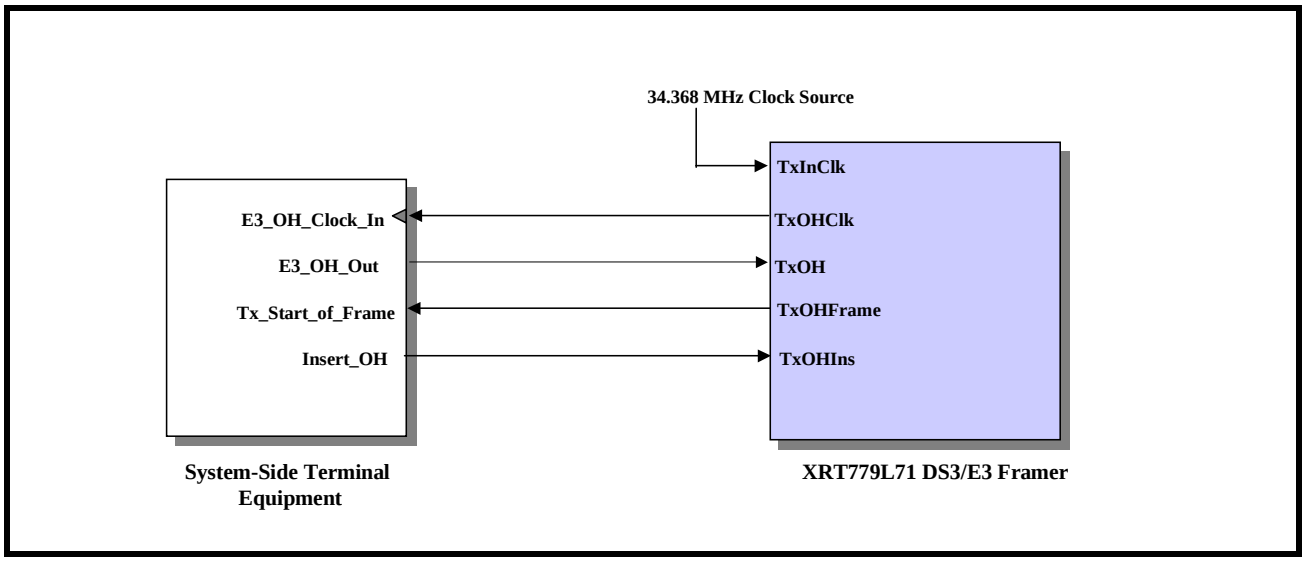
b. The Transmit Overhead Data Input Interface block will use the falling edge of the TxOHClk clock output signal to sample and latch the data residing on the TxOH input pin.

To use Method 1, the System-Side Terminal Equipment will need to interface to the following Transmit Overheads Data Input Interface pins.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

If Method 1 is used then the users system must be designed such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in Figure 139.

FIGURE 139. ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 1



Method 1 Operation of the Transmit Overhead Data Input Interface Block

To operate the Transmit Overhead Data Input Interface block, design/configure the System-Side Terminal Equipment to continuously and repeatedly execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the state of the TxOHFrame output pin (from the XRT79L71) upon the rising edge of the TxOHClk clock signal (which is also output from the XRT79L71). Whenever the System-Side Terminal Equipment samples the TxOHFrame output pin "High" then it will know that the Transmit Overhead Data Input Interface block is ready to accept overhead bits for a new E3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHFrame signal, it must also keep track of the number of rising edges (within the TxOHClk signal) that have occurred since the last time TxOHFrame was

sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bit is currently being processed by the Transmit Overhead Data Input Interface block at any given TxOHClk clock period. When the System-Side Terminal Equipment knows which overhead bit is being processed within a given TxOHClk clock period, it can decide when to insert the appropriate bit-value into the Transmit Overhead Data Input Interface block (and in-turn, force the Transmit DS3/E3 Framer block to insert this same bit-value into the appropriate overhead bit-position within the outbound E3 data-stream). From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin of the XRT79L71.

Table 40 relates the number of rising clock edge, within the TxOHClk output signal, since the TxOHFrame output signal was sampled "High" to the E3 Overhead Bit being processed by the Transmit Overhead Data Input Interface block. The user can use this table as a guide for inserting the appropriate overhead bits, within the outbound E3 data-stream, for Method 1.

TABLE 40: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
0 (TxOHClk, Clock Edge is coincident with the TxOHFrame signal being sampled "High")	FAS, Bit 1 (MSB)	NO
1	FAS, Bit 2	NO
2	FAS, Bit 3	NO
3	FAS, Bit 4	NO
4	FAS, Bit 5	NO
5	FAS, Bit 6	NO
6	FAS, Bit 7	NO
7	FAS, Bit 8	NO
8	FAS, Bit 9	NO
9	FAS, Bit 10 (LSB)	NO
10	A Bit	YES
11	N Bit	YES
12	BIP-4, Bit 1 (MSB)*	NO
13	BIP-4, Bit 2*	NO
14	BIP-4, Bit 3*	NO
15	BIP-4, Bit 4*	NO

NOTES:

1. *The shaded rows designate those Overhead bits that the XRT79L71 can be used to insert into the outbound E3 data-stream via the Transmit Overhead Data Input Interface block. The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound E3 data-stream.*
2. *The asterisk (*) indicates that these particular overhead bits will only be processed if BIP-4 processing is enabled, within the Transmit E3 Framer Block.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of clock edges from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal (by pulling it "High"). Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit (to be inserted into the outbound E3 data-stream) onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data (residing on the TxOH input pin) upon the very next falling edge of the TxOHClk output signal.

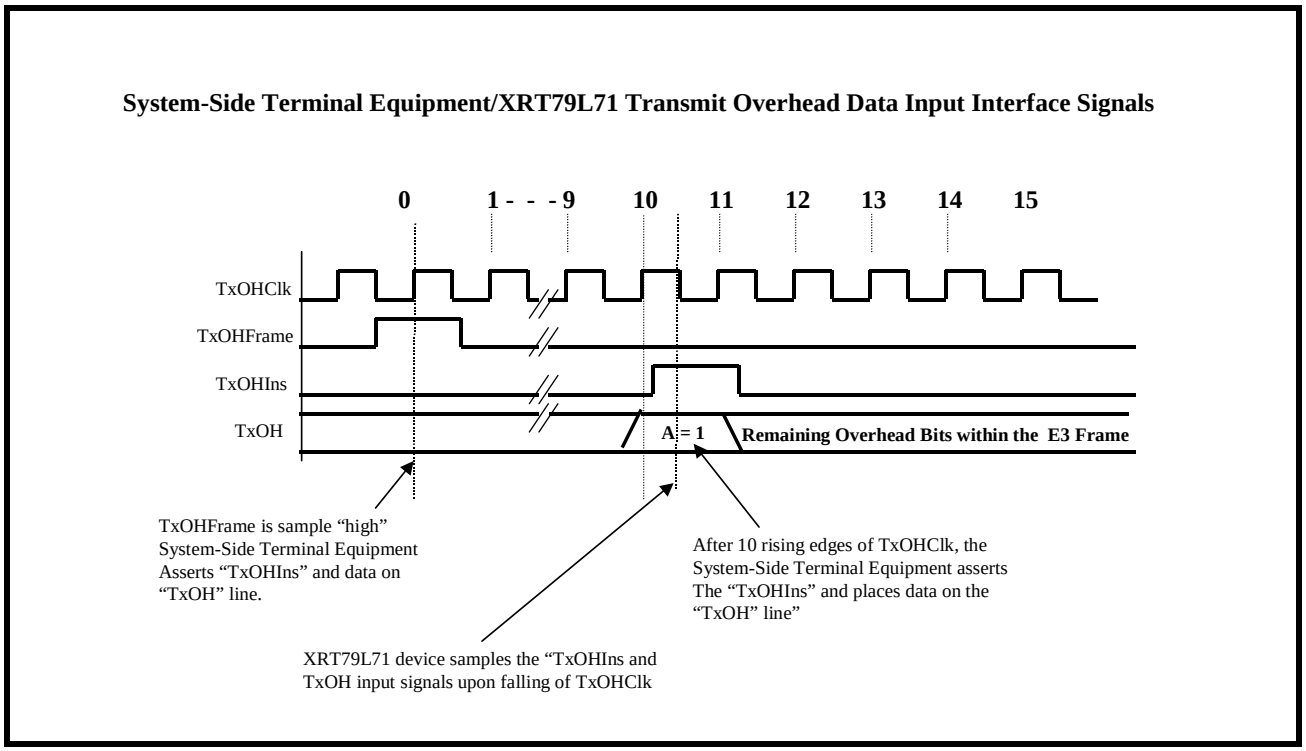
TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until the next rising edge of TxOHClk is detected. Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHClk period come up, for inserting an overhead bit into the outbound E3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 1 in order to transmit the FERF/RDI indicator to the remote terminal equipment.

For E3, ITU-T G.751 applications, the FERF/RDI indication is transmitted by setting the "A" bit-field (of each outbound E3 frame) to "1".

If we were to assume that the connection between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface block (of the XRT79L71) is as illustrated in Figure 139, then Figure 140 presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface, when using Method 1.

FIGURE 140. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 1)



In Figure 140, the System-Side Terminal Equipment samples the TxOHFrame signal being "High" during Rising Edge Clock Edge # 0 with the TxOHClk signal. At this point, the System-Side Terminal Equipment knows that the XRT79L71 is just about to process the very first overhead bit within a given E3 frame. According to Table 40, the "A" bit will be processed at TxOHClk clock edge # 10. In order to facilitate the transmission of the FERF/RDI indicator, the System-Side Terminal Equipment must (1) wait for 10 TxOHClk clock periods, and (2) then set this particular bit-field to "1". Once the System-Side Terminal Equipment has

waited through the 10 TxOHClk clock periods, it sets the "A" bit-field to "1" by implementing the following two tasks concurrently.

TASK 1 - The System Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "1".

After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample both the TxOHIns and the TxOH input pins being "High" during the very next falling edge of TxOHClk. Once the XRT79L71 has sampled these two signals, it will insert a "1" into the "A" bit-position within the outbound E3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as Clock Edge # 11, in Figure 140), the System-Side Terminal Equipment will negate (or de-assert) the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface for the remainder of this particular E3 frame period. Afterwards, the System-Side Terminal Equipment will repeat all of the steps that have been outlined in this case study.

5.2.2.2 Operating the Transmit Overhead Data Input Interface block using Method 2 - The TxInClk/ TxOHEnable Method

This particular method is referred to as the TxInClk/TxOHEnable method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHEnable output pin (from the Transmit Overhead Data Input Interface block) to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface block at any given time.

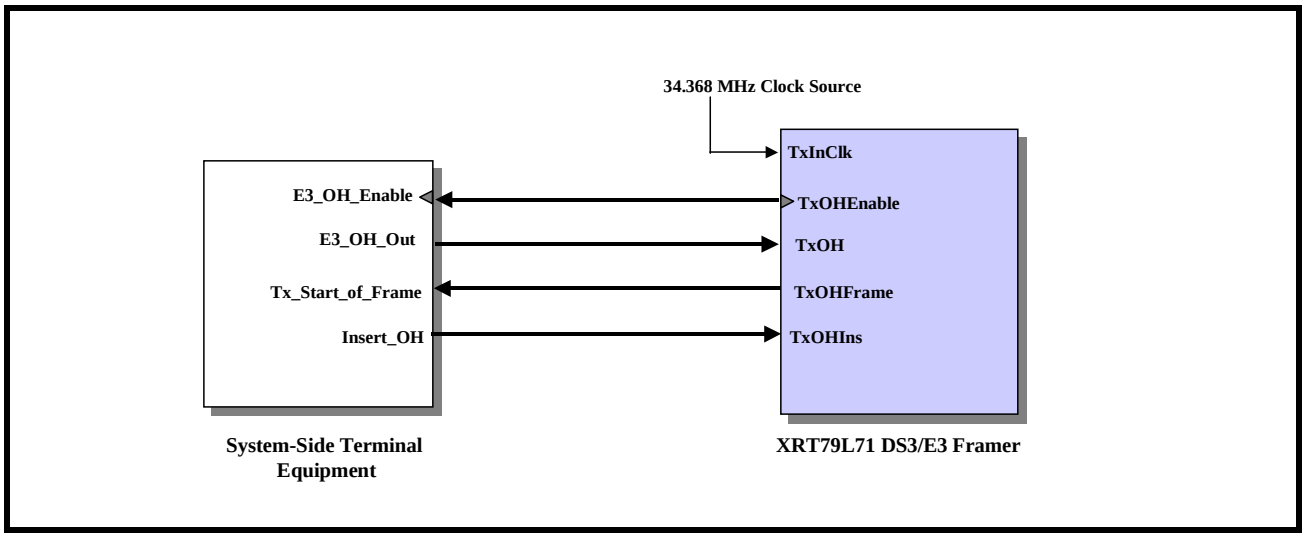
b. The Transmit Overhead Data Input Interface block will use the rising edge of TxInClk in order to sample and latch the data residing on the TxOH input pin.

If Method 2 is used, the System-Side Terminal Equipment will need to interface to the following Transmit Overhead Data Input Interface pins.

- TxOH
- TxOHFrame
- TxInClk
- TxOHEnable
- TxOHIns

If Method 2 is used then the users system must be designed such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in Figure 141

FIGURE 141. ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 2



Method 2 Operation of the Transmit Overhead Data Input Interface Block

To operate the Transmit Overhead Data Input Interface block per Method 2, design/configure the System-Side Terminal Equipment to continuously and repeatedly execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the states of both theTxOHFrame and the TxOHEnable output pins (from the XRT79L71) upon the falling edge of the TxInClk clock input signal. The XRT79L71 will pulse the TxOHEnable output pin "High" for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface block is processing an overhead bit. Therefore, if the System-Side Terminal Equipment samples the TxOHEnable output pin "High", then it knows that an overhead bit insertion opportunity (via the Transmit Overhead Data Input Interface block) is just about to occur. If the System-Side Terminal Equipment samples both the TxOHEnable and the TxOHFrame output pins "High" at the same time, then it knows that the Transmit Overhead Data Input Interface block is just about to process the very first overhead bit within a new E3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHEnable and TxOHFrame signals, it must also keep track of the number of times that the TxOHEnable output pin has been sampled "High" since the last time both the TxOHEnable and the TxOHFrame output pins have been sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bits are currently being processed by the Transmit Overhead Data Input Interface block at any give TxOHEnable assertion. When the System-Side Terminal Equipment knows which overhead bit is being processed within a given TxOHEnable assertion period, it can decide when to insert the appropriate bit-value to the Transmit Overhead Data Input Interface block (and in-turn, force the Transmit DS3/E3 Framer block to insert this bit into the appropriate overhead bit-position within the outbound E3 data-stream). From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin of the XRT79L71.

Table 41 relates the number of TxOHEnable output pulses that have occurred since both the TxOHFrame and the TxOHEnable pins were both sampled "High", to the E3 Overhead bit being processed by the Transmit Overhead Data Input Interface block. The user can use this table as a guide for inserting the appropriate overhead bits, within the outbound E3 data-stream for Method 2.

TABLE 41: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
0 (TxOHEnable and TxOHFrame are sampled "High" simultaneously)	FAS, Bit 1 (MSB)	NO
1	FAS, Bit 2	NO
2	FAS, Bit 3	NO
3	FAS, Bit 4	NO
4	FAS, Bit 5	NO
5	FAS, Bit 6	NO
6	FAS, Bit 7	NO
7	FAS, Bit 8	NO
8	FAS, Bit 9	NO
9	FAS, Bit 10 (LSB)	NO
10	A Bit	YES
11	N Bit	YES
12	BIP-4, Bit 1 (MSB)*	NO
13	BIP-4, Bit 2*	NO
14	BIP-4, Bit 3*	NO
15	BIP-4, Bit 4*	NO

NOTES:

1. *The shaded rows designate those Overhead bits that the XRT79L71 can be used to insert into the outbound E3 data-stream via the Transmit Overhead Data Input Interface block. The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound E3 data-stream.*
2. *The asterisk (*) indicates that these particular overhead bits will only be processed if BIP-4 processing is enabled, within the Transmit E3 Framer Block.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of TxOHEnable pulses from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal (by pulling it "High"). Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit (to be inserted into the outbound E3 data-stream) onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data (residing on the TxOH input pin) upon the second rising edge of TxInClk (after TxOHEnable was sampled "High").

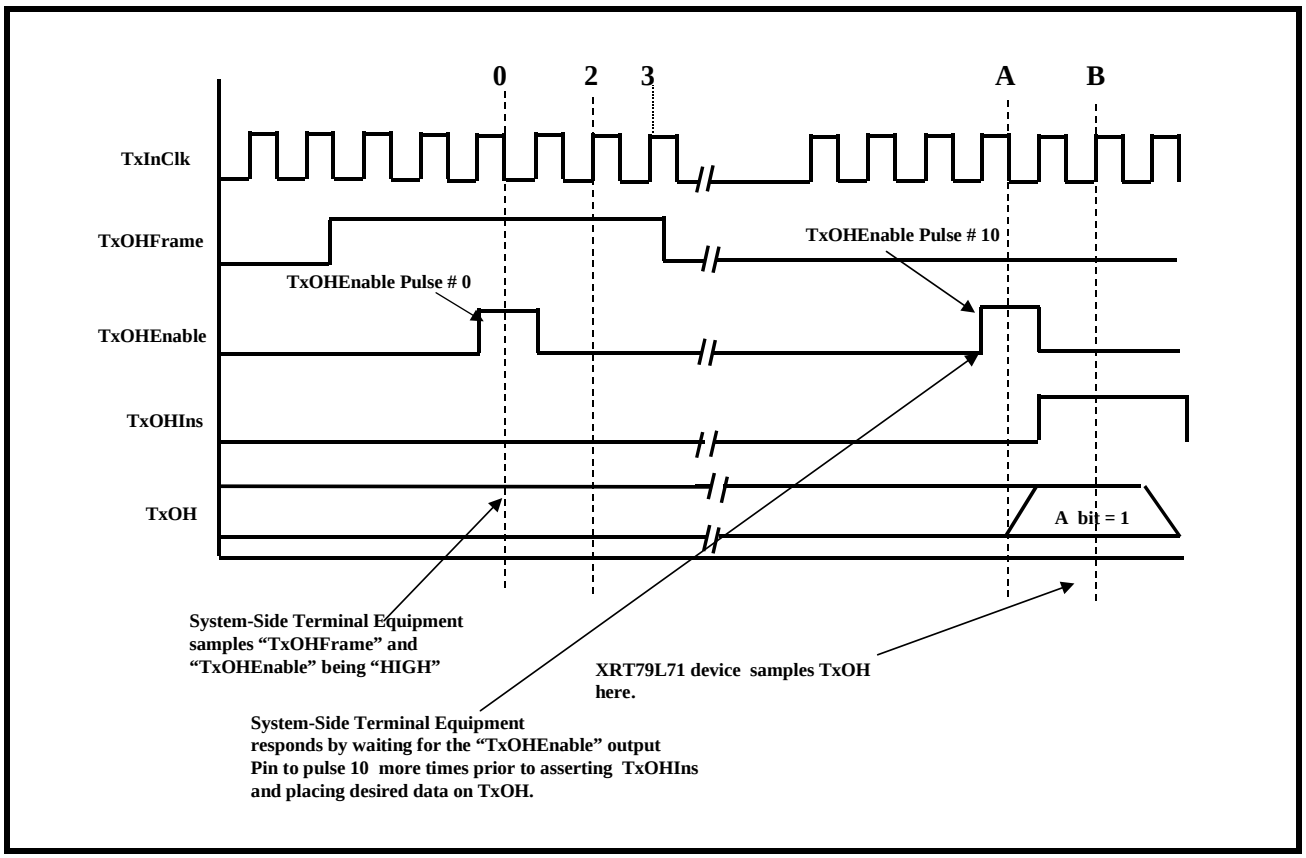
TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until the next time that the TxOHEnable output pin is sampled "High". Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHEnable pulsing period comes up, for inserting an overhead bit into the outbound E3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 2 in order to transmit the FERF/RDI indicator to the remote terminal equipment.

For E3 ITU-T G.751applications, the FERF/RDI indication is transmitted by setting the A bit-field within f each outbound E3 frame to "1".

If we were to assume that the connection between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface block (of the XRT79L71) is as illustrated in Figure 141, then Figure 142 presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface when using Method 2.

FIGURE 142. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 2)



In Figure 142, the System-Side Terminal Equipment samples the TxOHFrame and TxOHEnable output pins being "High" during Falling Clock Edge # 0 within the TxInClk signal. At this point, the System-Side Terminal Equipment know that the XRT79L71 is just about to process the very first overhead bit within a given outbound E3 frame. In order to facilitate the transmission of the FERF/RDI indicator, the System-Side Terminal Equipment must set the "A" bit-field (within the outbound E3 frame) to "1". According to Table 41, the "A" bit-field will be ready for processing coincident to the 10th assertion of the TxOHEnable output signal, following the most recent assertion of TxOHFrame.

The System-Side Terminal Equipment can externally set the "A" bit-field (within each outbound E3 frame) to "1" by implementing the following tasks in the sequence as depicted below.

TASK # 1 - Continuously sample both the TxOHFrame and the TxOHEnable output pins with each falling edge of TxInClk. Whenever the System-Side Terminal Equipment samples both of these input pin "High", then it should move on to **TASK # 2**.

TASK # 2 - Continue to sample both the TxOHEnable and TxOHFrame output pins with each falling edge of TxInClk. The System-Side Terminal Equipment should increment an internal counter each time it samples the TxOHEnable output pin "High". The System-Side Terminal Equipment should execute this task until the counter has reach the value "10". Once this counter has reached the value "10" then the System-Side Terminal Equipment should move on to **TASK # 3**.

TASK # 3 - The System-Side Terminal Equipment MUST execute all of the following two sub-tasks simultaneously.

TASK # 3a - The System-Side Terminal Equipment must assert the TxOHIns input pin by setting it "High".

TASK # 3b - The System-Side Terminal Equipment must set the TxOH input pin to "1".

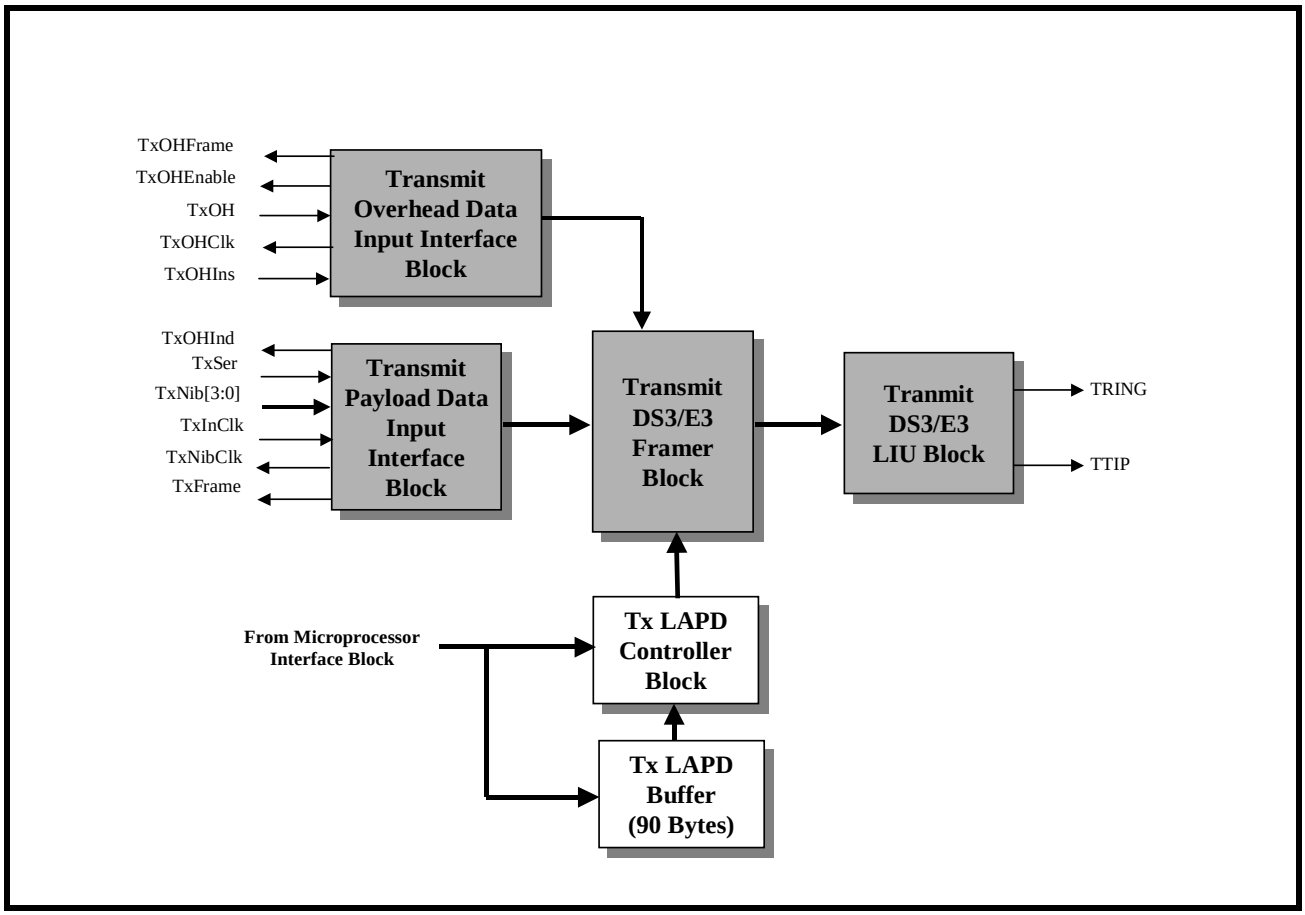
After the System-Side Terminal Equipment has executed these two sub-tasks, it must now move on to **TASK # 4**.

TASK # 4 - The System-Side Terminal Equipment must continue to sample the TxOHEnable output pin (from the XRT79L71). Whenever the System-Side Terminal Equipment samples the TxOHEnable "High" (once again) then it must negate the TxOHIns input pin (e.g., by setting it to "0"). Afterwards, the System-Side Terminal Equipment should then reset the internal TxOHEnable counter to "0", and return to **TASK # 1**.

5.2.3 TRANSMIT LAPD CONTROLLER BLOCK

The Transmit LAPD Controller block is the third functional block (within the Transmit Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. **Figure 143** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the Transmit LAPD Controller block highlighted.

FIGURE 143. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT LAPD CONTROLLER BLOCK HIGHLIGHTED).



The Transmit LAPD Controller block consists of the following sections.

- The Transmit LAPD Message Buffer
- The Transmit LAPD Controller

Each of these sections is described in some detail below.

The Transmit LAPD Message Buffer

The purpose of the Transmit LAPD Message Buffer is to permit the user to write and temporarily store the contents of the very next outbound LAPD Message that is to be transmitted. The Transmit LAPD Message Buffer is actually a 90 byte FIFO that is located at Address location 0x11B0 within the XRT79L71 address space.

The Transmit LAPD Controller

The Transmit LAPD Controller permits the user to transmit path maintenance data link (PMDL) messages to the remote terminal equipment via the outbound E3 Frames. In this case the message bits are inserted into and carried by the "N" bit within each outbound E3 frame. The on-chip Transmit LAPD Controller permits the user to transmit both standard and non-standard PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer), to store the contents of the outbound PMDL message to be transmitted. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in **Figure 144**.

FIGURE 144. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-BITS)	C/R	EA
TEI (7 BITS)		EA
CONTROL (8-BITS)		
76 OR 82 OR ANY-SIZE BYTES OF INFORMATION (PAYLOAD)		
FCS - MSB		
FCS - LSB		
FLAG SEQUENCE (8-BITS)		

Where for standard (Bellcore GR-499-CORE) applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the Transmit LAPD Controller is not currently transmitting a LAPD Message, it will instead be transmitting a continuous stream of Flag Sequence bytes via the "N" bit within each outbound E3 frame.

SAPI - Service Access Point Identifier

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for standard (Bellcore GR-499-CORE) applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3 and E3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For standard (Bellcore GR-499-CORE) applications the user must use the Control byte the value "0x03". Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The Information Payload is the 76 bytes, 82 bytes or any number of 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 Address 0x11B0).

A Special Note about the Information Payload when transmitting standard (Bellcore GR-499-CORE) type LAPD Messages

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 0x11B0 within the XRT79L71). The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. **Table 42** presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT79L71 and the corresponding octet value that the user must write into the first octet position within the Transmit LAPD Message buffer.

TABLE 42: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD (FOR STANDARD 76 OR 82 BYTE MESSAGES)

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

NOTE: *If the LAPD Message and information payload that is to be transmitted is of a size other than 76 or 82 bytes, then there are no restrictions on the value that should be written to the first byte within the information payload.*

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, x16 + x12 + x5 + 1. Afterwards, this FCS value is inserted into the two-octet FCS value position, within the LAPD Message frame. The LAPD Receiver (at the remote terminal) will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see **SEE"THE RECEIVE LAPD CONTROLLER BLOCK" ON PAGE 392.** on how the Receive LAPD Controller Block handles and processes incoming LAPD Message frames.

Operation of the Transmit LAPD Controller

As mentioned earlier, the Transmit LAPD Controller permits the user to transmit either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

The procedure for transmitting these types of LAPD Messages is presented below.

5.2.3.1 Transmitting Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) write the contents of an outbound PMDL Message, into the Transmit LAPD Message Buffer, and (2) command the Transmit LAPD Controller to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "0", as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Configure the "N" bit (within each outbound E3 Frame) to function as the LAPD Channel.

The user can accomplish this by setting Bits 4 and 3 (TxNSrcSel[1:0]) within the Transmit E3 Configuration Register - G.751 to [1, 0], as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This write operation will configure the Transmit LAPD Controller block to use the "N" bit as the LAPD Channel. In this setting, the Transmit LAPD Controller block will fragment the contents of the outbound LAPD/PMDL Message into bits and it will insert each of these bytes into the "N" bit-field position within each outbound E3 frame.

STEP 3 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit E3 LAPD Configuration Register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	X	X	1

NOTES:

1. *For normal operation, it is imperative that the user also make sure that bit 4 (Reserved) and bit 7 (LAPD Any) within this register are both set to "0".*
2. *Once the user executes the above mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle (Flag) Sequence (e.g., a continuous, repeating string of octets, of the value 0x7E) via the "N" bit within each outbound E3 frame.*

STEP 4 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a LAPD/PMDL Message at one-second intervals. To configure the Transmit LAPD Controller block to do this,

then the user should write a "1" into Bit 3 (Auto Retransmit) within the Transmit E3 LAPD Configuration Register. This action is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	1	0	X	1

STEP 5 - Specify the type of LAPD Message to be transmitted.

At this stage it is necessary to specify the type of LAPD Message, which is to be transmitted. The Transmit and Receive LAPD Controller blocks within the XRT79L71 are capable of supporting the following standard types of LAPD Messages.

- Test Signal
- Idle Signal
- CL Path Identification
- ITU Path Identification

The ITU Path Identification type of PMDL Message consists of an 82-byte information payload. All of the remaining standard types of PMDL/LAPD Messages contain 76 byte information payloads. To transmit an 82 byte message (e.g., the ITU Path Identification type of LAPD Message), then the user should set Bit 1 (Transmit LAPD Message Length), within the Transmit E3 LAPD Configuration Register to "1". For all of the remaining types of LAPD Messages (which are 76 bytes in length), the user should set Bit 1 (Transmit LAPD Message Length) to "0". This operation is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	0	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle LAPD/PMDL messages that contain 76-byte sized Information Payloads.*

To transmit an ITU Path Identification type of PMDL Message, write the value "1" into the Transmit LAPD Message Length bit-field, as depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/W	R/W	R/O	R/W	R/W
0	0	0	0	X	0	1	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle 82-byte sized PMDL Messages.*

STEP 6 - Load the Transmit LAPD Message Buffer

The next step is to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in **SEE"RECEIVING STANDARD-TYPE (76 OR 82 BYTE SIZE) LAPD MESSAGES" ON PAGE 395.**.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 6a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer

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

Sub-STEP 6a.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte (of Indirect Address 0x00) within the Transmit LAPD Message Buffer.

Sub-STEP 6a.2 - At Address Location 0x11C1 (the LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 6a.1, above) this will be the very first byte (of Indirect Address 0x00) within the Transmit LAPD Message Buffer.

STEP 6b - Write the value for SAPI, C/R and EA0 into the second byte-position, within the Transmit LAPD Message Buffer

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

NOTE: *Bellcore mandates that the user use the values "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.*

Sub-STEP 6b.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x01".

This step will cause the internal LAPD Message Buffer pointer to point to the second byte (of Indirect Address 0x01) within the Transmit LAPD Message Buffer.

Sub-STEP 6b.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write in the value "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.

This step will cause the value "0x3C" or "0x3E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6b.1, above) this will be the second byte (of Indirect Address 0x01) within the Transmit LAPD Message Buffer

STEP 6c - Write the value for TEI and EA1 into the third byte-position, within the Transmit LAPD Message Buffer

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

NOTE: *Bellcore mandates that the user use the value of "0x01" for this byte.*

Sub-STEP 6c.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x02".

This step will cause the internal LAPD Message Buffer pointer to point to the third byte (of Indirect Address 0x02) within the Transmit LAPD Message Buffer.

Sub-STEP 6c.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x01".

This step will cause the value "0x01" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6c.1, above) this will be the third byte (of Indirect Address 0x02) within the Transmit LAPD Message Buffer.

STEP 6d - Write in the CONTROL BYTE value of 0x03 into the fourth byte-position, within the Transmit LAPD Message buffer

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

Sub-STEP 6d.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x03".

This step will cause the internal LAPD Message Buffer pointer to point to the fourth byte (of Indirect Address 0x03) within the Transmit LAPD Message Buffer.

Sub-STEP 6d.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x03"

This step will cause the value "0x03" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6d.1, above) this will be the fourth byte (of Indirect Address 0x03) within the Transmit LAPD Message Buffer.

STEP 6e - Write in a specific value, which the remote terminal equipment can use to identify the PMDL Message type.

Therefore, the exact value that the user must write in depends upon the type of LAPD Message being transmitted to the Remote Terminal Equipment. **Table 43** presents a mapping of the value to be written into this byte, with the corresponding PMDL Message type, to be transmitted.

TABLE 43: A MAPPING OF THE VALUE TO BE WRITTEN INTO INDIRECT ADDRESS LOCATION 0X11B0 AND THE CORRESPONDING PMDL MESSAGE

PMDL MESSAGE TYPE	VALUE TO BE WRITTEN INTO ADDRESS LOCATION 0X11B0 (DURING STEP 5E)
Test Signal	0x32
Idle Signal	0x34

TABLE 43: A MAPPING OF THE VALUE TO BE WRITTEN INTO INDIRECT ADDRESS LOCATION 0X11B0 AND THE CORRESPONDING PMDL MESSAGE

PMDL MESSAGE TYPE	VALUE TO BE WRITTEN INTO ADDRESS LOCATION 0X11B0 (DURING STEP 5E)
CL Path Identification	0x38
ITU Path Identification	0x3F

The user can accomplish all of this by executing the following two sub-steps.

Sub-STEP 6e.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x04".

This step will cause the internal LAPD Message Buffer pointer to point to the fifth byte (of Indirect Address 0x04) within the Transmit LAPD Message Buffer.

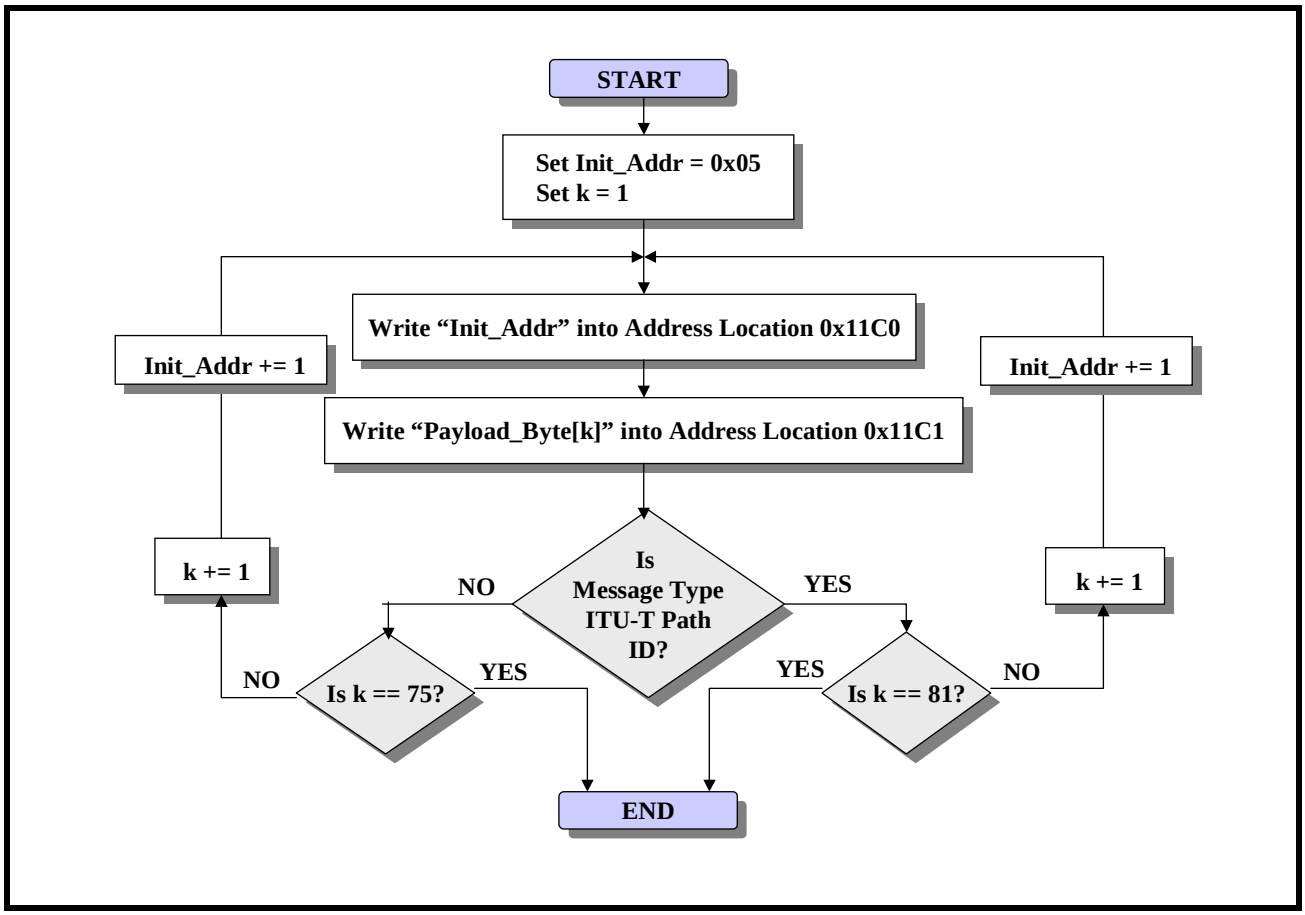
Sub-STEP 6e.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the appropriate value (per Table 43) into the location that is being identified by the LAPD Message Buffer pointer.

In this case (per Sub-STEP 6e.1, above) this will be the fifth byte (of Indirect Address 0x04) within the Transmit LAPD Message Buffer.

STEP 6f - Write in the remaining 75 or 81 byte of the Information Payload (within the PMDL Message) into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented in the following flow-chart.

FIGURE 145. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING THE PAYLOAD PORTION OF THE LAPD/PMDL MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



STEP 7 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller has completed its transmission of a given PMDL Message. The purpose of this interrupt is to alert the system µC/µP that the Transmit LAPD Controller has completed transmitting its most recent LAPD/PMDL message, and that it is now available to transmit a different LAPD Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 7a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation, at the Operational Block Level.

STEP 7b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

This step enables the Transmit DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 7c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit E3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 8 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit E3 LAPD Status/ Interrupt Register, as depicted below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0-> 1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will "zero-stuff" the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin (in a bit-by-bit manner) inserting the resulting PMDL Message into the "N" bit-fields, within the outbound E3 frames.

NOTES:

1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised to (some time later) to execute another write operation (to this register) that sets the Initiate Transmission of LAPD/PMDL Message bit back to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field (within the Transmit E3 LAPD Status/Interrupt register) to "1", as illustrated below.*

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will then toggle back to "0".

5.2.3.2 Transmitting Non-Standard Variable Length (e.g., up to 82 bytes) LAPD Messages

The user can (1) write the contents of the outbound PMDL Message into the Transmit LAPD Message buffer and (2) command the Transmit LAPD Controller to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Framing format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1", as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Configure the "N" bit (within each outbound E3 Frame) to function as the LAPD Channel.

The user can accomplish this by setting Bits 4 and 3 (TxNSrcSel[1:0]) within the Transmit E3 Configuration Register - G.751 to [1, 0], as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	1	0	0	0	0

This write operation will configure the Transmit LAPD Controller block to use the "N" bit as the LAPD Channel. In this setting, the Transmit LAPD Controller block will fragment the contents of the outbound LAPD/PMDL Message into bits and it will insert each of these bytes into the "N" bit-field position within each outbound E3 frame.

STEP 3 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit E3 LAPD Configuration register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	X	X	1

NOTE: *Once the user executes the above-mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle (Flag) Sequence (e.g., a repeating string of 0x7E) via the "N" bit within each outbound E3 frame.*

STEP 4 - Configure the Transmit LAPD Controller to transmit a non-standard size LAPD Message.

This is accomplished by setting Bit 7 (LAPD Any) within the Transmit E3 LAPD Configuration Register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	X	X	X	1

STEP 5 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a PMDL Message at one-second intervals. To configure the Transmit LAPD Controller to do this, write a "1" into Bit 3 (Auto-Retransmit) within the Transmit E3 LAPD Configuration register. This operation is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	1	0	X	1

STEP 6 - Specify the size of the outbound LAPD Message

This is accomplished by writing the size of the information payload (in terms of number of bytes) into the Transmit LAPD Byte Count Register, as depicted below.

Transmit LAPD Byte Count Register (Address = 0x1183)

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

STEP 7 - Load the Transmit LAPD Message Buffer

The next step is to load into the contents of the outbound LAPD/PMDL Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents o the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ an Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in **SEE"RECEIVING "NON-STANDARD" VARIABLE LENGTH (E.G., UP TO 82 BYTES) LAPD MESSAGES)" ON PAGE 400.**.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 7a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer.

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

Sub-STEP 7a.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte (of Indirect Address = 0x00) within the Transmit LAPD Message Buffer.

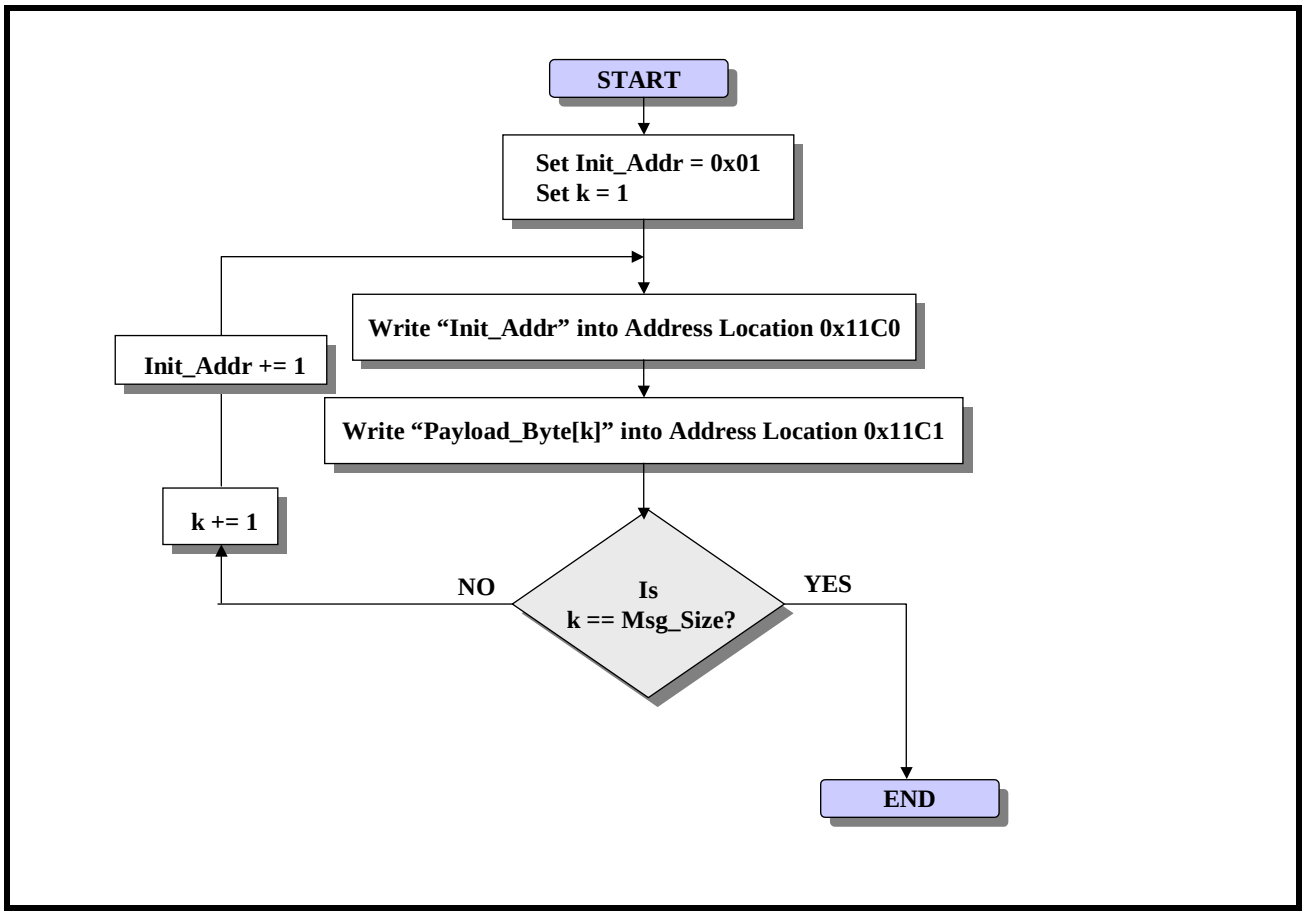
Sub-STEP 7a.2 - At Address Location 0x11C1 (the LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 7a.1, above) this will be the very first byte (of Indirect Address 0x00) within the Transmit LAPD Message Buffer.

STEP 7b - Write in the remaining bytes of this outbound message into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 146. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING IN THE REMAINING BYTES OF THE OUTBOUND MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



NOTE: *About* **Figure 146***: The value Msg_Size (within the decesion diamond) is the value that the user writes into the Transmit LAPD Byte Count Register, during STEP 6.*

STEP 8 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller has completed its transmission of a given PMDL Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 8a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

STEP 8b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

STEP 8c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit E3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 9 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit E3 LAPD Status/ Interrupt Register, as depicted below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0->1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will zero-stuff the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin (in a byte-by-byte manner) inserting the resulting PMDL Message into the "N" bit-fields (depending upon user selection during STEP 2), within the outbound E3 frames.

NOTES:

1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised to (some time later) to execute another write operation (to this register) that sets the Initiate Transmission of LAPD/PMDL Message bit back to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field (within the Transmit E3 LAPD Status/Interrupt register) to "1", as illustrated below.*

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will toggle back to "0".

The Mechanics of Transmitting a New LAPD Message, if the Transmit LAPD Controller has been configured to retransmit the LAPD Message frame, repeatedly at one-second intervals

If the Transmit LAPD Controller has been configured to retransmit the LAPD Message repeatedly at one-second intervals, then it will do the following at one-second intervals.

- Parse through the contents of the Transmit LAPD Message buffer and Zero-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.

To transmit another (e.g., different) PMDL message to the remote Receive LAPD Controller, the user will have to write this new message into the Transmit LAPD Message buffer, via the Microprocessor Interface section of the channel. However, the user must be careful when writing in this new message. If the user writes this message into the Transmit LAPD Message buffer at the wrong time (with respect to these one-second LAPD Message frame transmissions), the user's action could interfere with these transmissions, thereby causing the Transmit LAPD Controller to transmit a corrupted message to the remote Receive LAPD Controller. 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 DS3/E3 Framer Block to automatically reset activated interrupts***

The user can do this by writing a "1" into Bit 1 (Enable Interrupt Auto-Clear) within the Operating Mode Register - Byte 2, as depicted below.

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	1	0

This action will prevent the Transmit LAPD Controller from generating its own one-second interrupts.

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 = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	0	1

3. ***Write the new message into the Transmit LAPD Message buffer immediately after the occurrence of the One-Second interrupt.***

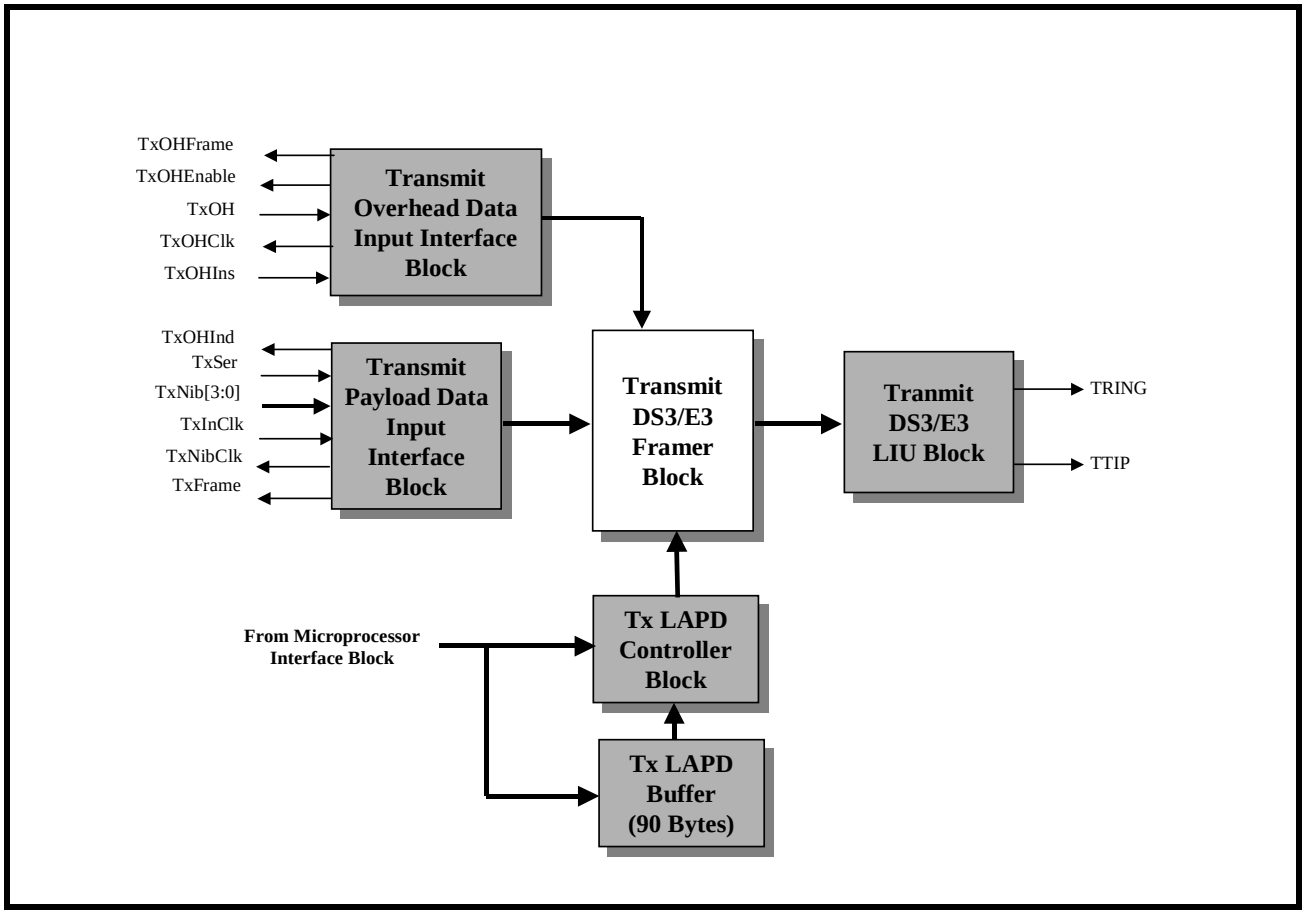
By timing the writes to the Transmit LAPD Message buffer to occur immediately after the occurrence of the One-Second interrupt, the user avoids conflicting with the one-second transmissions of the LAPD Message frame, and will transmit the correct messages to the remote Receive LAPD Controller.

5.2.3.3 Transmit -LAPD Contro;ller Block Interrupt

5.2.4 TRANSMIT E3 FRAMER BLOCK

The Transmit DS3/E3 Framer block is the fourth functional block (within the Transmit Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. Figure 147 presents an illustration of the "Transmit Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the "Transmit DS3/E3 Framer" block highlighted.

FIGURE 147. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "TRANSMIT DS3/E3 FRAMER" BLOCK HIGHLIGHTED)



The purpose of the Transmit E3 Framer block is to accept data from the Transmit Payload Data Input Interface block and the Transmit LAPD Controller, and to create an E3 data-stream. Afterwards, the Transmit E3 Framer block will route this E3 data-stream to the Transmit E3 LIU Block (for transmission to the remote terminal equipment). The Transmit E3 Framer block also supports the following functions.

- Transmitting the LOS Pattern (under Software control)
- Transmitting the AIS Pattern (under Software control)
- Transmitting the FERF (RDI) indicator (under Software Control, if BIP-4 Calculation and Insertion is disabled)
- Transmitting the FEBE (REI) indicator (automatically, if BIP-4 Calculation and Insertion is enabled).

5.2.4.1 TRANSMITTING THE LOS PATTERN

The Transmit DS3/E3 Framer block (within the XRT79L71) permits the user to transmit the LOS (Loss of Signal) pattern to the remote terminal equipment. In this case, the Transmit E3 Framer block will generate and transmit an "All Zeros" pattern to the remote terminal equipment. The procedure for transmitting the LOS Pattern is presented below.

Transmitting the LOS Pattern

The user can configure the Transmit DS3/E3 Framer block to transmit an LOS pattern by transmitting an "All Zeros" pattern. The user can accomplish this by setting Bit 1 (Tx LOS Enable), within the "Transmit E3 Configuration Register - G.751" to "1" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	1	0

Once this step is executed, then the Transmit DS3/E3 Framer block will override all of the outbound E3 data (payload and overhead bits), with an "All Zeros" Pattern.

5.2.4.2 TRANSMITTING THE E3 AIS PATTERN

The Transmit DS3/E3 Framer block permits the user to (upon Software Control) transmit the E3 AIS indicator (which is an "Unframed All Ones" Pattern) to the remote terminal equipment. The user can invoke this feature by setting Bit 2 (TxAIS Enable) within the "Transmit E3 Configuration" Register, to "1" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	1	0	0

Once this step is executed, then the Transmit DS3/E3 Framer block will override the contents of all "outbound" E3 data, with this E3 AIS (e.g., an "Unframed, All Ones") Pattern.

5.2.4.3 TRANSMITTING THE FERF/RDI INDICATOR

The Transmit E3 Framer block can be configured to transmit the FERF/RDI indicator to the remote terminal, under software control, provided that it (the Transmit E3 Framer block) is not also configured to compute and insert the BIP-4 value into each "outbound" E3 frame.

More specifically, the following items must be noted about the Transmit E3 Framer blocks, when it comes to generating the FERF/RDI indicator (while operating in the ITU-T G.751 Framing format).

1. The Transmit E3 Framer block CANNOT be configured to automatically generate the FERF/RDI indicator, in response to a "Service-Affecting" defect condition being declared by the corresponding Receive E3 Framer block (This is in contrast to the E3, ITU-T G.832 and the two DS3 framing formats).
2. The Transmit E3 Framer block CAN ONLY be configured to generate the FERF/RDI indicator, upon Software Control (provided that it is NOT configured to compute and insert the BIP-4 value within each outbound E3 frame)
3. If the Transmit E3 Framer block is configured to compute and insert the BIP-4 value into each outbound E3 frame, then it will NOT be able to transmit the FERF/RDI indicator. (In this case, the "A" bit will take on the role of FEBE/REI, as described in Section 5.2.4.2).

Forced Transmission of the FERF/RDI Indicator

The XRT79L71 permits the user to force the Transmit E3 Framer block to transmit the FERF/RDI indicator to the remote terminal equipment. The user can accomplish this by executing the following steps.

STEP 1 - Set Bit 7 (TxBIP-4 Enable), within the "Transmit E3 Configuration" Register, to "0" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

STEP 2 - Set Bit 0 (RxBIP-4 Enable), within the "Receive E3 Configuration and Status Register # 1" to "0", as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERFAlgo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	0

STEP 3 - Set Bit 1 (TxA), within the "Transmit E3 Service Bits" Register, to "1" as depicted below.

Transmit E3 Service Bits Register - G.751 (Address = 0x1135)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						TxA	TxN
R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	1	0

STEP 4 - Set Bit 5 and 6 (TxASrcSel[1:0]) within the "Transmit E3 Configuration" Register, to "[0, 0]" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4Enable	TxASrc-Sel[1:0]	TxNSrc-Sel[1:0]	TxAIS Enable	TxLOS Enable	TxFASSou-rceSel		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This step configures the Transmit E3 Framer block to read out the contents of Bit 1 (TxA) within the "Transmit E3 Service Bits" Register, as set the "A" bit (within each outbound E3 frame) to this value.

In this case, since we have set Bit 1 (TxA) within the "Transmit E3 Service Bits" Register to "1", then the Transmit E3 Framer block will now set the "A" bit-field (within each outbound E3 frame) to "1" (denoting the FERF/RDI indication).

5.2.4.4 TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR

If the Transmit DS3/E3 Framer block is configured to support the E3, ITU-T G.751 framing format, then it can be capable of transmitting the FEBE/REI indicator to the remote terminal equipment.

If the Transmit DS3/E3 Framer block is configured to support the transmission of the FEBE/REI indicator to the remote terminal equipment, then the purpose of this indication is two-fold.

1. It permits a Terminal (which is transmitting an E3 data-stream to the remote terminal) to determine whether or not this remote terminal is receiving its E3 data, in an error-free manner.
2. It permits a Terminal (which is receiving an E3 data-stream from a remote terminal) to inform this remote terminal when it is receiving "erred" E3 frames.

A NOTE ABOUT TRANSPORTING THE FEBE/REI INDICATOR VIA THE A-BIT

If the Transmit DS3/E3 Framer block is configured to transmit the FEBE/REI indicator, then the following conditions/restrictions apply.

1. If the Transmit DS3/E3 Framer block is configured to support the transmission of the FEBE/REI indicator, then it WILL NOT support the transmission of the FERF/RDI indicator.
2. The Transmit DS3/E3 Framer block will only support the transmission of the FEBE/REI indicator, if it (and the corresponding Receive DS3/E3 Framer block) has been configured to support the computation/ insertion and verification of the BIP-4 value within each E3 frame.

The role of the "A" bit-field (within the E3 frame, when configured to carry the "FEBE/REI" indicator) is best presented in the practical example below.

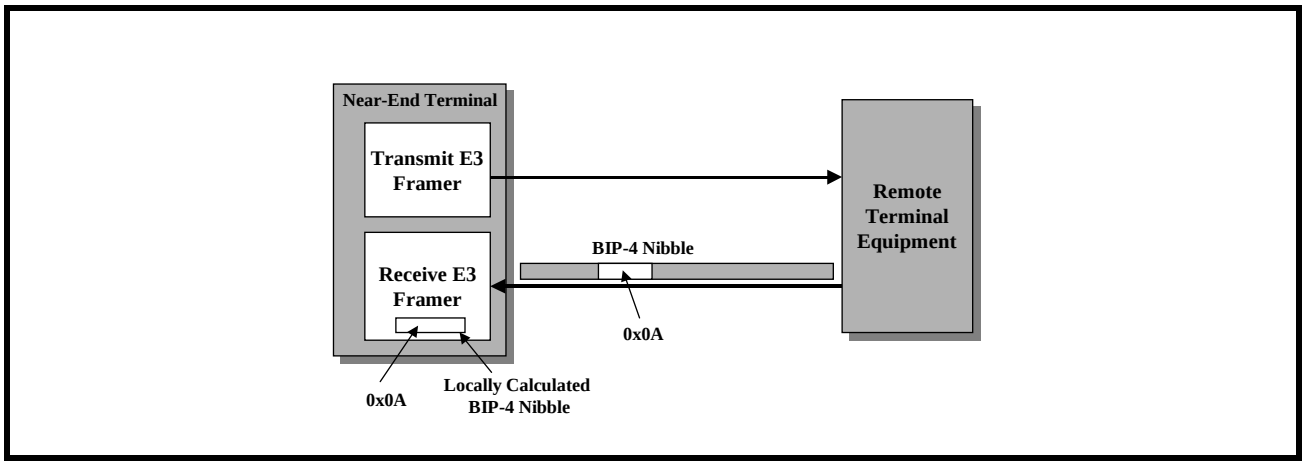
Example:

Consider a "Near-End" terminal that is communicating with a remote terminal. This "Near-End" terminal consists of the "Transmit DS3/E3 Framer" block and the "Receive DS3/E3 Framer" block within the XRT79L71, as depicted below in **Figure 148**.

The "Transmit DS3/E3 Framer" block will generate and transmit E3 frames to the remote terminal. Likewise, the "Receive DS3/E3 Framer" block will receive and process E3 frames, originating from the remote terminal. The "Near-End" Receive DS3/E3 Framer block (e.g., the Receive DS3/E3 Framer block within this particular device) is going to verify the values of the BIP-4 nibble within the incoming E3 frames (from the remote terminal equipment). If the "Near-End" Receive DS3/E3 Framer block detects no BIP-4 errors in the incoming E3 frame, then it will notify the remote terminal of this fact by forcing the "Near-End" Transmit DS3/E3 Framer block, to set the "A" bit, within the very next "outbound" E3 frame (which is destined for the remote terminal) to "0". This phenomenon is illustrated in **Figure 148** and **Figure 149**, below.

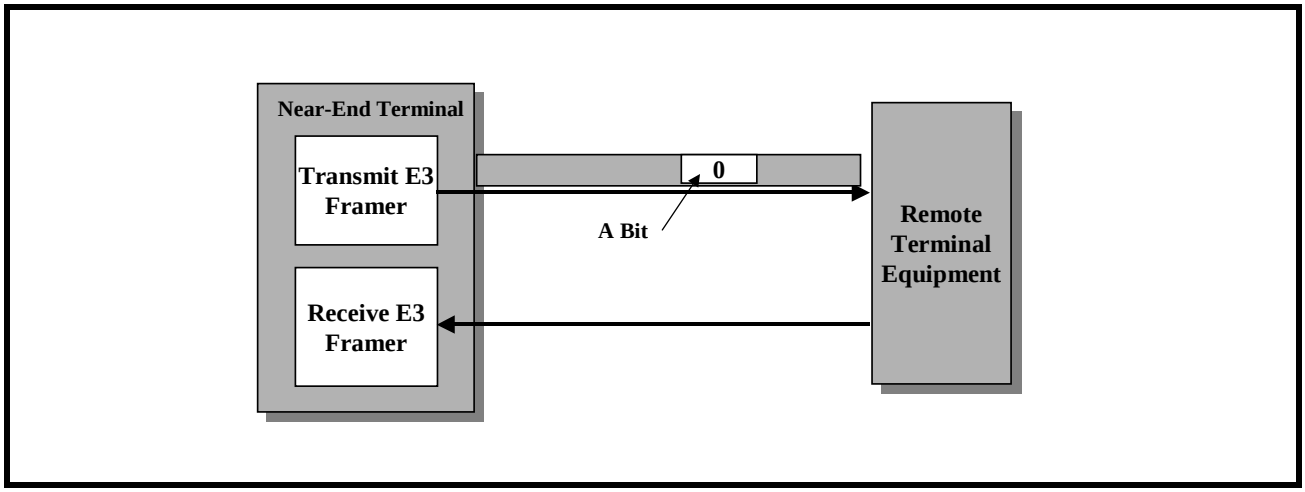
Figures _ illustrates the "Near-End" Receive DS3/E3 Framer block receiving a "error-free" E3 frame. In this figure, the Receive DS3/E3 Framer block has locally computed a BIP-4 value of "0x0A" (that was computed over a given incoming E3 frame). **Figure 148** indicates that the BIP-4 nibble value (which resides within the very next incoming E3 frame) is of the value "0x0A". As a consequence there is no BIP-4 nibble error being detected at this time.

FIGURE 148. A SIMPLE ILLUSTRATION OF A "NEAR-END" TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER.



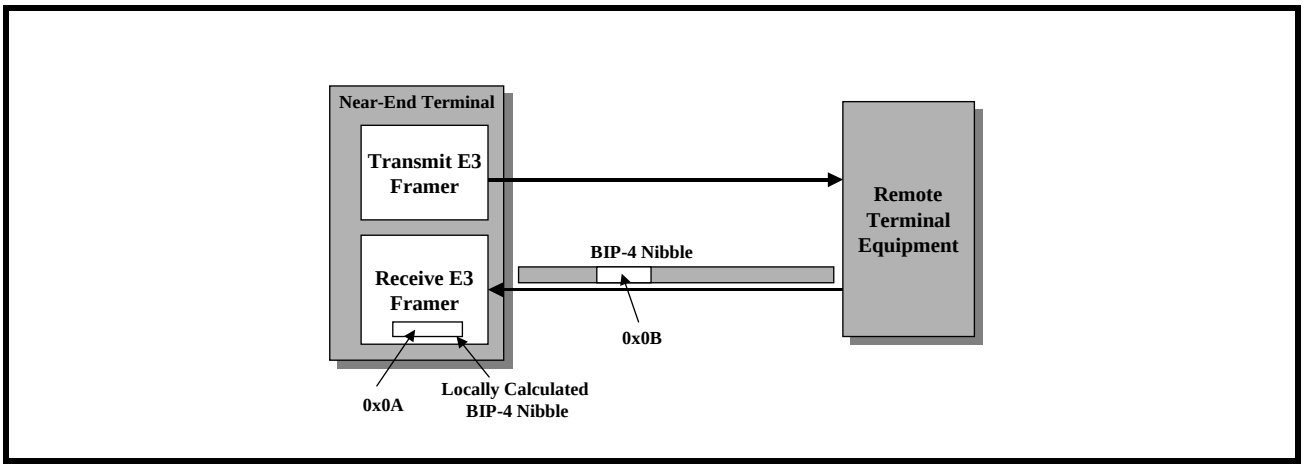
In response to this "un-erred" condition, the Transmit DS3/E3 Framer block (within the "Near-End" Terminal) will respond by setting the "A" bit-field to "0". Figure 149 presents an illustration of the Transmit DS3/E3 Framer block sending this "un-erred" indicator to the Remote Terminal Equipment.

FIGURE 149. A SIMPLE ILLUSTRATION OF THE "NEAR-END" TERMINAL TRANSMITTING THE "UN-ERRED" INDICATION TO THE REMOTE TERMINAL EQUIPMENT



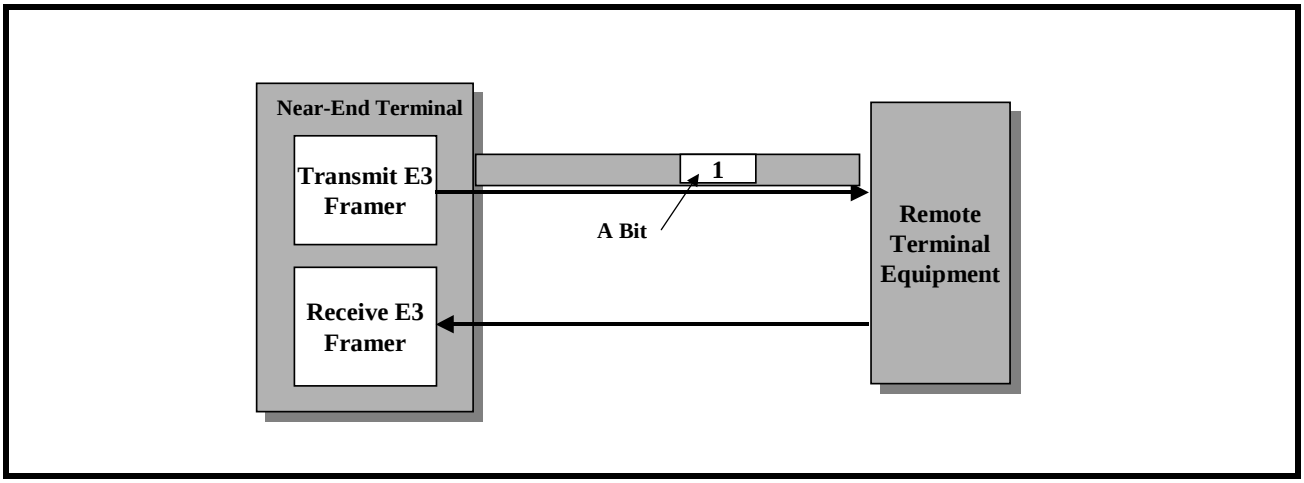
Next, Figure 150 presents an illustration of a given "Near-End" Terminal that is detecting a BIP-4 nibble error, within its incoming E3 signal. More specifically, the Receive DS3/E3 Framer block has locally computed a BIP-4 nibble value of "0x0A" (that was computed over a given E3 frame). Figure 150 indicates that the BIP-4 value (which resides within the very next incoming E3 frame) is of the value "0x0B". As a consequence, a BIP-4 nibble error is being detected at this time.

FIGURE 150. A SIMPLE ILLUSTRATION OF A "NEAR-END" TERMINAL DETECTING BIP-4 NIBBLE ERRORS WITHIN ITS INCOMING E3 SIGNAL



In response to this "erred" condition, the Transmit DS3/E3 Framer block (within the "Near-End" Terminal) will respond by setting the "A" bit, within the very next outbound E3 frame, to the value "1" in order to denote a FEBE/REI event. More specifically, the Transmit DS3/E3 Framer block (within the "Near-End" Terminal) will transmit an E3 frame (with the "A" bit-field set to "1") each time the corresponding Receive DS3/E3 Framer block (also within the "Near-End" Terminal) detects a BIP-4 nibble within an E3 frame. **Figure 151** presents an illustration of the Transmit DS3/E3 Framer block sending the FEBE/REI indicator to the remote terminal equipment.

FIGURE 151. A SIMPLE ILLUSTRATION OF THE "NEAR-END" TERMINAL EQUIPMENT TRANSMITTING THE FEBE/REI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



For information on how the Receive DS3/E3 Framer block processes the "A" bit-fields within each incoming E3 frame, please see **SEE"DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS" ON PAGE 390.**.

The Transmit DS3/E3 Framer block provides the user with two options associated with transmitting the FEBE/REI indicator (via the "A" bit).

- Forced transmission of the FEBE/REI indicator (e.g., under Software Control)
- Automatic transmission of the FEBE/REI indicator (in response to the Receive DS3/E3 Framer block detecting BIP-4 nibble errors)

5.2.4.4.1 Forced Transmission of the FEBE/REI Indicator

The XRT79L71 permits the user to force the Transmit DS3/E3 Framer block to force the transmission of the FEBE/REI indicator, via the "A" bit (upon Software Control) to the remote terminal equipment. The user can accomplish this by executing the following two-step procedure.

STEP 1 - Set Bit 1 (TxA), within the "Transmit E3 Service Bits" Register to "1" as depicted below.

Transmit E3 Service Bits Register - G.751 (Address = 0x1135)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused						TxA	TxN
R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	1	0

STEP 2 - Set Bits 5 and 6 (TxASrcSel[1:0]), within the "Transmit E3 Configuration" Register, to "[0, 0]" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS Source Sel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once these two steps are executed, then the Transmit DS3/E3 Framer block will begin to read out the contents of "Bit 1" (TxA), within the "Transmit E3 Service Bits" Register, and it will be setting the "A" bit (within each outbound E3 frame) to this particular value. This action will result in the transmission of the "FEBE/REI" indicator being transmitted to the remote terminal equipment.

NOTE: *In this configuration, the value of the "A" bit-field, that is being generated by the Transmit DS3/E3 Framer block does not reflect the health of the E3 signal that is being received by the corresponding "Near-End" Receive DS3/E3 Framer block.*

5.2.4.4.2 Automatic Transmission of the FEBE/REI Indicator

The XRT79L71 permits the user to configure the Transmit E3 Framer block to automatically transmit the FEBE/REI indicator, in response to whenever the Receive DS3/E3 Framer block detects BIP-4 nibble errors.

Configure the Transmit DS3/E3 Framer block to automatically transmit the FEBE/REI indicator to the remote terminal equipment (via the A bit) by executing the following two steps.

STEP 1 - Set Bit 0 (RxBIP-4 Enable), within the "Receive E3 Configuration and Status Register # 1 - G.751" to "1" as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERFAlgo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	1

STEP 2 - Set Bit 7 (TxBIP-4 Enable), within the "Transmit E3 Configuration" Register to "1" as depicted below.

Transmit E3 Configuration Register - G.751 (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxBIP-4 Enable	TxASrcSel[1:0]		TxNSrcSel[1:0]		TxAIS Enable	TxLOS Enable	TxFAS SourceSel
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	0	0	0

Once these two steps are executed, then the Receive DS3/E3 Framer block will be configured to compute and verify the BIP-4 nibble values within each incoming E3 frame. Further, the Transmit DS3/E3 Framer block will be configured to automatically set the "A" bit (within its outbound E3 frame) to "1", anytime the Receive DS3/E3 Framer block detects and flags BIP-4 errors.

5.2.4.5 Setting the Transmit E3 Framer Block Timing Reference

When designing a system, a decision must be made to configure the system to operate in either the Local-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from a "local" [or in-system] clock source) or in the Loop-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from the "remote terminal equipment's" clock source. The XRT79L71 can be configured to support either of these applications.

The XRT79L71 supports the following three (3) different "Timing Reference" Modes.

- Local-Timing/Asynchronous
- Local-Timing/TxFrameRef
- Loop-Timing

5.2.4.5.1 Local-Timing/Asynchronous Mode

If the XRT 79L71 is configured to operate in the Local-Timing/Asynchronous Mode, then the following are true.

- The Transmit E3 Framer block will use the clock signal that is applied to the "TxInClk" input pin as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.
- E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT 79L71.

Configuring the XRT 79L71 to operate in the Local-Timing/Asynchronous Mode

The XRT79L71 can be configured to operate in the "Local-Timing/Asynchronous" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[1, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT79L71 is configured to operate in the "Local-Timing/Asynchronous" Mode, do the following.

- Apply a 34.368MHz clock signal to the TxInClk input pin (Ball C10).
- Tie the TxFrameRef input pin (Ball A11) to GND.

5.2.4.5.2 Local-Timing/TxFrameRef Mode

If the XRT 79L71 is configured to operate in the Local-Timing/TxFrameRef Mode, then the following are true.

· The Transmit E3 Framer block will use the clock signal that is applied to the "TxInClk" input pin as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.

· The Transmit E3 Framer block will initiate the generation (and transmission) of a new E3 frame, anytime it detects a rising edge at the TxFrameRef input pin. In this case E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) will be synchronized to the signal that is applied to the TxFrameRef input pin.

Configuring the XRT79L71 to operate in the Local-Timing/TxFrameRef Mode

The XRT 79L71 can be configured to operate in the "Local-Timing/TxFrameRef" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT79L71 is configured to operate in the "Local-Timing/TxFrameRef" Mode, do the following.

- Apply a 34.368MHz clock signal to the TxInClk input pin (Ball C10).
- Apply a 22.375kHz clock signal to the TxFrameRef input pin (Ball A11).
- This 22.375kHz clock signal (that is being applied to the TxFrameRef input pin) must be synchronous with the 34.368MHz clock signal (that is being applied to the TxInClk input pin).

5.2.4.5.3 Loop-Timing Mode

If the XRT79L71 is configured to operate in the "Loop-Timing Mode, then all of the following are true.

- The Transmit E3 Framer block will use the "Recovered" clock signal (from the Receive DS3/E3 LIU Block) as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.
- E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT79L71.

Configuring the XRT 79L71 to operate in the Loop-Timing Mode

The XRT 79L71 can be configured to operate in the "Local-Timing/TxFrameRef" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 0]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Loop-Timing" Mode

If the XRT79L71 is configured to operate in the "Loop-Timing" Mode, do the following.

- Insure that either one of the following conditions are true.
  - **a.** That the Receive DS3/E3 LIU Block is receive a proper E3 line signal from the remote terminal equipment, or
  - **b.** That the SFM Synthesizer block (within the Receive DS3/E3 LIU Block) is configured to generate a 34.368MHz clock (to the remainder of the Receive DS3/E3 LIU Block circuitry) from either an externally supplied 12.288MHz or a 34.368MHz clock signal. (Please see Section 5.3.1.5 for details on how to accomplish this)
- · Tie the TxFrameRef input pin (Ball A11) to GND.

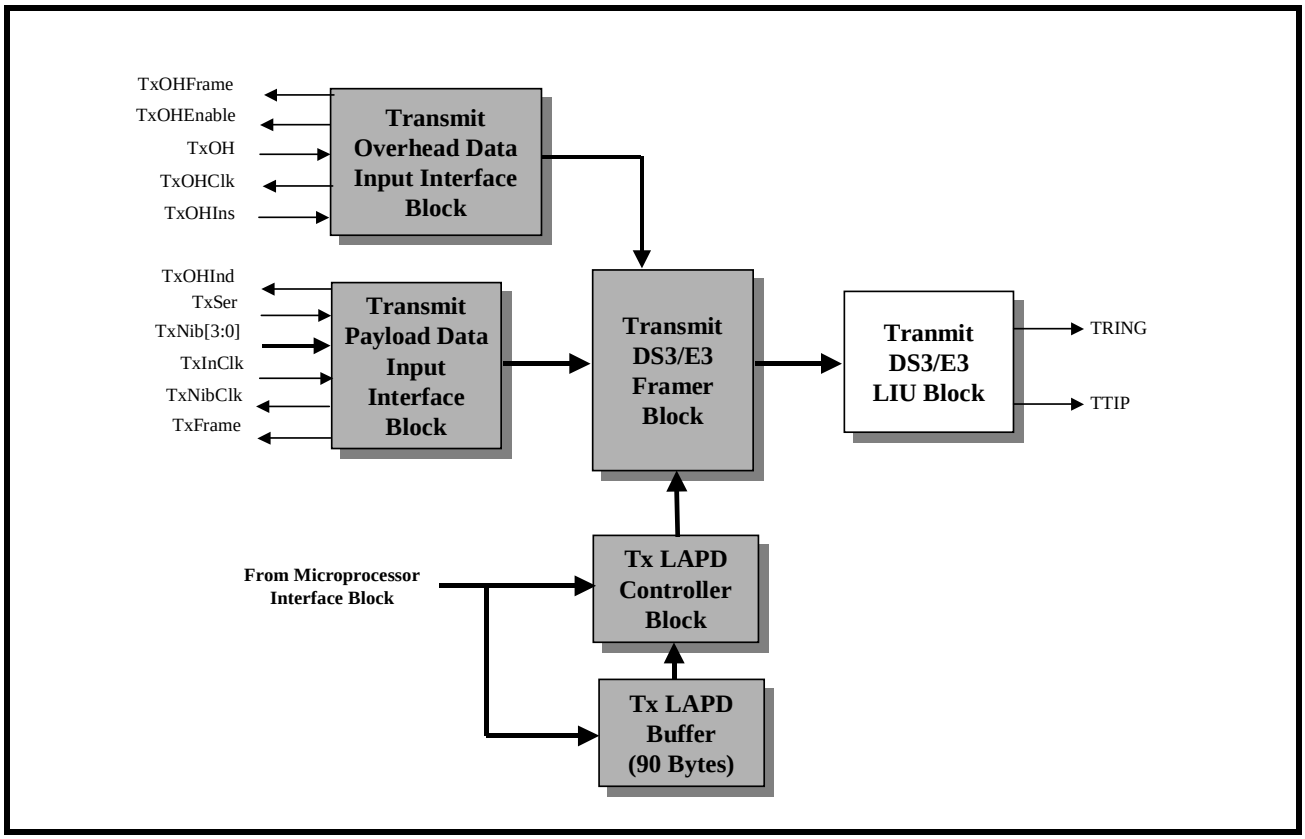
NOTE: *In order to permit the Microprocessor Interface to function (for Revision A silicon) the user is still required to supply a sufficiently high frequency clock signal to the "TxInClk" input pin, even if the XRT 79L71 is configured to operate in the "Loop-Timing" Mode.*

5.2.4.6 Controlling the State of the N-Bit within the outbound E3 data-stream

5.2.5 TRANSMIT DS3/E3 LIU BLOCK - E3 APPLICATIONS

The Transmit DS3/E3 Framer block is the fifth functional block (within the Transmit Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. **Figure 152** presents an illustration of the "Transmit Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the "Transmit DS3/E3 LIU" block highlighted.

FIGURE 152. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE "E3, ITU-T G.751 CLEAR-CHANNEL FRAMER" MODE (WITH THE "TRANSMIT DS3/E3 FRAMER" BLOCK HIGHLIGHTEDL



The purpose of the Transmit DS3/E3 LIU Block is to accept a DS3/E3 data-stream from the Transmit DS3/E3 Framer block and to perform all of the following operations on this data.

- To convert this "outbound" data into the B3ZS line code (for DS3 applications) or into the HDB3 line code (for E3 applications).
- For DS3 Applications, to further convert this data into a DS3 line signal such that each pulse that is generated by the Transmit DS3 LIU block will comply with the Isolated Pulse Template requirements per Bellcore GR-499-CORE.
- For E3 Applications, to further convert this data into an E3 line signal such that each pulse that is generated by the Transmit E3 LIU block will comply with the ITU-T G.703 Pulse Template requirements for E3 applications.

This particular section will describe the functionality and configuration options of the Transmit DS3/E3 LIU Block for E3 applications. The functionality and configuration options of the Transmit DS3/E3 LIU Block for DS3 applications are discussed in Section 4.2.6.

Figure 153 presents a more detailed illustration of the Transmit DS3/E3 LIU Block within the XRT79L71.

FIGURE 153. ILLUSTRATION OF THE TRANSMIT DS3/E3 LIU BLOCK WITHIN THE XRT79L71

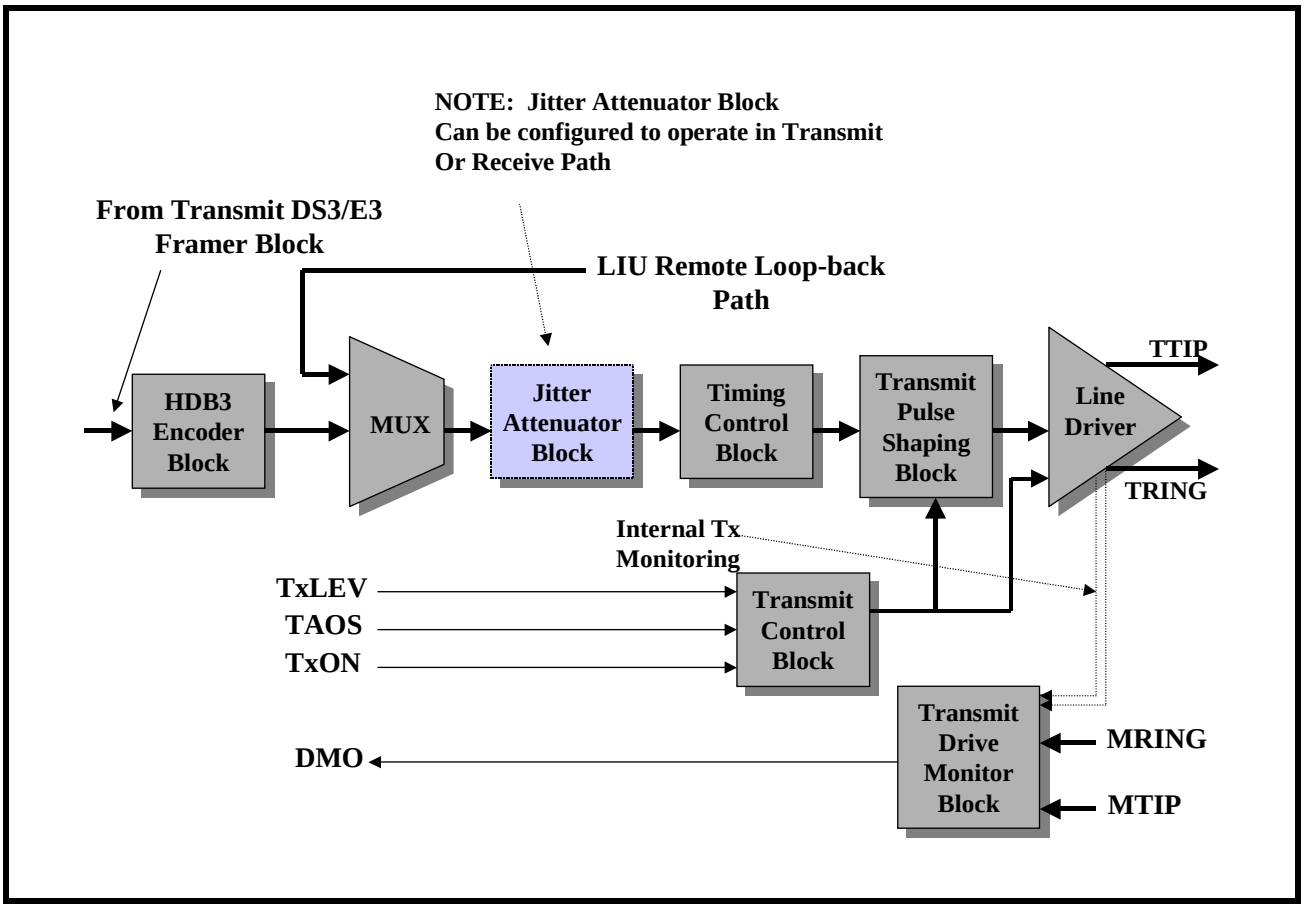


Figure 153 indicates that the Transmit DS3/E3 LIU Block consists of the following functional blocks.

- The HDB3 Encoder Block
- The Jitter Attenuator (which can be configured to operate in either the Transmit or Receive Directions)
- The Transmit Control Block
- The Transmit Pulse Shaping Block
- The Transmit Line Driver Block
- The Transmit Drive Monitor Block

5.2.5.1 The HDB3 Encoder Block

The purpose of the HDB3 Encoder block is to encode the "outbound" E3 traffic into the "HDB3 Line Code". In the case of the XRT79L71, the HDB3 Encoder block will always be enabled, and the user has no ability to disable the HDB3 Encoder block.

5.2.5.2 The Jitter Attenuator Block

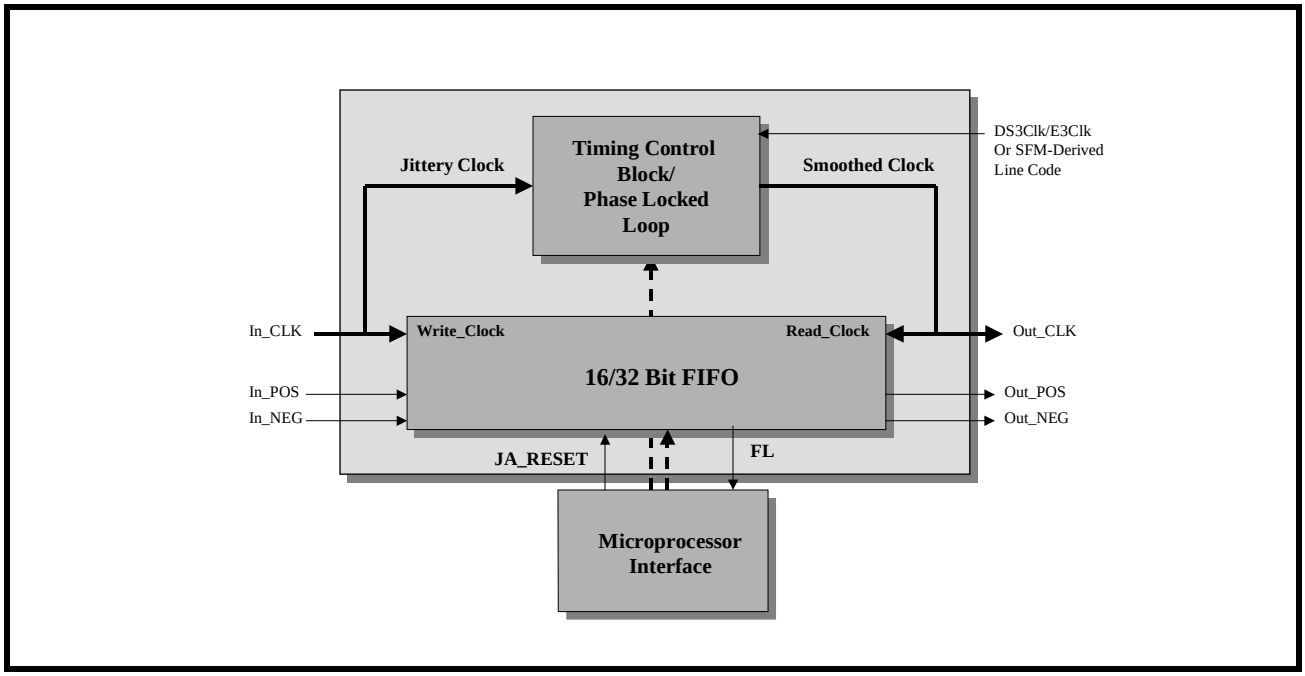
The XRT79L71 includes a Jitter Attenuator block that can be configured to operate in either the Transmit Direction (e.g., within the Transmit DS3/E3 LIU Block) or in the Receive Direction (e.g., within the Receive DS3/E3 LIU Block). The purpose of the Jitter Attenuator block is to permit the XRT79L71 to comply with all of the following "Jitter Transfer Characteristic" requirements.

- Bellcore GR-499-CORE "Category II to Category II Interfaces" (DS3 Applications)
- TBR-24 34Mbps D34U and D34S System Requirements (for E3 Applications)

Each of these requirements, and how the XRT79L71 performs against these requirements is described in detail below.

5.2.5.2.1 The Basic Architecture of the Jitter Attenuator Block

FIGURE 154. AN ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE JITTER ATTENUATOR BLOCK



According to Figure 154, the Jitter Attenuator consists of the following functional blocks.

- The Timing Control/Phase-Locked Loop Block
- The Jitter Attenuator FIFO

How the Jitter Attenuator Works

The Jitter Attenuator Block (within the XRT79L71) will accept "jittery" clock (In_Clk) and data signals (In_POS and In_NEG) from either the HDB3/B3ZS Encoder Block (if the Jitter Attenuator has been configured to operate in the Transmit Direction) or from the "Clock and Data Recovery" Block (if the Jitter Attenuator has been configured to operate in the Receive Direction). The Jitter Attenuator block will latch the data, residing on internal signals "In_POS" and "In_NEG" into the "Jitter Attenuator FIFO" upon the appropriate edge of the In_Clk signal.

In parallel, the In_Clk signal will also be routed to the "Timing Control/PLL" Block. This block consists of a Narrow-band PLL which has the responsibility of attenuating much of the jitter within the "In_Clk" signal. The "smoothed" version of this clock signal will be output from the Jitter Attenuator block (and is designated as "Out_Clk" in Figure 154). Further, the "Jittery Data" (which was latched into the Jitter Attenuator FIFO via the "In_Clk" signal) will now be clocked out of the Jitter Attenuator block via the "Out_Clk" signal.

NOTES:

1. *If the Jitter Attenuator block has been configured to operate in the "Transmit Direction" and if the XRT79L71 has been configured to operate in the "Local-Timing" Mode, then the "In_CLK" signal (as depicted in Figure 154, above) will be a buffered version of clock signal being applied to the "TxInClk" input pin.*
2. *If the Jitter Attenuator block has been configured to operate in the Transmit Direction" and if the XRT79L71 has been configured to operate in the "Loop-Timing" Mode, then the "In_CLK" signal (as depicted in FFigure 154, above) will be a buffered version of the "Recovered" clock signal (from the Receive DS3/E3 LIU Block).*

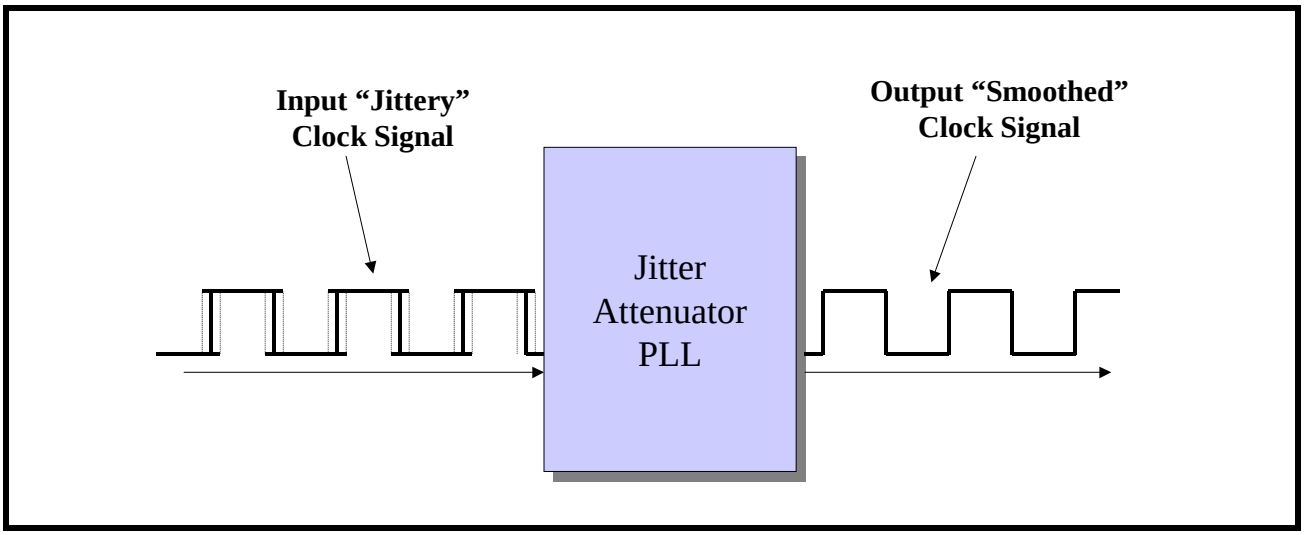
3. *If the Jitter Attenuator block has been configured to operate in the "Receive Direction", then the "In_CLK" signal (as depicted in* **Figure 154***, above) will be a buffered version of the "Recovered" clock signal (from the "Receive DS3/E3 LIU Block).*

Now that we have briefly described how the Jitter Attenuator block functions, we can now go into more details on how the individual functional blocks (within the Jitter Attenuator block) function.

5.2.5.2.1.1 The Jitter Attenuator PLL

The purpose of the Jitter Attenuator PLL block is to (1) receive and lock onto an input "jittery" clock signal, and (2) to regenerate a clock signal, of the exact same frequency, but with considerably less jitter. In performing this task, the Jitter Attenuator PLL block is said to be "attenuating jitter". A very simplistic illustration of what the Jitter Attenuator PLL has been designed to accomplish is presented below in **Figure 155**.

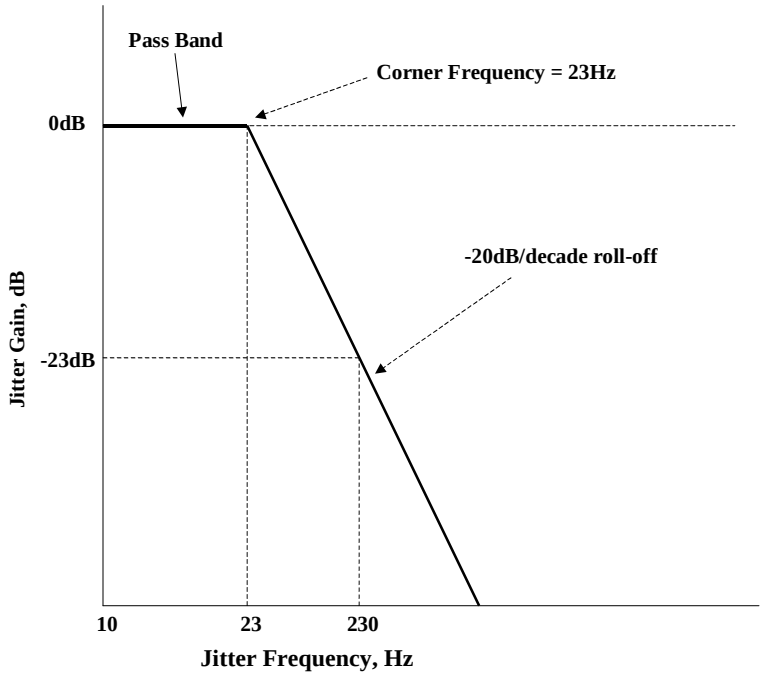
FIGURE 155. A SIMPLISTIC ILLUSTRATION OF THE ROLE/FUNCTION OF THE JITTER ATTENUATOR PLL BLOCK WITHIN THE XRT79L71.



The Jitter Attenuator PLL receives the input "jittery" clock signal via the "In_CLK" input pin. It outputs the "smoothed" (e.g., clock signal with reduced jitter) clock signal via the "Out_CLK" output pin.

The Jitter Attenuator PLL accomplishes this "Jitter Attenuation" task because it is a "very narrow-band" PLL. The corner (e.g., -3dB) frequency of the Loop-Filter (within the Jitter Attenuator PLL) is about 23Hz. A simple illustration of the "Jitter Gain (or Attenuation) Transfer Characteristics of the Jitter Attenuator PLL (within the XRT79L71) is illustrated below in **Figure 156**.

FIGURE 156. A SIMPLE ILLUSTRATION OF THE JITTER TRANSFER CHARACTERISTICS OF EACH JITTER ATTENUATOR PLL (WITHIN THE XRT79L71)



All of this means that if the Jitter Attenuator PLL receives a clock signal, that contains jitter with a frequency of about 23Hz, the Jitter Attenuator PLL will reduce the amplitude of this "23Hz jitter component" by about 3dB or 50%. The "Pass-Band" of the Loop-filter (within the Jitter Attenuator PLL) is any frequency below 23Hz. This means that the Jitter Attenuator PLL (within the XRT79L71) will not provide much attenuation on any jitter (within the input In_CLK clock signal) that is of frequencies less than 23Hz. As a consequence, much of the low-frequency jitter that appears at the "In_CLK" input, will also appear at the "Out_CLK" output pins. The Jitter Attenuator PLL will provide more than 3dB of jitter attenuation for jitter with frequencies greater than 23Hz. The Jitter Transfer Characteristics (of the Jitter Attenuator PLL, within the XRT79L71) is such that for frequencies greater than 23Hz, it imposes a -20dB/decade roll-off in the "Gain versus Frequency" curve (as presented in **Figure 156**). Therefore, if the Jitter Attenuator PLL block receives a signal (via the "In_CLK_n" input pin) that contains jitter which is of a frequency of 230Hz, then the Jitter Attenuation PLL will reduce the amplitude of this jitter by 23dB or by 95%. In general, the higher the frequency of the jitter (within the In_CLK_n input signal), the greater the jitter amplitude will be attenuated.

DISABLING THE JITTER ATTENUATOR PLL

The XRT79L71 permits the user to disable the Jitter Attenuator PLL. If the Jitter Attenuator PLL is disabled, then the Jitter Attenuator block will perform no jitter attenuation on the "In_CLK" input signal. More specifically, the Jitter Attenuator FIFO Block will also be by-passed, and the signal path (through the Jitter Attenuator block) will proceed to emulate the behavior of three wires. In this mode, the "In_CLK" input signal (within the Jitter Attenuator block) will essentially be shorted to the "Out_CLK" output signal. Further, the "In_POS" input signal will essentially be shorted to the "Out_POS" output signal; and the "In_NEG" input signal will essentially be shorted to the "Out_NEG". Data residing on the "In_POS" and "In_NEG" input pins will not be clocked into the Jitter Attenuator FIFO via the In_CLK input signal. Further, data will not be clocked out (via the "Out_POS" and "In_NEG_n" output pins) upon the Out_CLK_n output signal. This data will simply propagate through the Jitter Attenuator block; just as if it were three wires.

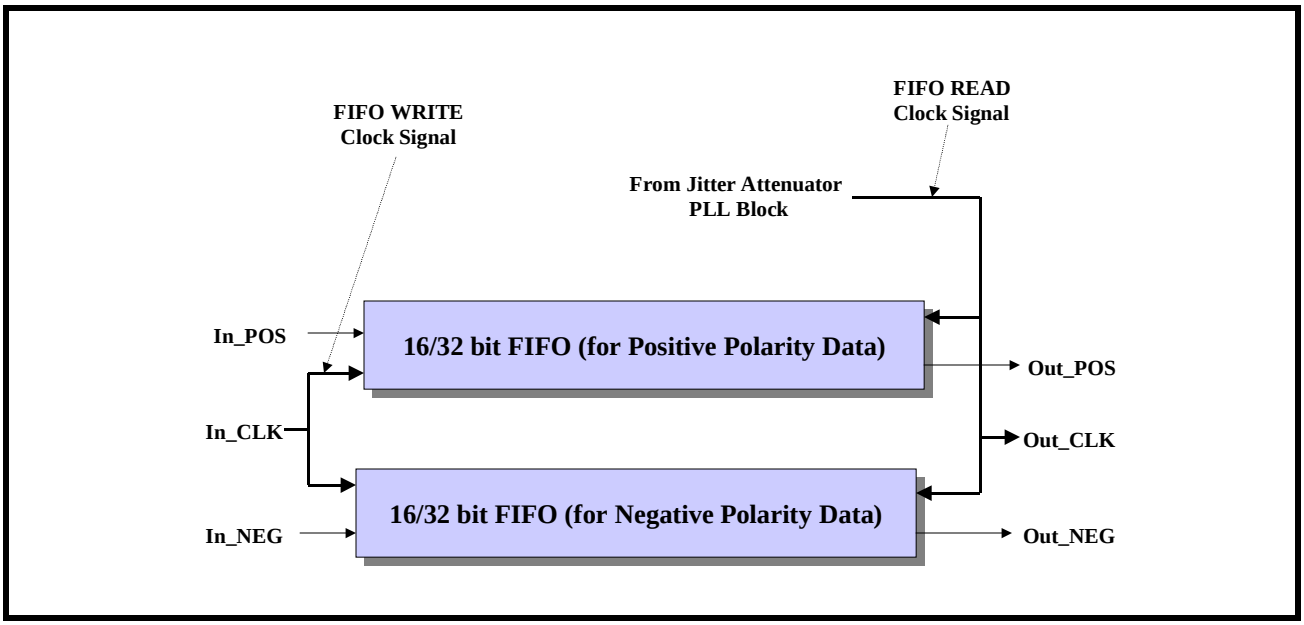
To enable or disable the Jitter Attenuator PLL, within the XRT79L71 by execute the procedure presented in SEE"ENABLING THE JITTER ATTENUATOR BLOCK AND SELECTING THE JITTER ATTENUATOR FIFO SIZE" ON PAGE 341..

5.2.5.2.1.2 The Jitter Attenuator FIFO

The Jitter Attenuator Block contains a 2-Channel FIFO. The purpose of this FIFO is to permit the Jitter Attenuator to absorb any instantaneous frequency differences between the In_CLK input signal and the Out_CLK output signal and to mitigate the occurrences of bit-errors.

The Jitter Attenuator FIFO actually consists of 2 FIFO channels, in the sense that one FIFO channel is dedicated for the "Positive-Polarity" Data (e.g., the In_POS to Out_POS path) and the other FIFO channel is dedicated for the "Negative-Polarity" Data (e.g., the In_NEG to Out_NEG path). The Physical Architecture of the 2-Channel Jitter Attenuator FIFO is presented below in Figure 157.

FIGURE 157. ILLUSTRATION OF THE PHYSICAL ARCHITECTURE OF 2-CHANNEL JITTER ATTENUATOR FIFO ARCHITECTURE WITHIN THE JITTER ATTENUATOR BLOCK



NOTE: *The Logical Architecture of the Jitter Attenuator FIFO is presented in the next section.*

SELECTING THE FIFO SIZE

The user can configure the Jitter Attenuator FIFO to operate with a depth of either 16 or 32 bits. A description upon (a) how to configure the FIFO to operate with a 16 or 32 depth, and (b) its operation, is presented below in SEE"ENABLING THE JITTER ATTENUATOR BLOCK AND SELECTING THE JITTER ATTENUATOR FIFO SIZE" ON PAGE 341..

Operating with a Jitter Attenuator FIFO Depth of 16 bits

If the Jitter Attenuator FIFO is configured to operate with a depth of 16 bits, then the following is true.

a. When the XRT79L71 first powers up, or experiences a "Hardware RESET", then the location of the FIFO_READ and FIFO_WRITE pointers will be 8 bits (or one-half the FIFO Size) apart from each other.

b. As a consequence, data, which is applied to the Jitter Attenuator block (via the In_POS_n and In_NEG_n input pins) will be written into the FIFO (into a location determined by the FIFO_WRITE pointer). This same data will be read out of the FIFO approximately 8 bit periods later once the "FIFO_READ" pointer has incremented around to this particular position within the FIFO. Hence, for 16-bit mode operation, the Jitter Attenuator FIFO imposes a nominal latency of 8 bit periods.

Configuring the Jitter Attenuator FIFO Depth to 16 bits:

The user can configure the Jitter Attenuator FIFO to operate with a depth of 16 bits by executing the procedure presented in SEE"ENABLING THE JITTER ATTENUATOR BLOCK AND SELECTING THE JITTER ATTENUATOR FIFO SIZE" ON PAGE 341..

Operating with a Jitter Attenuator FIFO depth of 32 bits.

If the Jitter Attenuator FIFO is configured to operate with a depth of 32 bits, then the following is true.

a. When the XRT79L71 first powers up, or experiences a "Hardware RESET", then the location of the FIFO READ and FIFO WRITE pointers will be 16 bits (or one-half the FIFO size) apart from each other.

b. As a consequence, data, which is applied to the Jitter Attenuator block (via the In_POS and In_NEG input pins) will be written into the FIFO (into a location determined by the FIFO WRITE pointer). This same data will be read out of the FIFO approximately 16 bit periods later once the "FIFO_READ" pointer has incremented around to this particular position within the FIFO. Hence, for 32-bit mode operation, the Jitter Attenuator FIFO imposes a nominal latency of 16 bit periods.

Configuring the Jitter Attenuator FIFO Depth of 32 bits:

To configure the Jitter Attenuator FIFO to operate with a depth of 32 bits, execute the procedure presented in Section 4.3.1.5.2.2.

WRITING DATA INTO THE FIFO

The Jitter Attenuator block accepts data via the In_POS and In_NEG input pins. Data on these pins are sampled and written into the "2-Channel" Jitter Attenuator FIFO upon the appropriate edge of the "In_CLK" input signal. The exact location (within the FIFO) that this data is written to, depends upon the location of the "FIFO_WRITE" pointer. Once a given sample of data has been loaded into the FIFO (at the location specified by the "FIFO_WRITE" pointer), the location of the "FIFO_WRITE" pointer will then be incremented to the next position within the FIFO, and the process repeats during the next period of In_CLK. It is appropriate to think of the Jitter Attenuator FIFO as a "circular-buffer"; in the sense that once the "FIFO_WRITE" pointer has reached the last bit, within the FIFO, it will "wrap-around" back to the first bit, within the FIFO. This concept is discussed in greater detail, in the FIFO Limit Alarm section of this data sheet.

It is important to note that the writing of data into the FIFO, and the incrementing of the "FIFO_WRITE" pointer is synchronized to the "In_CLK" (jittery clock) input signal.

READING DATA FROM THE FIFO

The Jitter Attenuator block, within the XRT79L71, reads out data from the Jitter Attenuator FIFO, and outputs this data via the "Out_POS" and "Out_NEG" output pins. Data is output via these pins, upon the appropriate edge of the "Out_CLK" output signal. The exact location (within the FIFO) that this data is read from, depends upon the location of the "FIFO_READ" pointer. Once a given sample of data has been extracted from the FIFO (at the location specified by the "FIFO_READ" pointer), the "FIFO_READ" pointer will then be incremented to the next position within the FIFO, and the process repeats, during the next period of Out_CLK. As in the case of the "FIFO_WRITE" pointer, it is appropriate to think of the Jitter Attenuator FIFO as a "circular buffer" in the sense that once "FIFO_READ" pointer has reached the last bit, within the FIFO, it will "wrap-around" back to the first bit, within the FIFO. This concept is discussed in greater detail, in the "FIFO Limit Alarm" section of this data sheet.

It is important to note that the reading of data from the FIFO, and the incrementing of the "FIFO_READ" pointer is synchronized to the "Out_CLK" (smoothed clock) output signal.

THE FIFO LIMIT ALARM

Whenever the XRT79L71 is initially powered-up, or experiences a "Hardware RESET", the locations of the "FIFO_WRITE" and "FIFO_READ" pointers (within the Jitter Attenuator block) will initially be 8 bit positions (if the Jitter Attenuator FIFO depth is configured to be 16-bits) or 16 bit positions apart from each other (if the Jitter Attenuator FIFO depth is configured to be 32-bits).

In the previous section, we mentioned that the "FIFO_WRITE" pointer is incremented (within the Jitter Attenuator FIFO) with each period of the In_CLK (e.g., the jittery clock) input signal. Further, we also indicated

that the "FIFO_READ" pointer is incremented (within the Jitter Attenuator FIFO) with each period of the Out_CLK (e.g., the smoothed clock) output signal.

Additionally, we also mentioned that the Jitter Attenuator PLL block accepts a "jittery" clock signal, via the "In_CLK" input signal and from this input clock signal, it synthesizes another clock signal (of the exact same frequency, but with considerably less jitter). This "synthesized" clock signal is output via the "Out_CLK" output pin.

Therefore, on the average, the frequencies of the "In_CLK" input signal and the "Out_CLK" output signal are identical. As a consequence, on the average, the FIFO_READ and FIFO_WRITE pointers will each increment (throughout the Jitter Attenuator FIFO) at the same rate and the two pointers will typically maintain a constant distance from each other. Figure 158 presents an illustration of the Logical Architecture of the Jitter Attenuator FIFO, within the Jitter Attenuator block. This particular figure illustrates the relationship between the "FIFO_WRITE" and "FIFO_READ" pointers.

FIGURE 158. ILLUSTRATION OF THE JITTER ATTENUATOR FIFO AND THE FIFO_WRITE AND FIFO_READ POINTERS.

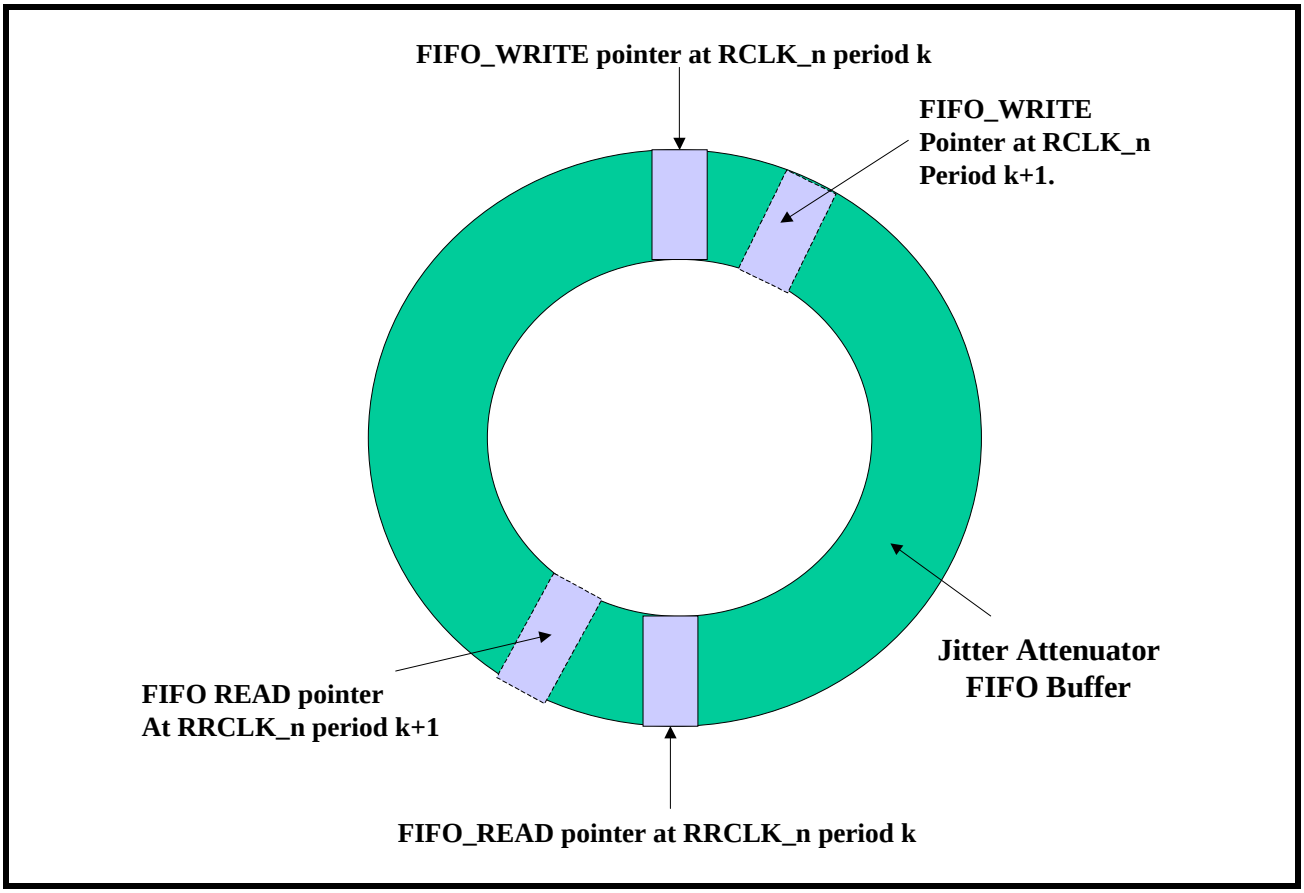


Figure 158 presents an illustration of the behavior of the "FIFO_READ" and the "FIFO_WRITE" pointers within the Jitter Attenuator FIFO. For the sake of discussion, this figure indicates that the initial position (e.g., "In_CLK" period "k") of the "FIFO_WRITE" pointer is at the "12 o'clock" position. This figure also indicates that during the very next "In_CLK" period, that the FIFO_WRITE pointer will have moved (in a clockwise direction) to the next position (which is labeled "FIFO_WRITE pointer at In_CLK period k + 1"). If the FIFO depth is configured to be 16 bits, then in 16 In_CLK cycles, after power-up or reset, the FIFO_WRITE pointer will have revolved around the circular buffer and will have incremented its location right back to its initial (or the "12 o'clock") position as is shown in Figure 13. Similarly, if the FIFO depth is configured to be 32 bits, then in 32 In_CLK cycles, after power-up or reset, the "FIFO_WRITE" pointer will have revolved around the circular buffer and will incremented its location back to its initial (e.g., "12 o'clock") position as shown in Figure 158.

Figure 158 also indicates that the initial position of the "FIFO_READ" pointer is at the "6 o'clock" position. Further, Figure 158 indicates that during the very next "Out_CLK" period the "FIFO_READ" pointer will have moved (in a clockwise direction) to the position labeled "FIFO_READ pointer at Out_CLK period k + 1".

If the FIFO Depth is configured to be 16 bits, then in 16 Out_CLK cycles, after power-up or reset, the "FIFO_READ" pointer will have revolved around the circular buffer and have incremented its position right back to the "6 o'clock" position) as is shown in Figure 158. Similarly, if the FIFO depth is configured to be 32 bits, then in 32 Out_CLK cycles, after power-up or reset, the FIFO_READ pointer will have revolved around the circular buffer and will have incremented its location back to its initial (e.g., "6 o'clock") position.

Recall that the "Out_CLK" (smoothed) output clock signal is derived from a PLL that has a very narrow bandwidth. Therefore, the Jitter Attenuator PLL will not be very responsive to instantaneous phase or frequency deviations that occur within the "In_CLK" (jittery) input clock signal. As the Jitter Attenuator PLL experiences large phase variations (via the In_CLK input pin) it will typically not pass along these large phase changes and variations to the Out_CLK output signal. This phenomenon will result in some modulation in the distance between the "FIFO_WRITE" and "FIFO_READ" pointers. If the modulation in the distance (between the "FIFO_WRITE" and "FIFO_READ" pointers) is such that the distance between these two pointers approach 0 bits, then bit-errors will occur. This modulation in the distance between the FIFO_WRITE and FIFO_READ pointers and its effect on the integrity of the data is described in some detail below.

The FIFO Under-run Condition

If the jitter within the "In_CLK" input clock signal is such that, for a very short period of time, the instantaneous frequency of "In_CLK" deviates, in the negative direction, (e.g., exhibits a lower frequency from that of the "Out_CLK" output clock signal) then the following events will happen.

- The Jitter Attenuator PLL will continue to generate the "Out_CLK" output clock signal at virtually a constant rate (e.g., with very little change in frequency). As a consequence, the FIFO_READ pointer will continue to increment and revolve about the circular buffer at largely a fixed rate.
- Because the In_CLK input clock signal is, for a very short period, of a lower frequency, the FIFO_WRITE pointer will now be incremented, around the circular buffer, at a lower rate than that of the "FIFO_READ" pointer.

If the above-mentioned condition were to exist long enough, the "FIFO_READ" pointer will eventually revolve around the circular buffer, and "catch-up" with the "FIFO_WRITE" pointer. If this happens, then a "FIFO Under-run condition" will be said to have occurred. In a "FIFO Under-run Condition", all new data that has been written into the Jitter Attenuator FIFO has been read out and depleted. At this point, the Jitter Attenuator block is now reading out data which has already been read out, at least once. This phenomenon will cause the XRT79L71 to transmit erroneous data to the remote terminal equipment.

The FIFO Overflow Condition

If the jitter within the "In_CLK" input clock signal is such that, for a very short period of time, the instantaneous frequency of "In_CLK" deviates, in the positive direction, (e.g., exhibits a much higher frequency from that of the "Out_CLK" output clock signal) then the following events will happen.

- The Jitter Attenuator PLL will continue to generate the "Out_CLK" at virtually a constant rate (e.g., with very little change in frequency). As a consequence, the FIFO_READ pointer will continue to increment and revolve about the circular buffer at largely a fixed rate.
- Because the "In_CLK" input clock signal is, for a very short period, of a higher frequency, the FIFO_WRITE pointer will now be incremented, around the circular buffer, at a higher rate than that of the "FIFO_READ" pointer.

If the above-mentioned condition were to exist long enough, the "FIFO_WRITE" pointer will eventually revolve around the circular buffer, and "catch-up" with the "FIFO_READ" pointer. If this happens, then an "Overflow condition" will be said to have occurred. In an "Overflow condition", new data has been written into locations within the Jitter Attenuator FIFO and overwriting data that the XRT79L71 has yet had an opportunity to read it out. This phenomenon will cause the XRT79L71 to drop or lose data.

Please see SEE"ALARM CONDITIONS OCCURRING WITHIN THE JITTER ATTENUATOR BLOCK" ON PAGE 342. for information on how the Jitter Attenuator block responds to a "FIFO Overflow" or "Underflow" condition.

The next few sections describe how to configure and use the Jitter Attenuator block

5.2.5.2.2 Enabling the Jitter Attenuator Block and selecting the Jitter Attenuator FIFO Size

The user will both enable the Jitter Attenuator block and select (and implement) a FIFO size by writing the appropriate values into Bits 0 (Jitter Attenuator PLL/FIFO Operation Mode - Bit 0) and 2 (Jitter Attenuator PLL/FIFO Operation Mode - Bit 2), as depicted below.

Jitter Attenuator Control Register (Address = 0x1307)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			DFL	Jitter Attenuator FIFO Pointer RESET	Jitter Attenuator PLL/FIFO Operating Mode - Bit 1	Jitter Attenua-tor in Transmit Path	Jitter Attenua-tor PLL/FIFO Operating Mode - Bit 0
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	X	0	X

Table 44 presents the relationship between the states of Bits 2 and 0 (within the "Jitter Attenuator Control" Register) and the (1) Enable/Disable State of the Jitter Attenuator, and (2) the Size of the Jitter Attenuator FIFO.

TABLE 44: THE RELATIONSHIP BETWEEN THE STATES OF BITS 2 AND 0 (WITHIN THE "JITTER ATTENUATOR CONTROL" REGISTER) AND THE (1) ENABLE/DISABLE STATE OF THE JITTER ATTENUATOR, AND (2) THE SIZE OF THE JITTER ATTENUATOR FIFO

BIT 2	BIT 0	JITTER ATTENUATOR STATE	FIFO SIZE
0	**0**	**ENABLED**	**16 BITS**
0	1	Disabled	N/A
1	0	Enabled	32 bits
1	1	Disabled	N/A

If the Jitter Attenuator is enabled, then it will perform its "Jitter Attenuation" processing on any DS3 or E3 signal that passes through it. However, if the Jitter Attenuator is disabled, then the "Narrow-band" PLL (within the Jitter Attenuator) will be by-passed. In this case, the Jitter Attenuator block will operate in a "transparent" mode (e.g., the DS3 or E3 signal will pass through the Jitter Attenuator block with no jitter attenuation processing performed on it, at all).

5.2.5.2.3 Configuring the Jitter Attenuator to operate in the Transmit or Receive Path

To configure the Jitter Attenuator block to operate in either the Transmit or Receive Direction, write the appropriate value into Bit 1 (Jitter Attenuator in Transmit Path), within the "Jitter Attenuator Control" Register as depicted below.

Jitter Attenuator Control Register (Address = 0x1307)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			DFL	Jitter Attenuator FIFO Pointer RESET	Jitter Attenuator PLL/FIFO Operating Mode - Bit 1	Jitter Attenuator in Transmit Path	Jitter Attenuator PLL/FIFO Operating Mode - Bit 0
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	X	X	X

Setting this bit-field to "0" configures the Jitter Attenuator to operate in the "Receive Path". Conversely, setting this bit-field to "1" configures the Jitter Attenuator to operate in the "Transmit Path".

5.2.5.2.4 Alarm Conditions occurring within the Jitter Attenuator Block

As mentioned in **SEE"THE JITTER ATTENUATOR FIFO" ON PAGE 337.**, the FIFO Under-run and Overflow conditions occur if the distance between the "FIFO_READ" and "FIFO_WRITE" pointers reach 0 bits. In either of these conditions an error-condition is said to have occurred. The XRT79L71 contains some circuitry to alert the system that the distance between the FIFO_WRITE and FIFO_READ pointers has fallen to at least two (2) bit-positions. If this condition were to occur, then Jitter Attenuator FIFO is on the verge of experiencing either a "FIFO Under-run" or "Overflow" condition. In response to this condition, the XRT79L71 will notify the Line Card circuitry of this phenomenon by declaring a "FIFO Limit" Alarm condition. The XRT79L71 will indicate that it is declaring a "FIFO Limit Alarm" condition by doing all of the following.

1. It will set Bit 3 (FIFO Limit Alarm Declared), within the LIU Alarm Status Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

2. It will also generate the "Change of FL Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by
   a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")
   b. Setting Bit 3 (Change of FL Condition Interrupt Status) within the "LIU Interrupt Status" Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Clearing the FIFO Limit Defect Condition

Similarly, once the Jitter Attenuator Block has recovered from the "FIFO Limit" Alarm condition, then it indicate that it is clearing the "FIFO Limit" defect condition by doing all of the following.

1. It will set Bit 3 (FIFO Limit Alarm Declared), within the LIU Alarm Status Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

2. It will also generate the "Change of FL Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")

b. Setting Bit 3 (Change of FL Condition Interrupt Status) within the "LIU Interrupt Status" Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

5.2.5.2.5 The Jitter Transfer Characteristic for E3 Applications

This section presents the Jitter Transfer Characteristics of the XRT79L71, if the Jitter Attenuator PLL has been enabled. As we present the Jitter Transfer Characteristics of the XRT79L71, we will also present some of the various Jitter Transfer Characteristic requirements that the XRT79L71 complies with.

5.2.5.3 The Transmit Control Block

The purpose of the "Transmit Control" Block is to accept any of the following signals from the on-chip Registers, and route the state of these signals to either the Transmit Pulse Shaping Block and the Line Driver block.

- TAOS - Transmit All OneS Pattern
- TxON - Transmit Driver ON/OFF

The role that "TxON" plays is address in *Section 3.3.1.5.6.5*. However, none of these sections address the "TAOS" signal. As a consequence, TAOS will be addressed in this section.

TAOS - Transmit All OneS

The user can configure the Transmit DS3/E3 LIU Block transmit an un-framed "All Ones" pattern via the TTIP and TRING output pins. This can be accomplished by setting Bit 2 (TAOS), within the "LIU Transmit Control" Register, to "1" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	1	0	0

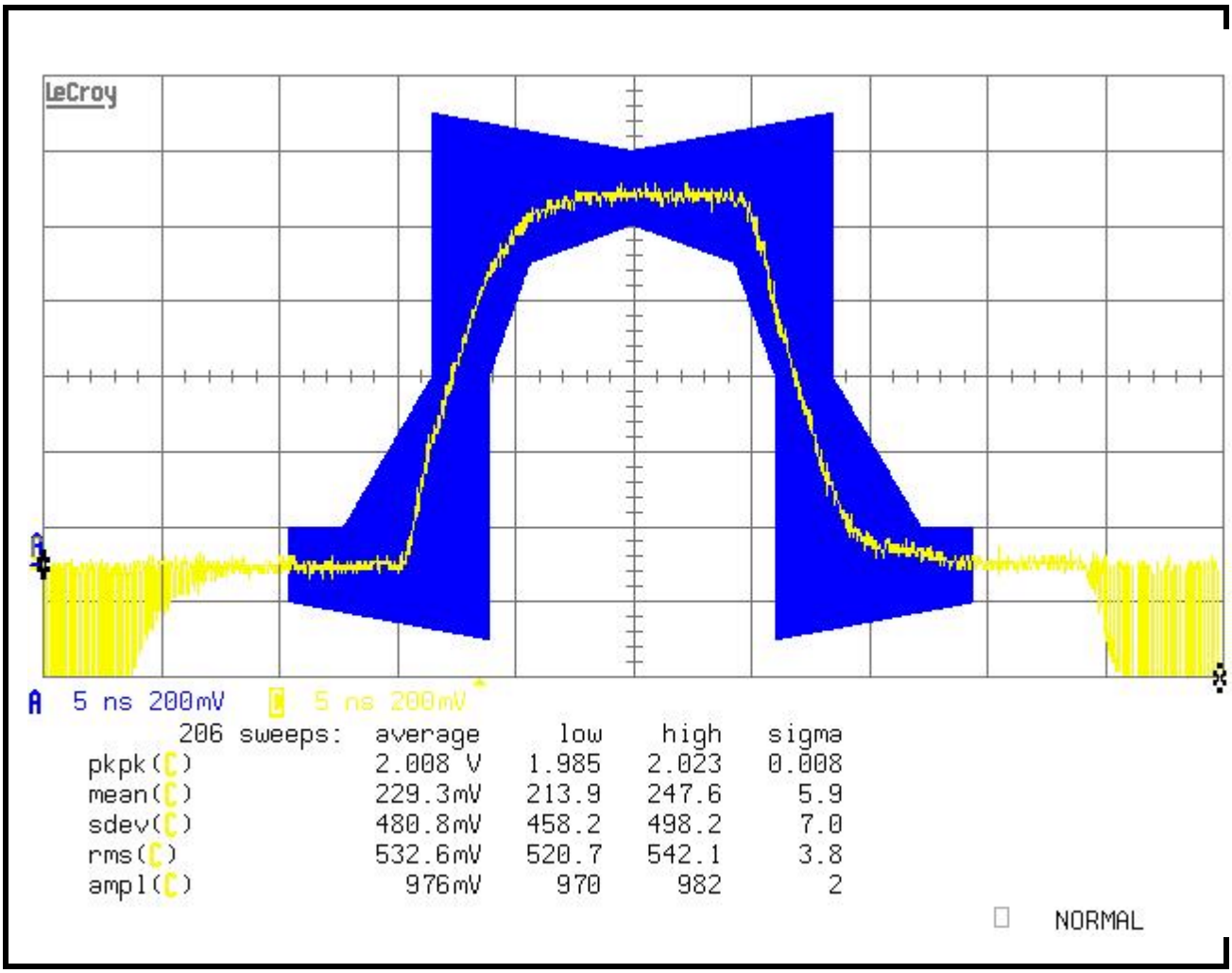
If the TAOS bit-field is set to "1", then the Transmit DS3/E3 LIU Block will overwrite the contents of the outbound DS3 or E3 data-stream that it receives from the Transmit DS3/E3 Framer block, with an unframed All Ones pattern.

Conversely, if the TAOS bit-field is set to "0", then the Transmit DS3/E3 LIU Block will NOT overwrite the contents of the outbound DS3 or E3 data-stream that it receives from the Transmit DS3/E3 Framer block, with an unframed All Ones pattern, as it transmit this DS3 or E3 data out onto the line.

5.2.5.4 The Transmit Pulse Shaping Block

For E3 Applications, the purpose of the "Transmit Pulse Shaping" block is to permit the Transmit DS3/E3 LIU Block to generate pulses (within the outbound E3 line signal) that comply with the ITU-T G.703 Pulse Template requirements for E3 applications. **Figure 159** presents the results of some actual pulse template measurements for E3 applications.

FIGURE 159. E3 PULSE TEMPLATE MEASUREMENT - TAKEN WITH 0 FEET OF CABLE LOSS



5.2.5.5 The Transmit Line Driver Block

The Transmit Line Driver block is the very last circuit that an "outbound" E3 signal will pass through prior to being output via the TTIP and TRING output pins. The Transmit Line Driver block permits the user to either enable or disable the transmission of a E3 line signal via the TTIP and TRING output pins.

If the Transmit Line Driver block is enabled, then make sure that all of the following is true.

1. That Bit 0 (TxON) within the "LIU Transmit APS/Redundancy Control" Register is set to "1" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

2. That the "TxON" input pin (Ball R15) is pulled "high".

Conversely, if the "Transmit Line Driver" block is disabled, then set Bit 0 (TxON) within the "LIU Transmit APS/ Redundancy Control" Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

NOTES:

1. *If this configuration setting is implemented, then the TTIP and TRING output pins (of the XRT79L71) will be tri-stated.*
2. *In order to control the "ON/OFF" state of the Transmit Line Driver (within the XRT79L71), via software command the user MUST make sure that the "TxON" input pin (Ball R15) is pulled to a logic "high".*
3. *If the user intends to implement the XRT79L71 into a "E3 Redundancy" design, then executing a write to this particular register, (either enabling or disabling the Transmit Output) will be required.*

5.2.5.6 The Transmit Drive Monitor Block

The Transmit Drive Monitor block permits the user to monitor the "Transmit Output" signal, for continuous bipolar signal activity, and can be used to (perhaps) detect a fault condition, in the "Transmit Output" line.

Use of the Transmit Drive Monitor block is optional. However, if this feature is used, there are two ways that this feature can be implemented.

- Externally, and
- Internally

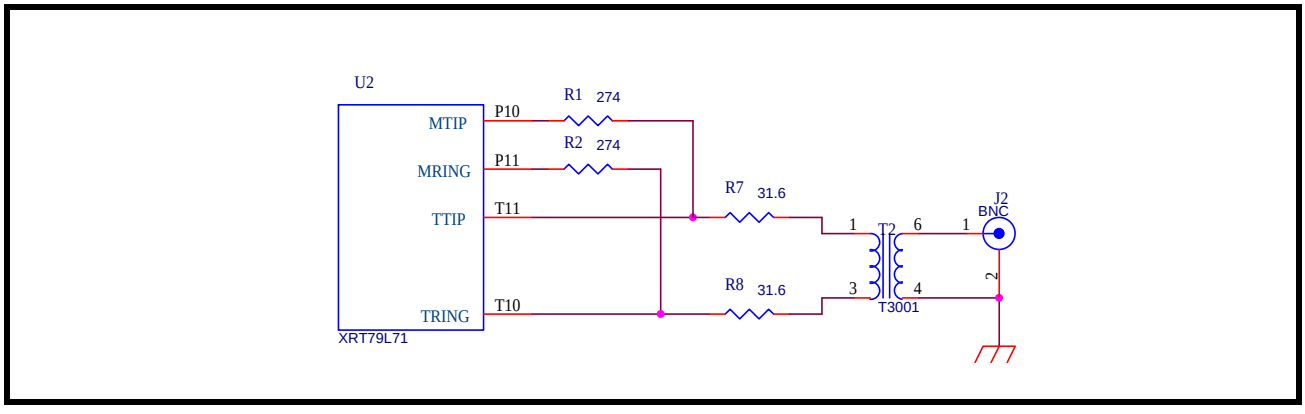
5.2.5.6.1 Implementing the Transmit Drive Monitor via "External" Means

To implement "Transmit Drive Monitoring" via "External Means, then this means that the Transmit Drive Monitor block will be monitoring (e.g., checking for bipolar activity) within the "Transmit Output" line signal via the "MTIP" and "MRING" input pins. If the "Transmit Drive Monitoring" via "External Means" is to be implemented, then the following steps must be executed.

STEP 1 - Design the Hardware such that (1) the "MTIP" ball is connected to the "TTIP" signal through a 274W resistor, connected in series, and (2) that the "MRING" ball is electrically connected to the "TRING" signal through a 274W resistor, connected in series.

These connections are also depicted in the Schematic design below in **Figure 160**,

FIGURE 160. A SCHEMATIC DESIGN, DEPICTING THE REQUIRED CONNECTIONS FOR "EXTERNAL" TRANSMIT DRIVE MONITORING



STEP 2 - Configure the Transmit Drive Monitor block into the "External" Mode.

This is accomplished by setting Bit 5 (Internal Transmit Drive Monitor) to "0" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	0	0	0	0	0	0

Once the user has implemented both of these steps, then the Transmit Drive Monitor block will proceed to check the TTIP/TRING lines for "bipolar activity" via the "MTIP" and "MRING" input pins. As long as the Transmit Drive Monitor block detects a regular stream of bipolar pulses via the MTIP/MRING pins, then it will negate the "Transmit DMO Condition" by continuing to set Bit 0 (Transmit DMO Condition), within the "LIU Alarm Status" Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

NOTE: *For most applications, the "Transmit DMO Condition" bit-field being set to "0" is a normal and a desirable condition.*

However, if the Transmit Drive Monitor block ever fails to detect any bipolar pulses (via the MTIP/MRING input pins) for approximately 128 bit periods, then the XRT79L71 will declare the "Transmit Drive Monitor" defect condition. Whenever the XRT79L71 declares the "Transmit Drive Monitor" defect condition, then it will do all of the following.

1. It will set Bit 0 (Transmit DMO Condition), within the "LIU Alarm Status" Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

2. It will also generate the "Change of DMO Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")

b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the "LIU Interrupt Status" Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

Clearing the Transmit DMO Condition

If the Transmit Drive Monitor block (while declaring the "Transmit Drive Monitor Defect Condition") detects at least one bipolar pulse via the MTIP/MRING input pins, then the XRT79L71 will clear the "Transmit Drive Monitor" defect condition. Whenever the XRT79L71 clears the "Transmit Drive Monitor" defect condition, then it will do all of the following.

1. It will set Bit 0 (Transmit DMO Condition), within the "LIU Alarm Status" Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared	Transmit DMO Condition
						Receive DS3/E3 LIU Block	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

2. It will also generate the "Change of DMO Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")

b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the "LIU Interrupt Status" Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

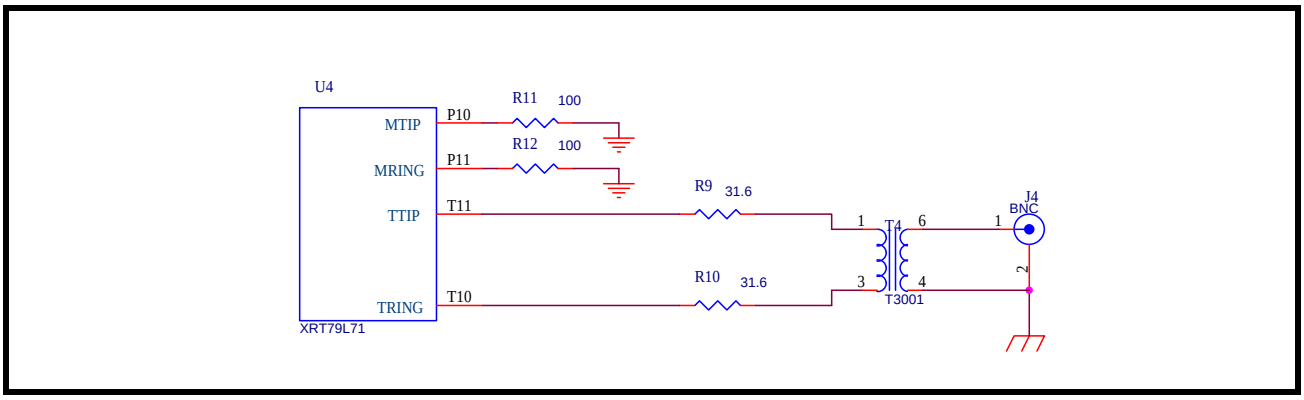
5.2.5.6.2 Implementing the Transmit Drive Monitor via "Internal" Means

If the Transmit Drive Monitoring via Internal Means is implemented, then the Transmit Drive Monitor block will be monitoring (e.g., checking for bipolar activity) within the "Transmit Output" line signal, via an internal connection to the TTIP/TRING output pins. If the "Transmit Drive Monitoring" is implemented via "Internal Means", then the following steps must be carried out.

STEP 1 - Design the Hardware such that (1) the "MTIP" and "MRING" balls are tied to GND via a 100W resistor connected in series.

These connections are also depicted in the Schematic design below in **Figure 161**.

FIGURE 161. A SCHEMATIC DESIGN, DEPICTING THE REQUIRED CONNECTIONS FOR "INTERNAL" TRANSMIT DRIVE MONITORING



STEP 2 - Configure the Transmit Drive Monitoring block into the "Internal" Mode.

The user can accomplish this step by setting Bit 5 (Internal Transmit Drive Monitor) to "1" as depicted below.

LIU Transmit Control Register (Address = 0x1304)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Internal Transmit Drive Monitor	Unused		TAOS	Unused	TxLEV
R/O	R/O	R/W	R/O	R/O	R/W	R/O	R/W
0	0	1	0	0	0	0	0

NOTE: *To implement "Transmit Output Drive Monitoring" via "Internal Means", then there is NO need to implement the external connections via the MTIP and TTIP and MRING and TRING pins. In this case, the Transmit Drive Monitor circuit will be monitoring (e.g., checking the TTIP/TRING signals for bipolar activity) by direction checking the pads of the TTIP/TRING outputs themselves.*

Once the user has implement this (above-mentioned) step, then the Transmit Drive Monitor block will proceed to "internally" check the TTIP/TRING lines for bipolar activity. As long as the Transmit Drive Monitor block detects a regular stream of bipolar pulses, then it will negate the "Transmit DMO Condition" by continuing to set Bit 0 (Transmit DMO Condition), within the "LIU Alarm Status" Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

NOTE: *For most applications, the "Transmit DMO Condition" bit-field being set to "0" is a normal and a desirable condition.*

However, if the Transmit Drive Monitor block ever fails to detect any bipolar pulses (via the TTIP/TRING output pads) for approximately 128 bit periods, then the XRT79L71 will declare the "Transmit Drive Monitor" defect condition. Whenever the XRT79L71 declares the "Transmit Drive Monitor" defect condition, then it will do all of the following.

1. It will set Bit 0 (Transmit DMO Condition), within the "LIU Alarm Status" Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

2. It will also generate the "Change of DMO Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by
   a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")
   b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the "LIU Interrupt Status" Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

Clearing the Transmit DMO Condition

If the Transmit Drive Monitor block (while declaring the "Transmit Drive Monitor Defect Condition") detects at least one bipolar pulse via the MTIP/MRING input pins, then the XRT79L71 will clear the "Transmit Drive Monitor" defect condition. Whenever the XRT79L71 clears the "Transmit Drive Monitor" defect condition, then it will do all of the following.

1. It will set Bit 0 (Transmit DMON Condition), within the "LIU Alarm Status" Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

2. It will also generate the "Change of DMO Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by

a. Asserting the Interrupt Request output pin (e.g., by toggling it "LOW")

b. Setting Bit 0 (Change of DMO Condition Interrupt Status) within the "LIU Interrupt Status" Register to "1" as depicted below.

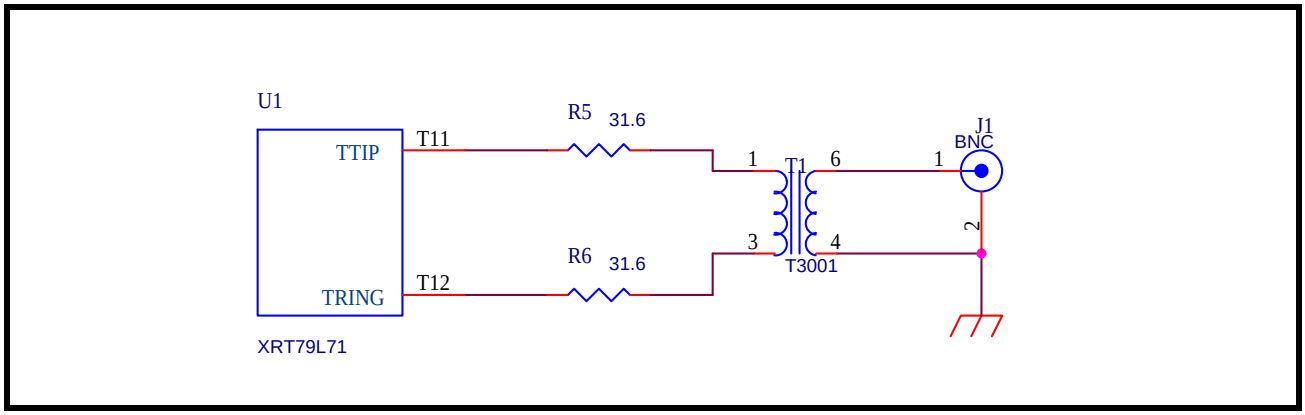
LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

5.2.5.7 Interfacing the Transmit DS3/E3 LIU Block to the Line

Figure 162 presents a schematic design that depicts our recommended approach to interfacing the Transmit DS3/E3 LIU Block to the line.

FIGURE 162. SCHEMATIC DESIGN, DEPICTING HOW TO INTERFACE THE TRANSMIT DS3/E3 LIU BLOCK (OF THE XRT79L71) TO THE LINE



NOTE: *For a more in-depth example on how to interface the XRT79L71 to the "outside" world, please see Appendix A for the "XRT79L71 Evaluation Board Schematic Design".*

5.3 THE RECEIVE DIRECTION

The next several sections will present an in-depth functional description of all of the blocks that are operating in the Receive Direction, within the XRT79L71, when configured to operate in the "Clear-Channel E3 Framer" Mode. Figure 163 presents a functional block diagram of the Receive Direction circuitry within the XRT79L71.

FIGURE 163. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3 CLEAR-CHANNEL FRAMER MODE

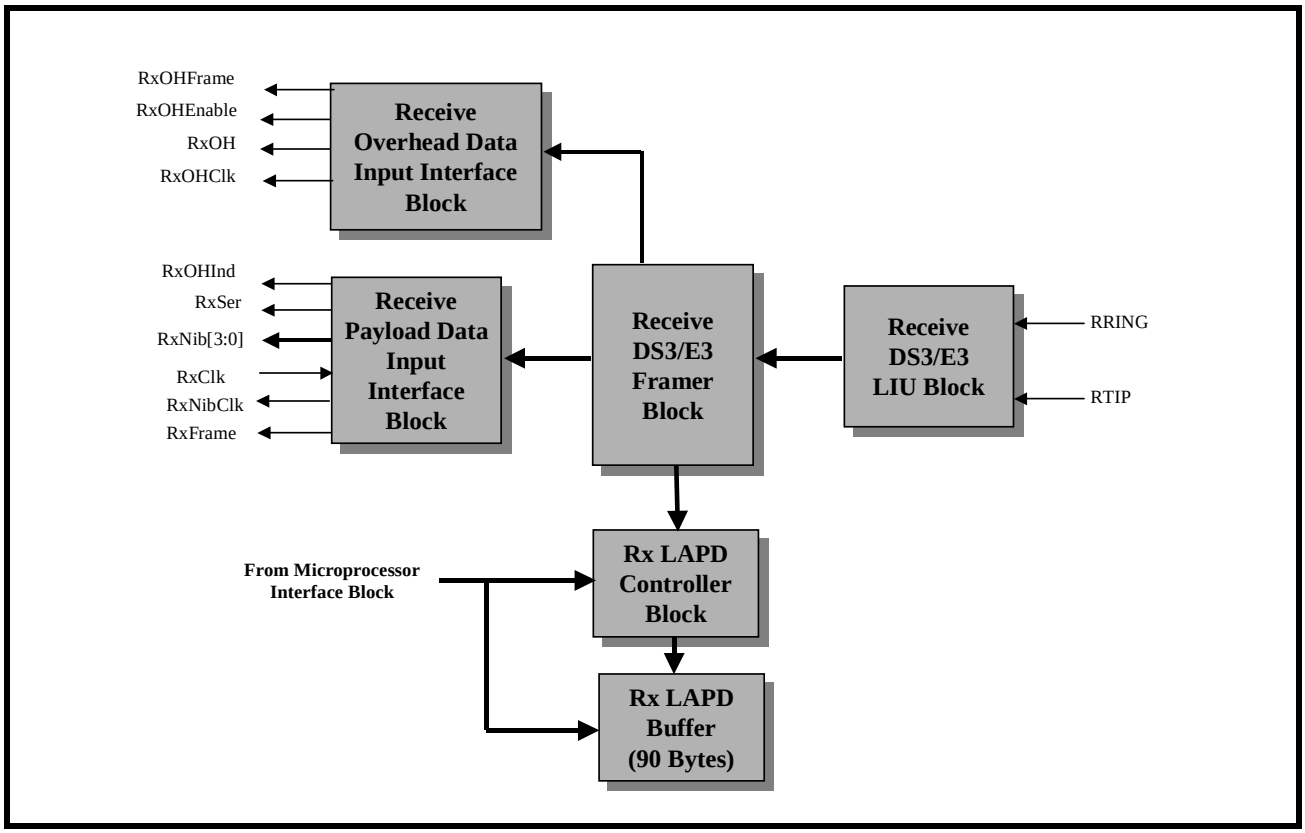


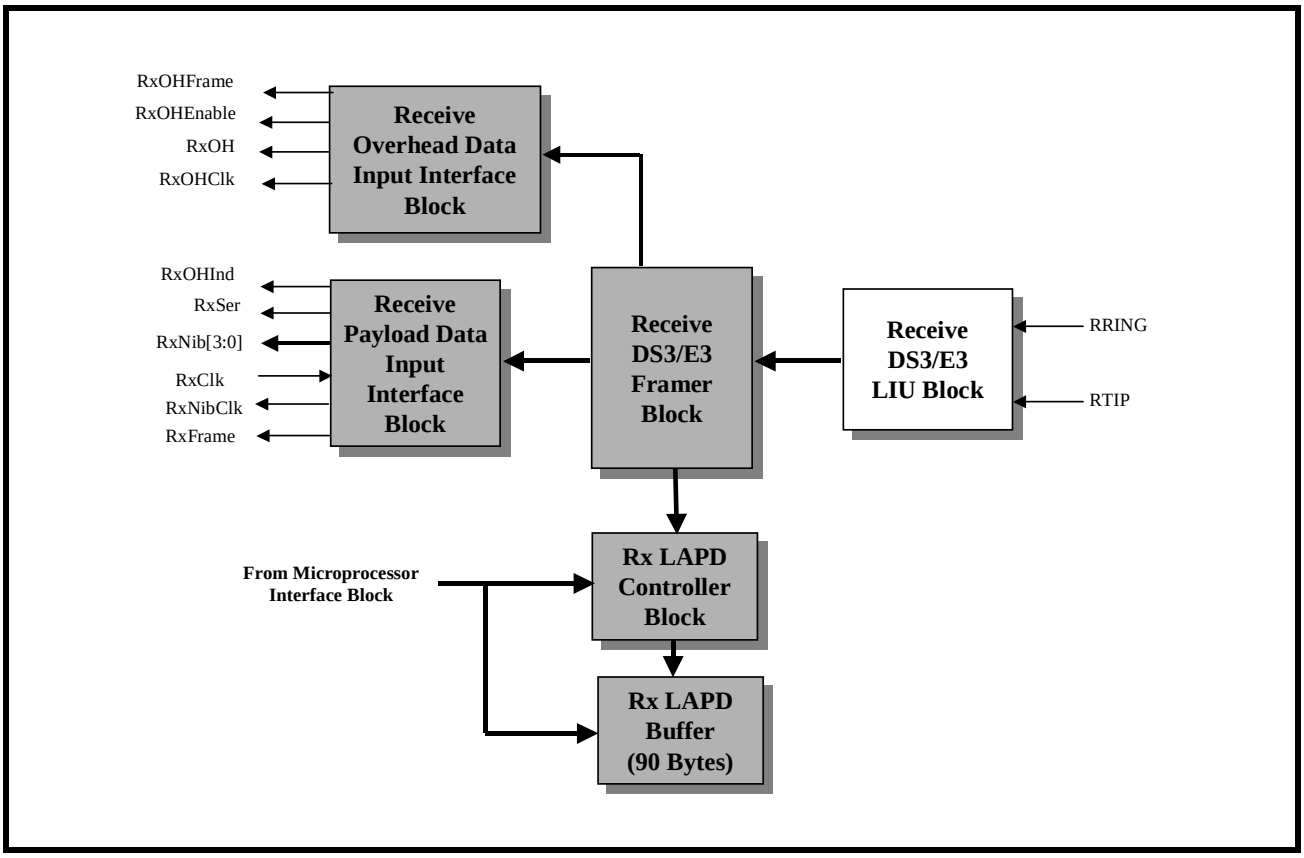
Figure 163 indicates that the Receive Direction circuitry consists of the following functional blocks.

- The Receive E3 LIU block
- The Receive E3 Framer Block
- The Receive LAPD Controller Block
- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block

5.3.1 THE RECEIVE E3 LIU BLOCK

The Receive E3 LIU Block is the first functiona block (within the Receive Direction) of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 164 presents an illustration of the "Receive Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the Receive E3 LIU Block highlighted.

FIGURE 164. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE DS3/E3 LIU" BLOCK HIGHLIGHTED).



The purpose of the Receive DS3/E3 LIU Block is to accept an DS3/E3 line signal and to perform all of the following operations on this data.

- To be able to receive a distorted E3 signal, that has been attenuated by at least 12dB of cable loss, along with an additional 6dB of flat (or resistive) loss in an error-free manner.
- To perform Clock and Data Recovery on this incoming DS3 line signal.
- To declare and clear the LOS (Loss of Signal) defect condition
- To declare and clear the LOL (Loss of Lock) defect condition
- To decode this incoming DS3 line signal from the HDB3 line code, back into a binary data-stream, prior to routing this signal to the Receive DS3/E3 Framer block.
- To be able to comply with the Jitter Tolerance Requirements per ITU-T G.823

This particular section will describe the functionality and configuration options of the Receive DS3/E3 LIU Block for E3 applications. The functionality and configuration options of the Receive DS3/E3 LIU Block for DS3 applications is discussed in **Figure 4.3.1**.

Figure 165 presents a more detailed illustration of the Receive DS3/E3 LIU Block within the XRT79L71.

FIGURE 165. ILLUSTRATION OF THE RECEIVE DS3/E3 LIU BLOCK WITHIN THE XRT79L71

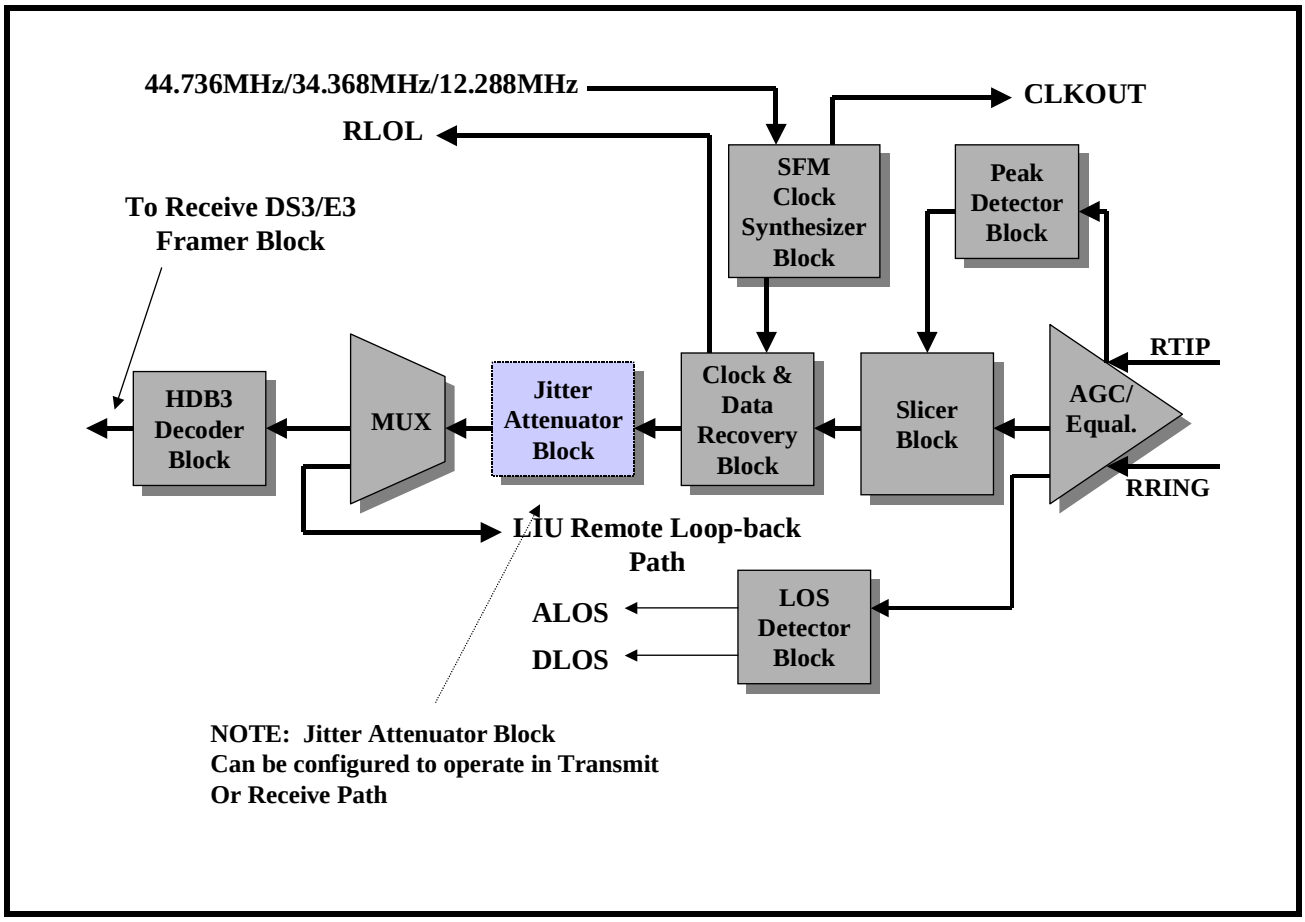


Figure 165 indicates that the Receive DS3/E3 LIU Block consists of the following functional blocks.

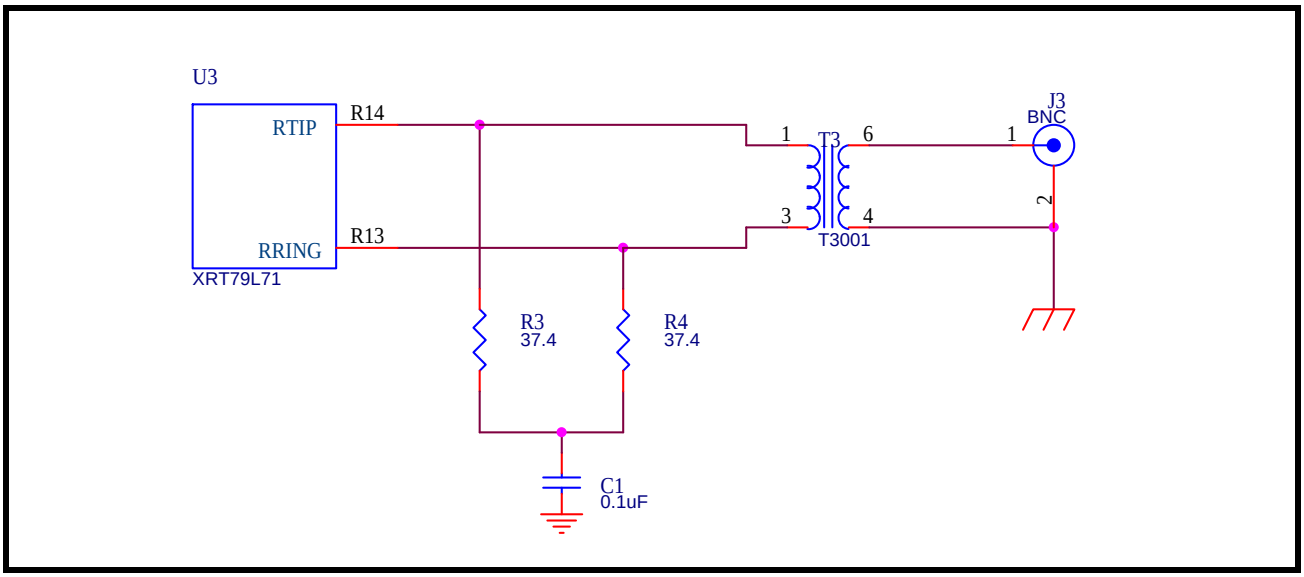
- The AGC (Automatic Gain Control) and Equalizer Block
- The Peak Detector Block
- The Slicer Block
- The SFM (Single Frequency Mode) Clock Synthesizer Block
- The Clock and Data Recovery Block
- The Jitter Attenuator Block
- The LOS Detector Block
- The HDB3 Decoder Block

5.3.1.1 Interfacing the Receive DS3/E3 LIU Block to the Line

Prior to "walking" through and discussing each of the various "sub-blocks" within the Receive DS3/E3 LIU Block, we will first present our recommendation of how to interface the Receive DS3/E3 LIU Block to the line.

Figure 166 presents a schematic design, depicting our recommended approach to interfacing the Receive DS3/E3 LIU Block to the line.

FIGURE 166. SCHEMATIC DESIGN, DEPICTING HOW TO INTERFACE THE RECEIVE DS3/E3 LIU BLOCK (OF THE XRT79L71) TO THE LINE



NOTE: *For a more in-depth example on how to interface the XRT79L71 to the "outside" world, please see Appendix A for the "XRT79L71 Evaluation Board Schematic Design".*

5.3.1.2 The Automatic Gain Control Block

The AGC (or Automatic Gain Control) Block is the very first sub-lock to receive either a DS3 or E3 signal from RTIP/RRING input pins. For E3 application, the purpose of the AGC block is to compensate for flat-loss that the incoming line signal may have experienced as it travels from the source (e.g., remote) terminal to the destination (e.g., local) terminal.

As the name of this sub-block implies, the AGC (Automatic Gain Control) block functions by automatically adjusting its ampification gain (of the incoming line signal) in order to insure that this amplitude of the incoming line signal (after being amplified by the AGC block) is within a certain desirable range for optimal "internal" signal processing within the remainder of the Receive DS3/E3 LIU Block circuitry.

Therefore, if the incoming DS3 or E3 line signal is of an amplitude that is less than this "desired" amplitude, then the AGC block will increase the amplitude of the incoming DS3/E3 signal by providing the appropriate amount of "Gain" to this signal.

If the amplitude of the "incoming" DS3 or E3 signal were to suddenly increase, then the AGC block will seek to make sure that the amplitude of this "internal" signal remains within the "desirable range" by lowering its gain to the appropriate level. Conversely, if the amplitude of the incoming DS3 or E3 signal were to suddenly decrease, then the AGC block will seek to make sure that the amplitude of this "internal" signal remains within the "desirable range" by increasing its gain to the appropriate level.

5.3.1.3 The Receive Equalizer Block

As a given pulse within a DS3 or E3 line rate signal travels from its "source" to the "destination" terminal via coaxial cable, it will experience a "frequency-dependent" loss, in which the "high-frequency" portions of this signal are more greatly attenuated than are the "lower-frequency" components of the signal. The result of this "frequency-dependent" loss is manifested by a change in shape of a given pulse within this DS3 or E3 line signal. More specifically, one will typically note that (as a given pulse travels along the communication medium) fast rising and falling edges give way to slower rising and falling edge. Pulses that originally of the "square-wave" shape (at the output of the "source" terminal) become less "square" and more "rounded" in shape as they travel along the coaxial cable. If the E3 data-stream travels a sufficiently long distance, then an entity that is responsible for properly receiving a given incoming E3 signal, will have trouble receiving this particular signal, due to phenomenon such as ISI (Inter-Symbol Interference).

The purpose of the Receive Equalizer block is to compensate for this "frequency-dependent" attenuation that occurs within a DS3 or E3 signal, traveling via coaxial cable. In essence, the Receive Equalizer block accomplishes this by applying higher gain to higher frequency signals, than it does for lower frequency signals.

The Register Set, within the XRT79L71, permits the user to either enable or disable the Receive Equalizer, within the Receive DS3/E3 LIU Block. In general, we strongly recommend that the user ALWAYS ENABLE the Receive Equalizer block for all applications.

The user can enable the Receive Equalizer block by setting Bit 0 (Receive Equalizer Enable), within the "LIU Receive Control" Register, to "1" as depicted below.

LIU Receive Control Register (Address = 0x1305)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Disable DLOS Detector	Disable ALOS Detector	Unused	LOSMUT Enable	Receive Monitor Mode Enable	Receive Equalizer Enable
R/O	R/O	R/W	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	1

NOTE: *By default, the Receive Equalizer block will (upon power up) be disabled. Therefore, the user MUST include "enabling the Receive Equalizer block" as a part of the "start-up/configuration" procedure.*

5.3.1.4 The Clock and Data Recovery Block

The purpose of the Clock and Data Recovery block (within the Receive DS3/E3 LIU Block) is two-fold.

- To acquire and maintain "phase-lock" with the incoming DS3 or E3 line signal.
- To insure that "downstream" circuitry, (such as the Receive DS3/E3 Framer block) is always provided with a line-rate clock signal (to use as it timing reference).

Upon power-up, the Clock and Data Recovery block will attempt to acquire "phase-lock" with the incoming DS3 or E3 line signal. Once the Clock and Data Recovery block has acquired "lock" with the incoming DS3 or E3 signal, then it will indicate this fact by doing all of the following.

- It will set Bit 2 (Receive LOL Defect Declared), within the "LIU Alarm Status"Register, to "0" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the "Change of LOL (Loss of Lock) Defect Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by (a) asserting the Interrupt Request output pin (by toggling it "LOW"), and (b) by setting Bit 2 (Change of LOL Condition Interrupt), within the "LIU Interrupt Status" Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

For the duration that the Clock and Data Recovery block is maintaining "lock" with the incoming E3 line signal, then all of the following will be true.

- The Clock and Data Recovery block will be synthesizing a 34.368MHz "recovered" clock signal that is derived from the incoming E3 line signal.
- The Clock and Data Recovery block will route this 34.368MHz clock signal to all down-stream circuitry (e.g., the Receive DS3/E3 Framer block, etc.). Each of these down-stream blocks will use this 34.368MHz clock signal as their timing source.
- The Clock and Data Recovery block will be continuously comparing the frequency of its Recovered Clock signal with a "Reference" Clock signal that it receives from the "SFM Synthesizer" Block. As long as the differences between these two frequencies is less than 0.5% (or 5000ppm), then the Clock and Data Recovery block will continue to operate in this "normal" mode.

NOTE: *Information on the "SFM Synthesizer" block can be found in Section 4.3.1.5.*

The LOL (Loss of Lock) Defect Declaration Criteria

As mentioned above, the Clock and Data Recovery block will be continuously monitoring the frequency of its recovered clock signal with a "Reference" Clock signal that it receives from the "SFM Synthesizer" block. As long as the differences between these two frequencies is less than 0.5% (or 5000ppm), then the Clock and Data Recovery block will continue to operate in this "normal" mode.

However, if the difference between these two frequencies exceeds 0.5% (or 5000ppm), then the Clock and Data Recovery block will declare the "LOL (Loss of Lock) Defect Condition". Whenever the Clock and Data Recovery block declares the "Loss of Lock Defect Condition", then it will cease its attempt to acquire and maintain "phase-lock" with the incoming E3 line signal. Instead, the Clock and Data Recovery will now lock onto the reference clock signal from the "SFM Synthesizer" block. The Clock and Data Recovery will make this transition, in order to guarantee that all "down-stream" circuitry (e.g., the Receive DS3/E3 Framer block) will be provided with a proper line-rate clock signal.

The XRT79L71 will inform the user that it is declaring the "Loss of Lock Condition" by doing all of the following.

- It will set Bit 2 (Receive LOL Defect Declared), within the "LIU Alarm Status" Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	1	0	0

- It will generate the "Change of LOL (Loss of Lock) Defect Condition" Interrupt. The XRT79L71 will indicate that it is generating this interrupt by (a) asserting the Interrupt Request output pin (by toggling it "LOW"), and (b) by setting Bit 2 (Change of LOL Condition Interrupt), within the "LIU Interrupt Status" Register, to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

For the duration that the Clock and Data Recovery block is declaring the "LOL Defect Condition" then all of the following will be true.

- The Clock and Data Recovery block will be synthesizing a 34.368MHz clock signal that is derived from the "Reference" Clock signal (which originates from the "SFM Synthesizer" block).
- The Clock and Data Recovery block will route this 34.368MHz clock signal to all down-stream circuitry (e.g., the Receive DS3/E3 Framer block, etc.). Each of these down-stream blocks will use this 34.368MHz clock signal as their timing source.

The Clock and Data Recovery Block and its "LOL Defect Declaration" Behavior during "Signal Present" and "No Signal Present" Conditions

Once the Clock and Data Recovery Block has declared the "LOL Defect Condition", then any subsequent behaviour of the Clock and Data Recovery Block, depends upon whether the AGC Block (within the Receive DS3/E3 LIU Block) determines that some sort of signal energy is present (at the RTIP/RRING input pins) or not.

If "No Signal is present" at the RTIP/RRING Input Pins

If the incoming E3 line signal were to be removed, such that there is absolutely no signal energy being applied to the RTIP/RRING input pins, then all of the following will happen.

- The Receive DS3/E3 LIU Block will declare the LOS Defect Condition (please see **SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.**for more information on how the Receive DS3/E3 LIU Block declares the LOS Defect Condition).

- The Clock and Data Recovery block (now having no incoming E3 line signal to lock onto) will begin to drift towards its VCO center frequency. As the Clock and Data Recovery block begins to do this, at some point, the "Recovered Clock" frequency (of the Clock and Data Recovery) will drift to beyond the point where it differs from the Reference Clock signal (being supplied by the "SFM Synthesizer" block) by 0.5% (or 5000ppm). Once the "Recovered Clock" signal differs from the "Reference Clock" signal by 5000ppm, or more, then it will declare the LOL Defect Condition.
- Once the Clock and Data Recovery block has declared the LOL Defect condition, it will now "lock" onto the "Reference Clock" signal (from the SFM Synthesizer block). In this case, the "Recovered" clock signal (from the Clock and Data Recovery block) will be derived from the "Reference Clock" signal (from the "SFM Synthesizer" block).

For the duration that the AGC Block is in the "High-Gain" Mode (which indicates that no signal energy is presented at the RTIP/RRING input pins), then the Clock and Data Recovery block will remain in the "above-mentioned" state. In other words, it will continue to declare the LOL Defect Condition. It will also continue to synthesize and route a 34.368MHz clock signal (to the down-stream circuitry, such as the Receive DS3/E3 Framer block) based upon the "Reference Clock" signal from the "SFM Synthesizer" block.

If "A Signal" is present at the RTIP/RRING Input Pins

If there is an impairment, within the incoming E3 line signal, such that signal energy was still present at the RTIP/RRING input pins, however "Recovered Clock" frequency (from the Clock and Data Recovery) differs from the "Reference Clock" Frequency by more than 5000ppm; then all of the following will happen.

- The Receive DS3/E3 LIU Block may (or may not) declare the LOS Defect condition (please see **SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.** for more information on how the Receive DS3/E3 LIU Block declares the LOS Defect Condition). This condition depends upon the amplitude of the signal being applied to the RTIP/RRING input pins.
- The Clock and Data Recovery block (now locking onto a signal that is more than 5000ppm off in frequency, from the "Reference Clock" signal (being supplied by the "SFM Synthesizer" block). As a consequence, the Clock and Data Recovery will declare the LOL Defect Condition.
- Once the Clock and Data Recovery block has declared the LOL Defect condition, it will now "lock" onto the "Reference Clock" signal (from the SFM Synthesizer block). In this case, the "Recovered" clock signal (from the Clock and Data Recovery block) will be derived from the "Reference Clock" signal (from the "SFM Synthesizer" block).

In this case, the AGC Block will not be in the "High-Gain" Mode (which indicates that there is some signal energy present at the RTIP/RRING input pins). Whenever this is the case, then the Clock and Data Recovery block will constantly be cycling through the following states.

STATE 1: The Clock and Data Recovery is locked onto a "line" signal that is more than 5000ppm off in frequency from the "Reference Clock" signal (that is supplied by the SFM Synthesizer block). As a consequence, the Clock and Data Recovery declares the LOL Defect Condition.

STATE 2: Now that the Clock and Data Recovery is declaring the LOL Defect condition, it (for the time-being) will cease to attempt to lock onto the incoming line signal (via the RTIP/RRING input pins), and will (instead) lock onto the "Reference Clock" signal (that originates from the SFM Synthesizer block). In this case, the difference between "Recovered" Clock and the "Reference Clock" is no longer be greater than 5000ppm. It will now be 0ppm (because the "Recovered" Clock is now being synthesized from the "Reference Clock"). As a consequence, the Clock and Data Recovery will now clear the LOL Defect condition.

STATE 3: Now that the Clock and Data Recovery has cleared the LOL Defect condition, it will now attempt to acquire lock with the incoming E3 line signal. As the Clock and Data Recovery does this, it will continue to make the comparison between the frequency of its "Recovered" Clock and the frequency of the "Reference Clock".

5.3.1.5 The SFM (Single-Frequency Mode) Synthesizer Block

The purpose of the SFM Synthesizer block is to provide the Clock and Data Recovery with a proper "Reference" clock signal that is of the frequency 44.736MHz (for DS3 applications) or 34.368MHz (for E3 applications). The Clock and Data Recovery block will use this "Reference Clock" signal in order to determine whether or not it should declare the "LOL" (Loss of Lock) Defect Condition.

The SFM Synthesizer block can be configured to operate in one of the following two modes.

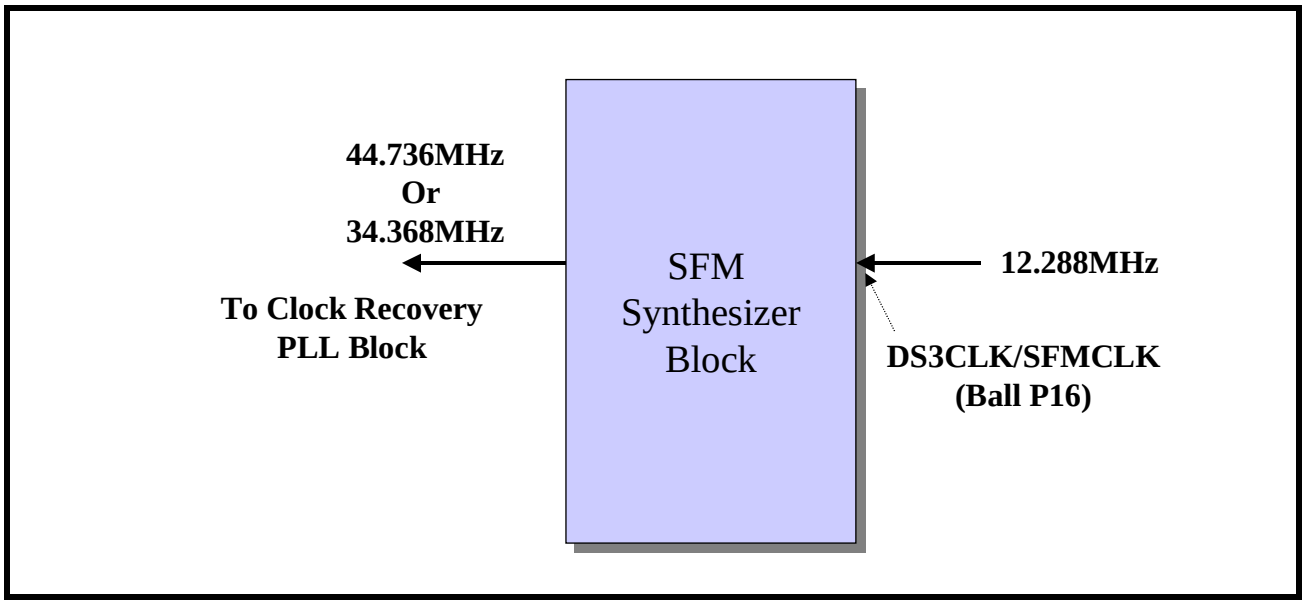
- The SFM (Single Frequency) Mode, or
- The Multiplexer Mode

5.3.1.5.1 Operating the "SFM Synthesizer" Block in the "SFM" (Single-Frequency) Mode

If the "SFM Synthesizer" block is configured to operate in the "SFM" Mode, then the user is expected to provide a 12.288MHz clock signal to the "DS3CLK/SFMCLK" input pin (Ball P16). The "SFM Synthesizer" block will then accept this 12.288MHz clock signal, and it will synthesize either a 44.736MHz clock signal (if the XRT79L71 has been configured to operate in the DS3 Mode), or it will synthesize a 34.368MHz clock signal (if the XRT79L71 has been configured to operate in the E3 Mode). The Clock and Data Recovery will, in turn, use this "synthesized" clock signal as its frequency reference, in order to determine whether or not it should declare the LOL defect condition.

This mode is referred to as the "Single-Frequency" Mode, because the user only needs to supply a single clock frequency (12.288MHz, in this case) to the DS3CLK/SFMCLK input pin, in order to permit the "SFM Synthesizer" block to fully support all operational requirements of the XRT79L71. Figure 167 presents a simple illustration that depicts how the "SFM Synthesizer" Block functions whenever it has been configured to operate in the "SFM" Mode.

FIGURE 167. A SIMPLE ILLUSTRATION THAT DEPICTS HOW THE "SFM SYNTHESIZER" BLOCK FUNCTIONS WHENEVER IT HAS BEEN CONFIGURED TO OPERATE IN THE "SFM" MODE



Configuring the SFM Synthesizer Block to operate in the SFM (Single-Frequency) Mode

The user can configure the "SFM Synthesizer" Block to operate in the SFM (Single-Frequency) Mode by executing the following steps.

STEP 1 - Apply a 12.288MHz clock signal to the DS3CLK/SFMCLK Input pin (Ball P16)

STEP 2 - Tie the E3CLK input pin (Ball M16) to GND.

STEP 3 - Set Bit 5 (SFM Enable) within the "LIU Channel Control" Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	1	0	0	0	0	0

Using the 44.736MHz/34.368MHz Synthesized Clock signal (from the SFM Synthesizer Block) as an External Clock Signal

As mentioned earlier, if the SFM Synthesizer block has been configured to operate in the "SFM (Single-Frequency) Mode", then it will (1) accept a 12.288MHz clock signal (via the DS3CLK/SFMCLK input pin), and it will use this 12.288MHz clock signal to (2) synthesize either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) and route this signal to the Clock and Data Recovery block.

In some applications there may be a desire to externally use the 44.736MHz or 34.368MHz clock signal that is synthesized by the "SFM Synthesizer" block elsewhere in the user's board design. If this is the case, then the XRT79L71 can support this requirement.

The XRT79L71 includes a pin (or ball) that is called "CLKOUT" (Ball K16). This output pin can be configured to output the 44.736MHz/34.368MHz clock signal that is synthesized by the "SFM Synthesizer" block.

Invoke this feature by setting Bit 6 (SFM Clock Out Enable), within the "LIU Channel Control" Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	1	1	0	0	0	0	0

Once this bit-field is set to "1", then the output driver (associated with the CLKOUT pin) will become active, and either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) will be output via this pin.

NOTE: *If this "CLKOUT" feature is invoked, the user must be aware that this clock signal is ultimately derived from the 12.288MHz clock signal (being applied to the DS3CLK/SFMCLK input poin) and is NOT the "Recovered" clock signal from the "Clock and Data Recovery" block.*

5.3.1.5.2 Operating the "SFM Sythesizer" Block in the "Multiplexer" Mode

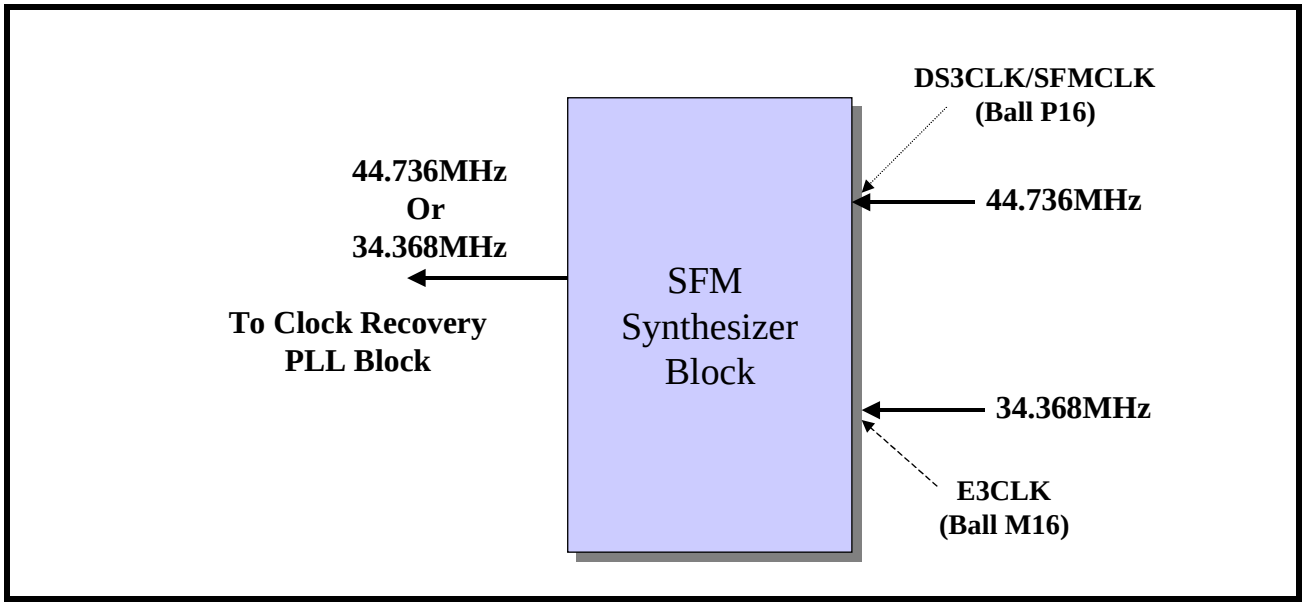
If the SFM Synthesizer block is configured to operate in the SFM Mode, then it is expected that any one of the following signals are provided to the following input pins.

- A 44.736MHz clock signal to the DS3CLK/SFMCLK input pin (Ball P16), or/and
- A 34.368MHz clock signal to the "E3CLK" input pin (Ball M16)

The "SFM Synthesizer" block will then multiplex and route the appropriate clock signal (e.g., the 44.736MHz or the 34.368MHz) to the Clock and Data Recovery, depending upon whether the XRT79L71 has been configured to operate in the DS3 or the E3 Mode.

This mode is referred to as the "Multiplexer" Mode, because the "SFM Synthesizer" block can accept up to both a 44.736MHz and a 34.368MHz via their respective inputs, and it will only output one of these clock signals (to the Clock and Data Recovery block) depending upon which mode (e.g., either DS3 or E3) that the XRT79L71 has been configured to operate in. Figure 168 presents a simple illustration that depicts how the "SFM Synthesizer" Block functions whenever it has been configured to operate in the "Multiplexer" Mode.

FIGURE 168. A SIMPLE ILLUSTRATION THAT DEPICTS HOW THE "SFM SYTHESIZER" BLOCK FUNCTIONS WHENEVER IT HAS BEEN CONFIGURED TO OPERATE IN THE "MULTIPLEXER" MODE



Configuring the SFM Synthesizer Block to operate in the "Multiplexer" Mode

Toconfigure the "SFM Synthesizer" Block to operate in the "Multiplexer" Mode execute the following steps.

STEP 1 - Apply a 44.736MHz clock signal (if available) to the "DS3CLK/SFMCLK" input pin (Ball P16), and apply a 34.368MHz clock signal (if available) to the "E3CLK" input pin (Ball M16).

STEP 2 - Set Bit 5 (SFM Enable) within the "LIU Channel Control" Register to "0" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	1	0	0	0	0	0

Using the 44.736MHz/34.368MHz Synthesized Clock signal (from the SFM Synthesizer Block) as an External Clock Signal

As mentioned earlier, if the SFM Synthesizer block has been configured to operate in the "Multiplexer Mode", then it will (1) accept a 44.736MHz clock signal (via the DS3CLK/SFMCLK input pin), and a 34.368MHz clock

signal (via the E3CLK input pin) and it (2) will output ether one of these clock signals to the Clock and Data Recovery block (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode).

In some applications there may be a desire to externally use the 44.736MHz or 34.368MHz clock signal that is synthesized by the "SFM Synthesizer" block elsewhere in the user's board design. If this is the case, then the XRT79L71 can support this requirement.

The XRT79L71 includes a pin (or ball) that is called "CLKOUT" (Ball K16). This output pin can be configured to output the 44.736MHz/34.368MHz clock signal that is synthesized by the "SFM Synthesizer" block.

Invoke this feature by setting Bit 6 (SFM Clock Out Enable), within the "LIU Channel Control" Register, to "1" as depicted below.

LIU Channel Control Register (Address = 0x1306)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	1	1	0	0	0	0	0

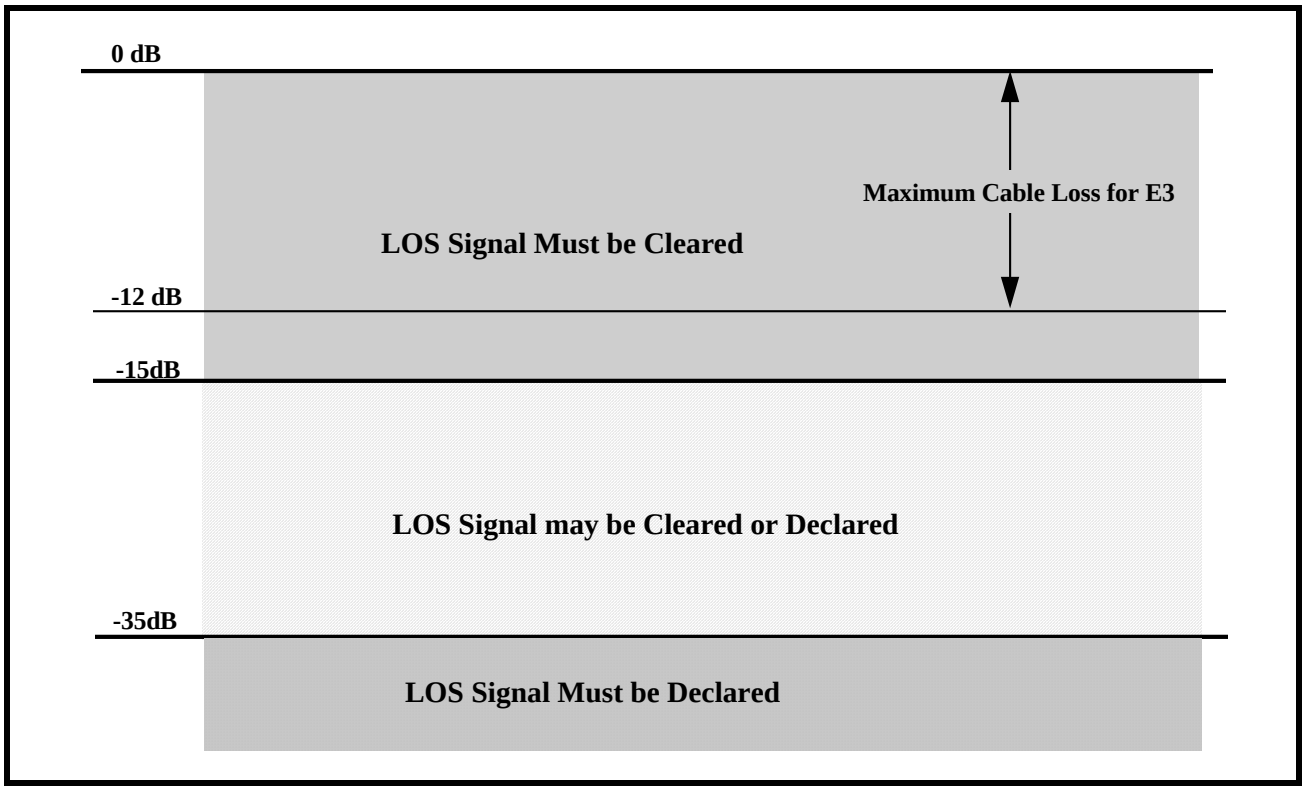
Once the user sets this bit-field to "1", then the output driver (associated with the CLKOUT pin) will become active, and either a 44.736MHz or 34.368MHz clock signal (depending upon whether the XRT79L71 has been configured to operate in the DS3 or E3 Mode) will be output via this pin.

5.3.1.6 The LOS Declaration and Clearance Criteria for E3 Applications

The Receive DS3/E3 LIU Block consists of a special type of LOS (Loss of Signal) Detectors that was specifically designed in order to insure that it declares and clears the LOS defect per the requirements in ITU-T G.775.

More specifically when the XRT79L71 is operating in the "E3 Mode", then the Receive DS3/E3 LIU Block will declare the LOS defect Condition if the signal amplitude drops to -35dB or below (where 0dB pertains to an E3 pulse that is of amplitude 1Vpk). Further, the Receive DS3/E3 LIU Block device will clear the LOS defect Condition if the signal amplitude rises back up to -15dB or above. **Figure 169** presents an illustration that depicts the signal levels at which the Receive DS3/E3 LIU Block will declare and clear the LOS defect.

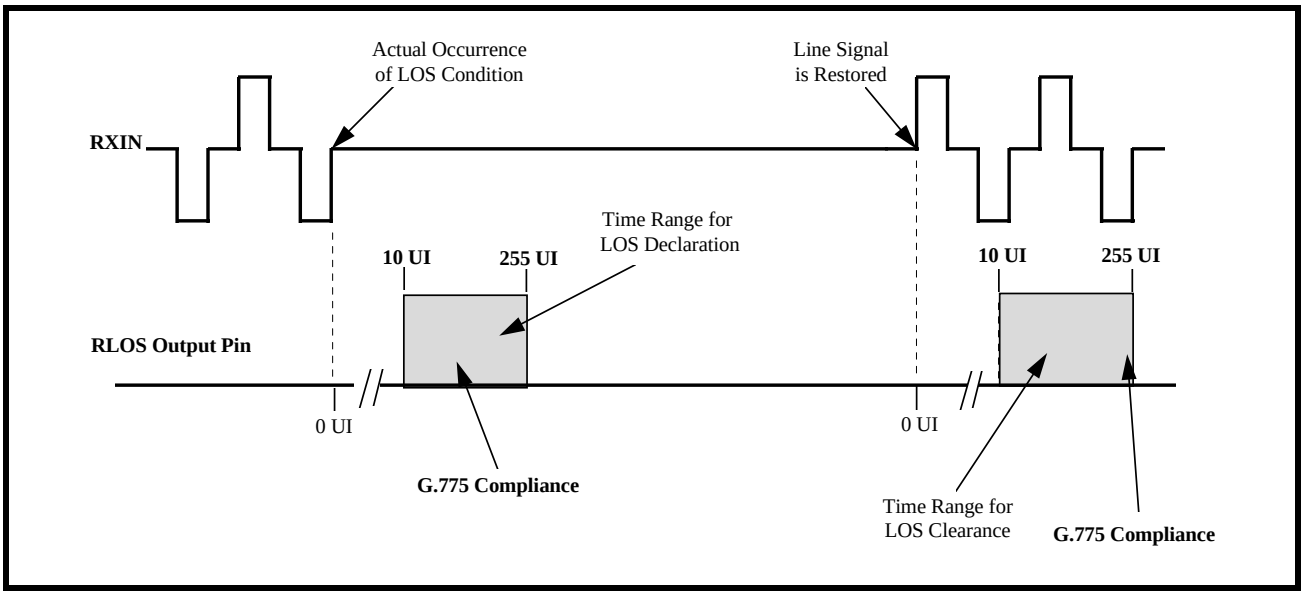
FIGURE 169. ILLUSTRATION OF THE SIGNAL LEVELS THAT THE RECEIVE DS3/E3 LIU BLOCK WILL DECLARE AND CLEAR THE LOS DEFECT CONDITION (FOR E3 APPLICATIONS).



Timing Requirements associated with Declaring and Clearing the LOS Defect Condition

The Receive DS3/E3 LIU Block (within the XRT79L71) was also designed to meet the ITU-T G.775 specification timing requirements for declaring and clearing the LOS defect condition. In particular, the Receive DS3/E3 LIU Block will declare the LOS defect condition, between 10 and 255 UI (or E3 bit-periods) after the actual time the LOS condition occurred. Further, the Receive DS3/E3 LIU Bloock will clear the LOS defect condition within 10 to 255 UI after restoration of the incoming line signal. **Figure 170**, illustrates the LOS Defect Condition Declaration and Clearance behavior, in response to first, the "Loss of Signal" event and then afterwards, the restoration of the signal.

FIGURE 170. THE BEHAVIOR THE LOS OUTPUT INDICATOR, IN RESPONSE TO THE LOSS OF SIGNAL, AND THE RESTORATION OF SIGNAL.



Declaring the LOS Defect Condition

Anytime the Receive E3 LIU Block declares the LOS Defect condition (per the criteria described above), then it will indicate (to the outside world) of this fact by doing all of the following.

- It will set Bit 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block), within the "LIU Alarm Status" Register, to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	1	0

- It will generate the "Change of LOS Condition Interrupt".

NOTE: *The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the "Interrupt Request" output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the "LIU Interrupt Status" Register to "1" as depicted below.*

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

- Clearing the LOS Defect Condition

The Receive DS3/E3 LIU Block will indicate (to the outside world) that it is clearing the LOS defect condition, by doing all of the following.

- It will set Bit 1 (Receive LOS Defect Declared - Receive DS3/E3 LIU Block), within the "LIU Alarm Status" Register to "1" as depicted below.

LIU Alarm Status Register (Address = 0x1303)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused		Digital LOS Defect Declared	Analog LOS Defect Declared	FL (FIFO Limit) Alarm Declared	Receive LOL Defect Declared	Receive LOS Defect Declared - Receive DS3/E3 LIU Block	Transmit DMO Condition
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- It will generate the "Change of LOS Condition Interrupt".

The XRT79L71 will indicate that it is generating this interrupt by (1) asserting the "Interrupt Request" output pin, and (2) by setting Bit 1 (Change of LOS Condition Interrupt), within the "LIU Interrupt Status" Register to "1" as depicted below.

LIU Interrupt Status Register (Address = 0x1302)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FL Condition Interrupt Status	Change of LOL Condition Interrupt Status	Change of LOS Condition Interrupt Status	Change of DMO Condition Interrupt Status
R/O	R/O	R/O	R/O	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

5.3.1.7 Jitter Attenuator Block

Please see Section 5.2.5.2 for a description of the Jitter Attenuator Block

5.3.1.8 The HDB3 Decoder Block

The purpose of the HDB3 Decoder block is to decode the "inbound" E3 traffic from the "HDB3 Line Code", into a digital data-stream. In the case of the XRT79L71, the HDB3 Decoder block will always be enabled and the user has no ability to disable the HDB3 Decoder block.

5.3.1.9 Performance Characteristics of the Receive E3 LIU Block

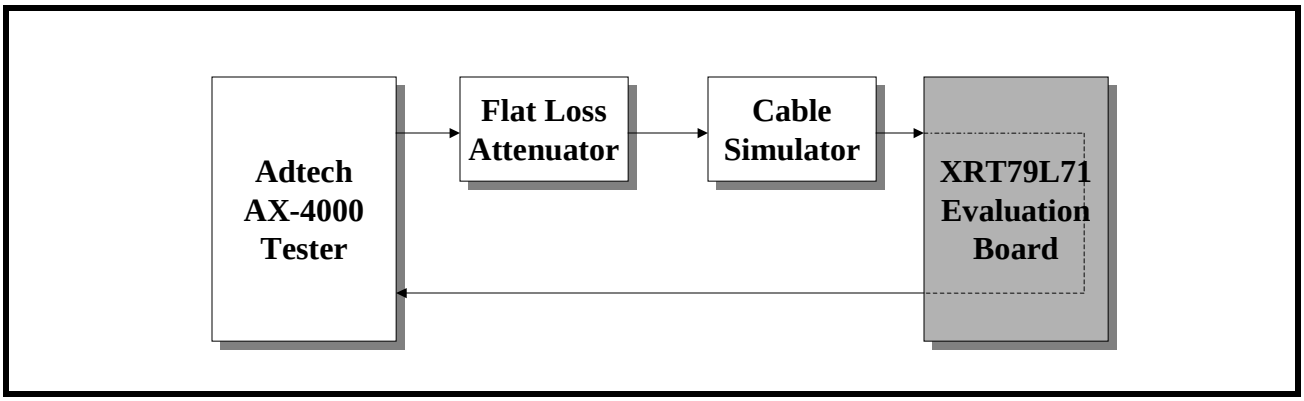
These next few sections will present the performance characteristics of the Receive DS3/E3 LIU Block, within the XRT79L71. In particular these sections will address the following parameters for E3 Applications.

- Receive Sensitivity
- Interference Margin
- Jitter Tolerance

5.3.1.9.1 Receive Sensitivity Capability of the Receive E3 LIU Block

For DS3 Applications, the Receive E3 LIU Block MUST be capable of receiving a "E3" signal that has been attenuated by anywhere from 0 to 12dB of cable loss, and at least 6dB of flat loss, in a un-erred manner. Table 45 summarizes the Receive Sensitivity of the Receive DS3/E3 LIU Block, within the XRT79L71.

FIGURE 171. ILLUSTRATION OF TEST SET-UP TO PERFORM THE "RECEIVE SENSITIVITY LOW-LEVEL" TEST



Test Approach

The Adtech AX-4000 tester was configured to generate ATM cells and to map these ATM cells into an E3 data stream, which was then to be output as a bipolar line signal.

The E3 line signal was then routed to the ME-1005 75W Coaxial Cable Simulator (from Mountain Engineering).

The Cable Simulator was configured to insert the "user-selected" amount of "shaped" (or cable) loss into this E3 line signal.

After this E3 line signal has been subjected to an appropriate amount of flat-loss and cable loss, it was then routed to the Receive Input of the XRT79L71 Evaluation Board. The XRT79L71 Evaluation Board was configured to operate in the "UTOPIA Loop-back" Mode. Therefore, the XRT79L71 Evaluation Board would handle the attenuated E3 line signals in the following manner.

- The Receive DS3/E3 LIU Block (within the XRT79L71) would receive this E3 line signal and perform "Clock" and "Data" Recovery on this line signal.
- This "Recovered" Clock and Data would then be routed to the Receive DS3/E3 Framer block, the Receive ATM Cell Processor and the Receive UTOPIA Interface block. The UTOPIA FPGA would then read out the contents of these ATM cells and it would then write these ATM cells back into the Transmit UTOPIA Interface block.

- The Transmit Section of the XRT79L71 would then accept these ATM cells and it would (once again) map these ATM cells back into an E3 data-stream. The XRT79L71 would output a E3 line signal, which would be routed back into the Receive Input of the Adtech AX-4000 Tester.

The Adtech AX-4000 Tester can determine whether or not bit-errors have occurred in the loop-back path (through the Flat-Loss Attenuators, the Cable Simulator, and the XRT79L71 Evaluation Board) by checking for E3 alarms, occurrences of HEC Byte errors (within ATM cells) and/or pattern sync errors within the ATM cell payload, etc.

Test Results

The Receive Sensitivity Test Results are summarized below in Table 45.

TABLE 45: RECEIVE SENSITIVITY TEST RESULTS (E3 APPLICATIONS)

TEST NUMBER	POWER SUPPLY VOLTAGE	RECEIVE SENSITIVITY	COMMENTS
1	3.15V	16.7dB of Cable Loss	
2	3.30V	16.7dB of Cable Loss	
3	3.45V	16.7dB of Cable Loss	

5.3.1.9.2 Interference Margin Capability of the Receive E3 LIU Block

The purpose of an Interference Margin Test (or specification) is to verify that the Receive DS3/E3 LIU Block (within the XRT79L71) can tolerance a certain amount of inband noise and still properly receive a DS3 or E3 signal.

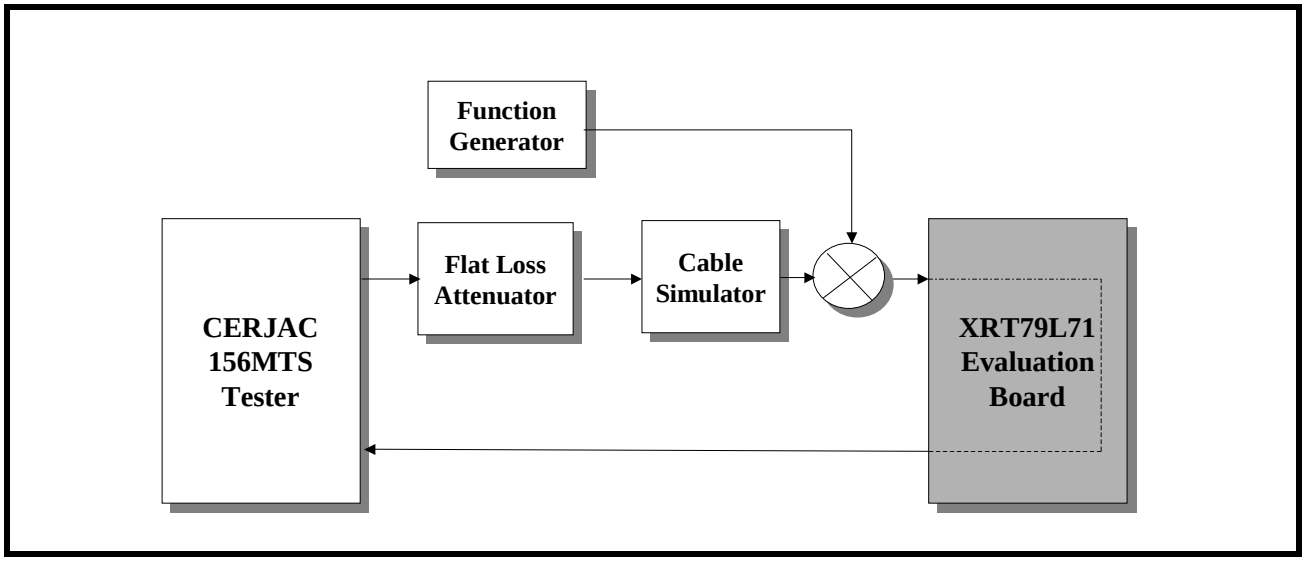
ITU-T G.703 mandates that an E3 Receiver be designed such that it will have an interference margin of at least 20dB. As a consequence, we have specifically designed the Receive DS3/E3 LIU Block such that it will have sufficient Interference Margin in order to meet both the ITU and our Customer's requirements.

This particular section documents the Interference Margin capability of the Receive DS3/E3 LIU Block, for E3 applications.

Test Approach

The basic test configuration used for "Interference Margin" measurements is illustrated below in Figure 172.

FIGURE 172. ILLUSTRATION OF TEST SET-UP USED TO TEST THE XRT79L71 FOR INTERFERENCE MARGIN



The CERJAC Tester was configured to internally generate a E3 signal, which was filled with an un-framed 2^23-1 PRBS pattern.

The E3 line signal was then routed to the ME-1005 75W Coaxial Cable Simulator (from Mountain Engineering). The Cable Simulator could either be configured to 0 to 12dB of "shaped" loss into the E3 line signal.

Next, this attenuated signal will be routed to one of the inputs of a summing network. The output of the function-generator (the output frequency of which is set at half the bit-rate) will be applied to the other input. This signal represents the "Interfering Tone" or "Noise" in this test. The summing network will add the "Noise" signal to the "distorted signal", and will output this composite signal to the "Receive Input" of the XRT79L71 Evaluation Board.

NOTE: *For E3 testing, the function generator was configured to generate a sinewave that has a frequency of 17.184MHz.*

Test Results

Table 46 presents the "Interference Margin" Test results for E3 Applications.

TABLE 46: INTERFERENCE MARGIN TEST RESULTS FOR E3 APPLICATIONS

TEST NUMBER	CABLE LENGTH (IN RECEIVE DIRECTION)	INTERFERENCE MARGIN TEST RESULTS
1	0dB	17dB
2	12.1dB	14dB

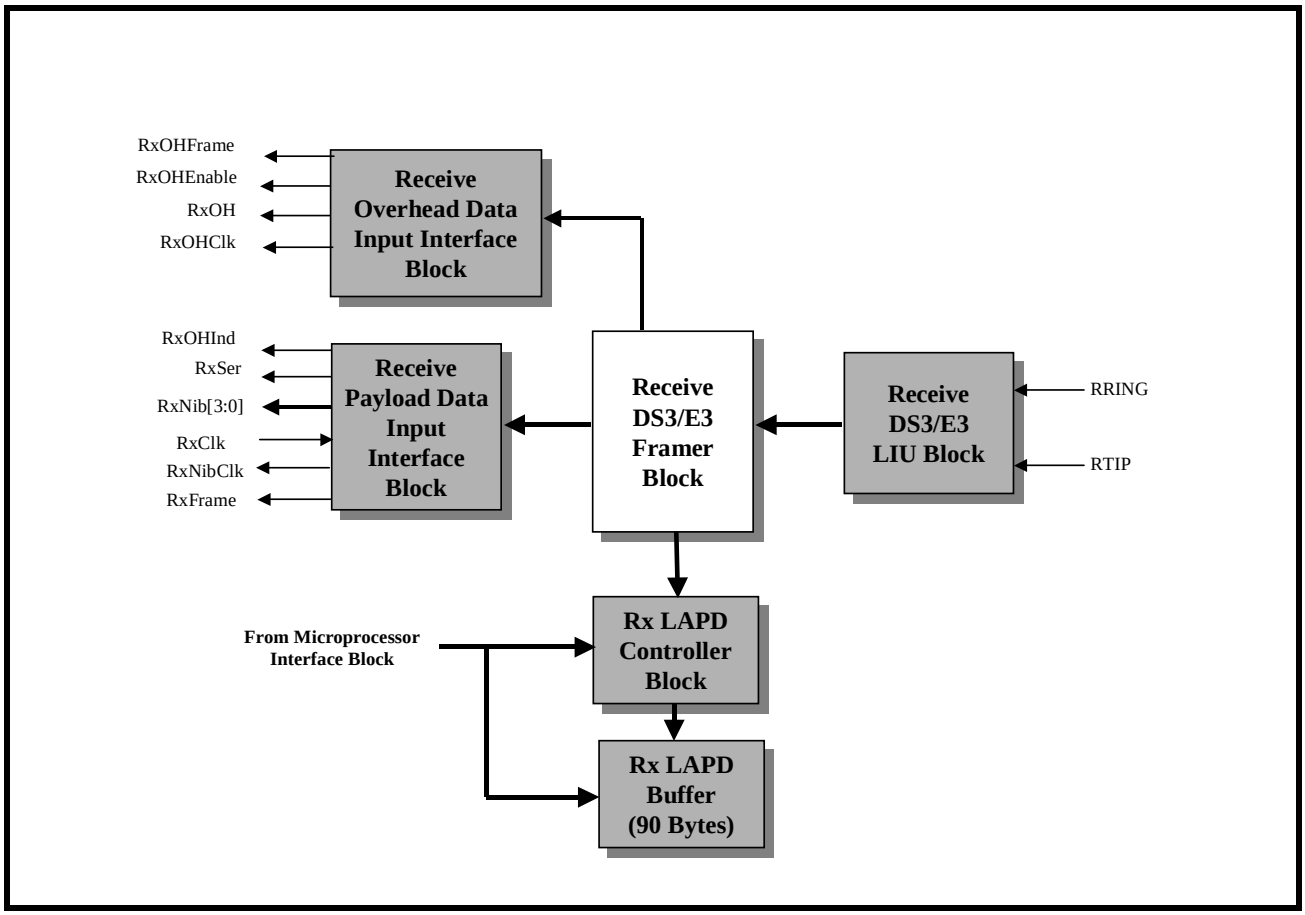
5.3.1.9.3 Jitter Tolerance Capability of the Receive E3 LIU Block

5.3.1.10 Receive DS3/E3 LIU Block Interrupts

5.3.2 THE RECEIVE E3 FRAMER BLOCK

The Receive DS3/E3 Framer block is the second functional block (within the Receive Direction) of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. 6 presents an illustration of the "Receive Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the Receive DS3/E3 Framer block highlighted.

FIGURE 173. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE DS3/E3 FRAMER" BLOCK HIGHLIGHTED).



The purpose of the Receive E3 Framer block (within the XRT79L71) is to accomplish the following.

- To acquire and maintain frame synchronization with the incoming E3 data-stream.
- To detect and declare all of the following defect/error conditions
  - **a.** LOS (Loss of Signal)
  - **b.** AIS (Alarm Indication Signal)
  - **c.** LOF (Loss of Frame)
  - **d.** FERF/RDI (Far-End Receive Failure or Remote Defect Indicator)
  - **e.** FEBE/REI (Far-End Block Error or Remote Error Indicator) Events (if configured to support BIP-4 verification)
  - **f.** BIP-4 Nibble Errors (if configured to support BIP-4 verification)

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

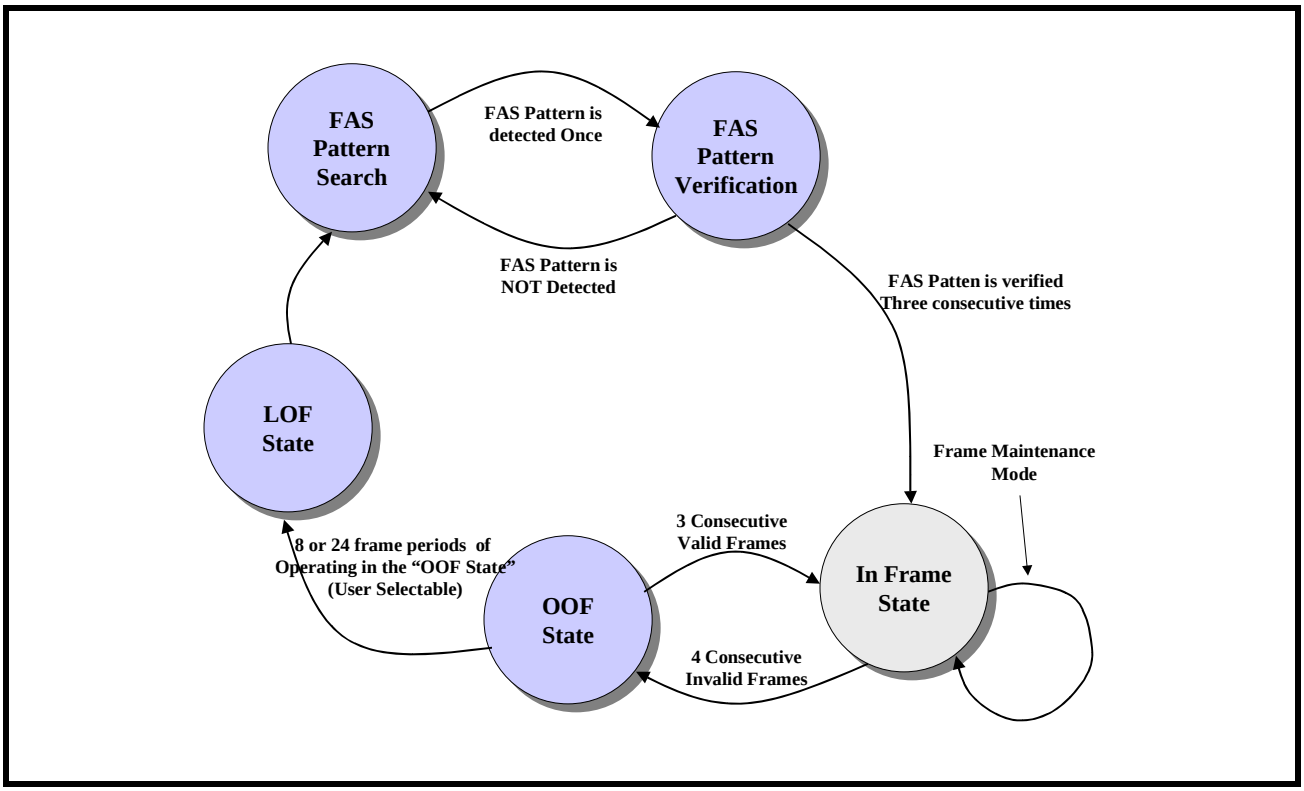
The Frame Maintenance Mode: In this mode, the Receive E3 Framer block is trying to maintain frame synchronization with the incoming E3 frames.

The operation of the Receive E3 Framer block, in both the Frame Acquisition and the Frame Maintenance Modes will be discussed, in considerable detail below.

5.3.2.1 THE FRAME-ACQUISITION MODES

The operation of the Receive E3 Framer block, while in the "Frame Acquisition" mode is best understood by reviewing the "Receive E3 Framer Block's Frame Acquisition/Maintenance Algorithm" state machine diagram, presented below in Figure 174.

FIGURE 174. THE RECEIVE E3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM - E3, ITU-T G.751 APPLICATIONS



Processing through the Frame Acquisition Modes

The Receive E3 Framer block will be performing "Frame Acquisition" operation while it is operating in the following states (per the "E3, ITU-T G.751 Frame Acquisition/Maintenance" algorithm State Machine diagram, as depicted in Figure 174.)

- FAS Pattern Search
- FAS Pattern Verification
- OOF State
- LOF State

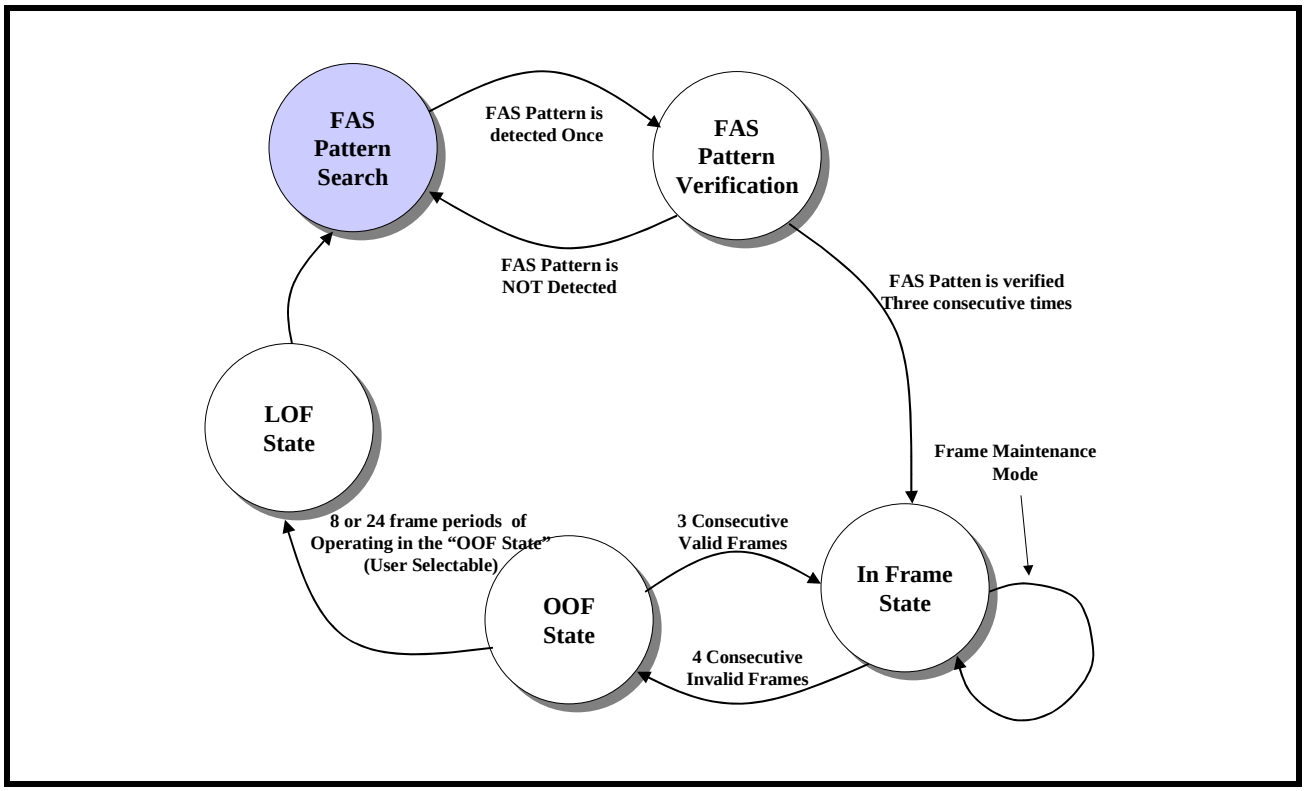
Once the Receive E3 Framer block enters the "In-Frame" state (per Figure 174), then it will begin "Frame Maintenance" Operation. The Receive E3 Framer block's operation in each of the "Frame Acquisition" states is presented below.

The "FAS Pattern Search" State

When the XRT79L71 is first powered up, or a "Hardware RESET" event occurs, the Receive E3 Framer block will be operating in the "Frame Acquisition" Mode. At this point, the very first thing that the Receive E3 Framer block will do is to begin to look for valid E3 frames (within the incoming E3 data-stream) by first searching for the FAS pattern. At this "initial point" the Receive E3 Framer block will be operating in the "FAS Pattern Search" state within the "E3 Frame Acquisition/Maintenance" Algorithm state machine diagram. In order to

clearly convey the mode that the Receive E3 Framer block is currently operating in, Figure 174 has been repeated below, with the "FAS Pattern Search" state shaded.

FIGURE 175. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S "FRAME ACQUISITION/MAINTENANCE" ALGORITHM (WITH THE "FAS PATTERN SEARCH STATE" SHADED)

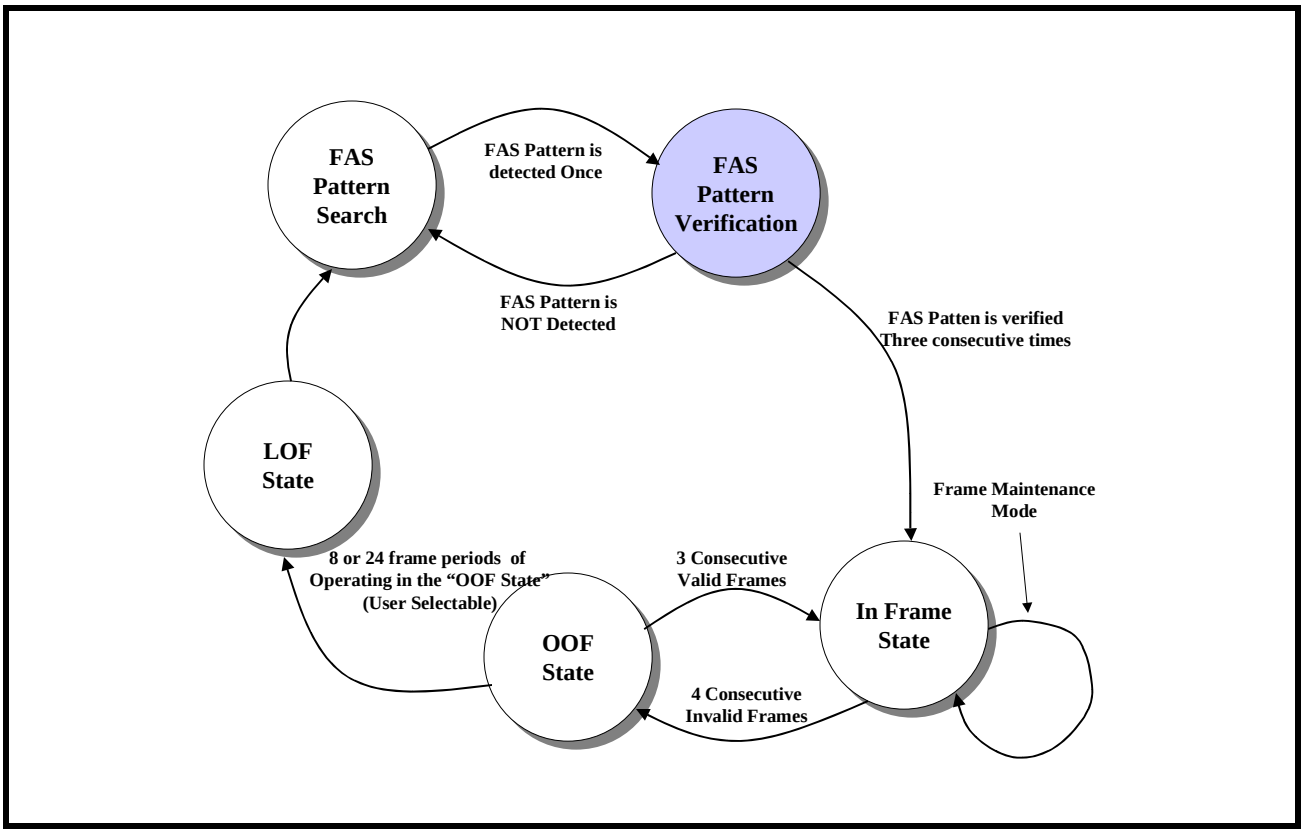


Recall from the discussion in SEE"DESCRIPTION OF THE E3, ITU-T G.751 FRAME STRUCTURE AND THE OVERHEAD BITS" ON PAGE 259., that each E3, ITU-T G.751 frame consists of the FAS pattern (which is of the value "[1,1,1,1,0,1,0,0,0,0]"). The Receive E3 Framer block will attempt to locate the FAS pattern by performing five (5) different searches in parallel. The "FAS Pattern Search" will be declared successful if the Receive E3 Framer block can locate a single instantiation of the FAS pattern. Once the Receive E3 Framer block detects this particular pattern, then it will declare this instantiation of the FAS Pattern (within the incoming E3 data-stream) as a "possible" or "candidate" FAS Pattern within the incoming E3 data-stream. Additionally, the Receive E3 Framer block will now transition to the "FAS Pattern Verification" State, within the "E3 Frame Acquisition/Maintenance" Algorithm (per Figure 174).

The "FAS Pattern Verification" State

Once the Receive E3 Framer block has detected the FAS Pattern, it must now verify that this pattern is indeed the FAS pattern, and not some other set of bits within the incoming E3 data-stream (that are mimicking the FAS pattern). Hence, the purpose of the "FAS Pattern Verification" state is to implement this additional check on the "Candidate FAS Pattern". In order to clearly convey the mode that the Receive E3 Framer block is now currently operating in, Figure 174 has been repeated below, with the "FAS Pattern Verification" state shaded.

FIGURE 176. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S "FRAME ACQUISITION/MAINTENANCE" ALGORITHM (WITH THE "FAS PATTERN VERIFICATION STATE" SHADED)



When the Receive E3 Framer block enterse this state, it will then quit performing the "bit-by-bit" search (within the incoming E3 data-stream) for the FAS pattern. Instead, the Receive E3 Framer block will read in a string of 10 bits that occur 1536 bits (e.g., one E3 frame period) later after the "Candidate FAS pattern" was first detected. The Receive E3 Framer block will perform this check during the next three (3) incoming E3 frame periods. If, in all three of these checks, the "sampled" (or "read-in") 10-bit string matched the "FAS Pattern", then the Receive E3 Framer block will "conclude" that it has TRULY found the FAS Pattern, and will now transition into the "In-Frame" state. However, if (in at least one of these "10-bit pattern check"), the sampled 10-bit pattern did NOT match the "FAS Pattern", then the Receive E3 Framer block will "conclude" that it has been fooled by data (within the incoming E3 data-stream) mimicking the FAS pattern, and will transition back into the "FAS Pattern Search" state.

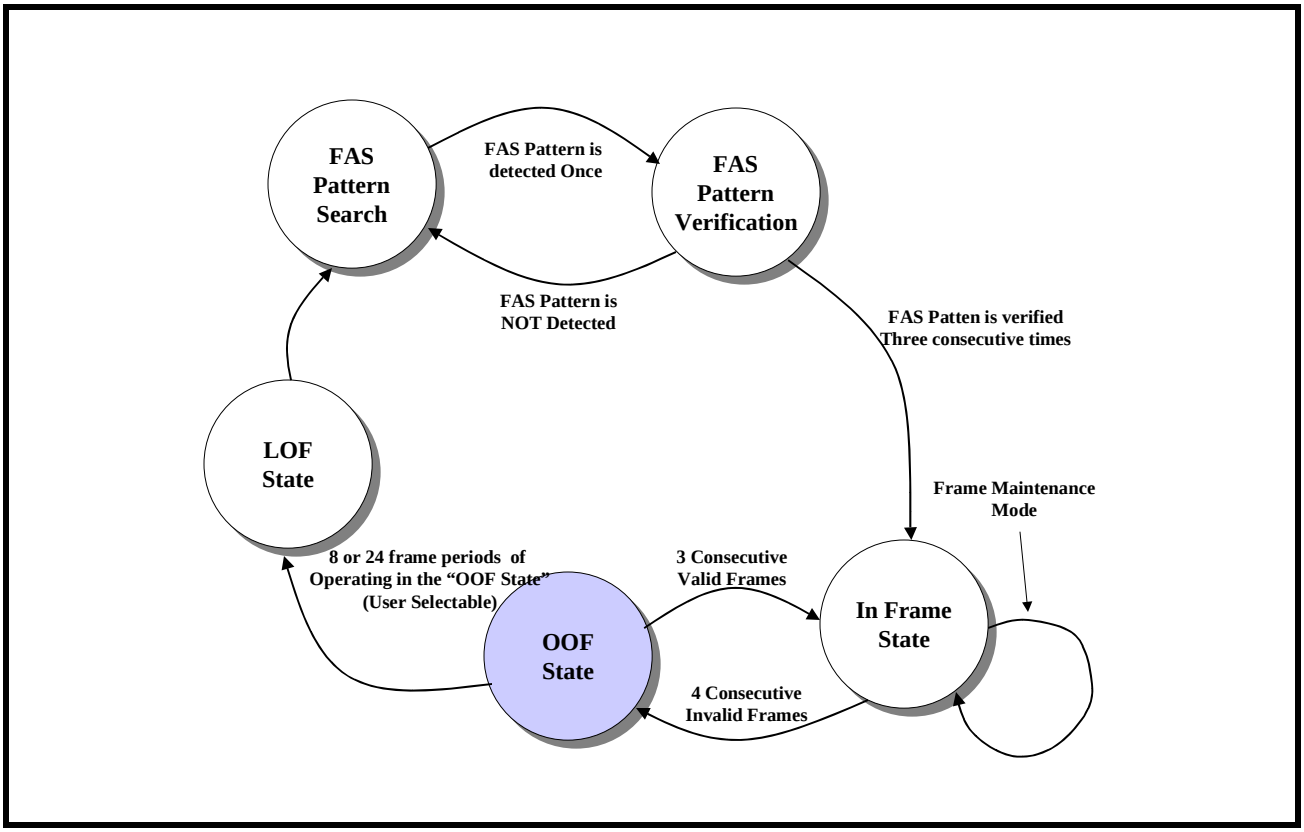
The "In-Frame" State

Once the Receive E3 Framer block (while operating in the "FAS Pattern Verification" state) detects the FAS Pattern, in the "correct" location (e.g., exactly one E3 frame period after the "Candidate" FAS Pattern was first detected, within three consecutive E3 frames), then it will "conclude" that it has correctly located the "FAS Pattern" and it will transition into the "In-Frame" state. Once the Receive E3 Framer block enters the "In-Frame" state, it will cease performing "Frame Acquisition" functions, and it will begin to perform "Frame Maintenance" functions. A detailed description of the Receive E3 Framer block, operating in the "In-Frame" state can be found in Section 5.3.2.1.

The "OOF" (Out of Frame) State

If the Receive E3 Framer block, while operating in the "In-Frame" state, receives four (4) consecutive "incoming" E3 frames, in which the "FAS Pattern" bits are erred, then it (the Receive E3 Framer block) will transition into the "OOF State". In order to clearly convey the mode that the Receive E3 Framer block is currently operating in, **Figure 174** has been repeated below, with the "OOF State" shaded.

FIGURE 177. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S "FRAME ACQUISITION/MAINTENANCE" ALGORITHM (WITH THE "OOF STATE" SHADED)



The Receive E3 Framer block's operation, while in the "OOF State" is a unique mix of the "Framing Maintenance" and "Frame Acquisition" operations. In the "OOF State" the Receive E3 Framer block will exhibit some "Frame Acquisition" operational characteristics by attempting to locate (once again) the FAS pattern. However, the Receive E3 Framer block will also exhibit some Frame Maintenance operational behavior by still using the most recent frame synchronization for its overhead and payload byte processing.

What happens when the Receive E3 Framer block transitions into the "OOF" State?

As the Receive E3 Framer block transitions from the "In-Frame" into the "OOF State", it will inform the Microprocessor of this fact by doing all of the following.

- **Declaring the "OOF Defect Condition"**

The Receive E3 Framer block will indicate that it is declaring the "OOF Defect Condition" by setting bit 5 (OOF Defect Declared), within the "Receive E3 Configuration and Status Register # 2" to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	1	0	0	0	0	0

- **Generating the "Change in OOF Defect Condition" Interrupt**

The Receive E3 Framer block will indicate that it is generating the "Change in OOF Defect Condition" Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "low".

b. Setting Bit 3 (Change in OOF Defect Condition Interrupt Status), within the "Receive E3 Interrupt Status Register # 1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

What happens if the Receive E3 Framer block transitions back into the "In-Frame" State?

If the Receive E3 Framer block (while operating in the "OOF State") is able to correctly locate the "FAS Pattern" in three (3) consecutive E3 frames, then it will transition back into the "In-Frame" State. The Receive E3 Framer block will inform the Microprocessor of this occurrence by doing all of the following.

- **Clearing the "OOF Defect Condition"**

The Receive E3 Framer block will indicate that it is clearing the "OOF Defect Condition" by setting Bit 5 (OOF Defect Declared), within the "Receive E3 Configuration and Status Register # 2", to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

- **Generating the "Change in OOF Defect Condition" Interrupt**

The Receive E3 Framer block will indicate that it is declaring the "Change in LOF Defect Condition" Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "low".

b. Setting Bit 3 (Change in OOF Defect Condition Interrupt Status) within the "Receive E3 Interrupt Status Register # 1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Transitioning into the "LOF State"

However, if the Receive E3 Framer block resides in the "OOF State" for more than the "User-Selectable" number of E3 frame periods (without being able to locate the FAS Pattern), then it (the Receive E3 Framer block) will automatically transition into the "LOF State".

Selecting the number of E3 Frame Periods, before transitioning into the "LOF State"

The user can configure the Receive E3 Framer block to reside in the "OOF State" and search the incoming E3 data-stream for each 8 or 24 E3 frame periods, prior to transitioning into the "LOF State". The user can accomplish this configuration setting by writing the appropriate value into Bit 7 (Receive LOF Algo), within the "Receive E3 Configuration and Status Register # 2" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	1	1	0	0	0	0	1

Setting this bit-field to "0" configures the Receive E3 Framer block to reside within the "OOF State" for up to 24 E3 frame periods before transitioning into the "LOF State". Conversely, setting this bit-field to "1" configures the Receive E3 Framer block to reside within the "OOF State" for up to 8 E3 frame periods before transitioning into the "LOF State".

The "LOF" (Loss of Frame) State

If the Receive E3 Framer block enters the "LOF Condition" state, then the following things will happen.

- The Receive E3 Framer block will discardc the most recent frame synchronization that it had, and
- The Receive E3 Framer block will make an unconditional transition to the "FAS Pattern Search" state, within the "Receive E3 Framer Block's - Frame Acquisition/Maintenance Algorithm".

Additionally, the Receive E3 Framer block will notify the Microprocessor/Microcontroller of this transition into the "LOF State", by doing the following.

- **Declaring the "LOF Defect Condition"**

The Receive E3 Framer block will indicate that it is declaring the "LOF Defect Condition" by setting Bit 6 (LOF Defect Condition Declared), within the "Receive E3 Configuration and Status Register # 2" to "1", as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	1	1	0	0	0	0	1

- **Generating the "Change in LOF Defect Condition" Interrupt**

The Receive E3 Framer block will indicate that it is declaring the "Change in LOF Defect Condition" Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "low".

b. Setting Bit 2 (Change in LOF Defect Condition Interrupt Status), within the "Receive E3 Interrupt Status Register # 1", to "1" as depicted below

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

5.3.2.2 THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DECLARATION CRITERIA

Once the Receive E3 Framer block has entered the "In-Frame" state, then it is considered to be operating in the "Frame Maintenance" Mode. As the Receive E3 Framer block transitions into the "In-Frame" state, it will notify the Microprocessor/Microcontroller of this fact by doing all of the following.

- Clearing the "OOF" (Out-of-Frame) and "LOF" (Loss of Frame) Defect Conditions

The Receive E3 Framer block will indicate that it is clearing both the "OOF" and "LOF Defect Conditions" by setting Bit 5 (OOF Defect Condition Declared) and 6 (LOF Defect Condition Declared) within the "Receive E3 Configuration and Status Register # 2" to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	1

- Generating the "Change of OOF Defect Condition" and "Change in LOF Defect Condition" Interrupts

The Receive E3 Framer block will indicate that it is generating the "Change in OOF Defect Condition" and the "Change in LOF Defect Condition" Interrupts, by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "low".

b. Setting Bits 2 (Change in LOF Defect Condition Interrupt Status) and 3 (Change OOF Defect Condition Interrupt Status) within the "Receive E3 Interrupt Status Register # 1" to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	1	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 Framing Alignment (FAS) Pattern is present at its proper location, within the incoming E3 data-stream. In general, as long as the FAS pattern is present at its proper location (with a small number of errors) the Receive E3 Framer block will continue to operate in the "Frame Maintenance" Mode. However, if this Receive E3 Framer block begins to detect a large number of FAS pattern errors within the incoming E3 data-stream, then it will exit the "In-Frame' state and will then declare the "OOF defect condition" whenever the Receive E3 Framer block receives at least four (4) consecutive E3 frames, in which the FAS pattern was erred.

Forcing a Reframe via Software Command

The Receive DS3/E3 Framer block permits the user to command a reframe procedure with the Receive E3 Framer block via software command. The user can accomplish this by inducing a "0" to "1" transition in Bit 0 (Reframe), within the "I/O Control Register" as depicted below. Once the user executes this step, then the Receive E3 Framer block will be forced into the Frame Acquisition Mode (or more specifically, into the "FAS Pattern Search" State, per **Figure 174**) and will begin to search for the FAS pattern. The XRT79L71 will also respond to this command by declaring both the "OOF" and "LOF Defect Conditions", and by generating the "Change in OOF Defect Condition" and the "Change in the LOF Defect Condition" interrupts.

I/O Control Register (Address = 0x1101)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	Unused				Reframe
R/W	R/O	R/W	R/O	R/O	R/O	R/O	R/W
1	0	1	0	1	0	0	0 -> 1

NOTE: *After the "0" to "1" transition within Bit 0 (Reframe) has been implemented, go back and induce a "1" to "0" transition within this bit-field.*

5.3.2.3 DECLARING AND CLEARING THE LOS DEFECT CONDITION

The Receive E3 Framer block has the responsibility for declaring and clearing the LOS (Loss of Signal) defect condition, within the incoming E3 data-stream, as described below.

5.3.2.3.1 Declaring the LOS Defect Condition

The Receive E3 Framer block will declare the "Loss of Signal" (LOS) Defect condition when it detects at least 32 consecutive "0s" within the incoming E3 data-stream, or if the Receive DS3/E3 LIU Block declares the LOS defect condition. The Receive E3 Framer block will indicate that it is declaring the LOS defect condition by doing all of the following.

- Setting Bit 4 (LOS Defect Declared) within the Receive E3 Configuration and Status Register # 2", to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	1	0	0	0	0

- The Receive E3 Framer block will also generate the "Change of LOS Defect Condition" Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 1 (Change in LOS Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1" to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

NOTE: *The LOS Defect Declaring Criteria for the Receive DS3/E3 LIU Block will be discussed in* ***SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.***.

5.3.2.3.2 Clearing the LOS Defect Condition

The Receive E3 Framer block will clear the "LOS Defect" condition when both of the following conditions are met.

a. When the Receive E3 Framer block (while declaring the LOS Defect Condition) receives a stream of 32 consecutive E3 bits, which does not contain a string of four (4) consecutive "0s".

b. When the Receive DS3/E3 LIU Block clears the LOS defect condition.

The Receive E3 Framer block will indicate that it is clearing the LOS defect condition by:

- Setting Bit 4 (LOS Defect Condition Declared) within the Receive E3 Configuration and Status Register # 2" to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	0

The Receive E3 Framer block will also generate the "Change of LOS Defect Condition" Interrupt by asserting the Interrupt Output pin (e.g., by pulling it "LOW"), and setting Bit 1 (Change in LOS Defect Condition Interrupt Status), within the Receive E3 Configuration and Status Register # 2" to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

NOTE: *The LOS Clearance criteria for the Receive DS3/E3 LIU Block will be discussed in* ***SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.***.

Configuration Options for the LOS Declaration/Clearance Criteria

The Receive DS3/E3 Framer block (within the XRT79L71) permits the user to change the "LOS Declaration criteria" such that the LOS defect condition is declared only if the Receive DS3/E3 LIU Block declares the LOS defect condition. If the configuration selection is implemented, then the "internally-generated" LOS criteria of "32 consecutive 0s" will be disabled. The user can accomplish this configuration selection by writing a "0" to Bit 5 (Internal LOS Enable) within the "Framer Operating Mode Register" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	1

5.3.2.3.3 The Relationship between the LOS Defect Condition being declared and cleared in the Receive DS3/E3 LIU Block and in the Receive DS3/E3 Framer Block

5.3.2.4 DECLARING AND CLEARING THE AIS DEFECT CONDITION

The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the AIS (Alarm Indication Signal) defect, as described below.

5.3.2.4.1 Declaring the AIS Defect Condition

If the XRT79L71 has been configured to operate in the "E3, ITU-T G.751 Framing format, then the Receive E3 Framer block will declare the AIS defect condition anytime it receives an E3 data-stream that contains 7 or less "0s" within two consecutive E3 frame periods.

If the Receive E3 Framer block declares the AIS defect condition, then it will do all of the following.

- It will set Bit 3 (AIS Defect Condition Declared)

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	1	0	0	0

- The Receive E3 Framer block will also generate the "Change in AIS Defect Condition" Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 0 (Change in AIS Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1" to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

5.3.2.4.2 Clearing the AIS Defect Condition

The Receive E3 Framer block will clear the "AIS Defect" condition whenever it receives two consecutive E3 frames that contain 8 or more "0s", within the incoming E3 data-stream. The Receive E3 Framer block will indicate that it is clearing the AIS defect by:

- It will set Bit 3 (AIS Defect Condition Declared), within the Receive E3 Configuration and Status Register # 2" to "0", as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	0

- The Receive E3 Framer block will also generate the "Change in AIS Defect Condition" Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "LOW"), and setting Bit 0 (Change in AIS Defect Condition Interrupt Status), within the "Receive E3 Interrupt Status Register # 1" to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.751 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	R/O	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

5.3.2.5 DECLARING AND CLEARING THE FERF/RDI DEFECT CONDITION

If the Receive DS3/E3 Framer block has NOT been configured to compute and verify the BIP-4 Nibble value (within the incoming E3 data-stream), then it will have the responsibility for declaring and clearing the "FERF/RDI" defect condition.

Tto have the Receive E3 Framer block declare and clear the FERF/RDI defect conditions, the user MUST configure the Receive E3 Framer block to NOT compute and verify the BIP-4 Nibble value (within the incoming E3 data-stream). Implement this configuration, by setting Bit 0 (RxBIP-4 Enable), within the Receive E3 Configuration and Status Register # 1" to "0" as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERF Algo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	X	0	0	0	0

5.3.2.5.1 Declaring the FERF/RDI (Far-End Receiv Failure/Remote Defect Indicator) Defect Condition

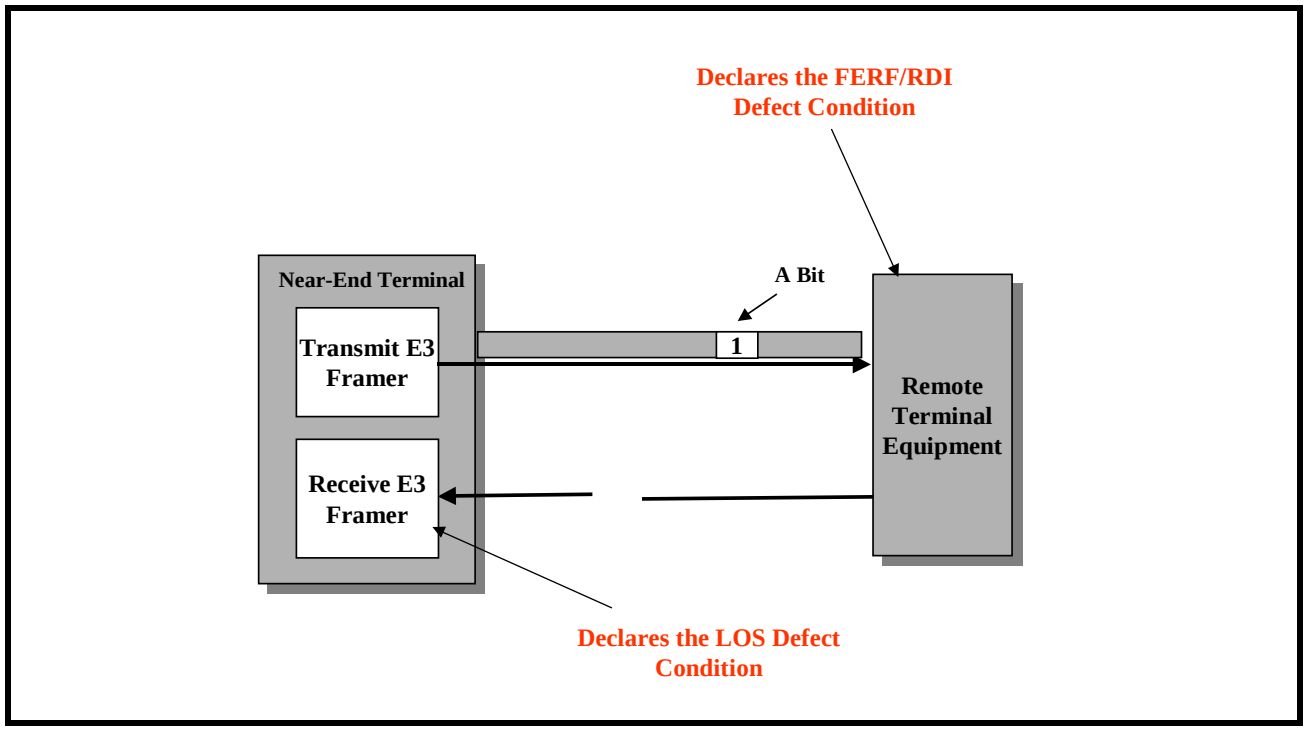
If the Receive E3 Framer block has NOT been configured to compute and verify the BIP-4 Nibble value (within the incoming E3 data-stream), then it will declares the FERF or RDI defectr condition, if it starts to receive E3 frames in which the "A" bit-field is set to "1".

NOTE: *If the Receive E3 Framer block has been configured to compute and verify the BIP-4 nibble (within the incoming E3 data-stream) then it will interpret the "A" bit-field being set to "1" as a FEBE/REI indicator. Please see*

SEE"DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS" ON PAGE 390. for details on how the Receive E3 Framer block handles the FEBE/REI indicators.

Recall, that back in SEE"TRANSMITTING THE FERF/RDI INDICATOR" ON PAGE 323., we described how one can configure the Transmit DS3/E3 Framer block to automatically transmit the FERF/RDI indicator to the remote terminal anytime (and for the duration) that the "Near-End" Corresponding Receive DS3/E3 Framer Block declares either the LOS, LOF/OOF or AIS defect condition. **Figure 178** recaps some of this discussion by presenting a figure that depicts the Transmit DS3/E3 Framer block automatically transmitting the FERF/RDI indicator to the "remote terminal equipment" (by setting the "FERF/RDI" bit within each "outbound" E3 frame to "1") because the "Near-End" Corresponding Receive DS3/E3 Framer block was declaring the LOS defect condition.

FIGURE 178. FA SIMPLE ILLUSTRATION OF THE "NEAR-END" TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT.



SEE"TRANSMITTING THE FERF/RDI INDICATOR" ON PAGE 323., we describe how a given Terminal will generate the FERF/RDI indicator. In this Section, we will describe how a given Terminal (that contains the XRT79L71) will respond to receiving the FERF/RDI indicator from the remote terminal equipment.

5.3.2.5.2 The FERF/RDI Defect Declaration Criteria

The Receive E3 Framer block will declare the FERF/RDI defect condition, if it receives a "user-selectable" number of consecutive E3 frames, in which the FERF/RDI bit-field is set to "1".

The appropriate number of consecutive incoming E3 frames (in which the "A" bit-field are set to "1") prior to the Receive E3 Framer block declaring the "FERF/RDI Defect condition" can be selected by writing the appropriate value into Bit 4 (RxFERF Algo) within the "Receive E3 Configuration and Status Register # 1" as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERF Algo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	X	0	0	0	0

Setting this bit-field to "0" configures the Receive E3 Framer block to declare the FERF/RDI defect condition, if it has received at least three (3) consecutive E3 frames, in which the "A" bit-field has been set to "1". Conversely, setting this bit-field to "1" configures the Receive E3 Framer block to declare the FERF/RDI defect condition, if it has received at least five (5) consecutive E3 frames, in which the FERF/RDI bit-field has been set to "1".

When the Receive E3 Framer block declares the "FERF" or "RDI" defect condition in the incoming E3 frame, then it will then do the following.

- It will set Bit 0 (FERF/RDI Defect Declared), within the Receive E3 Configuration and Status Register # 2" to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	1

This bit-field will remain asserted for the duration that the Receive E3 Framer block declares the FERF/RDI defect condition.

- The Receive E3 Framer block will also generate the "Change in FERF/RDI Defect Condition" Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 3 (Change in FERF/RDI Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 2" to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.751 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FERF/RDI Defect Condition Interrupt Status	Detection of BIP-4 Error Interrupt Status	Detection of FAS Bit Error Interrupt Status	Reserved
R/O	R/O	R/O	R/O	RUR	RUR	RUR	R/O
0	0	0	0	1	0	0	0

5.3.2.5.3 Clearing the FERF/RDI Defect Condition

The Receive E3 Framer block will clear the "FERF/RDI Defect Condition" whenever it receives a "User-Selectable" number of E3 frames, in which the "A" bit-field is set to "0".

Setting the FERF/RDI Defect Clearance Criteria

The user can specify the "FERF/RDI Defect Clearance" Criteria by writing the appropriate value into Bit 4 (RxFERF Algo), as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERF Algo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	X	0	0	0	0

If this bit-field is set to "0", then the Receive E3 Framer block will clear the "FERF/RDI" defect condition anytime it receives at least three (3) consecutive E3 frames, in which the "A" bit-field has been set to "0". Conversely, if this bit-field is set to "1", then the Receive E3 Framer block will clear the "FERF/RDI" defect condition anytime it receives at least five (5) consecutive E3 frames, in which the "A" bit-field has been set to "0".

Whenever the Receive E3 Framer block clears the FERF/RDI defect condition, it will do so, by doing all of the following.

- Setting Bit 0 (FERF/RDI Defect Condition Declared) within the "Receive E3 Configuration and Status # 2" to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	0

The Receive E3 Framer block will also generate the "Change in FERF/RDI Defect Condition" Interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 3 (Change of FERF/RDI Defect Condition Interrupt Status), within the "Receive E3 Interrupt Status Register # 2" to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.751 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FERF/RDI Defect Condition Interrupt Status	Detection of BIP-4 Error Interrupt Status	Detection of FAS Bit Error Interrupt Status	Reserved
R/O	R/O	R/O	R/O	RUR	RUR	RUR	R/O
0	0	0	0	1	0	0	0

5.3.2.6 DETECTING BIP-4 NIBBLE ERRORS

If the Receive E3 Framer block has been configured accordingly, it has the responsibility for detecting and flagging the occurrence of BIP-4 nibble errors.

If the Receive E3 Framer block is configured to compute and verify the BIP-4 nibbles, within the incoming E3 data-stream, then set Bit 0 (RxBIP-4 Enable) within the "Receive E3 Configuration and Status Register #1" to "1" as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERF Algo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	X	0	0	0	1

Assuming that the user implement the "above-mentioned" procedure, then the Receive E3 Framer block has the responsibility for detecting and flagging the occurrence of BIP-4 Nibbles, as described below.

Processing at the Remote Terminal Equipment

As the remote terminal equipment is generating and transmitting the E3 data-stream (that the local terminal equipment will ultimately receive and process), it will compute the BIP-4 value over an entire E3 frame. The results of this BIP-4 calculation will be inserted into the BIP-4 Nibble-position, within the very next "outbound" E3 frame. The purpose of the BIP-4 Nibble is to support Performance Monitoring and Error Detection within the E3 data-stream, as it is transported from one terminal equipment to another.

Processing at the Local Terminal Equipment

As the Receive E3 Framer block receives a given E3 frame (that was generated by the remote terminal equipment) it will locally-compute its own BIP-4 value for this incoming E3 frame. Afterwards, the Receive E3 Framer block will compare its "locally-computed" BIP-4 value with the contents of the BIP-4 nibble, within the very next incoming E3 frame. If these two values match, then the Receive E3 Framer block will "conclude" that it has received this particular E3 frame, in an un-erred manner. Conversely, if these two values DO NOT match, then the Receive E3 Framer block will "conclude" that is has received this particular E3 frame, in an erred manner.

If the Receive E3 Framer block determines that the BIP-4 Nibble (within a given E3 frame) is erred, then it will do the following.

- It will generate the "Detection of BIP-4 Error" Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 2 (Detection of BIP-4 Error Interrupt Status), within the Receive E3 Interrupt Status Register # 2" to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.751 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FERF/RDI Defect Condition Interrupt Status	Detection of BIP-4 Error Interrupt Status	Detection of FAS Bit Error Interrupt Status	Reserved
R/O	R/O	R/O	R/O	RUR	RUR	RUR	R/O
0	0	0	0	0	1	0	0

- It will increment the "PMON P-bit/Parity Error Count" Register once of each E3 frame that is detemined to have an erred BIP-4 nibble. The "PMON P-bit/Parity Error Count" Register is located at Address = 0x1154 and 0x1155. The bit-format for each of these registers is presented below.

PMON Parity/P-Bit Error Count Register - MSB (Address = 0x1154)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Parity/P-Bit Error Count Register - LSB (Address = 0x1155)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these "Performance Monitor" Registers, please see Section 2.5.*

- It will also increment the "One Second - P-bit/Parity Error Count - Accumulator" Register once for each incoming E3 frame that is determined to have an erred BIP-4 Nibble. The "One Second - P-Bit/Parity Error Count - Accumulator" Register is located at Address = 0x1170 and 0x1171. The bit-format for this 16-bit register is presented below.

One Second - Parity Error Accumulator Register - MSB (Address = 0x1170)

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

One Second - Parity Error Accumulator Register - LSB (Address = 0x1171)

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

NOTES:

1. *If the Transmit DS3/E3 Framer block has been configured to compute and insert the BIP-4 nibble into each outbound E3 frame, then the "Near-End" Transmit DS3/E3 Framer block (within this particular XRT79L71) will automatically be configured to set the "A" bit-field (within its very next outbound E3 frame) to "1" for each time in which the Receive E3 Framer block receives an E3 frame with an erred BIP-4 Nibble. Please see* ***SEE"TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR" ON PAGE 324.****for more information on this transmission of the FEBE/REI indicator.*
2. *For instructions on how to read out these "Performance Monitor" Registers, please see Section 2.5.*

5.3.2.7 DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS

If the Receive E3 Framer block has been configured accordingly, it has the responsibility for detecting and flagging the occurrence of FEBE/REI events.

To configure the Receive E3 Framer block to detect and flag FEBE/REI events, within the incoming E3 data-stream, then user must set Bit 0 (RxBIP-4 Enable) within the "Receive E3 Configuration and Status Register # 1" to "1" as depicted below.

Receive E3 Configuration and Status Register # 1 - G.751 (Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			RxFERF Algo	Unused			RxBIP-4 Enable
R/O	R/O	R/O	R/W	R/O	R/O	R/O	R/W
0	0	0	X	0	0	0	1

Assuming that the user implements the "above-mentioned" procedure, then the Receive E3 Framer block has the responsibility for detecting and flagging the occurrence of FEBE/REI events, as described below.

If the source and sink terminals are configured to support BIP-4 nibble verification, then the "A" bit will function as the "FEBE/REI" bit-field. The remote terminal equipment (which is generating the incoming E3 data-stream) will set the "A" bit-field to the appropriate value that indicates whether or not it (the remote terminal) is experiencing any BIP-4 nibble errors. If the remote terminal equipment is currently not detecting any BIP-4 Nibble errors, then it will set the "A" bit-field (within each outbound E3 frame) to "0". Hence, the FEBE/REI value (e.g., the "A" bit value) for an un-erred condition is "0".

Conversely, if the remote terminal equipment is detecting BIP-4 nibble errors, then it will proceed to set the "A" bit-field (within the very next outbound E3 frame) to "1".

If "BIP-4 Verification" has been enabled, and if the Receive E3 Framer block receives any E3 frames, in which the "A" bit is set to "1", then it will increment the "PMON FEBE/REI Event Count Registers" once for each E3 frame that it receives, in which the "A" bit set to "1". The bit-format and address locations these registers are presented below.

PMON FEBE Event Count Register - MSB (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FEBE Event Count Register - LSB (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these "Performance Monitor" Registers, please see Section 2.5.*

5.3.2.8 DETECTING FAS PATTERN ERRORS

Recall in SEE"THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DECLARATION CRITERIA" ON PAGE 379., we mentioned that in order to verify that the Receive E3 Framer block is maintaining proper Frame Synchronization with the incoming E3 data-stream, it will continuously check and verify that the FAS pattern (1) can be found in its proper location, and (2) that the FAS pattern is of the correct value. This Section went on to state that if the Receive E3 Framer block were to detect errors in the FAS pattern, within four (4) consecutive E3 frames, then it (the Receive E3 Framer block) would transition over into the "OOF State" and would declare the "OOF Defect" Condition.

In addition to checking and determining whether or not to declare the "OOF" or "LOF" defect condition, the Receive DS3/E3 Framer block will also flag and tally the occurrences of any FAS Pattern errors (that are detected within the incoming E3 data-stream), as described below.

While the Receive E3 Framer block is operating in the "Frame Maintenance" Mode, it will continue to check for valid FAS patterns. If the Receive E3 Framer block detects any errors in the FAS pattern, then it will do the following.

- It will generate the "Detection of FAS Pattern Error" Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "LOW") and setting Bit 1 (Detection of FAS Bit Error Interrupt Status), within the "Receive E3 Interrupt Status Register # 2" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.751 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Change of FERF/RDI Defect Condition Interrupt Status	Detection of BIP-4 Error Interrupt Status	Detection of FAS Bit Error Interrupt Status	Reserved
R/O	R/O	R/O	R/O	RUR	RUR	RUR	R/O
0	0	0	0	0	0	1	0

- It will increment the "PMON Framing Bit/Byte Error Count" Registers, once for each E3 frame that is determined to contain an erred FAS pattern. The "PMON Framing Bit/Byte Error Count" Register is located at Address = 0x1152/0x1153. The bit-format of these registers is presented below.

PMON Framing Bit/Byte Error Count Register - MSB (Address = 0x1152)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Upper_Byte[7:0]							
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 = 0x1153)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTES:

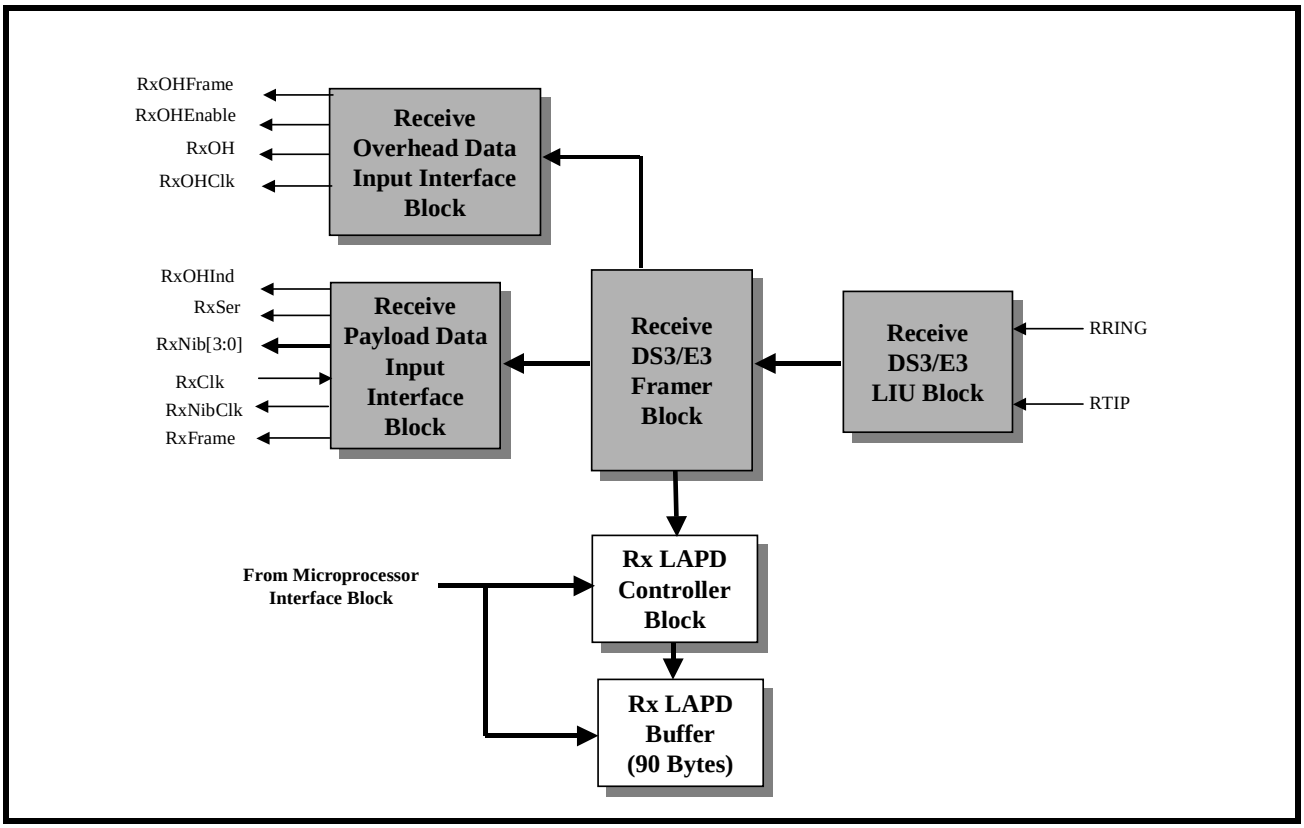
1. *These "Performance Monitor" Registers will cease to increment anytime the Receive E3 Framer block is declaring the "OOF" or "LOF" defect condition.*
2. *For instructions on how to read out these "Performance Monitor" Registers, please see Section 2.5.*

5.3.2.9 Monitoring the State of the N-Bit in the incoming E3 data stream

5.3.3 THE RECEIVE LAPD CONTROLLER BLOCK

The Receive LAPD Controller block is the third functional block (within the Receive Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. **Figure 179** presents an illustration of the "Receive Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the Receive LAPD Controller block highlighted.

FIGURE 179. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE LAPD CONTROLLER" BLOCK HIGHLIGHTED)



The Receive LAPD Controller block consists of the following sections.

- The Receive LAPD Message Buffer
- The Receive LAPD Controller

The Receive LAPD Message Buffer

The purpose of the Receive LAPD Message Buffer is to store the contents of LAPD/PMDL Messages that have been received by the Receive LAPD Controller block. The Receive LAPD Message Buffer will serve as a temporary storage location (of these incoming LAPD/PMDL Messages) until they can be read out by the Microprocessor. The Receive LAPD Message Buffer is actually a 90 byte FIFO that is located at Indirect Address Location 0x11C0 within the XRT79L71 address space.

The Receive LAPD Controller

The Receive LAPD Controller permits the user to receive path maintenance data link (PMDL) message from the remote terminal equipment via the "inbound" E3 frames. In this case, the PMDL Message is extracted out of the "N" bit-fields within each incoming E3 frame. The on-chip Receive LAPD Controller is capable of receiving both "standard" and "non-standard" PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the "Receive LAPD Message" buffer), to store the contents of newly received PMDL Messages. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in **Figure 180**.

FIGURE 180. LAPD MESSAGE FRAME FORMAT

Flag Sequence (8 bits)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		
76 or 82 or Any-size Bytes of Information (Payload)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

Where; for standard (Bellcore GR-499-CORE) applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the Receive LAPD Controller is not currently receiving a LAPD Message, it will instead be receiving a continuous stream of Flag Sequence bytes via the "N" bit within each "inbound" E3 frame.

- **SAPI - Service Access Point Identifier**

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for "standard" (Bellcore GR-499-CORE) applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For "standard" (Bellcore GR-499-CORE) applications the user must use the Control byte the value 0x03. Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The "Information Payload" is the 76 bytes, 82 bytes or any number of bytes of data (e.g., the PMDL Message) that the user has written into the on-chip "Receive LAPD Message" buffer (which is located at "Address" 0x11C0).

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, x16 + x12 + x5 + 1. Afterwards, this FCS value is inserted into the two-octet "FCS value" position, within the LAPD Message frame. The Receive LAPD Controller block will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see **SEE"TRANSMIT LAPD CONTROLLER BLOCK" ON PAGE 303.** on how the Transmit LAPD Controller block computes and inserts the FCS values into its "outbound" LAPD Message frames.

Operation of the Receive LAPD Controller

As mentioned earlier, the Receive LAPD Controller permits the user to receive either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

The procedure for receiving these types of LAPD Messages is presented below.

5.3.3.1 Receiving Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) configure the Receive LAPD Controller block to extract out the contents of the incoming PMDL Messages from the incoming E3 data-stream, and (2) to properly read out the contents of a newly received message (which is being stored in the "Receive LAPD Message" buffer), by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the "E3, ITU-T G.751" Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "0"; as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	InternalLOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Enable the Receive LAPD Controller Block

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive E3 LAPD Control Register" to "1" as depicted below.

Receive E3 LAPD Control Register - G.751 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	0	0

NOTES:

1. *For normal operation, it is imperative that the user also make sure that Bit 7 (Receive LAPD Any) within this register is set to "0".*
2. *Once the user executes the above-mentioned step, then the Receive LAPD Controller will begin to extract out the contents of any incoming LAPD/PMDL Message that is being transported via the "N-bit, within the incoming E3 data-stream. In most cases, the Receive LAPD Controller block will simply begin to receive the Flag Sequence octet (which is originating from the remote terminal).*

STEP 3 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming E3 data-stream, then it will assert Bit 0 (Flag Present) within the "Receive E3 LAPD Status" Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 4 - Enable the "Receive LAPD Interrupt" (Optional).

This step is optional. However, if tthis step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the "Receive LAPD Interrupt" is actually a "three-step" process.

STEP 4a - Enable the "DS3/E3 Framer" block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the "DS3/E3 Framer" block for interrupt generation; at the "Operational Block" Level.

STEP 4b - Enable the "Receive DS3/E3 Framer" block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the "Framer Block Interrupt Enable" Register, to "1", as illustrated below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the "Receive DS3/E3 Framer" block for interrupt generation, at the "Block" Level.

STEP 4c - Enable the "Receive LAPD Interrupt" at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the "Receive E3 LAPD Control" Register, as depicted below.

Receive E3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 5 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 6 - Service the Receive LAPD Interrupt

Please see **SEE"RECEIVE LAPD CONTROLLER BLOCK INTERRUPTS" ON PAGE 404.** of how to service the Receive LAPD Interrupt.

STEP 7 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive E3 LAPD Status Register" as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFC SError	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message (that is residing within the Receive LAPD Message Buffer) in an un-erred manner (e.g., there are no FCS

errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD Controller block has received this particular LAPD Message (that is residing within the Receive LAPD Message Buffer) in an erred manner.

NOTE: The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the "RxFCS Error" bit-field, prior to processing a given message.

STEP 8 - Identify the Type and Size of Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of Bits 4 and 5 (RxLAPDType[1:0]) within the Receive E3 LAPD Status Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFC SError	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	0	1	0

TABLE 47: THE RELATIONSHIP BETWEEN THE CONTENTS WITHIN BITS 4 AND 5 (RXLAPDTYPE[1:0]) AND THE TYPE OF LAPD/PMDL MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER

RXLAPDTYPE[1:0]	TYPE OF LAPD/PMDL MESSAGE RESIDING IN THE RECEIVE LAPD MESSAGE BUFFER	SIZE OF MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER(INFORMATION PAYLOAD/TOTAL MESSAGE SIZE)
00	CL Path Identification Message	76 Bytes/82 Bytes
01	Idle Signal Identification	76 Bytes/82 Bytes
10	Test Signal Identification	76 Bytes/82 Bytes
11	ITU-T Path Identification	82 Bytes/88 Bytes

Table 47 indicates that if the value within the "RxLAPDType[1:0]" bit-fields are set to "[1, 1]", then the size of total message residing within the Receive LAPD Message Buffer is 88 bytes, and that the size of the Information Payload (within these 88 bytes of data) is 82 bytes. Likewise, this table also indicates that if the value within the "RxLAPDType[1:0]" bit-fields are set to some value other than "[1, 1]", then the size of the total message residing within the Receive LAPD Message Buffer is 82 bytes, and that the size of the Information Payload (within these 82 bytes of data) is 76 bytes.

The Relationship between the Total LAPD/PMDL Message Size and Information Payload Size

The Relationship between the size of the Total (or complete) LAPD/PMDL Message, and the size of the Information Payload, is best explained by presenting (again) the byte-format of the LAPD Message 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 or Any-size Bytes of Information (Payload)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

Figure 181 indicates that the complete LAPD/PMDL Message will consist of an information payload of either 76 or 82 bytes, along with a total of six overhead bytes (consisting of the Flag Sequence, SAPI, TEI, Control and the two FCS bytes).

NOTE: *When receiving and processing the "Standard 76 or 82-byte" type of LAPD Messages, then the data residing within the Receive LAPD Message will be exactly of the byte-format, as presented above in* **Figure 181**.

Why determining the Size of the Incoming LAPD Message is important

If the Microprocessor has "learned" (based upon reading out the values of the "RxLAPDType[1:0]" bit-fields) that the size of the total LAPD/PMDL Message is 88 bytes, then the Microprocessor "knows" (when it comes time to read out the contents of the Message, residing within the Receive LAPD Message Buffer) that it can read out and process the contents of 88-bytes (out of 90 bytes) within this Buffer. Conversely, if the Microprocessor has "learned" that the size of the total LAPD/PMDL Message is only 82 bytes, then the Microprocessor must "know" that it can only read out and process the first 82 bytes of data within the Receive LAPD Message Buffer. The last eight bytes within the Buffer are simply "junk bytes" and have no value.

STEP 9 - Read out the contents of the Receive LAPD Message Buffer

The instructions that follow were written with the assumption that the user only wishes to extract out the Information Payload bytes, from the complete LAPD Message that is residing within the Receive LAPD Message Buffer. Whenever the reading out the contents of newly received PMDL/LAPD Messages from the Receive LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that is presented below.

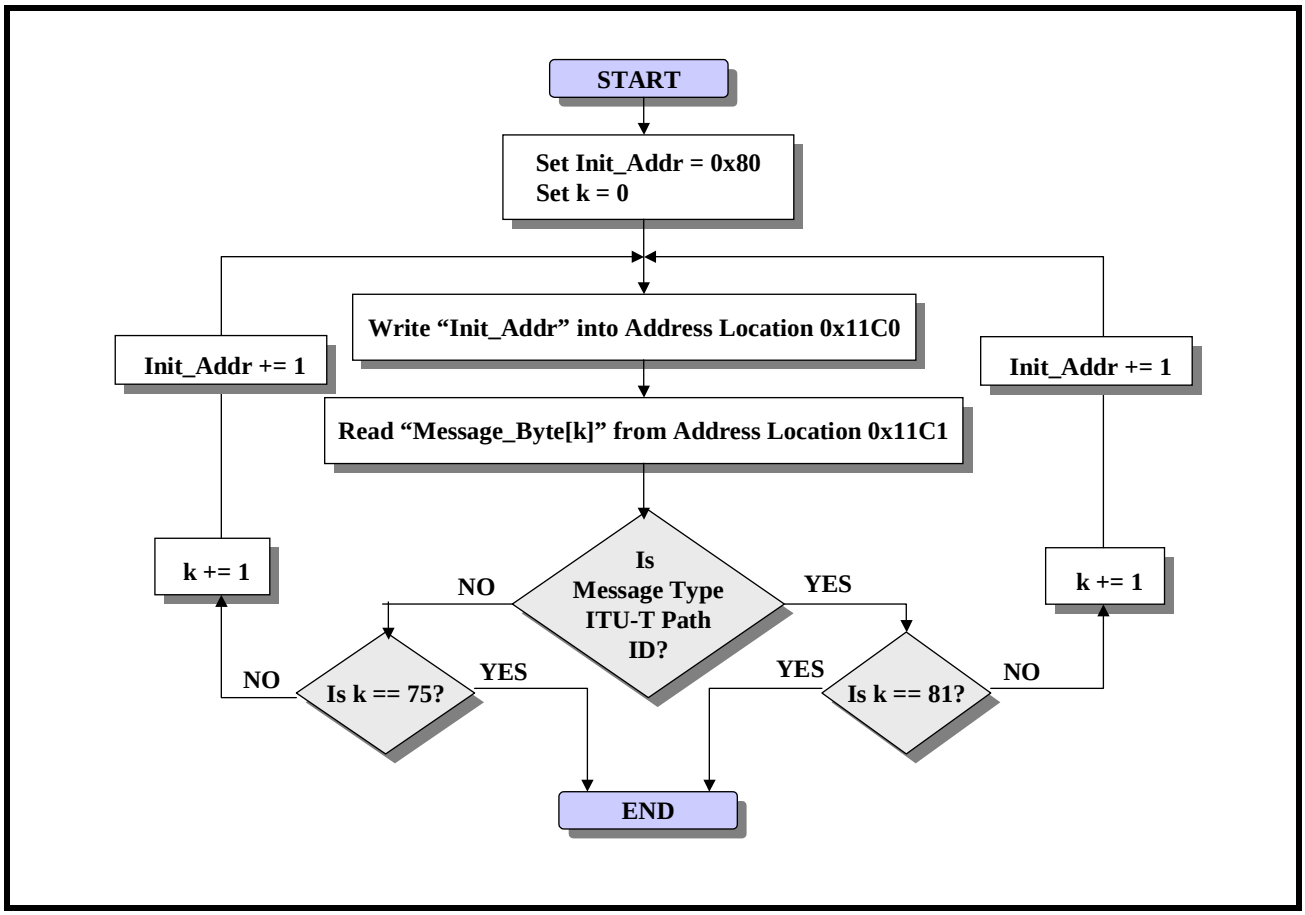
In order to begin the process of reading out the contents of the "incoming" PMDL Message, then the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Receive LAPD Message Buffer. The approach that one should use to access the Transmit LAPD Message Buffer is presented in **SEE"TRANSMITTING STANDARD-TYPE (76 OR 82 BYTE SIZE) LAPD MESSAGES" ON PAGE 306.**.

The approach to use when reading out the contents of the newly received PMDL Message from the Receive LAPD Message buffer is presented in the flow-chart below.

FIGURE 182. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE FOR READING OUT THE CONTENTS OF A NEWLY RECEIVED LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER



A NOTE ABOUT Figure 182:

The answer to the "Is Message Type ITU-T Path ID?" "Decision Diamond" (within the flow-chart) was obtained during STEP 8 (see above).

STEP 10 - Remove Header and FCS Bytes from this newly received (and read out) PMDL/LAPD Message

As mentioned above, the byte of the data, residing within the Receive LAPD Message buffer, will be as is presented within Figure 181. Therefore, if only processing the "Information Payload" portion of this LAPD/ PMDL Message, then remove these additional bytes from this block of data, before further processing.

5.3.3.2 Receiving "Non-Standard" Variable Length (e.g., up to 82 bytes) LAPD Messages)

The user can (1) enable the Receive LAPD Controller, and (2) read out a "non-Standard" incoming PMDL/ LAPD message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the "E3, ITU-T G.751" Framing Format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1"; as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Enable the Receive LAPD Controller

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive E3 LAPD Control Register" to "1" as depicted below.

Receive E3 LAPD Control Register - G.751 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	0	1	0	0

STEP 3 - Configure the Receive LAPD Controller to receive a "non-standard" size LAPD Message

This is accomplished by setting Bit 7 (RxLAPD Any) within the "Receive E3 LAPD Control" Register, to "1", as depicted below.

Receive E3 LAPD Control Register - G.751 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
1	0	0	0	0	1	0	0

STEP 4 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming E3 data-stream, then it will assert Bit 0 (Flag Present) within the "Receive E3 LAPD Status" Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFC SError	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 5 - Enable the "Receive LAPD Interrupt" (Optional).

This step is optional. However, if this step is executed, then the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the "Receive LAPD Interrupt" is actually a "three-step" process.

STEP 5a - Enable the "DS3/E3 Framer" block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the "DS3/E3 Framer" block for interrupt generation; at the "Operational Block" Level.

STEP 5b - Enable the "Receive DS3/E3 Framer" block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the "Framer Block Interrupt Enable" Register, to "1", as illustrated below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the "Receive DS3/E3 Framer" block for interrupt generation, at the "Block" Level.

STEP 5c - Enable the "Receive LAPD Interrupt" at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the "Receive E3 LAPD Control" Register, as depicted below.

Receive E3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 6 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 7 - Service the Receive LAPD Interrupt

Please see **SEE"RECEIVE LAPD CONTROLLER BLOCK INTERRUPTS" ON PAGE 404.** of how to service the Receive LAPD Interrupt.

STEP 8 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive E3 LAPD Status Register" as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFC SError	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message (that is residing within the Receive LAPD Message Buffer) in an un-erred manner (e.g., there are no FCS errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD Controller block has received this particular LAPD Message (that is residing within the Receive LAPD Message Buffer) in an erred manner.

NOTE: *The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the "RxFCS Error" bit-field, prior to processing a given message.*

STEP 9 - Determine the Size of the Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of the Receive LAPD Byte Count Register, as depicted below.

Receive LAPD Byte Count Register (Address = 0x1184)

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

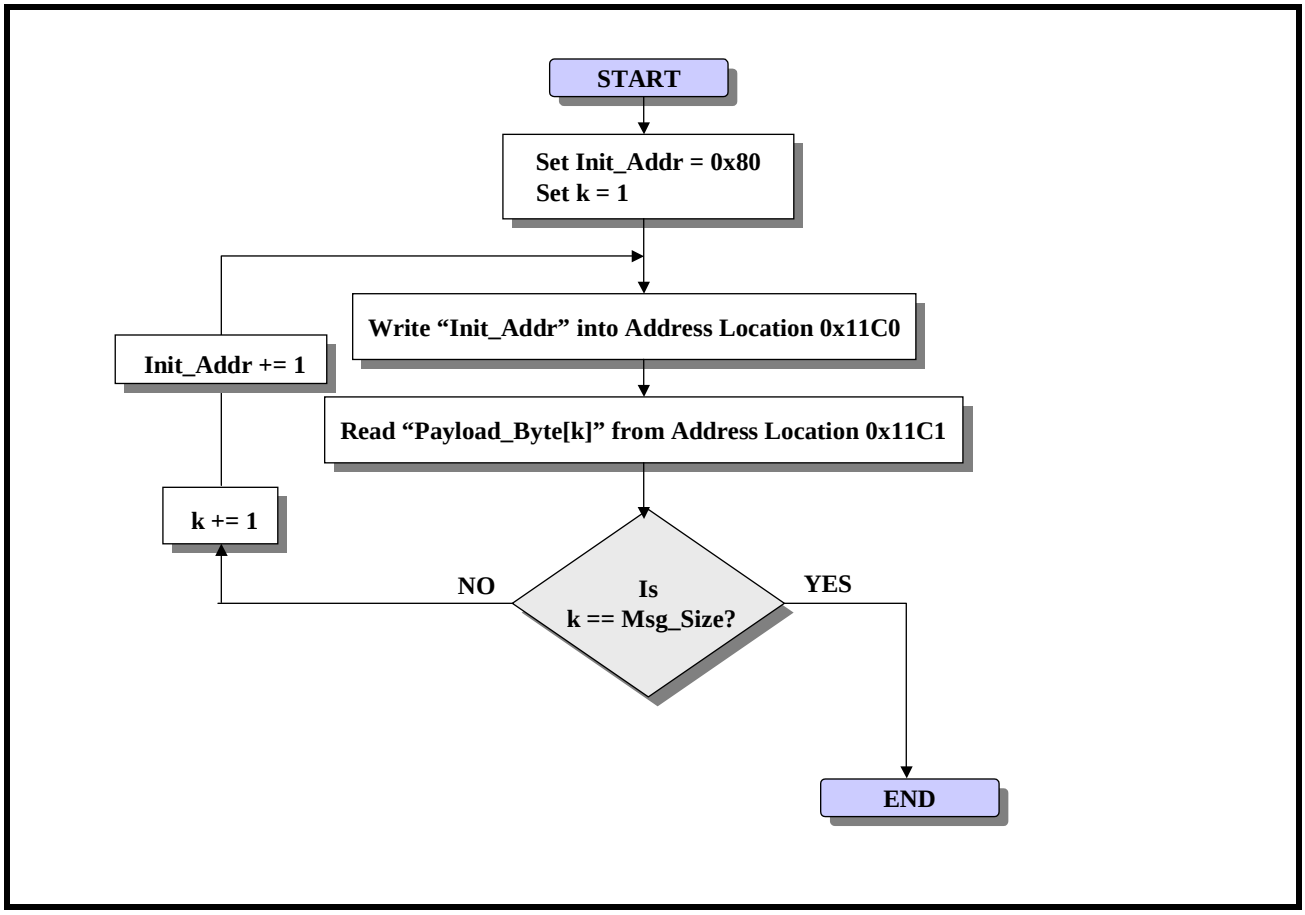
NOTES:

1. *The Receive LAPD Byte Count Register will have the size of the newly received LAPD/PMDL Message, in terms of bytes.*
2. *This register is only active if the Receive LAPD Controller is receiving a "non-standard" LAPD/PMDL Message.*

STEP 10 - Read out the contents of the Receive LAPD Message Buffer

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 183. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO READING OUT THE CONTENTS OF THE NEWLY RECEIVE LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER.



A NOTE ABOUT Figure 183:

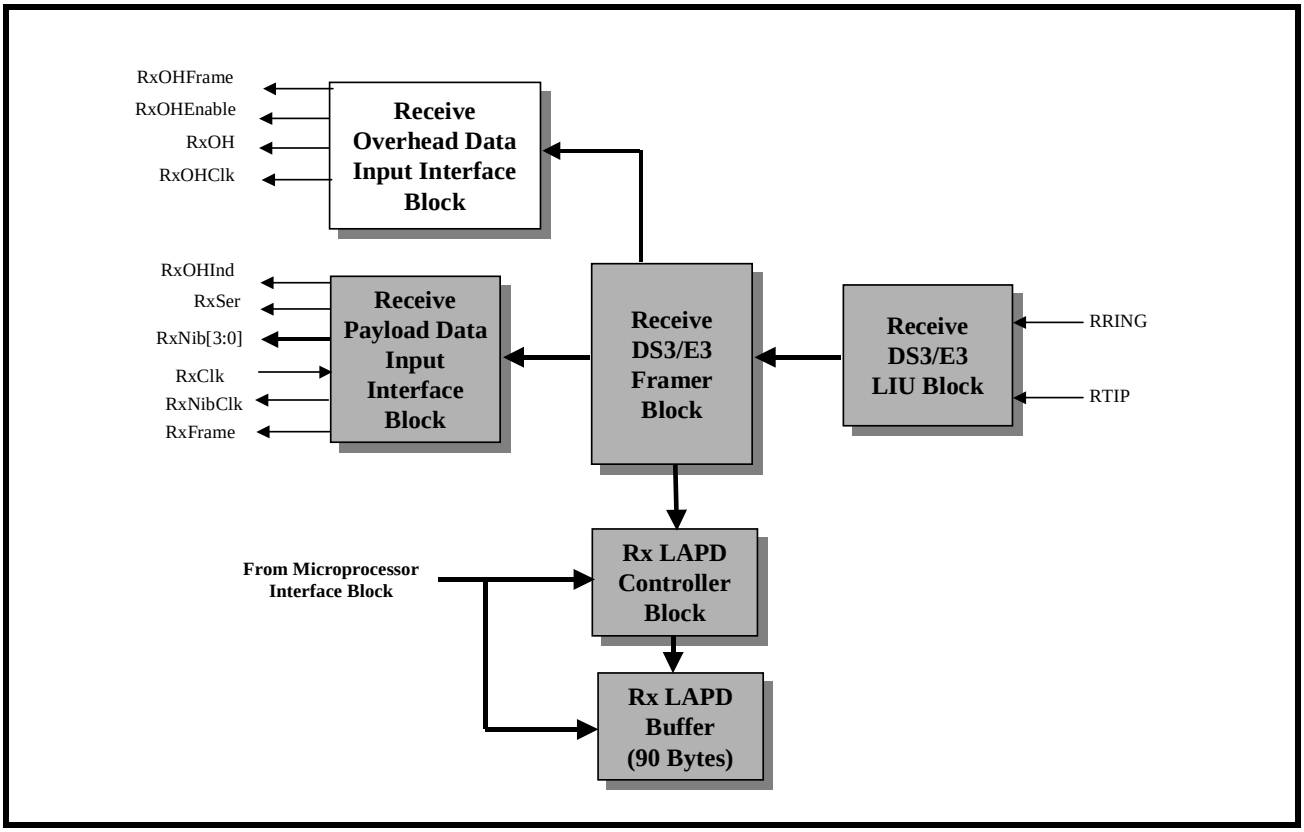
The value of the parameter "Msg_Size" within the "Is k=Msg_Sizes?" Decision Diamond (within the flow-chart) was obtained during STEP 9 (see above).

5.3.3.3 Receive LAPD Controller Block Interrupts

5.3.4 THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

The Receive Overhead Data Output Interface block is the sixth functional block (within the Receive Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. **Figure 184** presents an illustration of the "Receive Direction" circuitry, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the Receive Overhead Data Output Interface block highlighted.

FIGURE 184. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK HIGHLIGHTED)



Background Information

In order to fully understand the role of the Receive Overhead Data Output Interface, some background information needs to be discussed first.

As mentioned in **SEE"DESCRIPTION OF THE E3, ITU-T G.751 FRAME STRUCTURE AND THE OVERHEAD BITS" ON PAGE 259.**, 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 XRT79L71 has been designed to handle and process both the payload type and overhead type of bits/bytes for each E3 frame. Within the XRT79L71, the Receive Payload Data Output Interface Block (which is discussed in considerable detail in **SEE"THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK" ON PAGE 411.**) has been designed to output the payload data (that has been extracted from the incoming E3 data-stream) to the "System-Side" Terminal Equipment. 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 E3 data-stream. The XRT79L71 does not offer the user a means to shut off this transmission of overhead data. However, the Receive Overhead Data Output Interface block does not provide the user with the appropriate output signals for an external Data-Link Layer (or System-Side Terminal Equipment) to sample and process these overhead bits.

In order to accomplish this, the Receive Overhead Data Output Interface block has numerous output pins. Table 48 presents a list and a brief definition of each of these pins.

TABLE 48: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOH	C7	O	**Receive Overhead Data Output Interface block - Data Output pin:** The exact approach that one should sample this output pin depends upon whether one is using "Method 1" (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 1 - THE "RXOHCLK" METHOD" ON PAGE 408., below) or "Method 2" (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/ RXOHENABLE" METHOD" ON PAGE 409., below) for extracting data from the "Receive Overhead Data Output Interface block, as described below. **If Method 1 is used:** The XRT79L71 outputs the overhead bits, within the incoming E3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will output a given bit, upon the falling edge of RxOHClk. Hence, the "System-Side Terminal Equipment" should be designed (or configured) to sample the data, at this pin, upon the rising edge of RxOHClk. **If Method 2 is used:** The XRT79L71 outputs the overhead bits, within the incoming E3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will assert the "RxOHEnable" output pin (for one "RxClk" period) whenever the data, residing on the "RxOH" output pin has become stable and is safe for "sampling". In this case, the user should design (or configure) the System-Side Terminal Equipment to sample and latch the "RxOH" data upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high". The XRT79L71 will always output the E3 overhead bits via this output pin. There are no external input pins or register bits settings available that will disable this output pin.
RxOHClk	A7	O	**Receive Overhead Data Output Interface block - Clock Output pin:** This particular output pin is only used if the Method 1 is employed to extract overhead data from the "Receive Overhead Data Output Interface" port (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 1 - THE "RXOHCLK" METHOD" ON PAGE 408., below). The XRT79L71 will output the overhead bits (within the incoming E3 data-stream), via the "RxOH" output pin, upon the falling edge of this particular clock signal. As a consequence, the "System-Side Terminal Equipment" should be designed (or configured) to sample the data, at this pin, upon the rising edge of RxOHClk.NOTE: This output clock signal is always active.

TABLE 48: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOHFrame	A8	O	**Receive Overhead Data Output Interface block - Start of Frame Indicator Output pin:** The exact approach that one should sample this output pin depends upon whether one is using "Method 1" (see **SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 1 - THE "RXOHCLK" METHOD" ON PAGE 408.**, below) or "Method 2" (see **SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/RXOHENABLE" METHOD" ON PAGE 409.**, below) for extracting data from the "Receive Overhead Data Output Interface block, as described below. **If Method 1 is used:T** he XRT79L71 will pulse this output pin "high" (for one period of RxOHClk) coincident to whenever the Receive Overhead Data Output Interface block outputs the very first overhead bit of the most recently received E3 frame. This output pin will be "low" at all other times. The Receive Overhead Data Output Interface block will update this output pin, upon the falling edge of "RxOHClk". Hence, the System-Side Terminal Equipment should be designed (or configured) to sample the data, at this pin, upon the rising edge of "RxOHClk". **If Method 2 is used:** The Receive Overhead Data Output Interface block will assert the "RxOHEnable" output pin (for one "RxClk" period) whenever the data, residing on the "RxOH" output pin has become stable and is safe for "sampling". In this case, the user should design (or configure) the System-Side Terminal Equipment to sample and latch both the "RxOH" and the "RxOHFrame" output pins upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high".
RxOHEnable	B7	O	**Receive Overhead Data - Enable Output Pin:** This particular output pin is only used if Method 2 is employed (see **SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/RXOHENABLE" METHOD" ON PAGE 409.**, below). The Receive Overhead Data Output Interface will assert this signal (e.g., pulse it "high") for one "RxClk" period, coincident to whenever it is safe for the System-Side Terminal Equipment to sample the overhead bit that is being output via the "RxOH" output pin. This output pin will remain "low" at all other times. If the "Method 2" is to be used, then design (or configure) the System-Side Terminal Equipment such that it will sample and latch the "RxOH" output (from the XRT79L71), upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high".

The Two Methods of Extracting out Overhead Data from the Receive Overhead Data Output Interface Block

There are two methods that can be used to extract the overhead data from the Receive Overhead Data Output Interface block. One "Method" is referred to as "Method 1" or the "RxOHClk" Method, and the other "method" is referred to as "Method 2" or the "RxClk/RxOHEnable" Method. Each of these methods is described in considerable detail below.

5.3.4.1 Operating the Receive Overhead Data Output Interface block using Method 1 - The "RxOHClk" Method

This particular method is referred to as the "RxOHClk" method for the following reasons.

a. The System-Side Terminal Equipment will use the RxOHClk clock output signal (from the Receive Overhead Data Output Interface block) to sample and latch the "overhead" data via the "RxOH" output pin.

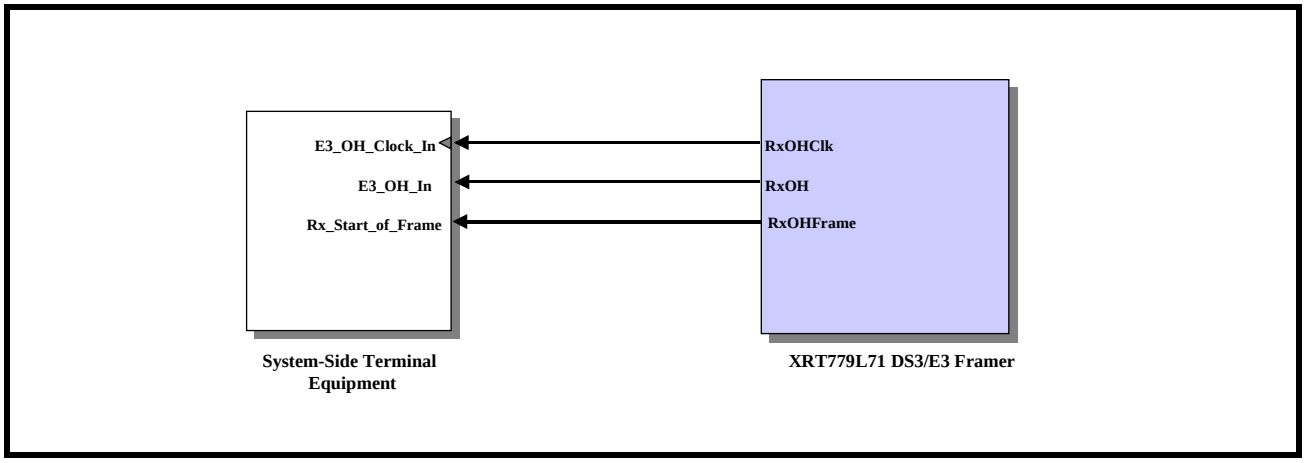
b. b.The "Receive Overhead Data Output Interface" block will update the data (via the RxOH output pin) upon the falling edge of the "RxOHClk" clock output signal.

If "Method 1" is used the "System-Side Terminal Equipment" will need to interface to the following "Receive Overhead Data Output Interface" pins.

- RxOH
- RxOHClk
- RxOHFrame

If "Method 1" is used, the users system must be designed such that the "System-Side" Terminal Equipment will be interfaced to the "Receive Overhead Data Output Interface" block in the manner as presented below in Figure 185.

FIGURE 185. ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK WHEN USING "METHOD 1"



Method 1 Operation of the Receive Overhead Data Output Interface Block

If the Receive Overhead Data Output Interface block is to be operated, then design/configure the "System-Side" Terminal Equipment to continuously execute the following tasks.

TASK # 1: The "System-Side" Terminal Equipment must sample the state of the "RxOHFrame" output pin (from the XRT79L71) upon the rising edge of the "RxOHClk" clock signal (which is also output from the XRT79L71). Whenever the "System-Side Terminal Equipment" samples the "RxOHFrame" output pin "high", then it will "know" that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit (within the most recently received E3 frame) via the "RxOH" output pin.

TASK # 2: As the "System-Side Terminal Equipment' samples the "RxOHFrame" output signal, it must also keep track of the number of rising edges (within the RxOHClk signal) that have occurred since the last time "RxOHFrame" was sampled "high". By doing this, the "System-Side" Terminal Equipment will be able to keep track of which overhead bit is being output via the "RxOH" output pin, at any given "RxOHClk" period. When the System-Side Terminal Equipment "knows" which overhead bit is being output (via the RxOH output pin) during a particular RxOHClk clock period, then it can "make sense" of these overhead bits and process these overhead bits as appropriate.

Table 49 relates the number of rising clock edges, within the "RxOHClk" output signal, since the "RxOHFrame" output signal was sampled "high" to the DS3 Overhead Bit being output via the RxOH output pin (of the Receive Overhead Data Output Interface block), for Method 1.

TABLE 49: THE RELATIONSHIP BETWEEN 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 PROCESSED BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN RXOHCLK SINCE RXOHFRAME BEING SAMPLED "HIGH	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
0 (RxOHClk, Clock Edge is coincident with the "RxO-HFrame" signal being sampled "high")	FAS, Bit 1 (MSB)
1	FAS, Bit 2
2	FAS, Bit 3
3	FAS, Bit 4
4	FAS, Bit 5
5	FAS, Bit 6
6	FAS, Bit 7
7	FAS, Bit 8
8	FAS, Bit 9
9	FAS, Bit 10 (LSB)
10	A Bit
11	N Bit
12	BIP-4, Bit 1 (MSB)*
13	BIP-4, Bit 2*
14	BIP-4, Bit 3*
15	BIP-4, Bit 4 (LSB)*

NOTE: *The asterisk (*) indicates that these particular overhead bits will only be processed if BIP-4 processing is enabled, within the Receive E3 Framer Block.*

5.3.4.2 Operating the Receive Overhead Data Output Interface block using Method 2 - The "RxClk/RxOHEnable" Method

This particular method is referred to as the "RxClk/RxOHEnable" method for the following reasons.

a. The System-Side Terminal Equipment will use the "RxOHEnable" output pin (form the Receive Overhead Data Output Interface block) to keep track of which overhead bit is being processed by the Receive Overhead Data Output Interface (via the RxOH output pin) as any given time.

b. The System-Side Terminal Equipment must use the falling edge of "RxClk" in order to sample and latch the data residing on the "RxOH" output pin.

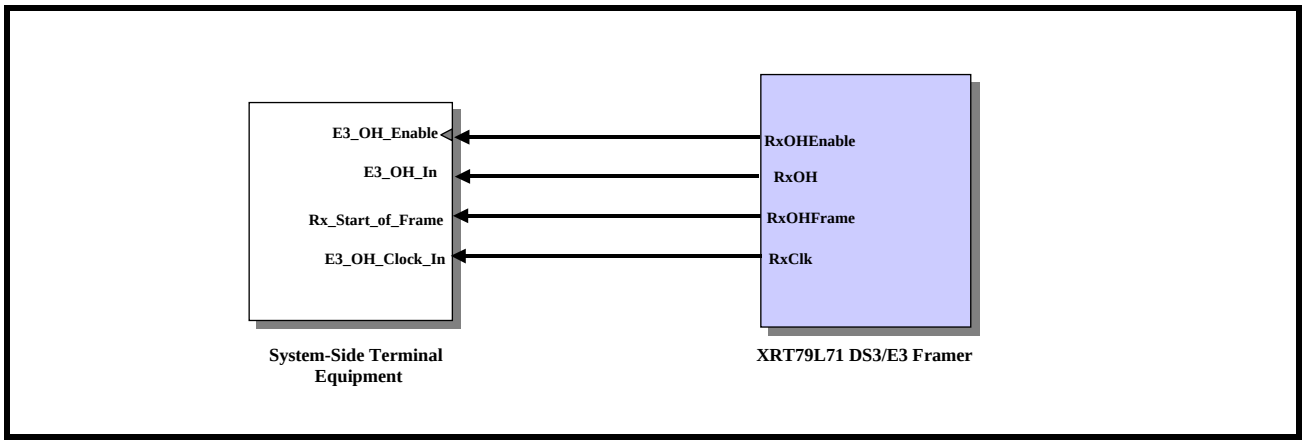
If "Method 2" is used, the "System-Side" Terminal Equipment will need to interface to the following "Receive Overhead Data Output Interface" pins.

- RxOH
- RxOHFrame

- RxOHEnable
- RxClk

If "Method 2" is used, the users system must be designed such that the "System-Side" Terminal Equipment will be interfaced to the "Receive Overhead Data Output Interface" block, in a manner as presented below in Figure 186.

FIGURE 186. ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK WHEN USING "METHOD 2"



Method 2 Operation of the Receive Overhead Data Output Interface Block

If the Receive Overhead Data Output Interface block is to be operated per Method 2", then design/configure the "System-Side" Terminal Equipment to continuously execute the following tasks.

TASK # 1: The "System-Side" Terminal Equipment must sample the states of the "RxOHFrame", "RxOH" and the "RxOHEnable" output pins (from the XRT79L71) upon the falling edge of "RxClk" clock output signal. The XRT79L71 will pulse the "RxOHEnable" output pin "high" for one "RxClk" period, coincident to whenever it is "safe" (or OK) to sample the states of the "RxOH" and the "RxOHFrame" output pins. If the System-Side Terminal Equipment samples both the "RxOHEnable" and the "RxOHFrame" output pin "high", then it "knows" that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit (within the most recently received E3 frame) via the "RxOH" output pin.

TASK # 2: As the "System-Side" Terminal Equipment samples the "RxOHEnable" and "RxOHFrame" output signals, it must also keep track of the number of times that the "RxOHEnable" output pin has been sampled "high" since the last time that both the "RxOHEnable" and "RxOHFrame" output pins have been sampled "high". By doing this, the "System-Side" Terminal Equipment will be able to keep track of which overhead bits are being output via the "RxOH" output at any time. This will permit the "System-Side Terminal Equipment to properly handle these overhead bits as appropriate.

Table 50 relates the number of "RxOHEnable" output pulses that have occurred since both the "RxOHFrame" and the "RxOHEnable" output pins were sampled "high", to the E3 Overhead Bit that is being output via the "RxOH" output pin. The user can use this table as a guide for extracting the appropriate overhead bits, via the "Receive Overhead Data Output Interface" block, whenever "Method 2" is used.

TABLE 50: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE "RXOHENABLE" SIGNAL, SINCE THE RXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING OUTPUT (VIA THE RXOH OUTPUT PIN) BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF PULSES IN RXOHENABLE SINCE RXOHFRAME BEING SAMPLED "HIGH	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
0 (RxOHEnable and RxOHFrame are sampled "high" simultaneously)	**FAS, BIT 1 (MSB)**
1	**FAS, BIT 2**
2	**FAS, BIT 3**
3	**FAS, BIT 4**
4	**FAS, BIT 5**
5	**FAS, BIT 6**
6	**FAS, BIT 7**
7	**FAS, BIT 8**
8	**FAS, BIT 9**
9	**FA2, BIT 10 (LSB)**
10	**A BIT**
11	**N BIT**
12	**BIP-4, BIT 1 (MSB)***
13	**BIP-4, BIT 2***
14	**BIP-4, BIT 3***
15	**BIP-4, BIT 4 (LSB)***

NOTE: *The asterisk (*) indicates that these particular overhead bits will only be processed if BIP-4 processing is enabled, within the Receive E3 Framer block.*

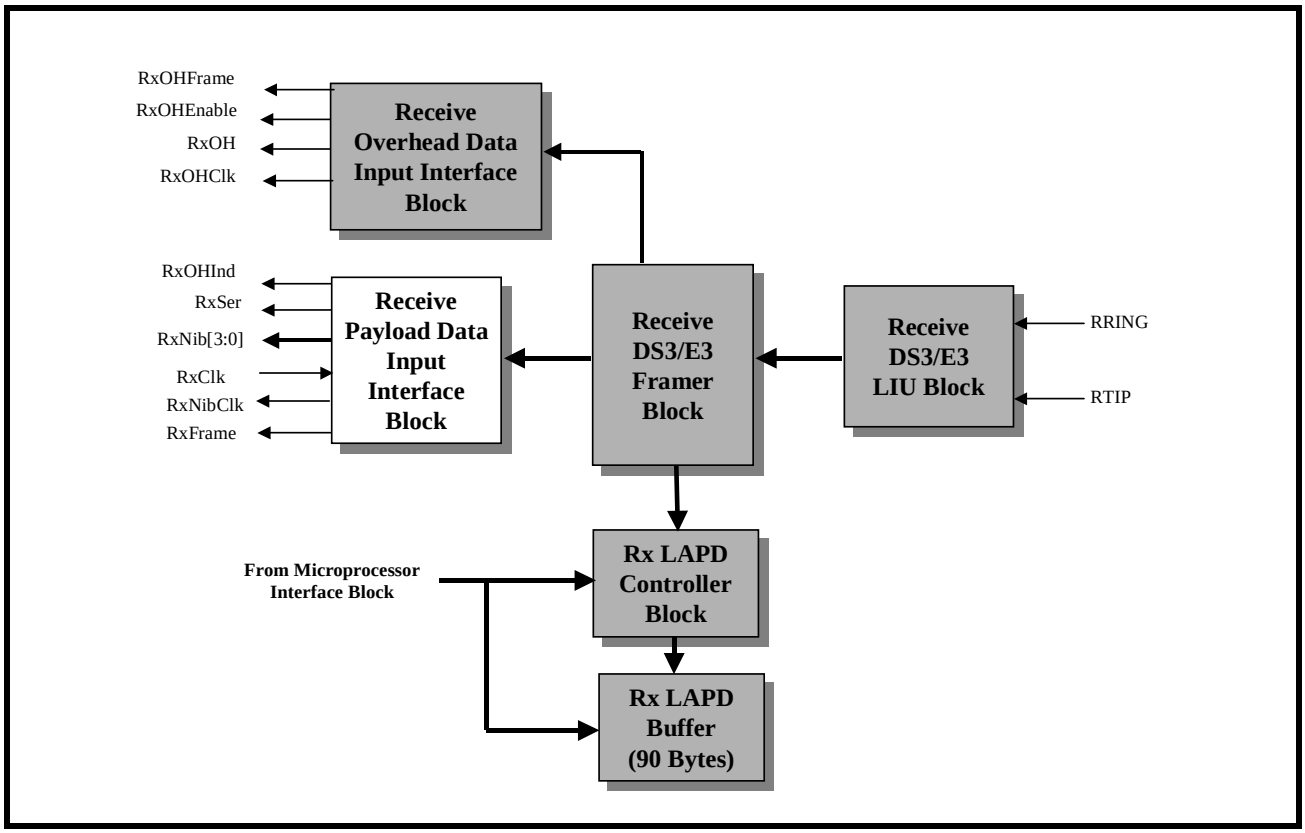
Figure 187 presents an illustration of the behavior of the "RxOH", "RxOHFrame", "RxOHEnable" and "RxClk" output signals, whenever "Method 2" is used.

Figure 187. An illustration of the behavior of Receive Overhead Data Output Interface block signals, whenever the "Method 2" approach to Data Extraction is used

5.3.5 THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

The Receive Payload Data Output Interface block is the seventh (and final) functional block (within the Receive Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.751 Clear-Channel Framer Applications. Figure 188 presents an illustration of the "Receie Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.751 Clear-Channel Framer Mode, with the "Receive Payload Data Output Interface" block highlighted.

FIGURE 188. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK HIGHLIGHTED).



The purpose of the "Receive Payload Data Output Interface" block is to output payload data that has been extracted from the incoming E3 data-stream that has been received and processed by the "Receive Direction" circuitry within the XRT79L71.

In order to accomplish this, the Receive Payload Data Output Interface block has numerous output pins. **Table 51** presents a list and a brief definition of each of these pins.

TABLE 51: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxSer	A5	O	**Receive Serial Payload Data Output pin:** If the Receive Payload Data Output Interface block is operated in the "Serial" Mode, then the XRT79L71 will output the payload data (that has been extracted from the incoming E3 data-stream), via this output pin. The XRT79L71 will update the data (on this pin) upon the rising edge of the RxCLK output clock signal. The user is advised to design (or configure) the System-Side Terminal Equipment such that it will sample the data that is output via this output pin, upon the falling edge of RxCLK. ***NOTES:*** *1. This signal is only active if the "NibIntf" input pin is pulled "low".* *2. In reality, for Serial Mode operation, the entire incoming E3 data-stream (payload bits and overhead bits) will be output via the "RxSer" output pin. The user will need to use the "RxOHInd/RxGapClk" signals in order to distinguish the "payload bits" from the "overhead bits" as they are output via the "RxSer" output pin.*
RxNib[3:0]	B4 A4 D6 C5	O	**Receive Nibble-Parallel Payload Data Output Pin:** If the Receive Payload Data Output Interface block is operated in the "Nibble-Parallel" Mode, then the XRT79L71 will output the payload data (that has been extracted from the incoming E3 data-stream), via these output pins, in a "Nibble-Parallel" manner. The XRT79L71 will update the data (via these four output pins) upon the falling edge of the RxCLK output signal. The user is advised to design (or configure) the System-Side Terminal Equipment such that it will sample this data upon the rising edge of RxCLK. ***NOTE:*** *These pins are only active if the "NibIntf" input pin is pulled "high".*

TABLE 51: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxCLK/ RxNibClk	A6	O	**Receive Payload Data Output Interface -Clock Output Pin:** The exact behavior of this signal depends upon whether the XRT79L71 has been configured to operate in the "Serial Mode" or in the "Nibble-Parallel Mode", as described below. **Serial Mode Operation - RxCLK** If the Receive Payload Data Output Interface block has been configured to operate in the Serial Mode, then this signal will be a 34.368MHz clock output signal. The Receive Payload Data Output Interface block will update the data (via the RxSer output pin) upon the rising edge of this clock signal. For "Serial Mode" operation, the user is advised to design (or configure) the System-Side Terminal Equipment to sample the data on the "RxSer" output pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation - RxNibClk** If the Receive Payload Data Output Interface block has been configured to operate in the Nibble-Parallel Mode, then the XRT79L71 will pulse this output pin 384 times for each inbound E3 frame. The Receive Payload Data Output Interface block will update the data (via the "RxNib[3:0]" output pins upon the falling edge of this clock signal. For "Nibble-Parallel Mode" operation, the user is advised to design (or configure) the System-Side Terminal Equipment to sample the data on the "RxNib[3:0]" output pins, upon the rising edge of this clock signal. ***NOTE:*** *This output clock signal is ultimately derived from the Recovered Clock signal (via the Receive DS3/E3 LIU Block).*

TABLE 51: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOHInd/ RxGapClk	C6	O	**Receive Overhead Bit Indicator Output/Receive Gap-Clock Output:** The function of this output pin depends upon whether or not the XRT79L71 has been configured to operate in the Gapped-Clock Mode. **Non-Gapped Clock Mode - RxOHInd:** This output pin will pulse "high", for one "RxClk" period, coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the "RxSer" output pin. This output pin will be held "low" at all other times. The purpose of this output pin is to alert the System-Side Terminal Equipment that the current bit (e.g., the one that is currently residing on the RxSer output pin) is an overhead bit and should not be processed by the System-Side Terminal Equipment. The XRT79L71 will update output signal upon the rising edge of RxClk. Therefore, the user is advised to design (or configure) the System-Side 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 active (in the "RxOHInd" role) if the Receive Payload Data Output Interface block has been configured to operate in either the Serial or the Nibble-Parallel Mode. This output pin will be held "low" at all times. This is in contrast for DS3 applications, in which is pin is only active for "Serial" Mode operation.* **Gapped Clock Mode - RxGapClk:** This particular output pin will function as a "payload bit" output clock signal. In other words, in this mode, the Receive Payload Output Interface block will only generate a clock pulse (via this output pin) coincident to whenever it outputs a payload bit via the RxSer output pin. The Receive Payload Data Output Interface block will NOT generate a clock edge (via this output pin) coincident to whenever it outputs an overhead bit via the "RxSer" output pin. As a consequence, there will be "gaps" within this particular clock output signal (hence the name "Gapped Clock Mode"). If the XRT79L71is configured to operate in the Gapped Clock Mode, then the user must design or configure the System-Side Terminal Equipment to sample and latch the "RxSer" data upon the falling edge of the "RxOHInd/RxGapClk" clock signal.
RxFrame	B6	O	**Receive Payload Data Output Interface - Receive Start of Frame Output Indicator:** The behavior of this output pin depends upon whether the XRT79L71 has been configured to operate in the "Serial" or in the "Nibble-Parallel" Mode. **Serial Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "high" (for one RxCLK period) coincident to whenever it outputs the very first bit of a new E3 frame via the "RxSer" output pin. This output pin will remain "low" at all other times. **Nibble-Parallel Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "high" (for one RxCLK or "Nibble-Period") coincident to whenever it outputs the very first nibble of a new E3 frame via the "RxNib[3:0]" output pins. This output pin will remain "low" at all other times.

Operation of the Receive Payload Data Output Interface Block

The Receive Payload Data Output Interface block permits the user to configure it to operate in either of the following modes.

- The "Serial" Mode
- The "Nibble-Parallel" Mode

5.3.5.1 Serial Mode Operation of the Receive Payload Data Output Interface

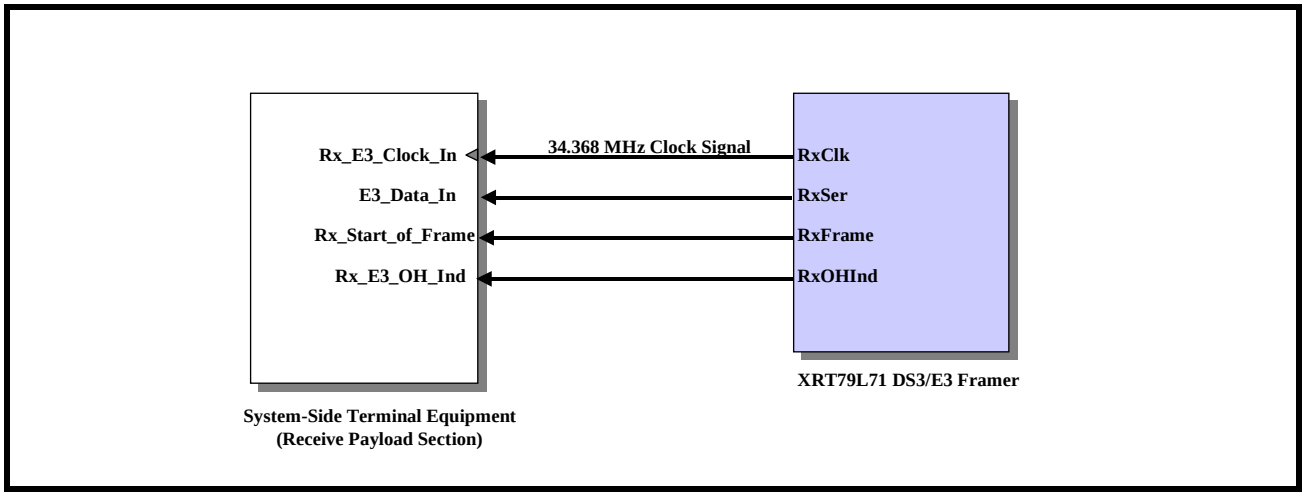
If the Receive Payload Data Output Interface block is configured to operate in the "Serial Mode" it can be further configured to operate in either the "Non-Gapped Clock" Mode or in the "Gapped Clock" Mode.

5.3.5.1.1 Operating the Receive Payload Data Output Interface Block in the "Non-Gapped Clock" Mode

If the Receive Payload Data Output Interface block has been configured to operate in the "Serial/Non-Gapped-Clock" Mode, then all of the following is true.

- The XRT79L71 will output the entire contents of the incoming E3 data-stream (consisting of both payload and overhead bits) via the RxSer output pin, upon the rising edge of the RxCLK signal (which is a 34.368MHz clock signal).
- The user will need to rely on the "RxOHInd/RxGapClk" output pin in order to distinguish a payload bit from an overhead bit, within the data that is output via the "RxSer" output pin.
- The XRT79L71 will pulse the "RxFrame" output pin "high" for one "RxCLK" period, coincident to whenever it outputs the very first bit within a given E3 frame via the "RxSer" output pin. The "RxFrame" output pin will be held "low" at all other times.

FIGURE 189. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR SERIAL MODE OPERATION



Whenever the XRT79L71 has been configured to operate in this mode, then the Receive Payload Data Output Interface block will function as the source of the 34.368MHz clock signal (via the RxCLK output signal). This clock signal is used as the "System-Side Terminal Equipment" clock source by both the "Receive Payload Data Output Interface block (of the XRT79L71) and the "System-Side Terminal Equipment" device or circuitry.

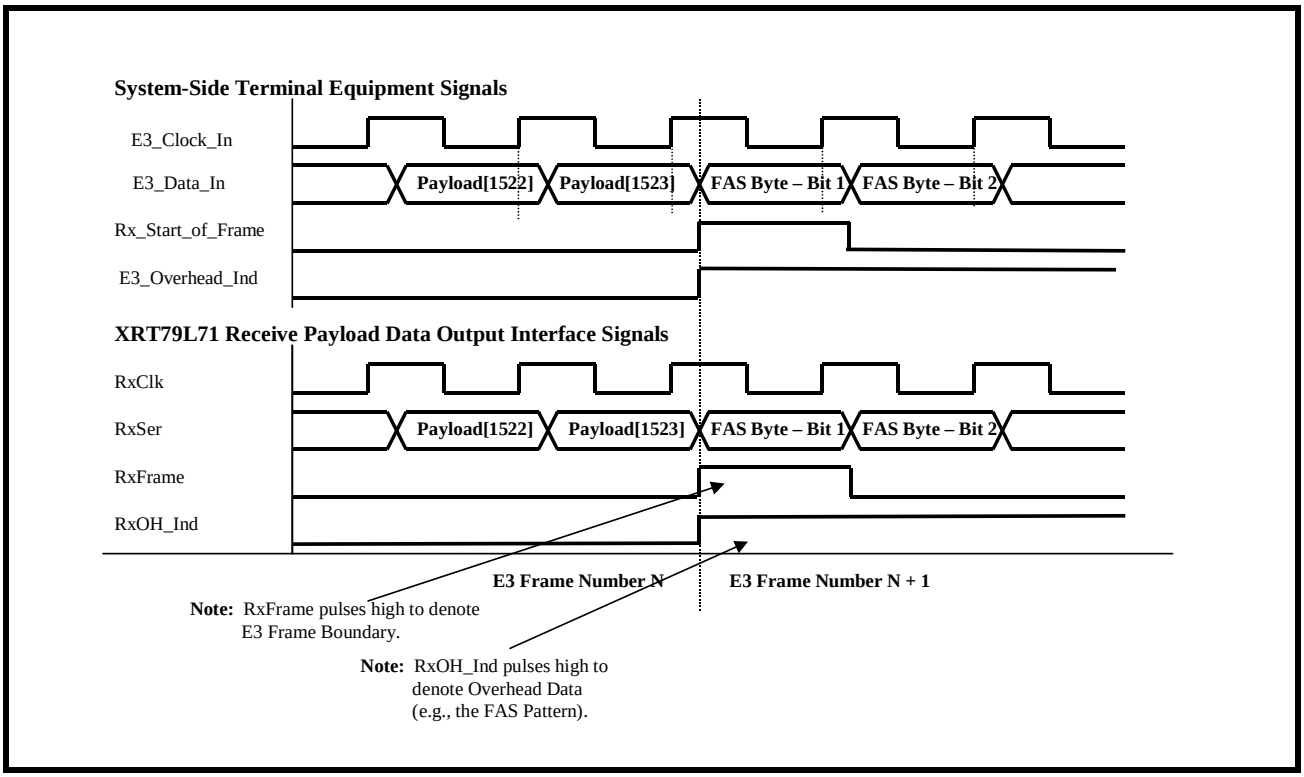
The Receive Payload Data Output Interface block will serially output the entire contents of the incoming E3 data-stream via the "RxSer" output pin. As mentioned earlier, the Receive Payload Data Output Interface block will output this data upon the rising edge of the RxCLK signal. As a consequence, the user is advised to design (or configure) the "System-Side Terminal Equipment" circuitry to sample and latch this data (via the "E3_Data_In" input pin) upon the falling edge of RxCLK (Rx_E3_Clock_In), as depicted below in **Figure 190**.

The Receive Payload Data Output Interface block (within the XRT79L71) will indicate that it is processing the very first bit of a given E3 frame by pulsing the "RxFrame" output pin "HIGH" for one bit-period. The "RxFrame" output pin will be held "LOW" at all other times.

Finally, the Receive Payload Data Output Interface block (within the XRT79L71) permits the System-Side Terminal Equipment to identify a given bit (that is being output via the "RxSer" output pin) as either an "overhead" or a "payload" bit by pulsing the "RxOH_Ind" output pin "HIGH" (for one bit-period) coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the "RxSer" output pin. Conversely, the Receive Payload Data Output Interface block will hold the "RxOH_Ind" output pin "LOW" coincident to whenever the Receive Payload Data Output Interface block outputs a payload bit via the "RxSer" output pin.

NOTE: *Since the E3, ITU-T G.751 framing format consists of 12 or 16 consecutive overhead bits, whenever the "RxOH_Ind" output pin pulses "high" (in order to denote an overhead bit), it will typically pulse high for 12 or 16 consecutive RxCLK periods*

FIGURE 190. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR "SERIAL MODE" OPERATION



Configuring the XRT79L71 to operate in Serial Mode.

The user can configure the XRT79L71 to operate in the Serial Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in Figure 189.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "LOW".

NOTE: *This step also configures the "Transmit Payload Data Input Interface" block to operate in the "Serial Mode".*

Operating the Receive Payload Data Output Interface in the "Non-Gapped Clock" Mode

If the Receive Payload Data Output Interface block (within the XRT79L71) has been configured to operate in the "Serial" Mode, then we have recommended that the user design or configure their "System-Side Terminal Equipment" to do the following, when receiving/accepting E3 data via the "RxSer" output pin.

- Check the state of the "RxOH_Ind" output pin (from the XRT79L71) upon the falling edge of the "RxClk" signal.
- Perform either of the following actions, depending upon the sampled state of the "RxOH_Ind" output pin, as described below.

If RxOH_Ind is sampled "LOW"

Then the "System-Side Terminal Equipment" should accept this particular data bit (that is being sampled from the RxSer output pin) and treat it as a payload bit.

If RxOH_Ind is sampled "HIGH"

Then the "System-Side Terminal Equipment" should NOT accept this particular data bit (that is being sampled from the RxSer output pin) and should definitely NOT treat it as a payload bit.

In this particular approach, the user must "gate" the acceptance of a particular data bit (being sampled via the RxSer output pin) based upon the corresponding sampled state of the "RxOH_Ind" output pin. While implementing such a design into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Receive Payload Data Output Interface block to operate in the "Gapped-Clock" Mode.

5.3.5.1.2 Operating the Receive Payload Data Output Interface block has been configured to operate in the "Gapped-Clock" Mode

As mentioned above, in order to simplify the task of interfacing the Receive Payload Data Output Interface block to certain devices, the XRT79L71 permits the user to configure the Transmit Payload Data Input Interface and the Receive Payload Data Output Interface blocks to operate in the "Gapped-Clock" Mode. If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then the role of the "RxOH_Ind" output pin will change from being the "Overhead Indicator" output pin, to now being a "payload data clock" output pin. In other words, If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then it (the Receive Payload Data Output Interface block) will only generate a clock pulse (via the "RxOH_Ind" output pin) coincident to whenever it outputs a payload bit via the "RxSer" output pin. Whenever the Receive Payload Data Output Interface block outputs an overhead bit (via the "RxSer" output pin) then it will NOT generate a clock pulse via the "RxOH_Ind" output pin. This action will result in the Receive Payload Data Output Interface block generating a "gapped" clock signal via the "RxOH_Ind" output pin (hence the term, "Gapped-Clock" Mode).

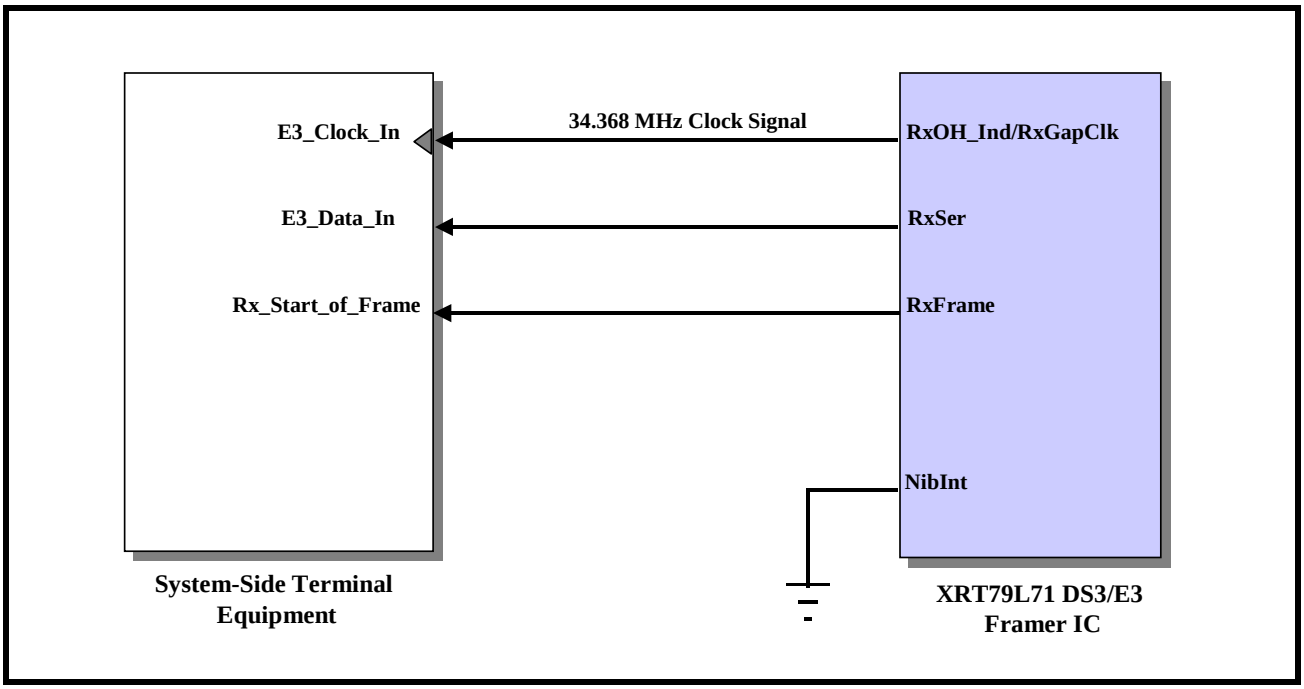
If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then the "System-Side" Terminal Equipment will be expected to sample the data (that is being output via the RxSer output pin) upon the rising edge of RxClk. In this case, there is no need to "check the state" of the certain output pin, then "gate" the acceptance/treatment of the next bit (output via the RxSer output pin) based upon the state of this particular output pin. The System-Side Terminal Equipment only needs to sample the "RxSer" output signal upon the rising edge of this particular "Gapped-Clock" signal.

Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode

To configure the Receive Payload Interface block to operate in the "Gapped-Clock" Mode, do all of the following.

STEP 1 - Interface the "System-Side" Terminal Equipment to the Receive Payload Data Input Interface block, in a manner as indicated below.

FIGURE 191. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE" TERMINAL EQUIPMENT TO THE "RECEIVE PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "GAPPED-CLOCK" MODE OPERATIONS



STEP 2 - Set Bit 6 (RxGapped Clock Mode Enable), within the "Framer Test Register" to "1" as depicted below.

Framer Test Register (Address = 0x110C)

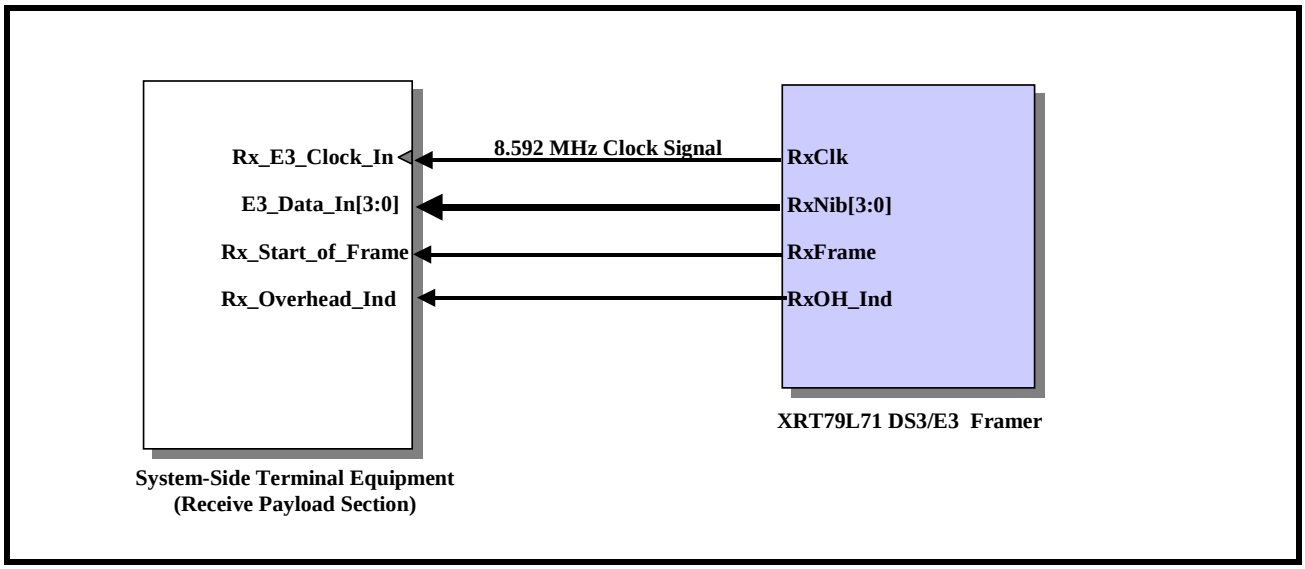
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	1	0	0	0	0	0	0

5.3.5.2 Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface

If XRT79L71 is configured to operate in the "Nibble-Parallel" Mode, then all of the following is true.

- The XRT79L71 will output the contents of both the payload bits and the overhead bits (within the incoming E3 data-stream) via the "RxNib[3:0]" output pins, upon the falling edge of the RxCLK signal.
- The XRT79L71 will pulse the "RxFrame" output pin "high" for one "RxCLK" (or Nibble) Period coincident to whenever it outputs the very first nibble within a given E3 frame via the "RxNib[3:0]" output pins. The "RxFrame" output pin will be held "low" at all other times.

FIGURE 192. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "NIBBLE-PARALLEL MODE" OPERATION



Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface Block

Whenever the XRT79L71 has been configured to operate in the "Nibble-Parallel" Mode, then the Receive Payload Data Output Interface block will function as the source of a Nibble Clock signal (via the "RxCLK" output signal).

The Receive Payload Data Output Interface block will output all of the payload and overhead data (that has been extracted out to the incoming E3 data-stream) in a "Nibble-Parallel" Mode via the "RxNib[3:0] output pins. As mentioned earlier, the Receive Payload Data Output Interface block will output this data upon the falling edge of the "RxCLK" signal. As a consequence, the user is advised to design (or configure) the "System-Side Terminal Equipment" circuitry to sample and latch this data (via the "E3_Data_In[3:0]" input pins) upon the rising edge of RxCLK (Rx_E3_Clock_In), as depicted below in **Figure 193**.

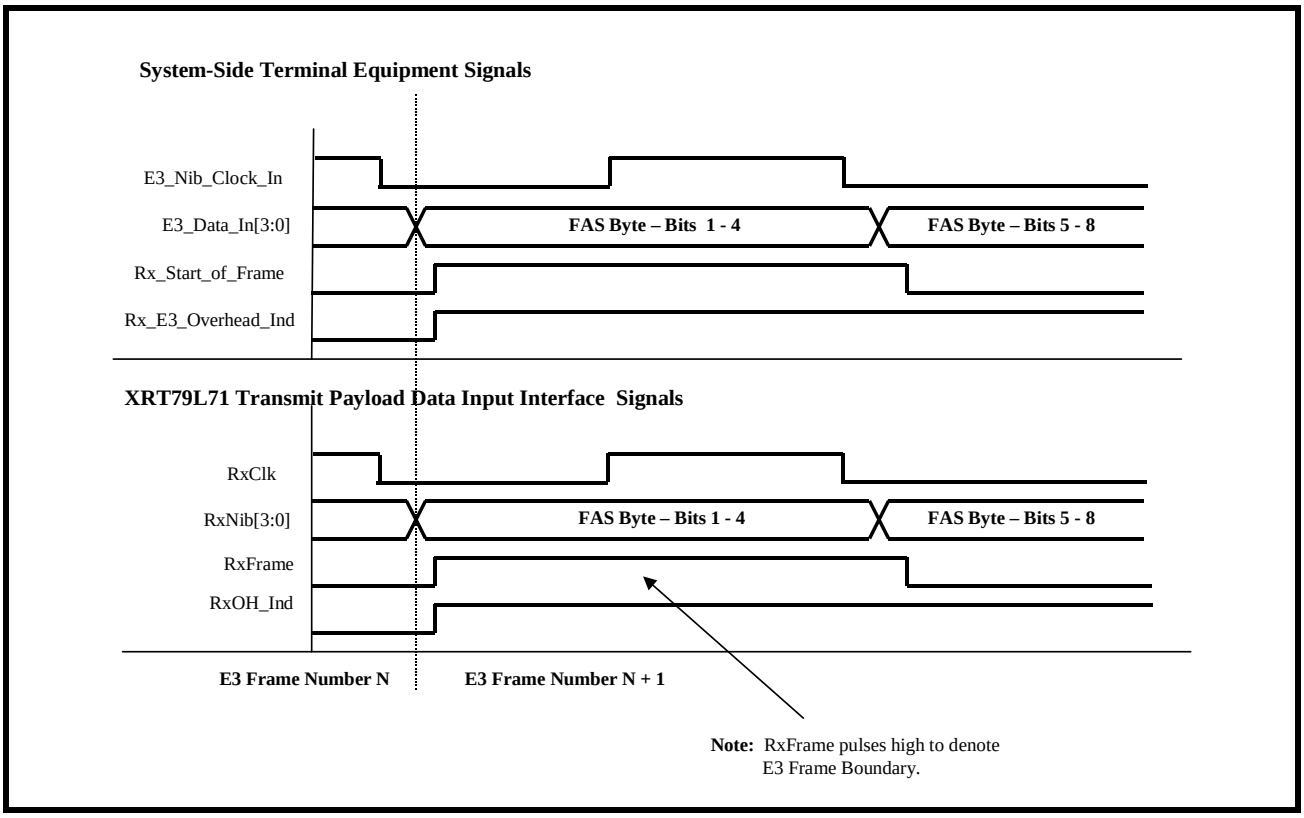
The Receive Payload Data Output Interface block (within the XRT79L71) will indicate that it is processing the very first payload nibble of a given E3 frame by pulsing the "RxFrame" output pin "HIGH" for one "nibble-period". The "RxFrame" output pin will be held "LOW" at all other times.

Finally, since (for E3 Applications) the Receive Payload Data Output Interface block (within the XRT79L71) does process and output overhead bits (in contrast to DS3, Nibble-Parallel Mode applications), then the Receive Payload Data Output Interface block will drive the "RxOH_Ind" output pin "high" coincident to whenever it outputs a nibble that contains "overhead" data. Conversely, the Receive Payload Data Output Interface block will drive the "RxOH_Ind" output pin "low" coincident to whenever it outputs a nibble that contains "payload" data.

The Frequency of the RxClk signal for E3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Receive Payload Data Input Interface block has been configured to operate in the "Nibble-Parallel" Mode, it will process both the E3 payload and overhead bits. As a consequence, the frequency of the RxClk signal (for Nibble-Parallel Mode applications) will be exactly 34.368MHz/4 (or 8.592MHz).

FIGURE 193. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR "NIBBLE-PARALLEL MODE" OPERATION



Configuring the XRT 79L71 to operate in the Nibble-Parallel Mode

The XRT79L71 can be configured to operate in the Nibble-Parallel Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in Figure 189.

STEP 2 - Configure the XRT 79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "HIGH".

NOTE: *This step also configures the "Transmit Payload Data Input Interface" block to operate in the "Nibble-Parallel Mode".*

6.0 ARCHITECTURAL/FUNCTIONAL DESCRIPTION OF THE XRT 79L71 - E3, ITU-T G.832 MODE OPERATION

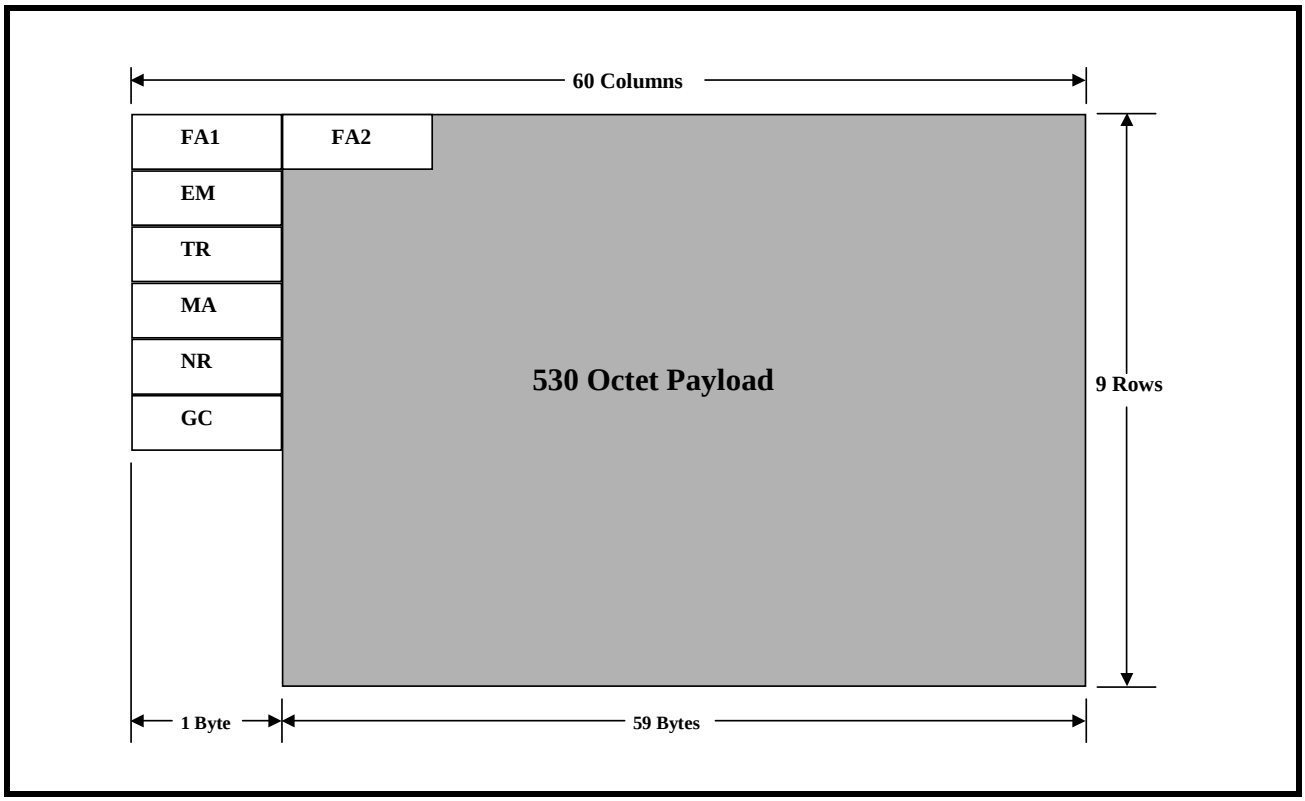
This particular section discusses Clear-Channel Framer over E3, ITU-T G.832 operation of the XRT79L71. Prior to discussing the architecture and the role of the DS3/E3 Framer blocks, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Mode; it is imperative to discuss the E3, ITU-T G.832 Frame structure.

6.1 DESCRIPTION OF THE E3, ITU-T G.832 FRAME STRUCTURE AND OVERHEAD BITS

The E3, ITU-T G.832 frame contains 537 bytes, of which 7 bytes are overhead bytes and the remaining 530 bytes are payload bytes.

These 537 octets are arranged in a 9 rows of 60 byte columns each, except for the last three rows, which contain only 59 byte 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 34.368Mbps.**Figure 194** presents an illustration of the E3, ITU-T G.832 Frame Format.

FIGURE 194. ILLUSTRATION OF THE E3, ITU-T G.832 FRAMING FORMAT



Configuring the XRT79L71 to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode

To configure the XRT79L71 to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, execute the following three steps.

STEP 1 - Configure the XRT79L71 to operate in the Clear-Channel Framer/HDLC Controller Mode

This can be accomplished by setting Bit 0 (Configuration Control) within the Operation Control Register - Byte 3 to "1" as depicted below.

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

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused							Configura-tion Control
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/W
0	0	0	0	0	0	0	1

STEP 2 - Configure the XRT79L71 to operate in the Clear-Channel Framer Mode.

The user can accomplish this by setting Bit 6 (HDLC Controller Enable), within the Payload HDLC Control Register, to "0" as depicted below.

Payload HDLC Control Register, Address = 0x1182

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer By-Pass	HDLC Controller Enable	HDLC CRC-32	Unused	HDLC Loop-back	Unused		
R/W	R/W	R/W	R/O	R/W	R/O	R/O	R/O
0	0	0	0	0	0	0	0

STEP 3 - Configure the XRT79L71 to operate in the E3, ITU-T G.832 Framing Format

To configure the XRT79L71 to operate in the E3, ITU-T G.832 Framing format write the appropriate data into Bits 2 (IsDS3) and 6 (Frame Format) within the Framer Operating Mode Register (Address = 0x1100), as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	1	1	1

The following table lists the relationship between the value of these bit-fields and the resulting framing format for the XRT79L71.

TABLE 52: THE RELATIONSHIP BETWEEN THE CONTENTS OF BITS 2 (FRAME FORMAT) AND 6 (ISDS3) WITHIN THE FRAMER OPERATING MODE REGISTER, AND THE RESULTING FRAMING FORMAT

BIT 6 (ISDS3)	BIT 2 (FRAME FORMAT)	RESULTING FRAMING FORMAT OF XRT79L71
0	0	E3, ITU-T G.751
0	1	E3, ITU-T G.832
1	0	DS3, C-bit Parity
1	1	DS3, M13/M23

he XRT79L71 will be operating in the E3, ITU-T G.751 Framing Format, upon power-up or hardware reset. Therefore, the user must set Bit 6 (Frame Format) to "1" in order to configure the XRT79L71 to operate in the E3, ITU-T G.832 Mode.

NOTE: *This bit setting configures the framing format for both the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block.*

Definition of the E3, ITU-T G.832 Overhead Bytes

In general, the E3, ITU-T G.832 Overhead bytes serve the following three purposes.

1. To support Frame Synchronization between the Near-End and remote E3 Terminals.
2. To provide Bit-Interleaved-Parity bits in order to facilitate performance monitoring and error detection with the E3 data-stream.
3. To support the transmission of Alarms, Status, and Data Link Information between the Near-End and the remote E3 Terminals.

The E3 Overhead bytes supporting each of these purposes are further defined below.

Frame Synchronization Bytes - The Frame Alignment Bytes (FA1 and FA2)

The FA1 and FA2 bytes are known as the Frame Alignment bytes, within a given E3, ITU-T G.832 Frame. A given Transmitting E3 Terminal Equipment will set the FA1 byte to the value "0xF6" and the FA2 byte to the value "0x28". The Receiving E3 Terminal Equipment will use these FA1 and FA2 bytes in order acquire and maintain frame synchronization with the incoming E3 data-stream. For more information on how the Receive DS3/E3 Framer block uses these bit-fields, please see SEE"DETECTING EM BYTE ERRORS" ON PAGE 530..

Performance Monitoring Byte - The Error Monitor Byte (EM)

Each E3 frame consists of a single EM byte. This EM byte carries the BIP-8 (Bit Interleaved Parity - 8) value of the previous E3 frame for performance monitoring. As a Transmitting E3 Terminal assembles an E3 frame prior to transmitting this E3 signal to the remote terminal equipment, it will compute the Bit-Interleaved-Parity - 8 (BIP-8) over this entire outbound E3 frame. The Transmitting E3 Terminal will then insert the resulting BIP-8 value into the EM byte-position within the very next outbound E3 frame.

As a Receiving E3 Terminal receives a given incoming E3 frame, it will locally compute its value for the EM byte. Afterwards, this Receiving E3 Terminal will compare its locally-computed BIP-8 value with the value of the EM byte within the very next incoming E3 frame. If these two EM byte values match, then the Receiving E3 Terminal will presume that the first of these two incoming E3 frames was received in an error-free manner. If these two EM byte values DO NOT match, then the Receiving E3 Terminal will presume that the first of these two incoming E3 frames was received in an erred manner.

For information on how the Receive DS3/E3 Framer block handles the EM bytes within the incoming E3 data-stream, please see SEE"DETECTING EM BYTE ERRORS" ON PAGE 530..

Alarm and Signaling-Related Overhead Bytes

The E3, ITU-T G.832 frame includes numerous bytes that are used to support the handling of alarms/defects and signaling information. Each of these byte-fields is defined below.

The Trail-Trace Byte (TR)

The Trail-Trace Byte is used to repetitively transmit a Trail-Access Point Identifier value so that a Receiving Terminal can verify its continued connection to the intended Transmitting Terminal. This Trail-Access Point Identifier consists of a repeating 16-byte message which consists of 15 ASCII characters and a CRC byte. The Byte-format of this Trail-Access Point Identifier Message is presented below in Table 53.

TABLE 53: THE BYTE-FORMAT OF THE TRAIL-TRACE MESSAGE THAT THIS BEING TRANSPORTED VIA AN E3 DATA-STREAM VIA THE TR BYTE

BYTE NUMBER		BIT 1 (MSB)	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8 (LSB)
1 (Frame Start Marker)		1	CRC_Value[6:0]						
2		0	TTM_Byte_1[6:0]						
*		0	TTM_Byte_N[6:0]						
15		0	TTM_Byte_15[6:0]						
16		0	TTM_Byte_16[6:0]						

Table 53 indicates that the very first byte of this 16-byte string is a Frame Start Marker byte, which 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 very first byte of this Trail Access Point Identifier Message). The remaining bits within this particular byte (e.g., C0 through C6) are the results of a CRC-7 calculation that was computed over the previous Trail-Access Point Identifier Message. All of the remaining 15

bytes will contain a "0" within their MSB. These remaining 15 bytes will transport 15 ASCII characters as is required by the E.164 Terminal Numbering format.

The XRT79L71 will support the transmission and reception of this Trail-Access Point Identifier Message, via the E3, ITU-T G.832 data-stream. For more details on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block within the XRT79L71 handle these messages, please see **SEE"TRANSMIT TRAIL-TRACE MESSAGE CONTROLLER BLOCK" ON PAGE 488.** and **SEE"RECEIVE TRAIL-TRACE MESSAGE CONTROLLER BLOCK" ON PAGE 534.**, respectively.

The Maintenance and Adaptation Byte (MA)

The exact role of the MA byte has been changed per the October 1998 Revision of the ITU-T G.832 specification document. However, the XRT79L71 supports both the Pre-October 1998 version of the MA byte and the Post-October 1998 version of the MA byte. Each of these versions of the MA byte is briefly described below.

The Post-October 1998 Version of the MA Byte

The Post-October 1998 Version of the MA byte contains the following bit-format.

FIGURE 195. THE BIT FORMAT OF THE POST OCTOBER 1988 VERSION OF THE MA BYTE

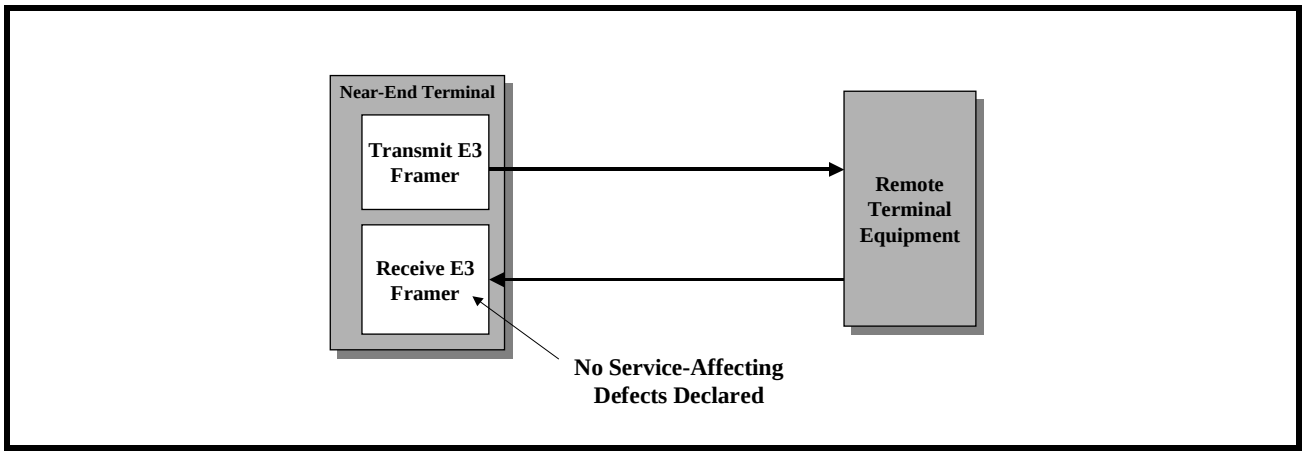
Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
FERF/RDI	FEBE/REI	Payload_Type[2:0]			SSM Multi-Frame Indicator[1:0]		SSM Bit

Bit 1 - FERF/RDI (Far-End Receive Failure/Remote Defect Indicator)

Bit 1 within the MA byte carries the FERF (Far-End Receive Failure)/RDI (Remote Defect Indicator) bit within the E3, ITU-T G.832 frame. The purpose of the FERF/RDI bit-field within the MA byte is to permit a given E3 Terminal Equipment to transmit a FERF/RDI indicator to the remote terminal equipment. Whenever a Near-End E3 Terminal declares a service-affecting defect condition (e.g., the LOS, LOF or the AIS defect condition) within its incoming E3 data-stream, then it will inform the remote terminal equipment (e.g., the source of this defective E3 signal) of this fact by transmitting the FERF/RDI indicator back out to the Remote Terminal Equipment via the outbound (returning) E3 signal. This Near-End E3 Terminal will indicate the FERF/RDI condition by setting the FERF/RDI bit-field, within each outbound E3 frame to "1" for the duration that the Receive-related defect condition exists. Conversely, this Near-End E3 Terminal will indicate that it is NOT transmitting the FERF/RDI indicator by setting the FERF/RDI bit-field to "0" for the duration that no Receive-related/Service-affecting defect conditions exists. This concept of transmitting the FERF/RDI indicator in response to certain defect conditions is reinforced in Figures **196** through **199**, below.

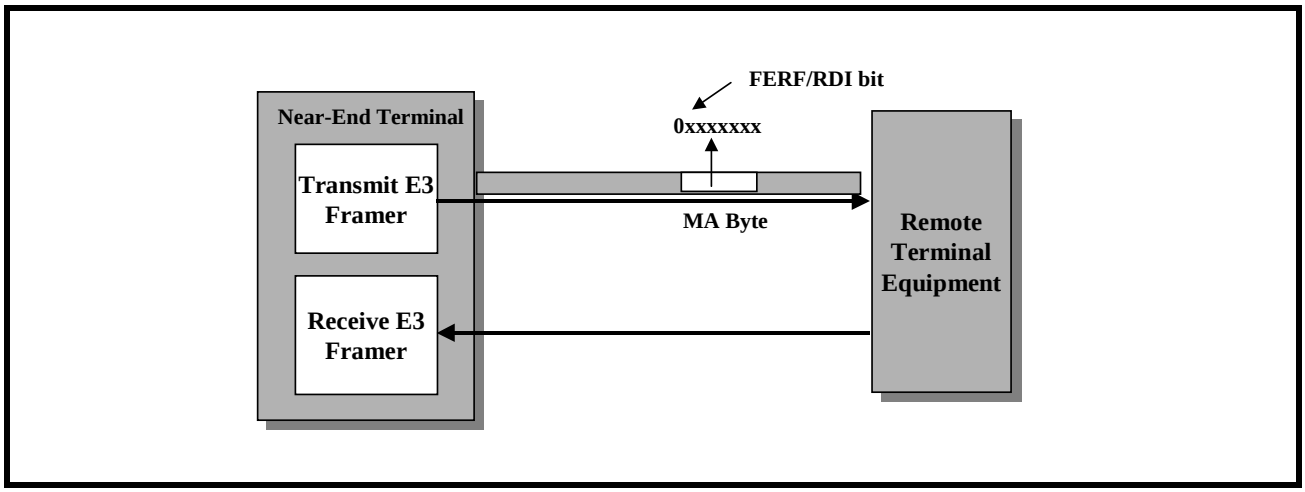
Figure 196 presents an illustration of a given Near-End Terminal that is exchanging E3 data with a remote terminal, in an un-erred manner.

FIGURE 196. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



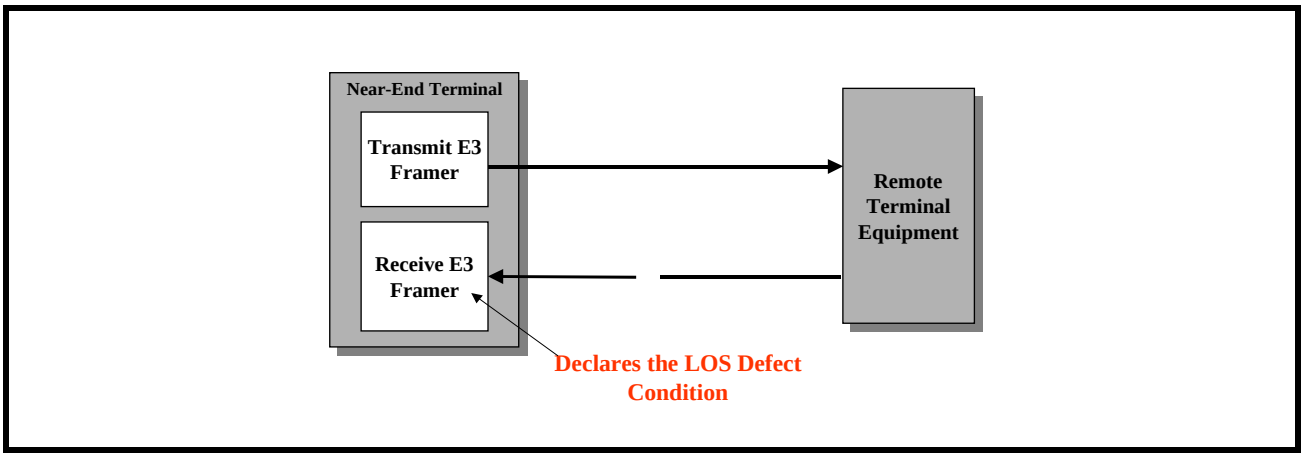
In response to this un-erred condition, the Transmit E3 Framer within the Near-End Terminal will respond by setting the FERF/RDI bit-field within each of its outbound E3 frames to "0", in order to denote an un-erred condition. Figure 197 presents an illustration of the Transmit E3 Framer sending the un-erred indication to the Remote Terminal Equipment.

FIGURE 197. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



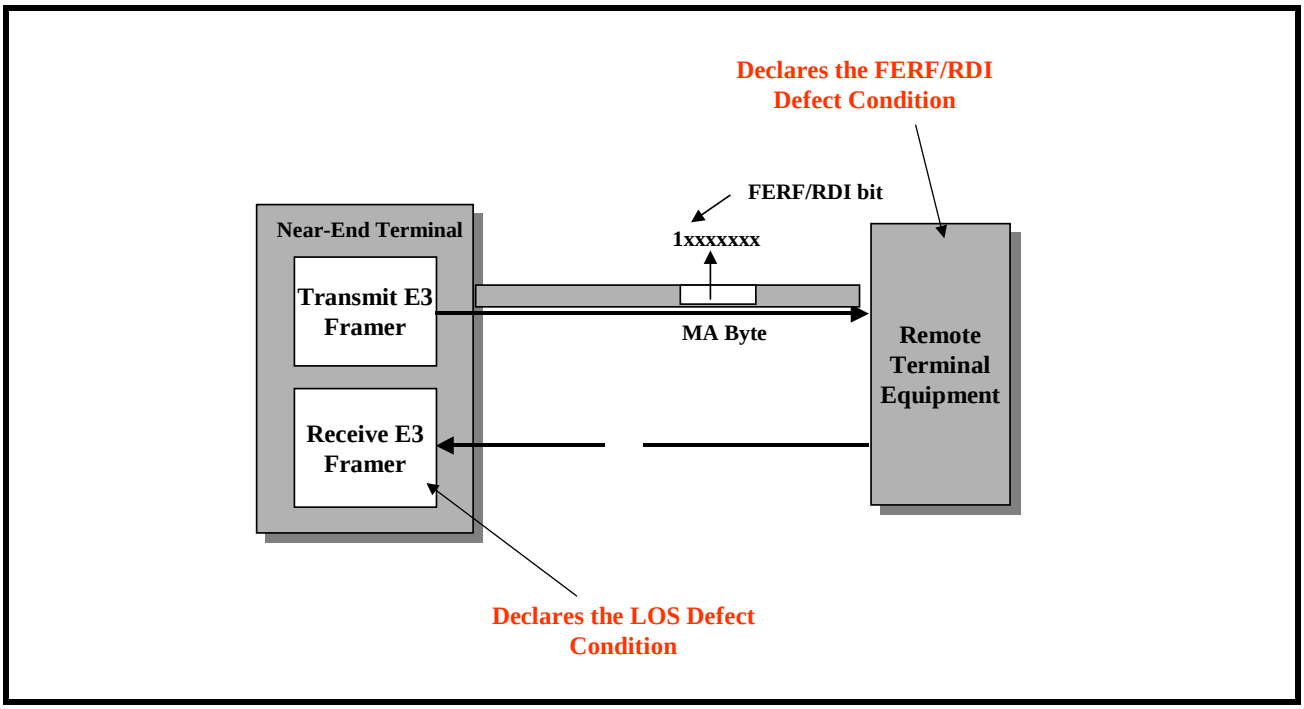
Next, Figure 198 presents an illustration of a given Near-End Terminal that is declaring the LOS defect with its incoming E3 signal.

FIGURE 198. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DECLARING THE LOS DEFECT CONDITION WITHIN ITS INCOMING E3 SIGNAL



In response to this erred condition, the Transmit E3 Framer within the Near-End Terminal will respond by setting the FERF/RDI bit, within each of its outbound E3 frames to "1" in order to denote a FERF (Far-End Receive Failure) or RDI (Remote Defect Indicator) condition. Figure 199 presents an illustration of the Transmit E3 Framer sending the FERF/RDI indicator to the Remote Terminal Equipment.

FIGURE 199. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the XRT79L71 Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block handle the FERF/RDI indication will be described in SEE"TRANSMITTING THE FERF/RDI INDICATOR" ON PAGE 499. and SEE"DECLARING AND CLEARING THE FERF/RDI DEFECT CONDITION" ON PAGE 527., respectively.

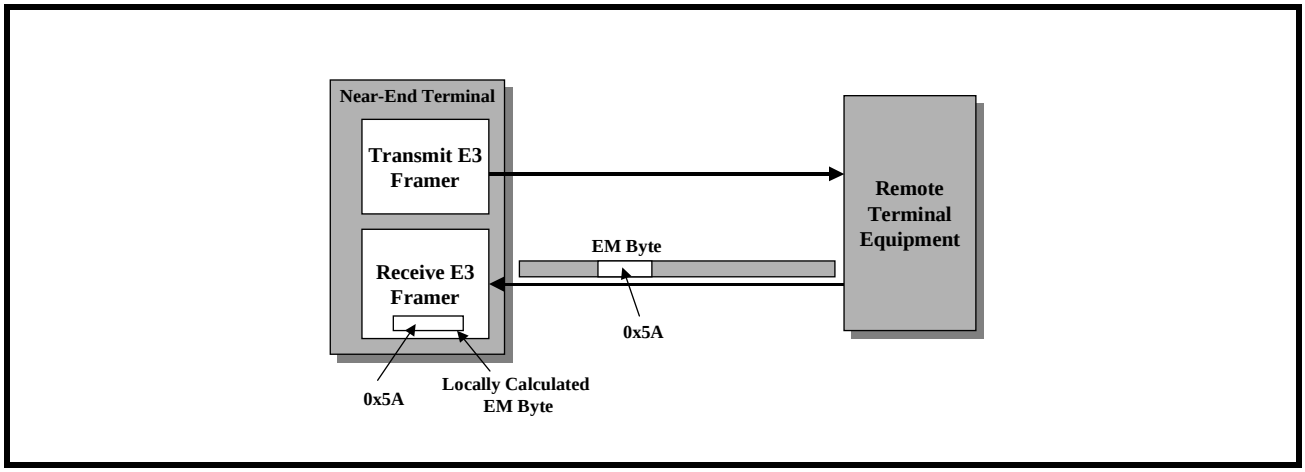
Bit 2 - FEBE/REI (Far-End Block Error/Remote Defect Indicator)

Bit 2 within the MA bytes carries the FEBE (Far-End Block Error)/REI (Remote Error Indicator) bit within the E3, ITU-T G.832 frame. If the Near-End XRT79L71 detects EM byte errors within the incoming (received) E3 data-stream, then it will inform the remote terminal of this fact by setting the FEBE/REI bit-field within the outgoing, or return E3 frame to the value "1" to indicate an erred condition. The Transmit Section of the XRT79L71 will set the FEBE/REI bit-field to "0" in order to denote an un-erred condition.

This concept of transmitting the FEBE/REI indicator in response to the detection of EM byte errors is reinforced in Figures 200 through 203 below.

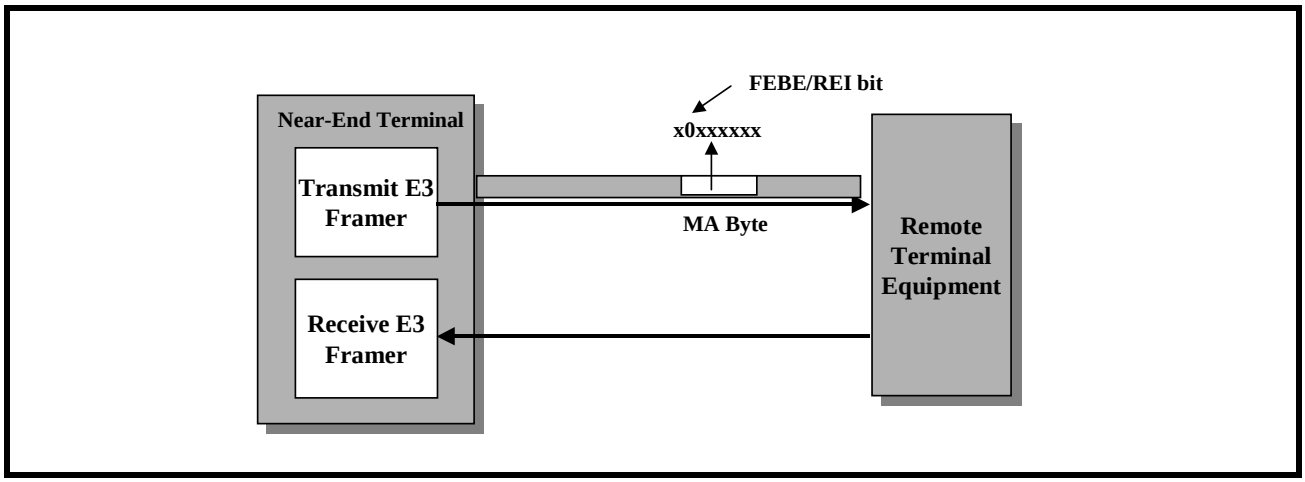
Figure 200 presents an illustration of a given Near-End Terminal that is exchanging E3 data with the remote terminal, in an un-erred manner. More specifically, the Receive E3 Framer block has locally computed an EM byte value of "0x5A" that was computed over a given incoming E3 frame. Figure 200 indicates that the EM byte value which resides within the very next incoming E3 frame is of the value "0x5A". As a consequence, there is no EM byte errors being detected at this time.

FIGURE 200. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



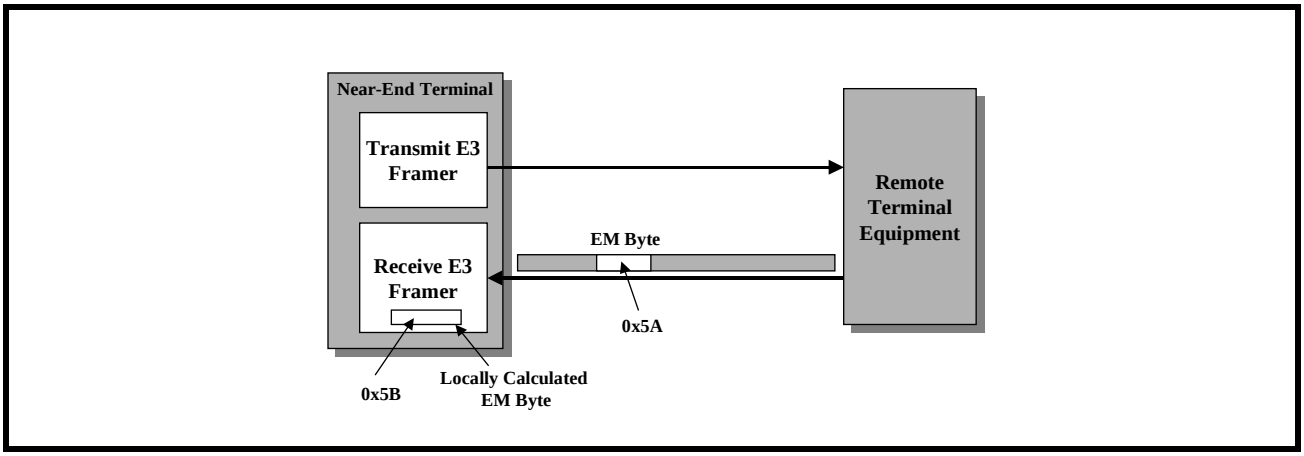
In response to this un-erred condition, the Transmit E3 Framer within the Near-End Terminal will respond by setting the FEBE/REI bit-field "0". Figure 201 presents an illustration of the Transmit E3 Framer sending this un-erred indication to the Remote Terminal Equipment.

FIGURE 201. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



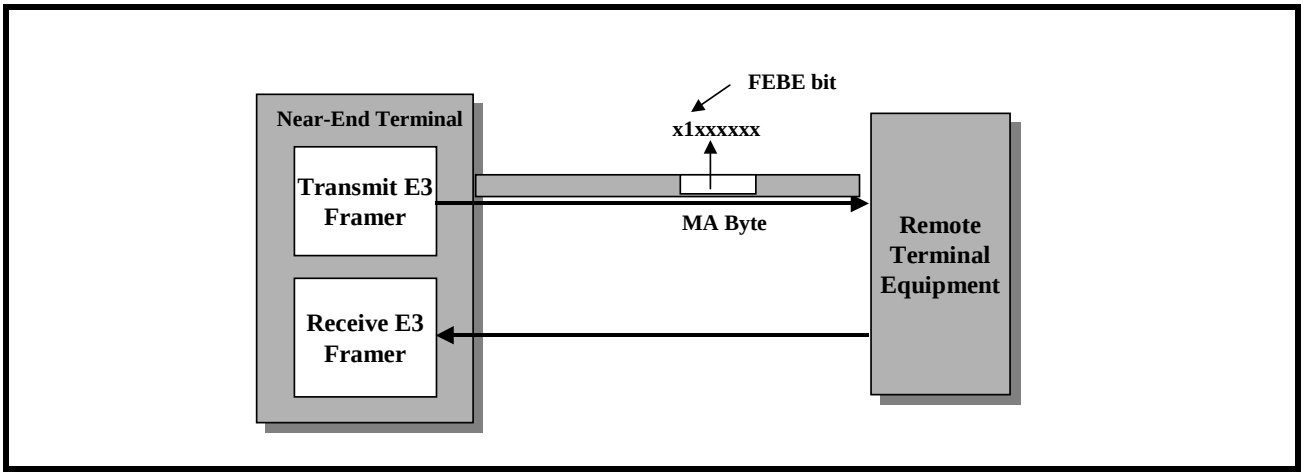
Next, Figure 202 presents an illustration of a given Near-End Terminal that is detecting an EM byte error within its incoming E3 signal. More specifically, the Receive E3 Framer block has locally computed an EM byte value of "0x5B" that was computed over a given incoming E3 frame. Figure 202 indicates that the EM byte value which resides within the very next incoming E3 frame is of the value "0x5A". As a consequence, an EM byte error is being detected at this time.

FIGURE 202. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DETECTING EM BYTE ERRORS WITHIN ITS INCOMING E3 SIGNAL



In response to this erred condition, the Transmit E3 Framer within the Near-End Terminal will respond by setting the FEBE/REI bit, within the very next outbound E3 frame to the value "1" in order to denote a FEBE/REI event. More specifically, the Transmit E3 Framer within the Near-End Terminal will transmit an E3 frame with the FEBE/REI bit set to "1" each time the corresponding Receive E3 Framer within the Near-End Terminal detects an EM byte errors within an E3 frame. Figure 203 presents an illustration of the Transmit E3 Framer sending the FEBE/REI indicator to the Remote Terminal Equipment.

FIGURE 203. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FEBE/REI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



A more detailed discussion on how the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block within the XRT79L71 handles the FEBE/REI bit can be found in SEE"TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR" ON PAGE 505. and SEE"DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS" ON PAGE 531..

Bits 3 through 5 - Payload Type[2:0]

The purpose of these three (3) bit-fields is to permit a given Terminal Equipment to identify the type of payload that is currently being transported via a given E3 data-stream. In this case, the role of these bit-fields are somewhat analogous to that of the C2 byte (Payload Label Byte) within SONET/SDH. The relationship between the contents within these bit-fields and the corresponding type of data being carried via the payload bytes is listed below in **Table 54**.

TABLE 54: THE RELATIONSHIP BETWEEN THE CONTENTS OF THE PAYLOAD TYPE[2:0] BIT-FIELDS AND THE TYPE OF DATA BEING TRANSPORTED VIA THE PAYLOAD BYTES WITHIN A GIVEN E3 DATA-STREAM

PAYLOAD_TYPE[2:0]	TYPE OF DATA BEING TRANSPORTED VIA E3 PAYLOAD
000	Unequipped
001	Equipped, Non-Specific
010	ATM
011	SDH TU-12s

For more details on how the Transmit DS3/E3 Framer and Receive DS3/E3 Framer blocks handle the Payload_Type[2:0] bit-fields, within the E3, ITU-T G.832 Framing format, please see **SEE"SETTING THE PAYLOAD-TYPE BIT-FIELDS WITHIN THE OUTBOUND E3 DATA-STREAM" ON PAGE 509.** and **SEE"DECLARING AND CLEARING THE PAYLOAD-TYPE MISMATCH DEFECT CONDITION" ON PAGE 534.**, respectively.

Bits 6 and 7 - SSM Multi-Frame Indicator[1:0]

The E3, ITU-T G.832 framing format permits the user to repeatedly transmit the SSM (Synchronization Status Message) from one terminal equipment to another. According to ITU-T G.707, the SSM is a four-bit value that is used to identify the quality-level of synchronization/timing that the Transmitting E3 Terminal Equipment is currently operating at. These two bit-fields, along with Bit 8 (SSM Bit) within the MA Byte are used to facilitate the repetitive transmissions of the SSM from one terminal equipment to another.

As mentioned above, the SSM (Synchronization Status Message) is a four-bit message that is continuously and repetitively transmitted from one terminal equipment to another, via the E3 data-stream. Bit 8 (the SSM bit) within the MA byte is actual bit that is used to repetitively transmit the SSM from one terminal equipment to another. Bits 6 and 7, are Multi-Frame Indicator bits that are used to identify which of the four-bits within the SSM is currently being transmitting within Bit 8, of this particular E3 frame. **Table 55** presents the relationship between the contents of these bits (e.g., the SSM Multi-Frame Indicator bits) and the individual SSM bit that is currently being transported via Bit 8 of the MA byte within this same E3 frame.

TABLE 55: THE RELATIONSHIP BETWEEN THE VALUES OF BITS 6 AND 7 (SSM MULTI-FRAME INDICATOR) AND THE SSM BIT THAT IS BEING TRANSPORTED VIA BIT 8 (SSM BIT) WITHIN THE MA BYTE OF THE CURRENT E3 FRAME

BITS 6 AND 7 (SSM MULTI-FRAME INDICATOR[1:0]	SSM BIT BEING TRANSPORTED IN BIT 8 (SSM BIT) OF MA BYTE WITHIN THE CURRENT E3 FRAME
00	SSM Bit # 1 (The Most Significant Bit) within the SSM
01	SSM Bit # 2
10	SSM Bit # 3
11	SSM Bit # 4 (The Least Significant Bit) within the SSM

Bit 8 - SSM Bit

The role of this particular bit-field is described above (Bits 6 and 7 - SSM Multi-Frame Indicator[1:0])

For information on how the Transmit DS3/E3 Framer and Receive DS3/E3 Framer blocks handle the MA byte, please see SEE"TRANSMIT SSM CONTROLLER BLOCK" ON PAGE 494. and SEE"RECEIVE SSM CONTROLLER BLOCK" ON PAGE 539., respectively.

The Pre-October 1998 Version of the MA Byte

The Pre-October 1998 Version of the MA byte contained the following bit-format.

Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
FERF/RDI	FEBE/REI	Payload_Type[2:0]			Payload Dependent[1:0]		Timing Marker

Bits 6 and 7 - Payload Dependent

Bit 8 - Timing Marker

The Network Operator Byte (NR)

This byte is allocated for maintenance purposes specific to individual network operators. The XRT79L71 permits the user to transport either a proprietary message or a LAPD/PMDL Message via the NR byte.

Information on how to transmit and receive a proprietary message via the NR byte can be found in SEE"USER CONTROL OVER THE NR BYTE WITHIN THE OUTBOUND E3 DATA-STREAM" ON PAGE 509. and SEE"MONITORING THE NR BYTE WITHIN THE INCOMING E3 DATA-STREAM" ON PAGE 534., respectively. Information on how to transmit and receive a LAPD/PMDL Message via the NR byte can be found in SEE"TRANSMIT LAPD CONTROLLER BLOCK INTERRUPT" ON PAGE 488. and SEE"RECEIVE LAPD CONTROLLER BLOCK" ON PAGE 549., respectively.

The General Purpose Communications Channel Byte (GC)

This byte is allocated to provide either a data or voice channel for maintenance purposes. The XRT79L71 permits the user to transport either a proprietary message or a LAPD/PMDL Message via the GC byte.

Information on how to transmit and receive a proprietary message via the GC byte can be found in SEE"USER CONTROL OVER THE GC BYTE WITHIN THE OUTBOUND E3 DATA-STREAM" ON PAGE 509. and SEE"MONITORING THE GC BYTE WITHIN THE INCOMING E3 DATA-STREAM" ON PAGE 534., respectively. Information on how to transmit and receive a LAPD/PMDL Message via the GC byte can be found in SEE"TRANSMIT LAPD CONTROLLER BLOCK INTERRUPT" ON PAGE 488. and SEE"RECEIVE LAPD CONTROLLER BLOCK" ON PAGE 549., respectively.

6.2 THE TRANSMIT DIRECTION - E3, ITU-T G.832 CLEAR-CHANNEL FRAMER APPLICATIONS

Now that the basics of the E3, ITU-T G.832 frame structure have been discussed; the next several sections present an in-depth functional description of all of the blocks that are operating in the Transmit Direction, within the XRT79L71, when configured to operate in the "Clear-Channel E3 Framer" Mode. Figure 204 presents a functional block diagram of the Transmit Direction circuitry within the XRT79L71.

FIGURE 204. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE

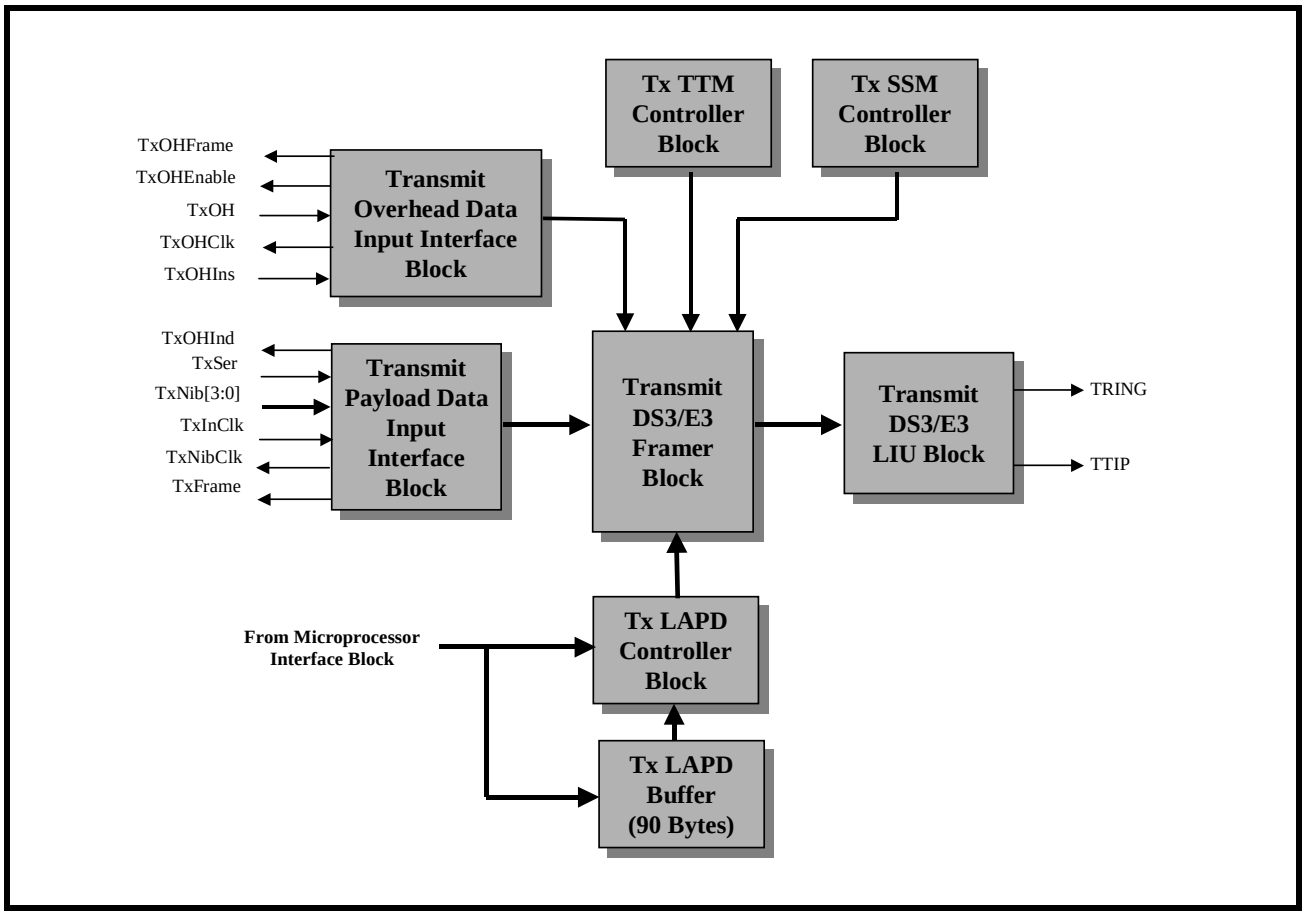


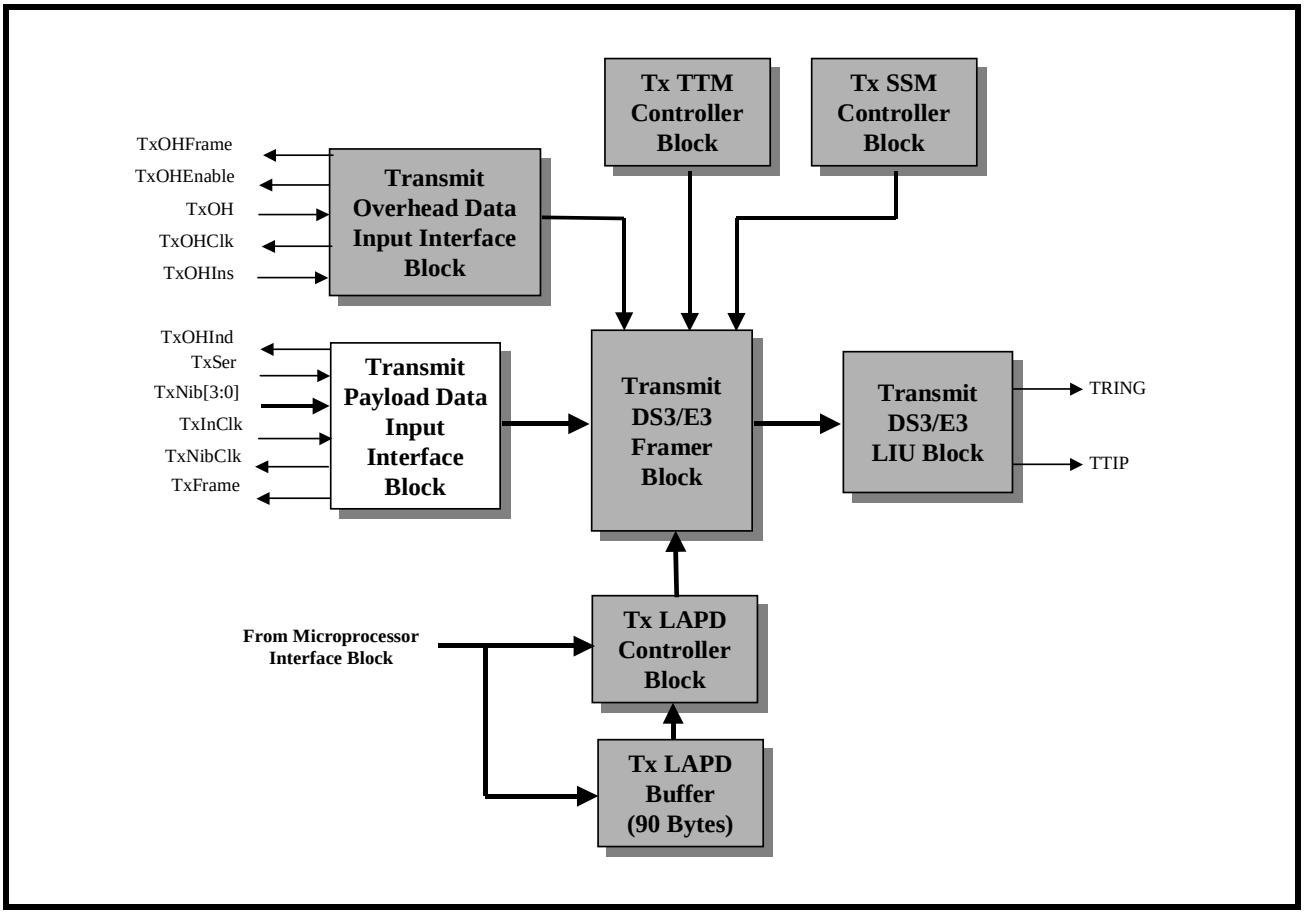
Figure 204 indicates that the Transmit Direction circuitry consists of the following functional blocks.

- The Transmit Payload Data Input Interface block
- The Transmit Overhead Data Input Interface block
- The Transmit LAPD Controller block
- The Transmit Trail-Trace Message Controller block
- The Transmit SSM Controller block
- The Transmit E3 Framer Block
- The Transmit E3 LIU Block

6.2.1 TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

The Transmit Payload Data Input Interface block is the very first functional block (within the Transmit Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. Figure 205 presents an illustration of the "Transmit Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, Clear-Channel Framer Mode, with the "Transmit Payload Data Input Interface" block highlighted.

FIGURE 205. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE' BLOCK HIGHLIGHTED).



The purpose of the "Transmit Payload Data Input Interface" block is to accept payload data from system-side or up-stream source and to "pass" this payload data along to the "Transmit E3 Framer" block that will ultimately map this payload data into the payload bytes within each outbound E3 frame.

In order to accomplish this, the Transmit Payload Data Input Interface block has numerous input and output pins. **Table 56** presents a list and a brief description of each of these pins.

TABLE 56: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxSer	C9	I	**Transmit Serial Payload Data Input Pin:** If the Transmit Payload Data Input Interface block is operated in the "Serial" Mode, then the "System-Side" equipment is expected to apply the payload data that is to be transported via the outbound E3 data-stream to this input pin, in a serial manner. The XRT79L71 samples the data that is on this input pin upon the rising edge of either the RxOutClk (for loop-timing applications) or the TxInClk signal (for local-timing applications). ***NOTE:*** *This signal is only active if the "NibIntf" input pin is pulled "LOW".*
TxNib[3:0]	B8 C8 D8 A9	I	**Transmit Nibble-Parallel Payload Data Input Pins:** If the Transmit Payload Data Input Interface block is operated in the Nibble-Parallel Mode, then the "System-Side" equipment is expected to apply the payload data that is to be transported via the outbound E3 data-stream to these input pins, in a nibble-parallel manner. The XRT79L71 samples the data that is placed on these input pins upon the third rising edge of TxInClk, following a given rising edge of the TxNibClk output pin. ***NOTE:*** *These signals are only active if the "NibIntf" input pin is pulled "HIGH".*
TxNibFrame	A10	O	**Transmit End of Frame Output Indicator - Nibble Mode:** The Transmit Section of the XRT79L71 pulses this output pin "high" for one nibble period whenever the Transmit Payload Data Input Interface block is processing the very last nibble within a given E3 frame. The purpose of this output pin is to alert the "System-Side" Terminal Equipment that is needs to begin transmission of a new E3 frame to the Transmit Payload Data Input Interface of the XRT79L71. ***NOTE:*** *This output pin is only active if the XRT79L71 has been configured to operate in the "Nibble-Parallel" Mode.*
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the "Transmit Circuitry" within the XRT79L71. Additionally, if the XRT79L71 has been configured to operate in both the Serial and Local-Timing Mode, then the XRT79L71 will sample the data, residing on the TxSer input pin, upon the rising edge of this input clock signal.
TxNibClk	D9	O	**Transmit Nibble-Mode Clock Output Signal:** If the XRT79L71 has been configured to operate in the "Nibble-Parallel" Mode, then the XRT79L71 will derive this output clock signal from either the "TxInClk" or the "LIU Recovered Clock" signal ***NOTE:*** *The frequency of this clock output signal is one-fourth of the TxInClk or the RxOutClk signals.* The user is advised to update the "Nibble-Parallel" data via the "TxNib[3:0]" output pins, upon the rising edge of this clock output signal. The XRT79L71 will sample the "TxNib[3:0]" input pins upon the third rising edge of the "TxInClk" clock input signals (following a rising edge in this particular signal).

TABLE 56: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxOHInd/TxGap-Clk	B9	O	**Transmit Overhead Bit Indicator Output/Transmit Gap-Clock Output:** The function of this output pin depends upon whether or not the XRT79L71 has been configured to operate in the Gapped-Clock Mode. **Non-Gapped Clock Mode - TxOHInd:** This output pin will pulse "high" one bit period prior to the time that the "Transmit Section" of the XRT79L71 is processing an Overhead bit. This output pin will be held "low" at all other times. The purpose of this output pin is to warn the System-Side Terminal Equipment that during the very next bit-period, the XRT79L71 is going to be processing an Overhead bits and will be (during this very next bit-period) ignoring any data that is applied to the TxSer input pin. **Gapped-Clock Mode - TxGapClk:** If the XRT79L71 has been configured to operate in the "Gapped-Clock" Mode, then this particular output pin will function as a "demand" output clock signal. In this case, the System-Side Terminal Equipment will be expected to update the data on the "TxSer" input pin, upon the rising edge of this particular output signal. The XRT79L71 will sample and latch the "TxSer" data, upon the falling edge of the "TxGapClk" signal. ***NOTE:*** *In the "Gapped-Clock" Mode, the XRT79L71 will only generate a clock edge (via this output pin) whenever the Transmit Payload Data Input Interface is processing "payload" data. The XRT79L71 will NOT generate a clock edge (via this output pin) whenever the Transmit Payload Data Input Interface is processing an "overhead" bit.*
TxFrame	B10	O	**Transmit End of Frame Output Indicator:** The Transmit Section of the XRT79L71 pulses this output pin "high" for one bit-period coincident to whenever the Transmit Payload Data Input Interface block is processing the last bit of a given E3 frame. The purpose of this output pin is to alert the System-Side Terminal Equipment that it needs to begin transmission of a new E3 frame to the Transmit Payload Data Input Interface block of the XRT79L71 (e.g., to permit the XRT79L71 to maintain Transmit E3 Framing alignment control over the System-Side Terminal Equipment).

TABLE 56: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxFrameRef	A11	I	**Transmit E3 Frame Reference Input::** The XRT79L71 allows the user to configure this input pin to function as the "Transmit E3 Frame Generation Reference" Input. If this particular configuration option is invoked, the TransmitE3 Framer block (within the XRT79L71) will initiate E3 frame generation anytime it detects a rising edge of this input pin. ***NOTE:*** *If tthis configuration option is implemented, it is imperative that this particular input signal is synchronous with the "TxInClk" input signal. Failure to do so will result in the transmission of erred E3 frames to the remote terminal equipment.*
RxOutClk	B5	O	**Loop-Timing Reference Clock Output Pin:** If the XRT79L71 is configured to operate in the Loop-Timing Mode, then the Transmit Section of the XRT79L71 will be configured to use the LIU Recovered Clock signal as its timing source. In this case, the XRT79L71 will output a 34.368MHz clock signal via this particular output pin. In this configuration, the TxInClk signal will be inactive and will NOT be used to sample and latch the data on the TxSer input pin. In this case, the XRT79L71 will now be configured to sample the TxSer input pin upon the rising edge of the "RxOutClk" signal. ***NOTE:*** *This output pin will always be active, in the sense that it will always generate a 34.368MHz clock signal (even when the XRT79L71 is NOT configured to operate in the Loop-Timing Mode). However, the XRT79L71 will only use this particular clock signal to sample and latch the data on the TxSer input pin whenever the XRT79L71 has been configured to operate in the Loop-Timing Mode.*

Operation of the Transmit Payload Data Input Interface Block

The Transmit Payload Data Input Interface block permits the user to configure it to operate in the following combination of modes.

- The Serial or Nibble-Parallel Interface Mode
- The Loop-Timing or Local-Timing Mode

If the XRT79L71 is configured to operate in the Local-Timing Mode, then there are two additional "sub-options".

- The "Frame-Master" or "Frame-Slave" Mode

With these three sets of configuration options, the Transmit Payload Data Input Interface block can be configured to operate in any one of the following six (6) Modes.

- Mode 1 - Serial/Loop-Timing Mode
- Mode 2 - Serial/Local-Timing/Frame Slave Mode
- Mode 3 - Serial/Local-Timing/Frame Master Mode
- Mode 4 - Nibble-Parallel/Loop-Timing Mode
- Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode
- Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode

Table 57 presents a "Quick-Look" Summary of each of these Modes

TABLE 57: A SUMMARY OF THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" MODES

MODE	NIBBLE-PARALLEL/ SERIAL MODE	SOURCE OF "SYSTEM-SIDE TERMINAL EQUIPMENT" CLOCK	FRAMING ALIGNMENT TIMING SOURCE
1	Serial	Loop-Timing Mode: The XRT79L71 will output a 34.368MHz clock signal via the "RxOutClk" output pin. This clock signal is ultimately derived from Recovered Line Clock (from Receive DS3/E3 LIU Block).	Asynchronous upon Power up.
2	Serial	Local-Timing Mode: The user is expected to apply a 34.368MHz clock signal to the TxInClk Input pin	TxFrameRef Input
3	Serial	Local-Timing Mode: The user is expected to apply a 34.368MHz clock signal to the TxInClk Input pin	Asynchronous upon Power up
4	Nibble-Parallel	Loop-Timing Mode: The XRT79L71 will output an 8.592MHz "Nibble-Clock" signal (via the "TxNibClk" output). This clock signal is ultimately derived from the Recovered Line Clock (from the Receive DS3/E3 LIU Block).	Asynchronous upon Power up
5	Nibble-Parallel	Local-Timing Mode: The user is expected to apply a 34.368MHz clock to the TxInClk input pin. The XRT79L71 will use the TxInClk signal to derive the 8.592MHz clock signal (which is output via the "TxNibClk" output pin).	TxFrameRef Input pin
6	Nibble-Parallel	Local-Timing Mode: The user is expected to apply a 34.368MHz clock signal to the TxInClk input pin. The XRT79L7 will use the TxInClk signal to derive the 8.592MHz clock signal (which is output via the "TxNibClk" output pin).	Asynchronous upon Power up

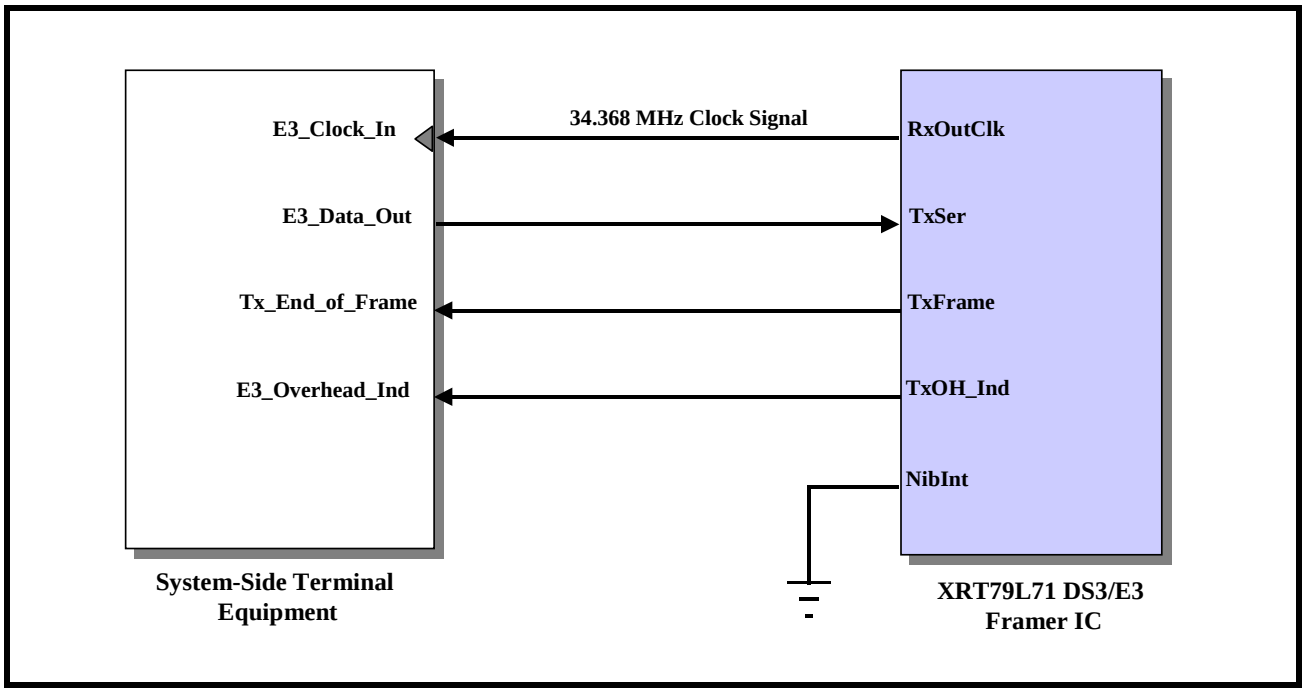
6.2.1.1 Mode 1 - Serial/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in "Mode 1" then all of the following is true.

· The XRT79L71 will be configured to operate in the Loop-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal (from the Receive E3 LIU Block) as its timing source.

- Since the XRT79L71 is configured to operate in the "Serial-Mode", it will sample and latch the data, being applied to the "TxSer" input pin upon the rising edge of the "RxOutClk" output signal.
- The XRT79L71 will pulse the "TxFrame" output pin "high" for one bit period coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given E3 frame.

FIGURE 206. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE" TERMINAL EQUIPMENT TO THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR MODE 1 (SERIAL/LOOP-TIMING) OPERATION



Mode 1 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of the 34.368MHz clock signal via the "RxOutClk" output signal.

NOTE: *The "RxOutClk" signal is a buffered version of the "Recovered Line Clock" signal, from the Receive DS3/E3 LIU Block).*

This clock signal is used as the "System-Side Terminal Equipment" clock source by both the Transmit Payload Data Input Interface block (of the XRT79L71) and the "System-Side Terminal Equipment" device or circuitry.

The "System-Side" Terminal Equipment should serially output the payload data (that is to be transported via the "outbound" E3 data-stream) via its "E3_Data_Out" output pin. The user is advised to design the "System-Side Terminal Equipment" circuitry such that it will update the data (via the "E3_Data_Out" output pin) upon the rising edge of the 34.368MHz clock signal (at its "E3_Clock_In" input pin) as depicted below in **Figure 207**.

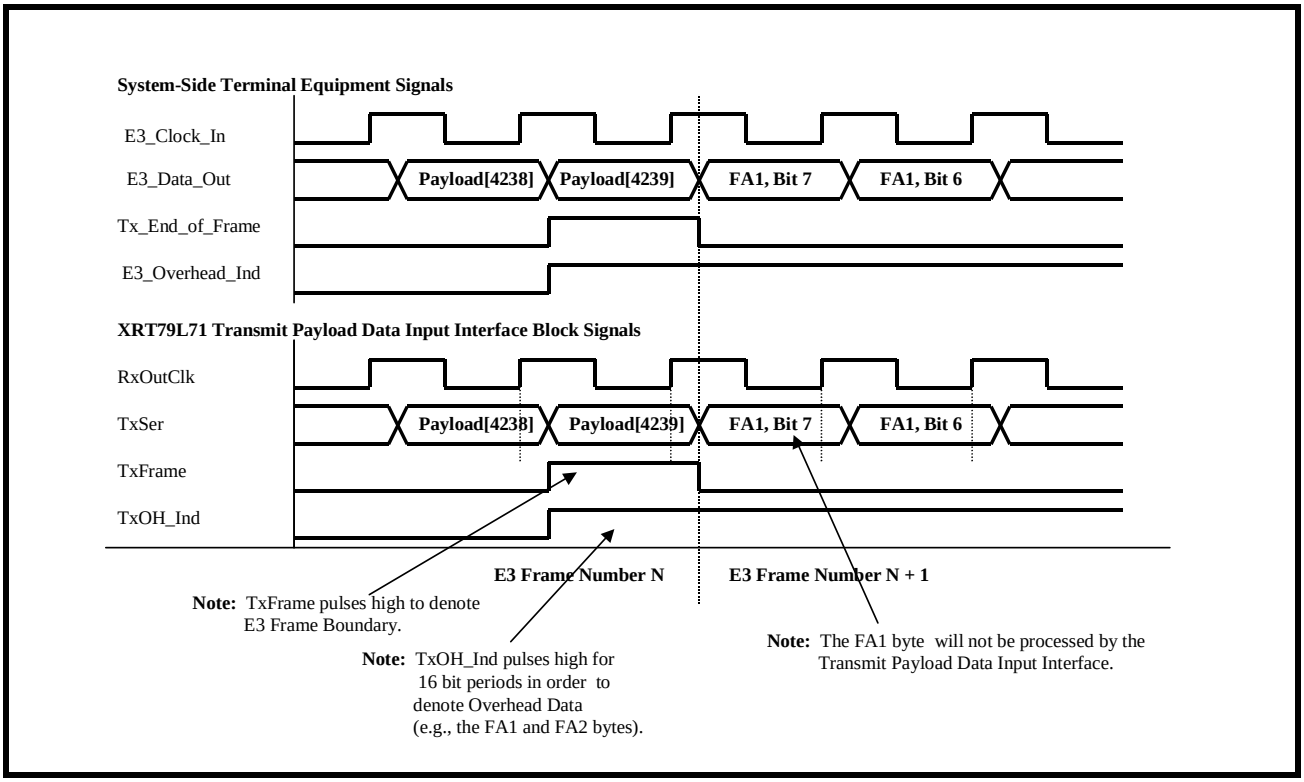
The XRT79L71 will latch the contents of the "TxSer" input pin, upon the rising edge of the "RxOutClk" signal. The XRT79L71 will indicate that it is processing the very last bit of a given E3 frame by pulsing its "TxFrame" output pin "HIGH" for one bit-period. The "TxFrame" output pin will have held "LOW" at all other times. Whenever the System-Side Terminal Equipment detects this pulse at its "Tx_Start_of_Frame" input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the "E3_Data_Out" output (or the TxSer input) pin.

Finally, the Transmit Payload Data Input Interface block (within the XRT79L71) will indicate that it is about to process an overhead bit by pulsing the "TxOH_Ind" output pin "HIGH" for one period prior to its processing. In **Figure 206**, the "TxOH_Ind" output pin (of the XRT79L71) is connected to the "E3_Overhead_Ind" input pin of the "System-Side Terminal Equipment" circuitry. Whenever the "E3_Overhead_Ind" input pin is pulsed "high" the "System-Side" Terminal Equipment is expected to NOT transmit an E3 payload bit upon the very next rising edge of "E3_Clock_In". Instead, the System-Side Terminal Equipment is expected to delay its transmission of the very next payload bit by one "RxOutClk" clock period.

NOTES:

1. *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead bits), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for 16 consecutive bit-periods (when processing the FA1 and FA2 bytes), and it will do so for 8 consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "RxOutClk", and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*
2. *2.If the "Transmit Payload Data Input Interface" block is to be operated in the "Gapped-Clock" Mode, then refer to* ***SEE"OPERATING THE TRANSMIT PAYLOAD DATA INPUT INTERFACE IN THE GAPPED CLOCK MODE" ON PAGE 453.****.*

FIGURE 207. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 1 (SERIAL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 1 (Serial/Loop-Timing)

The user can configure the XRT79L71 to operate in Mode 1 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 206.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "LOW".

NOTE: *This step also configures the "Receive Payload Data Output Interface" to operate in the "Serial Mode".*

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[0, 0]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

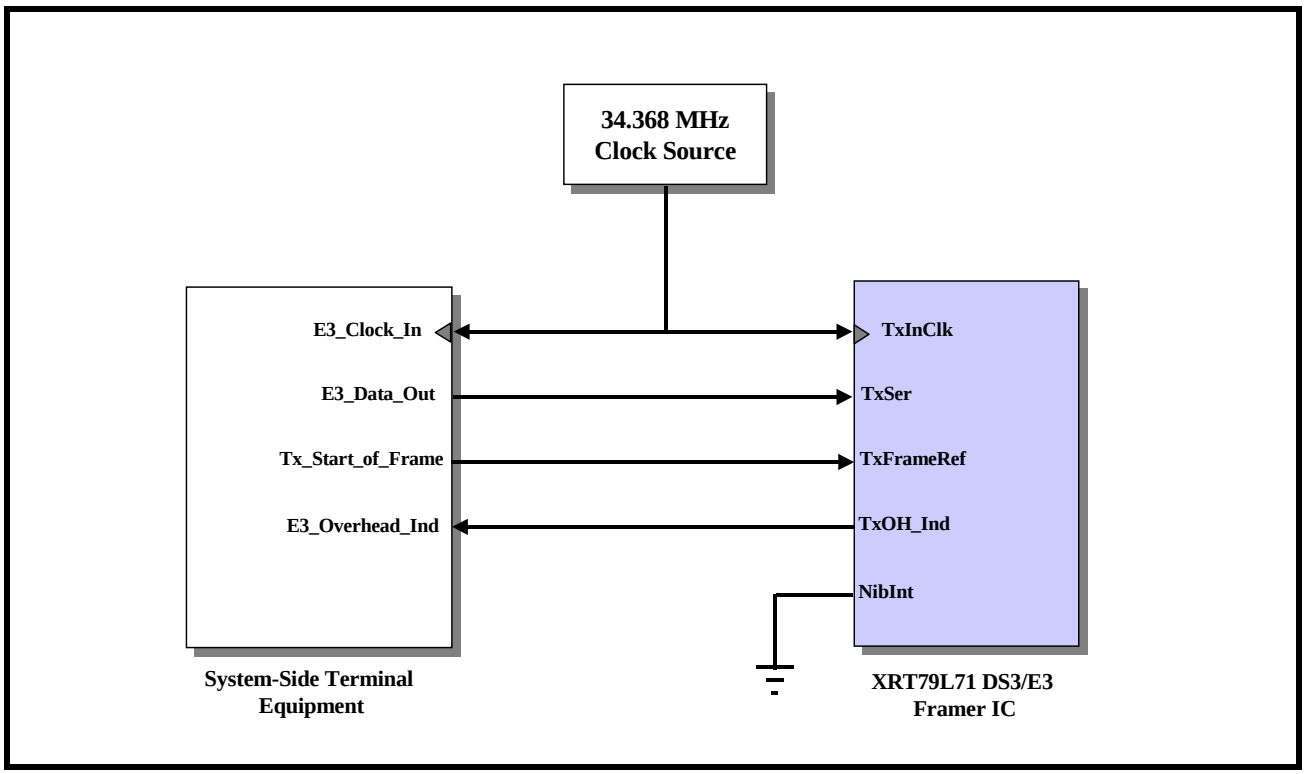
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	0

6.2.1.2 Mode 2 - Serial/Local-Timing/Frame Slave Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in "Mode 2" then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the "TxInClk" input signal as its timing source.
- The Transmit Section of the XRT79L71 will initiate the generate and transmission of a new E3 frame anytime it detects a rising edge at the "TxFrameRef" input pin.
- The XRT79L71 will still pulse the "TxFrame" output pin coincident to whenever the Transmit Payload Data Input Interface block is processing the very last bit within a given E3 frame.

FIGURE 208. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 2 (SERIAL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Mode 2 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 34.368MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More

specifically, this 34.368MHz clock signal will be applied to both the "E3_Clock_In" input of the "System-Side" Terminal Equipment and the "TxInClk" input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data (that is to be transported via the "outbound" E3 data-stream) via its "E3_Data_Out" output pin. The user is advised to design the "System-Side Terminal Equipment" circuitry such that it will update the data (via the "E3_Data_Out" output pin) upon the rising edge of the 34.368MHz clock signal (at its "E3_Clock_In" input pin) as depicted below in **Figure 208**.

The XRT79L71 will latch the contents of the "TxSer" input pin, upon the rising edge of the "TxInClk" signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a "Framing Reference" signal to the XRT79L71 by pulsing its "TxFrameRef" input pin "HIGH" for one bit-period, coincident with the first bit a new "outbound" E3 frame being applied to the "TxSer" input pin. Once the XRT79L71 detects the rising edge of the input at its "TxFrameRef" input pin, it will begin to generate and transmit a new E3 frame.

NOTES:

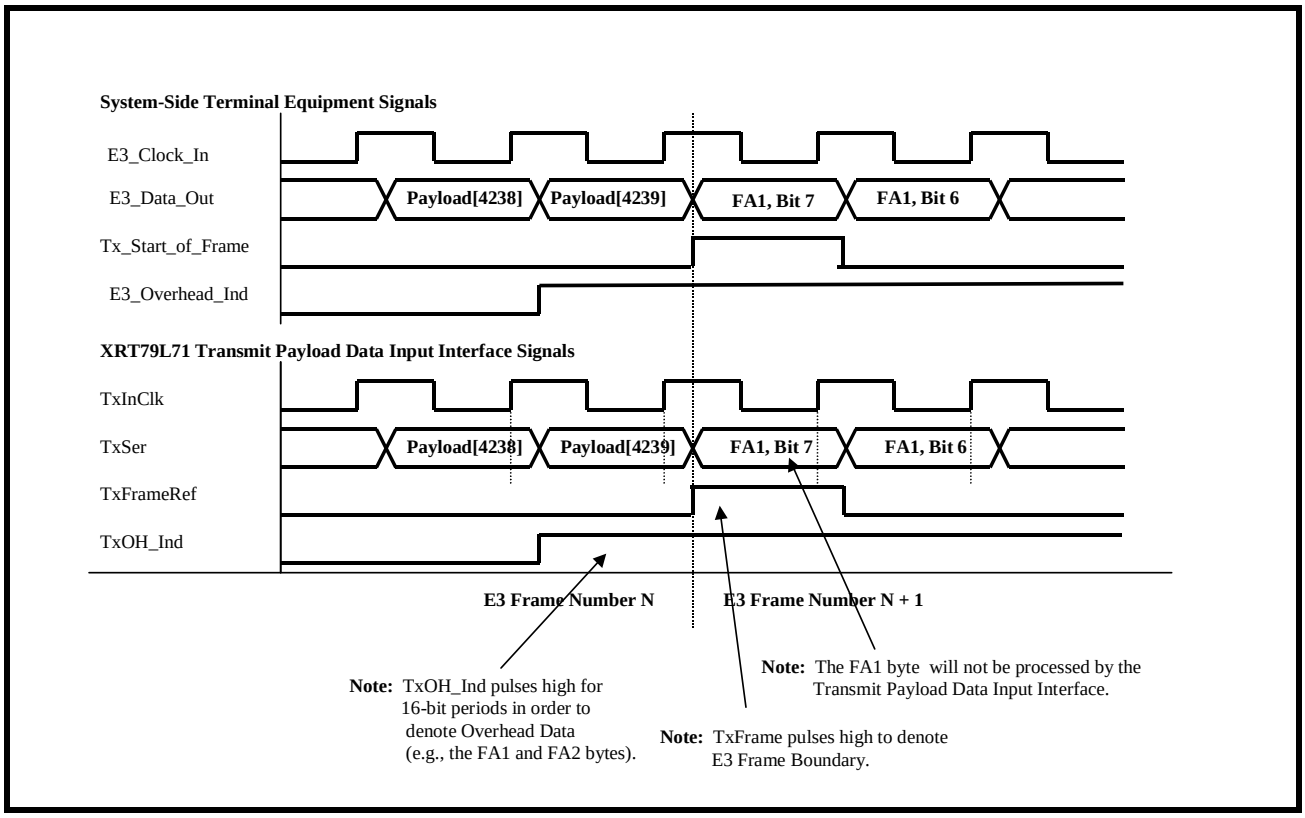
1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of "Frame Generation" and is referred to as the "Frame Master". Since the XRT79L71 does not control or dictate the instant that it will generate a new E3 frame, but is dictated by the "System-Side Terminal Equipment" it is referred to as the "Frame Slave".*
2. *It the XRT79L71 is configured to operate in the "Mode 2" then it is imperative that the "Tx_Start_of_Frame" or "TxFrameRef" signal is synchronized to the "TxInClk" input clock signal. If the user fails to insure that the TxFrameRef input signal is synchronized to the "TxInClk" input clock signal, then the XRT 79L71 will transmit erred E3 data to the remote terminal equipment.*

Finally, the XRT79L71 pulses its "TxOHInd" output pin "HIGH" one bit-period prior to its processing a given overhead bit within the outbound E3 frame. Since the "TxOH_Ind" output pin of the XRT79L71 is electrically connected to the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment); whenever the XRT79L71 pulses its "TxOH_Ind" output pin "HIGH", it will also drive the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment) "HIGH". Whenever the "System-Side Terminal Equipment" detects this pin toggling "high" it should delay transmission of the very next E3 payload bit by one "TxInClk" clock period.

NOTES:

1. *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead bits), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for 16 consecutive bit-periods (when processing the FA1 and FA2 bytes), and it will do so for 8 consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "TxInClk" and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*
2. *2.If the Transmit Payload Data Input Interface block is to be configured into the "Gapped Clock" Mode, then refer to* ***SEE"OPERATING THE TRANSMIT PAYLOAD DATA INPUT INTERFACE IN THE GAPPED CLOCK MODE" ON PAGE 453.****.*

FIGURE 209. ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT/TRANSMIT PAYLOAD DATA INPUT INTERFACE" SIGNALS FOR MODE 2 OPERATION.



Configuring the XRT79L71 to operate in Mode 2 (Serial/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 2 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 208.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "LOW".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[0, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

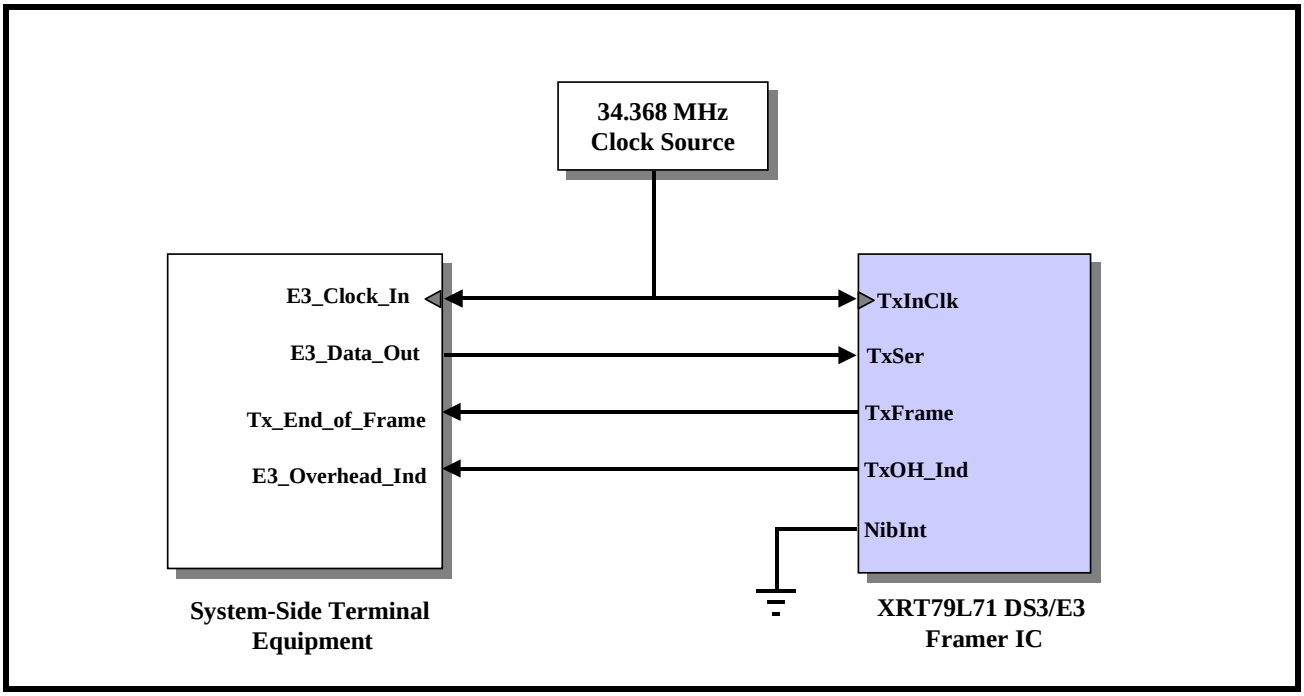
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	0	1

6.2.1.3 Mode 3 - Serial/Local-Timing/Frame Master Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in "Mode 3", then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the "TxInClk" input signal as its timing source.
- Since the XRT79L71 is configured to operate in the "Serial-Mode", it will sample and latch the data, being applied to the "TxSer" input pin upon the rising edge of the "TxInClk" input signal.
- The XRT79L71 will still pulse the "TxFrame" output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given E3 frame.

FIGURE 210. AN ILLUSTRATION AS TO HOW SHOULD INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-SLAVE) MODE OPERATION



Mode 3 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, then one is required to supply a 34.368MHz clock signal to both the System-Side Terminal Equipment circuitry and the XRT79L71. More specifically, this 34.368MHz clock signal will be applied to both the "E3_Clock_In" input of the "System-Side" Terminal Equipment and the "TxInClk" input pin of the XRT79L71, in parallel.

The System-Side Terminal Equipment will serially output the payload data (that is to be transported via the "outbound" E3 data-stream) via its "E3_Data_Out" output pin. The user is advised to design the "System-Side Terminal Equipment" circuitry such that it will update the data (via the "E3_Data_Out" output pin) upon the rising edge of the 34.386MHz clock signal (at its "E3_Clock_In" input pin) as depicted below in Figure 211.

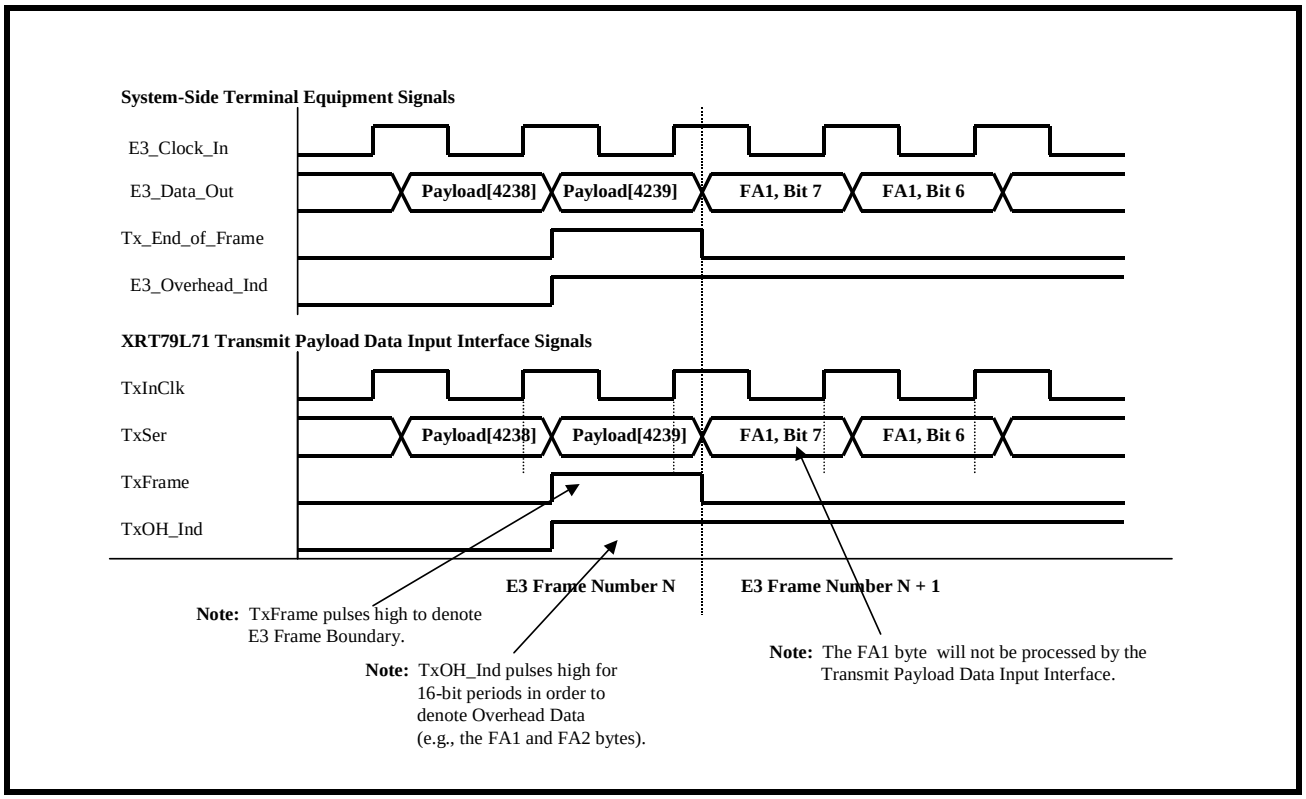
The XRT79L71 will latch the contents of the "TxSer" input pin upon the rising edge of the "TxInClk" signal.

Finally, the XRT79L71 pulses its "TxOH_Ind" output pin "HIGH" one bit-period prior to it processing a given overhead bit within the outbound E3 frame. Since the "TxOH_Ind" output pin of the XRT79L71 is electrically connected to the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment); whenever the XRT79L71 pulses its "TxOH_Ind" output pin "HIGH", it will also drive the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment) "HIGH". Whenever the "System-Side Terminal Equipment" detects this pin toggling "high" it should delay transmission of the very next E3 payload bit by one "TxInClk" clock period.

NOTES:

1. *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead bits), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for 16 consecutive bit-periods (when processing the FA1 and FA2 bytes), and it will do so for 8 consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "TxInClk" and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*
2. *2.In this particular mode, the Transmit Direction circuitry (within the XRT 79L71) is dictating the instant that it will initiate the generation of a new E3 frame; and is referred to as the "Frame Master". As a consequence, this particular mode is referred to as the "Frame Master" Mode.*
3. *3.In contrast to "Mode 2" operation, if the XRT 79L71 is configured to operate in "Mode 3", then it is NOT imperative that the "TxFrameRef" input be synchronized to the "TxInClk" input clock signal. In this case, we recommend that the user tie the "TxFrameRef" input pin to GND.*
4. *4.ITo operate the Transmit Payload Data Input Interface block in the "Gapped-Clock" Mode, Refer to* ***SEE"OPERATING THE TRANSMIT PAYLOAD DATA INPUT INTERFACE IN THE GAPPED CLOCK MODE" ON PAGE 453.****.*

FIGURE 211. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 3 (SERIAL/LOCAL-TIMING/FRAME-MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 3 (Serial/Local-Timing/Frame-Master Mode)

The user can configure the XRT79L71 to operate in Mode 3 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 210.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode.

This can be accomplished by setting the "NibIntf" input pin to a logic "LOW".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[1, X]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

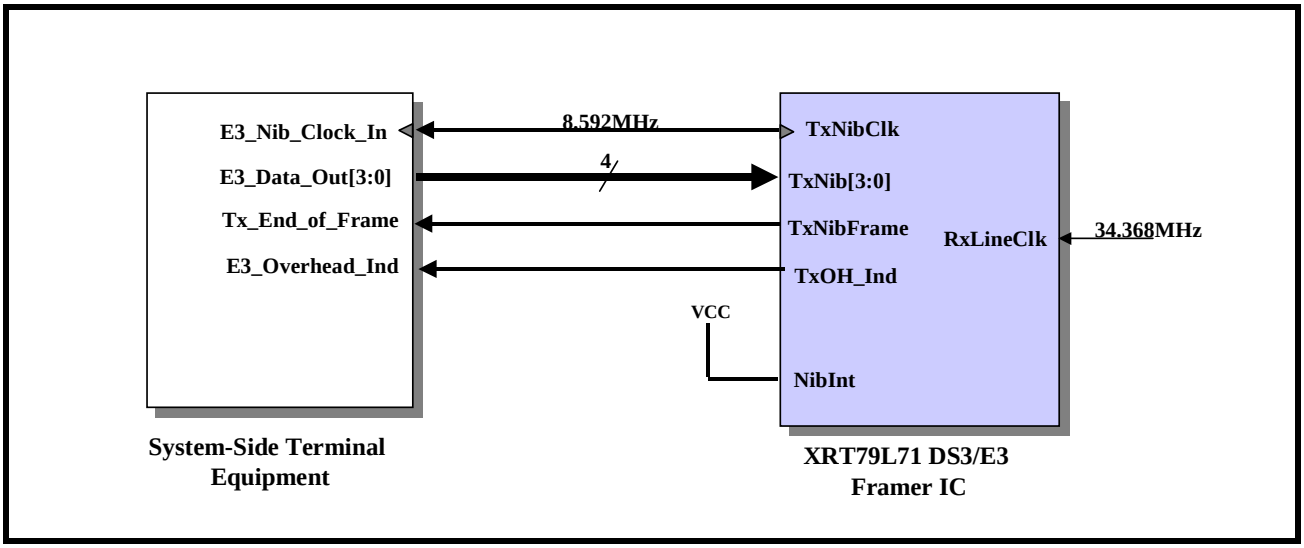
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	0	1	X

6.2.1.4 Mode 4 - Nibble-Parallel/Loop-Timing Mode Operation of the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in "Mode 4" then all of the following is true.

- The XRT79L71 will be configured to operate in the "Loop-Timing" Mode. In other words, the Transmit Section of the XRT79L71 will use the Recovered Clock signal (from the Receive E3 LIU Block) as its timing source.
- In this mode, the XRT79L71 will use the LIU Recovered Clock signal to derive the "TxNibClk" signal.
- For E3 Applications, the "TxNibClk" frequency is exactly one-fourth of the LIU Recovered Clock signal (or 8.592MHz).
- Since the XRT79L71 is configured to operate in the "Nibble-Parallel" Mode, it will sample and latch the data, being applied to the "TxNib[3:0]" input pins upon the third rising edge of the "RxOutClk" output clock signal, following a given rising edge of the "TxNibClk" output clock signal.
- The XRT79L71 will pulse the "TxNibFrame" output pin "high" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble of a given E3 frame.

FIGURE 212. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR MODE 4 (NIBBLE-PARALLEL/ LOOP-TIMING) OPERATION



Mode 4 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of both the 34.368MHz clock signal (via the "RxOutClk" output signal) and a Nibble Clock signal (via the "TxNibClk" output signal).

The "System-Side Terminal Equipment" should output the payload data (that is to be transported via the "outbound" E3 data-stream) in a "Nibble-Parallel" manner via its "E3_Data_Out[3:0]" output pins. The user is advised to design (or configure) the System-Side Terminal Equipment circuitry such that it will update the data (via the "E3_Data_Out[3:0]" output pins) upon the rising edge of the "TxNibClk" clock signal (at its "E3_Nib_Clock_In" input pin) as depicted below in **Figure 213**.

The XRT79L71 will latch the contents of the "TxNib[3:0]" input pins, upon the third rising edge of the "RxOutClk" signal following a given rising edge in the "TxNibClk" signal. The XRT79L71 will indicate that it is processing the very last nibble of a given E3 frame by pulsing its "TxNibFrame" output pin "HIGH" for one nibble-period. Whenever the System-Side Terminal Equipment detects this pulse at its "Tx_Start_of_Frame" input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the "E3_Data_Out[3:0]" output (or "TxNib[3:0]" input pins).

The Transmit Payload Data Input Interface block's handling of E3 Overhead bits when configured to operate in the Nibble-Parallel Mode

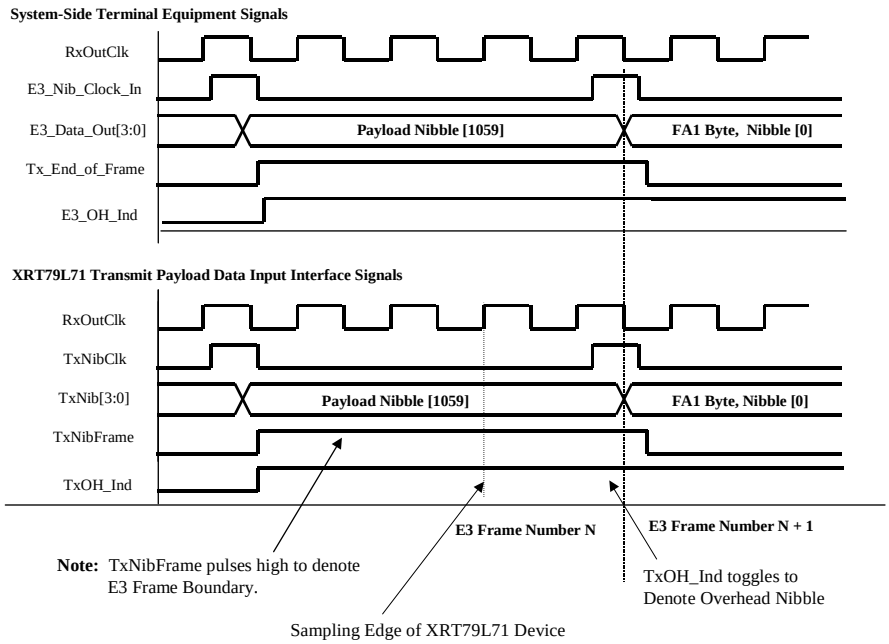
In contrast to the DS3 Framing formats (which are "bit-oriented" framing formats), the E3, ITU-T G.832 framing format is a "byte-oriented" framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface (within the XRT79L71) will be processing an "E3 overhead nibble", and the "TxOH_Ind" output pin (in this case) DOES have meaning. In "Mode 4" Operation, the XRT79L71 will pulse its "TxOH_Ind" output pin "HIGH" one nibble-period prior to the instant that it will process a given "Overhead" nibble within the outbound E3 frame. Since the "TxOH_Ind" output pin of the XRT79L71 is electrically connected to the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment); whenever the XRT79L71 pulses its "TxOH_Ind" output pin "HIGH", it will also drive the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment) "HIGH". Whenever the "System-Side Terminal Equipment" detects this pin toggling "high" it should delay the transmission of the very next E3 payload nibble by one "TxNibClk" clock period.

NOTE: *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead nibbles), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for four (4) consecutive nibble-periods (when processing the FA1 and FA2 bytes) and it will do so for two (2) consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "TxNibClk" and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the "RxOutClk" signal.

FIGURE 213. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 4 (NIBBLE-PARALLEL/LOOP-TIMING) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 4 (Nibble-Parallel/Loop-Timing) Mode

The user can configure the XRT79L71 to operate in Mode 4 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 212.

STEP 2 - Configure the XRT79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "HIGH".

STEP 3 - Configure the XRT79L71 to operate in the Loop-Timing Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[0, 0]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

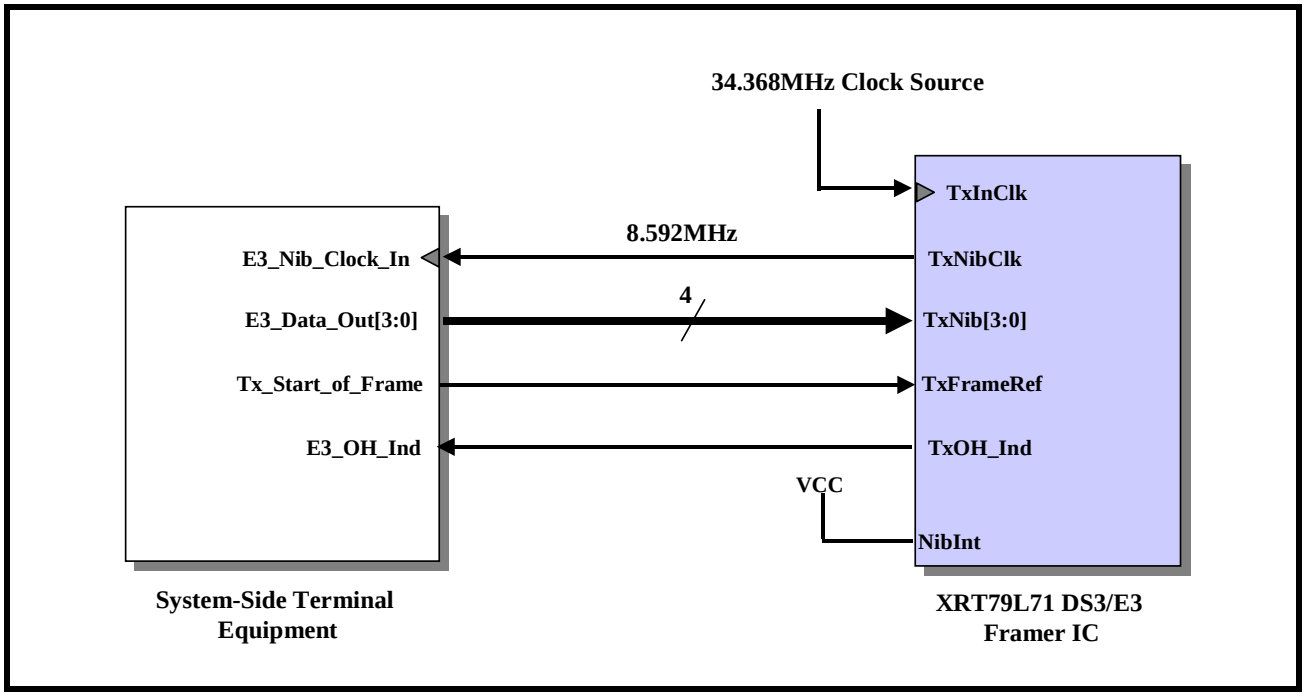
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	1	0	0

6.2.1.5 Mode 5 - Nibble-Parallel/Local-Timing/Frame Slave Mode Operation for the Transmit Payload Data Input Interface Block

If the XRT79L71 is configured to operate in "Mode 5" then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the "TxInClk" input signal has its timing source.
- In this mode, the XRT79L71 will use the TxInClk signal to derive the "TxNibClk" signal.
- For E3 Applications, the "TxNibClk" frequency is exactly one-fourth of the "TxInClk" clock input signal (or 8.592MHz).
- Since the XRT79L71 is configured to operate in the "Nibble-Parallel" Mode, it will sample and latch the data, being applied to the "TxNib[3:0]" input pins upon the third rising edge of the "TxInClk" input clock signal, following a given rising edge of the "TxNibClk" output clock signal.
- The Transmit Section of the XRT79L71 will initiate the generate and transmission of a new E3 frame anytime it detects a rising edge in the "TxFrameRef" input pin.
- The XRT79L71 will pulse the "TxNibFrame" output pin "high" for one nibble-period coincident to whenever the Transmit Payload Data Input Interface is processing the very last nibble within a given E3 frame.

FIGURE 214. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Mode 5 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble Clock signal (via the "TxNibClk" output signal).

NOTE: *For "Mode 5" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The "System-Side Terminal Equipment" should output the payload data (that is to be transported via the "outbound" E3 data-stream) in a "Nibble-Parallel" manner via its "E3_Data_Out[3:0]" output pins. The user is advised to design (or configure) the System-Side Terminal Equipment circuitry such that it will update the data (via the "E3_Data_Out[3:0]" output pins) upon the rising edge of the "TxNibClk" clock signal (at its "E3_Nib_Clock_In" input pin) as depicted below in **Figure 215**.

The XRT79L71 will latch the contents of the "TxNib[3:0]" input pins, upon the third rising edge of the "TxInClk" signal following a given rising edge in the "TxNibClk" signal. In this particular mode, the System-Side Terminal Equipment also has the responsibility of providing a "Framing Reference" signal to the XRT79L71 by pulsing its "TxFrameRef" input pin "HIGH" for one nibble-period, coincident with the first nibble of a new "outbound" E3

frame being applied to the "TxNib[3:0]" input pins. Once the XRT79L71 detects the rising edge of the input at its "TxFrameRef" input pin, it will begin to generate and transmit a new E3 frame.

NOTES:

1. *In this particular mode, the System-Side Terminal Equipment is controlling the start of "Frame Generation" and is referred to as the "Frame Master". Since the XRT79L71 does not control nor dictate the instant that it will generate a new E3 frame, but is driven by the "System-Side" Terminal Equipment, it is referred to as the "Frame Slave".*
2. *If the XRT79L71 is configured to operate in "Mode 5", then it is imperative that the "Tx_Start_of_Frame" or "TxFrameRef" signal is synchronized to the "TxInClk" input clock signal. Failure to do this will result in the transmission of erred E3 data to the remote terminal equipment.*

The Transmit Payload Data Input Interface block's handling of E3 Overhead Bytes when configured to operate in the "Nibble-Parallel" Mode

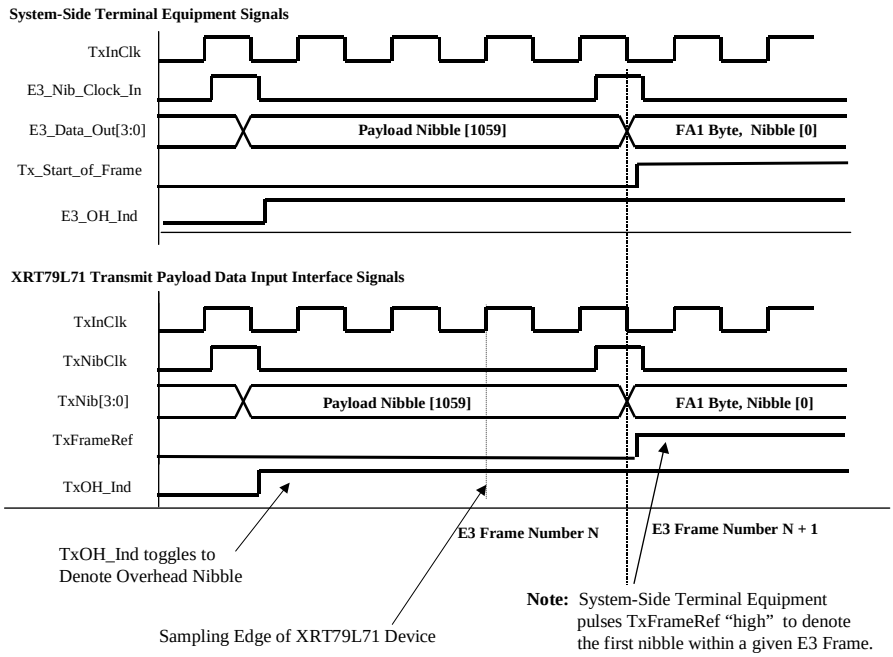
In contrast to the DS3 Framing formats (which are "bit-oriented" framing formats), the E3, ITU-T G.832 framing format is a "byte-oriented" framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface (within the XRT79L71) will be processing an "E3 overhead nibble", and the "TxOH_Ind" output pin (in this case) DOES have meaning. In "Mode 5" Operation, the XRT79L71 will pulse its "TxOH_Ind" output pin "HIGH" one nibble-period prior to the instant that it will process a given "Overhead" nibble within the outbound E3 frame. Since the "TxOH_Ind" output pin of the XRT79L71 is electrically connected to the "E3_Overhead"_Ind" input pin (of the System-Side Terminal Equipment); whenever the XRT79L91 device pulses its "TxOH_Ind" output pin "HIGH", it will also drive the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment) "HIGH". Whenever the "System-Side Terminal Equipment" detects this pin toggling "high" it should delay the transmission of the very next E3 payload nibble by one "TxNibClk" clock period.

NOTE: *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead nibbles), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for four (4) consecutive nibble-periods (when processing the FA1 and FA2 bytes) and it will do so for two (2) consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "TxNibClk" and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the "TxInClk" signal.

FIGURE 215. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 5 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 5 (Nibble-Parallel/Local-Timing/Frame-Slave Mode)

The user can configure the XRT79L71 to operate in Mode 5 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 214.

STEP 2 - Configure the XRT79L71 to operate in the "Nibble-Parallel" Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "HIGH".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Slave Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[0, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

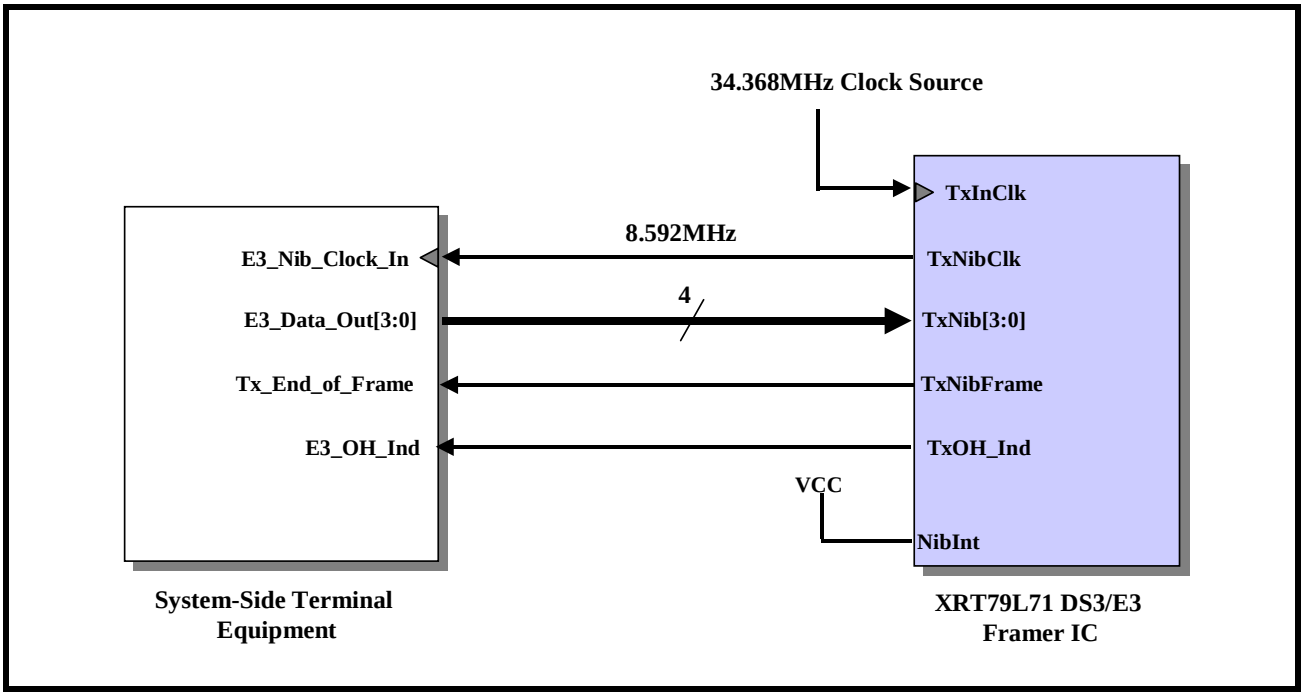
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	1	1	0	1

6.2.1.6 *Mode 6 - Nibble-Parallel/Local-Timing/Frame Master Mode Operation for the Transmit Payload Data Input Interface Block*

If the XRT79L71 is configured to operate in "Mode 6" then all of the following is true.

- The XRT79L71 will be configured to operate in the Local-Timing Mode. In other words, the Transmit Section of the XRT79L71 will use the "TxInClk" input signal as its timing source.
- In this mode, the XRT79L71 will use the "TxInClk" signal to derive the "TxNibClk" signal.
- For E3 Applications, the "TxNibClk" frequency is exactly one-fourth of the "TxInClk" clock signal (or 8.592MHz).
- Since the XRT79L71 is configured to operate in the "Nibble-Parallel" Mode, it will sample and latch the data, being applied to the "TxNib[3:0]" input pin upon the third rising edge of "TxInClk" input clock signal, following a given rising edge of the "TxNibClk" output clock signal.
- The XRT79L71 will pulse the "TxNibFrame" output pin coincident to whenever the Transmit Payload Data Input Interface is processing the very last bit within a given E3 frame.

FIGURE 216. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Mode 6 Operation of the Transmit Payload Data Input Interface Block

Whenever the XRT79L71 has been configured to operate in this mode, it will function as the source of a Nibble-Clock signal (via the "TxNibClk" output signal).

NOTE: *For "Mode 6" Operation, the "TxNibClk" output signal is ultimately derived from the "TxInClk" input signal.*

The "System-Side" Terminal Equipment should output the payload data (that is to be transported via the "outbound" E3 data-stream) in a "Nibble-Parallel" manner via its "E3_Data_Out[3:0]" output pins. The user is advised to design (or configure) the System-Side Terminal Equipment circuitry such that it will update the data (via the "E3_Data_Out[3:0]" output pins) upon the rising edge of the "TxNibClk" clock signal (at its "E3_Nib_Clock_In" input pins) as depicted below in **Figure 217**.

The XRT79L71 will latch the contents of the "TxNib[3:0]" input pins, upon the third rising edge of the "TxInClk" signal following a given rising edge in the "TxNibClk" signal. The XRT79L71 will indicate that it is processing the very last nibble of a given E3 frame by pulsing its "TxNibFrame" output pin "HIGH" for one nibble-period. Whenever the System-Side Terminal Equipment detects this pulse at its "Tx_End_of_Frame" input pin, then it is expected to begin the transmission of the contents of the very next outbound E3 frame, via the "E3_Data_Out[3:0]" output (or "TxNib[3:0]" input pins).

The Transmit Payload Data Input Interface block's handling of E3 Overhead Bytes when configured to operate in the "Nibble-Parallel" Mode

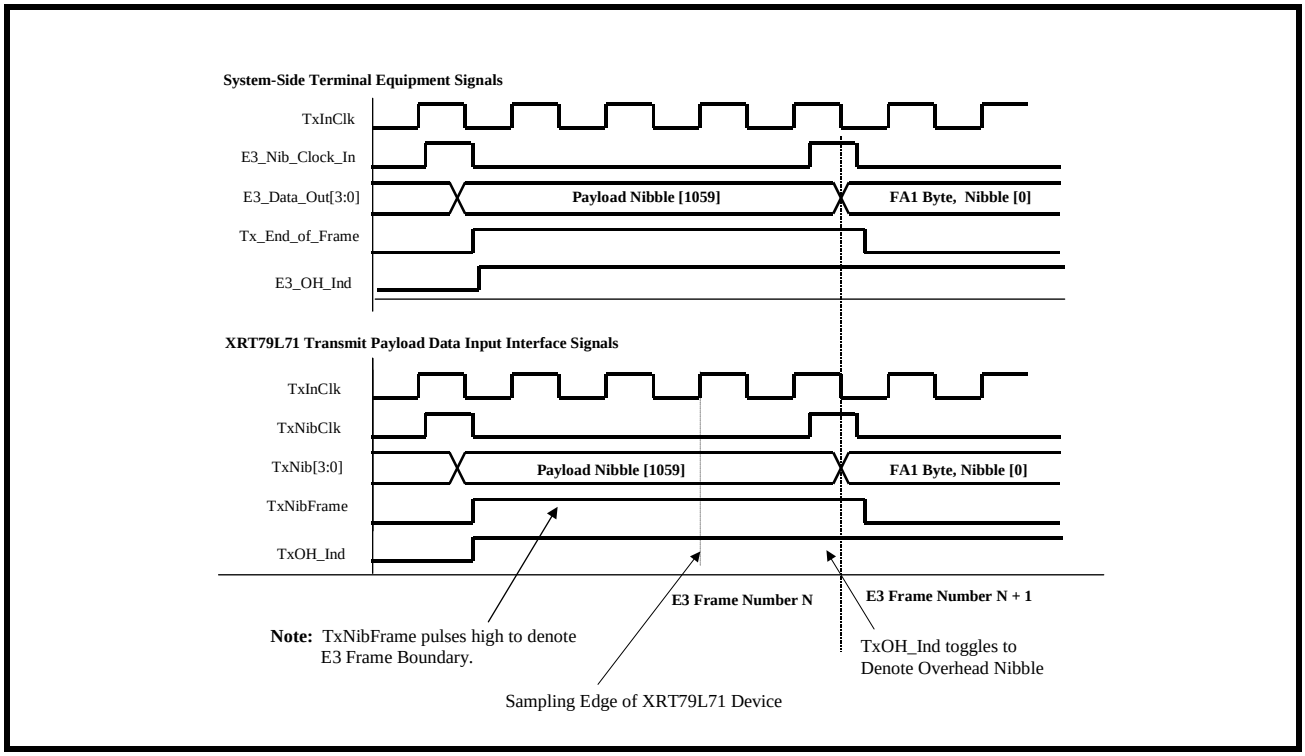
In contrast to the DS3 Framing formats (which are "bit-oriented" framing formats), the E3, ITU-T G.832 framing format is a "byte-oriented" framing format. As a consequence, there will be cases in which the Transmit Payload Data Input Interface (within the XRT79L71) will be processing an "E3 overhead nibble", and the "TxOH_Ind" output pin (in this case) DOES have meaning. In "Mode 6" Operation, the XRT79L71 will pulse its "TxOH_Ind" output pin "HIGH" one nibble-period prior to the instant that it will process a given "Overhead" nibble within the outbound E3 frame. Since the "TxOH_Ind" output pin of the XRT79L71 is electrically connected to the "E3_Overhead"_Ind" input pin (of the System-Side Terminal Equipment); whenever the XRT79L91 device pulses its "TxOH_Ind" output pin "HIGH", it will also drive the "E3_Overhead_Ind" input pin (of the System-Side Terminal Equipment) "HIGH". Whenever the "System-Side Terminal Equipment" detects this pin toggling "high" it should delay the transmission of the very next E3 payload nibble by one "TxNibClk" clock period.

NOTE: *Since the E3, ITU-T G.832 Frame consists of overhead bytes (in lieu of overhead nibbles), whenever the "TxOH_Ind" output pin (of the XRT79L71) pulses "high" it will do so for four (4) consecutive nibble-periods (when processing the FA1 and FA2 bytes) and it will do so for two (2) consecutive bit-periods, when processing the remaining five (5) overhead bytes. Therefore, for the E3, ITU-T G.832 framing format, whenever the "System-Side Terminal Equipment" detects the "TxOH_Ind" output pin being pulled "high", it is expected to (1) continuously sample the state of the "TxOH_Ind" output pin with each rising edge of "TxNibClk" and (2) to NOT transmit an E3 payload bit (to the Transmit Payload Data Input Interface block) until it samples the "TxOH_Ind" output pin toggling "low".*

The Frequency of TxNibClk for E3, Nibble-Parallel Mode Operation

In contrast to that for the DS3 framing formats, for E3 Applications (both ITU-T G.832 and ITU-T G.751 framing formats) the frequency of the TxNibClk clock signal is exactly one-fourth of the frequency of the "TxInClk" signal.

FIGURE 217. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR MODE 6 (NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER) MODE OPERATION



Configuring the XRT79L71 to operate in Mode 6 (Nibble-Parallel/Local-Timing/Frame Master Mode)

The user can configure the XRT79L71 to operate in Mode 6 by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Transmit Payload Data Input Interface in the manner as depicted above in Figure 216.

STEP 2 - Configure the XRT79L71 to operate in the "Nibble-Parallel" Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "HIGH".

STEP 3 - Configure the XRT79L71 to operate in the Local-Timing/Frame Master Mode

This can be accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to "[1, X]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	1	X

6.2.1.7 Operating the Transmit Payload Data Input Interface in the Gapped Clock Mode

In contrast to the DS3 Mode, if the XRT79L71 has been configured to operate in the "E3, ITU-T G.832" Mode, then the Transmit Payload Data Input Interface block can be configured to operate in either the "Gapped-Clock" or the "Non-Gapped Clock" Mode as described below.

6.2.1.7.1 Operating the Transmit Payload Data Input Interface in the "Serial/Gapped-Clock" Mode

This particular section discusses both the "Non-Gapped Clock" and the "Gapped-Clock" Modes of operation of the Transmit Payload Data Input Interface block.

If the Transmit Payload Data Input Interface block has been configured to operate in the "Non-Gapped Clock" Mode

If the Transmit Payload Data Input Interface (within the XRT79L71) has been configured to operate in any one of the "Serial" Modes (e.g., Modes 1 through 3, as described in **SEE"MODE 1 - SERIAL/LOOP-TIMING MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 437.**, **SEE"MODE 2 - SERIAL/LOCAL-TIMING/FRAME SLAVE MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 440.** and **SEE"MODE 3 - SERIAL/LOCAL-TIMING/FRAME MASTER MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 442.**), then we have recommended that the user design or configure their "System-Side" Terminal Equipment to perform the following procedure, when supplying payload data to the "TxSer" input pin.

- Check the state of the "TxOH_Ind" output pin (from the XRT79L71) upon the falling edge of either the "TxInClk" or the "RxOutClk" signal.
- Perform either of the following actions, depending upon the sampled state of the "TxOH_Ind" output pin, as described below.

If TxOH_Ind is sampled "LOW"

Then the "System-Side Terminal Equipment" should proceed to place the very next "payload" bit on the "TxSer" input pin upon the very next rising edge of either the "TxInClk" or "RxOutClk" signal.

If TxOH_Ind is sampled "HIGH"

Then the "System-Side Terminal Equipment should NOT proceed to place the very next "payload" bit on the "TxSer" input pin, upon the very next rising edge of either the "TxInClk" or "RxOutClk" signal. In this case, the

"System-Side Terminal Equipment" should "hold" (or NOT advance the very next payload bit) to the "TxSer" input pin (of the XRT79L71) until it samples the "TxOH_Ind" output "LOW" once again.

In this particular approach, the user must design in the appropriate State Machine circuitry within the "System-Side" Terminal Equipment in order to properly respond to the state of the "TxOH_Ind" output pin, while providing the payload data to the Transmit Payload Data Input Interface. While designing such a "State Machine" into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Transmit Payload Data Input Interface block to operate in the "Gapped-Clock" Mode.

If the Transmit Payload Data Input Interface block has been configured to operate in the "Gapped-Clock" Mode

If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then the role of the "TxOH_Ind" output pin will change from being the "Overhead Indicator" output pin, to now being a "demand-clock" output pin. In other words, If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then it (the Transmit Payload Data Input Interface block) will generate a clock pulse (via the "TxOH_Ind" output pin) if and only if it is ready to accept and process a payload bit. If the Transmit Payload Data Input Interface block is about to process an overhead bit, then it will not generate a clock pulse via the "TxOH_Ind" output pin. This action will result in the Transmit Payload Data Input Interface block generating a "gapped" clock signal via the "TxOH_Ind" output pin (hence the term "Gapped-Clock" Mode).

If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then the "System-Side" Terminal Equipment will be expected to update the data on the "TxSer" input pin (of the XRT79L71) upon the rising edge of the "TxOH_Ind" output signal. The XRT79L71 will sample and latch the "TxSer" data, upon the falling edge of the "TxOH_Ind" output signal. In this case, there is no need to "check the state" of a certain output pin, and then "gate" the placement of the next payload bit (on the TxSer input pin) with the sampled state of this particular signal. The System-Side Terminal Equipment only needs to respond to the rising edge of this particular "Gapped-Clock" signal.

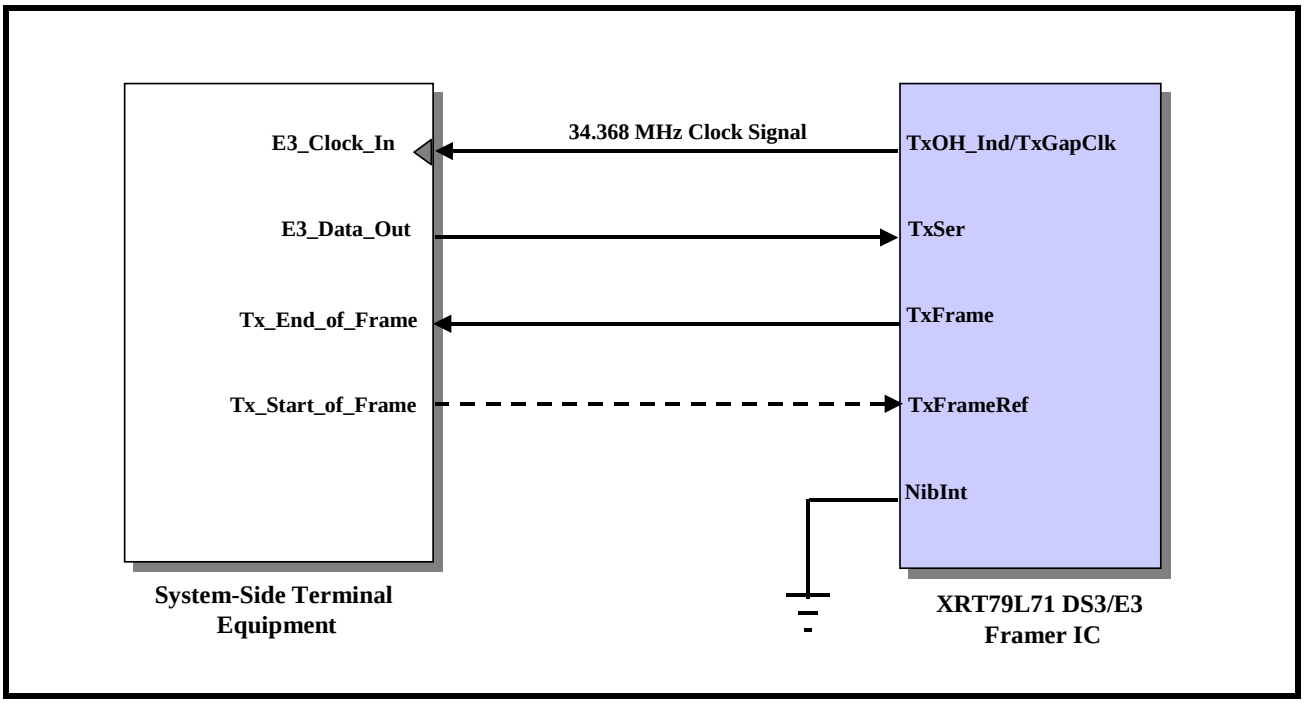
If the XRT79L71 has been configured to operate in the "Loop-Timing" Mode (e.g., Mode 1), then this "Gapped-Clock" signal (from the TxOH_Ind output pin) will be derived from the "LIU Recovered Clock signal (from the Receive DS3 LIU Block). Similarly, if the XRT79L71 has been configured to operate in the "Local-Timing" Mode, then this "Gapped-Clock" signal will be derived form the "TxInClk" input signal.

Configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode

If the Transmit Payload Data Input Interface block is to be configured to operate in the "Gapped-Clock" Mode, then do all of the following.

STEP 1 - Interface the "System-Side" Terminal Equipment to the Transmit Payload Data Input Interface block, in a manner as indicated below in Figure 218.

FIGURE 218. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE" TERMINAL EQUIPMENT TO THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "GAPPED-CLOCK" MODE OPERATIONS



STEP 2 - Set Bit 5 (TxGapped Clock Mode Enable), within the "Framer Test Register" to "1" as depicted below.

Framer Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	0	1	0	0	0	0	0

NOTE: *This setting only configures the Transmit Payload Data Input Interface block to operate in the "Gapped-Clock" Mode. Configuring the Receive Payload Data Input Interface block to operate in the "Gapped-Clock" Mode is discussed separately in* ***SEE"NIBBLE-PARALLEL MODE OPERATION OF THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE" ON PAGE 579.***.

6.2.1.7.2 Operating the Transmit Payload Data Input Interface in the "Nibble-Parallel/Gapped-Clock" Mode

This particular section discusses both the "Non-Gapped Clock" and the "Gapped-Clock" Modes of operation of the Transmit Payload Data Input Interface block.

If the Transmit Payload Data Input Interface block has been configured to operate in the "Non-Gapped Clock" Mode

If the Transmit Payload Data Input Interface (within the XRT79L71) has been configured to operate in any one of the "Nibble-Parallel" Modes (e.g., Modes 4 through 6, as described in **SEE"MODE 4 - NIBBLE-PARALLEL/**

LOOP-TIMING MODE OPERATION OF THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 445., SEE"MODE 5 - NIBBLE-PARALLEL/LOCAL-TIMING/FRAME SLAVE MODE OPERATION FOR THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 447. and SEE"MODE 6 - NIBBLE-PARALLEL/LOCAL-TIMING/FRAME MASTER MODE OPERATION FOR THE TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 450.), then we have recommended that the user design or configure their "System-Side" Terminal Equipment to do the following, when supplying payload data to the "TxNib[3:0]" input pins.

- Check the state of the "TxOH_Ind" output pin (from the XRT79L71) upon the falling edge of the "TxNibClk" signal.
- Perform either of the following actions, depending upon the sampled state of the "TxOH_Ind" output pin, as described below.

If TxOH_Ind is sampled "LOW"

Then the "System-Side Terminal Equipment" should proceed to place the very next "payload" nibble on the "TxNib[3:0]" input pins upon the very next rising edge of either the "TxNibClk" signal.

If TxOH_Ind is sampled "HIGH"

Then the "System-Side Terminal Equipment should NOT proceed to place the very next "payload" nibble on the "TxNib[3:0]" input pins, upon the very next rising edge of either the "TxNibClk" signal. In this case, the "System-Side Terminal Equipment" should "hold" (or NOT advance the very next payload nibble) to the "TxNib[3:0]" input pin (of the XRT79L71) until it samples the "TxOH_Ind" output "LOW" once again.

In this particular approach, the user must design in the appropriate State Machine circuitry within the "System-Side" Terminal Equipment in order to properly respond to the state of the "TxOH_Ind" output pin, while providing the payload data to the Transmit Payload Data Input Interface. While designing such a "State Machine" into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Transmit Payload Data Input Interface block to operate in the "Gapped-Clock" Mode.

If the Transmit Payload Data Input Interface block has been configured to operate in the "Gapped-Clock" Mode

If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then the role of the "TxOH_Ind" output pin will change from being the "Overhead Indicator" output pin, to now being a "demand-clock" output pin. In other words, If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then it (the Transmit Payload Data Input Interface block) will generate a clock pulse (via the "TxOH_Ind" output pin) if and only if it is ready to accept and process a payload nibble. If the Transmit Payload Data Input Interface block is about to process an overhead nibble, then it will not generate a clock pulse via the "TxOH_Ind" output pin. This action will result in the Transmit Payload Data Input Interface block generating a "gapped" clock signal via the "TxOH_Ind" output pin (hence the term "Gapped-Clock" Mode).

If the Transmit Payload Data Input Interface block is configured to operate in the "Gapped-Clock" Mode, then the "System-Side" Terminal Equipment will be expected to update the data on the "TxNib[3:0]" input pin (of the XRT79L71) upon the rising edge of the "TxOH_Ind" output signal. The XRT79L71 will sample and latch the "TxNib[3:0]" data, upon the falling edge of the "TxOH_Ind" output signal. In this case, there is no need to "check the state" of a certain output pin, and then "gate" the placement of the next payload nibble (on the TxNib[3:0] input pins) with the sampled state of this particular signal. The System-Side Terminal Equipment only needs to respond to the rising edge of this particular "Gapped-Clock" signal.

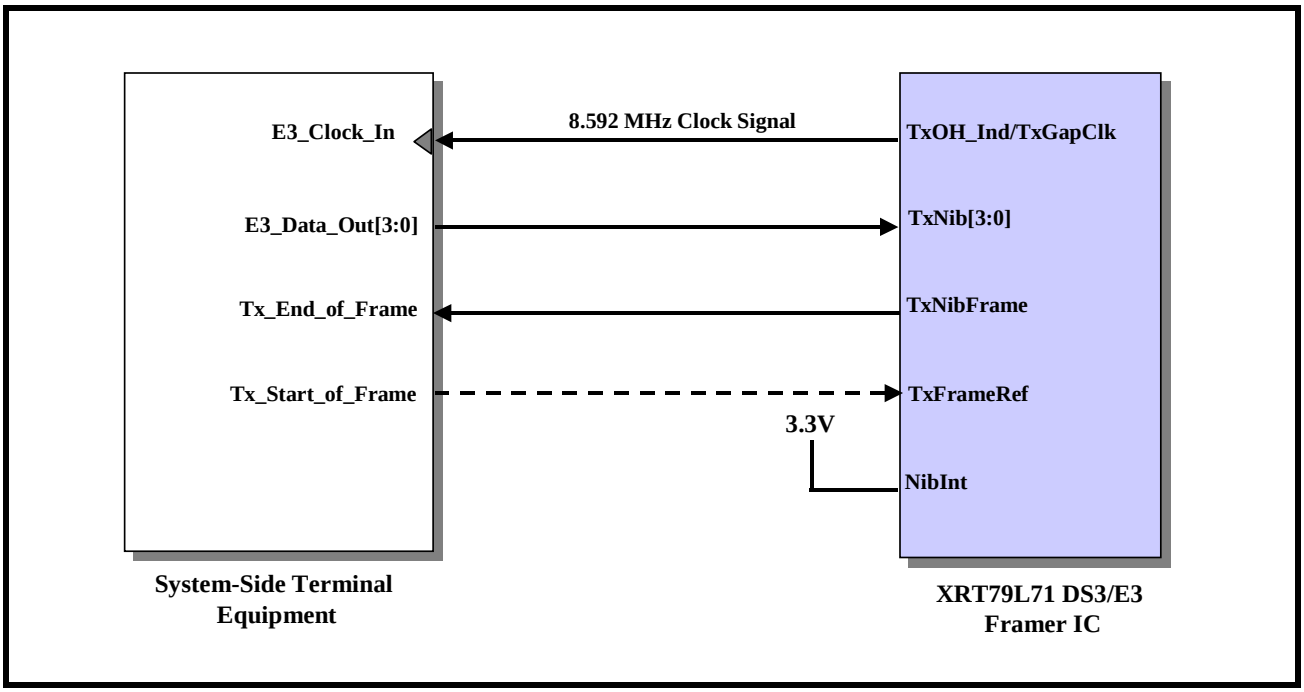
If the XRT79L71 has been configured to operate in the "Loop-Timing" Mode (e.g., Mode 4), then this "Gapped-Clock" signal (from the TxOH_Ind output pin) will be derived from the "LIU Recovered Clock signal (from the Receive E3 LIU Block). Similarly, if the XRT79L71 has been configured to operate in the "Local-Timing" Mode, then this "Gapped-Clock" signal will be derived form the "TxInClk" input signal.

Configuring the Transmit Payload Data Input Interface block to operate in the Gapped-Clock Mode

If the Transmit Payload Data Input Interface block is to be configured to operate in the "Gapped-Clock" Mode, then do all of the following.

STEP 1 - Interface the "System-Side" Terminal Equipment to the Transmit Payload Data Input Interface block, in a manner as indicated below in Figure 219.

FIGURE 219. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE" TERMINAL EQUIPMENT TO THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR NIBBLE-PARALLEL GAPPED-CLOCK MODE OPERATIONS



STEP 2 - Set Bit 5 (TxGapped Clock Mode Enable), within the "Framer Test Register" to "1" as depicted below.

Framer Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	0	1	0	0	0	0	0

NOTE: *This setting only configures the Transmit Payload Data Input Interface block to operate in the "Gapped-Clock" Mode. Configuring the Receive Payload Data Input Interface block to operate in the "Gapped-Clock" Mode is discussed separately in* ***SEE"NIBBLE-PARALLEL MODE OPERATION OF THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE" ON PAGE 579.****.*

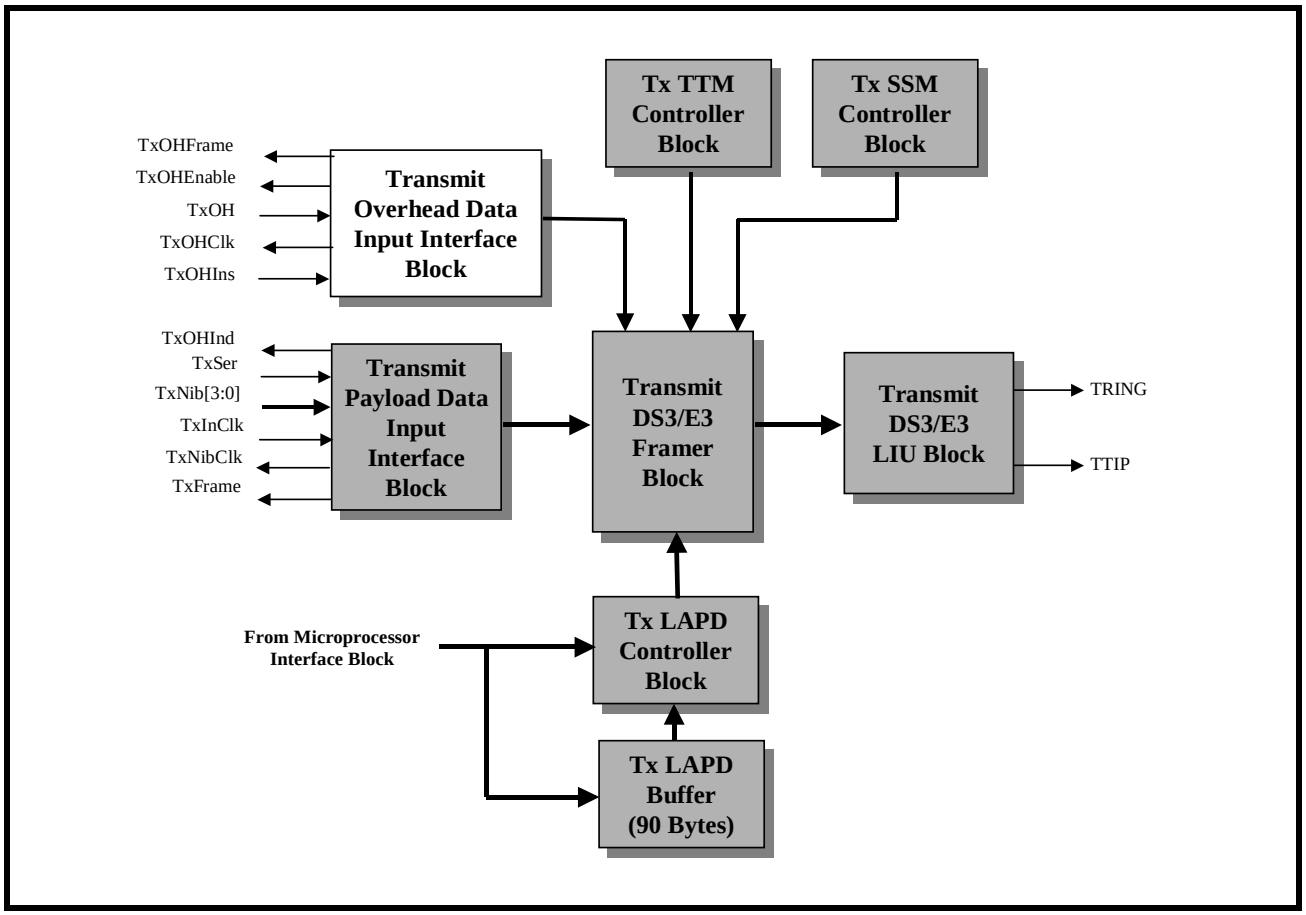
6.2.1.8 Accepting and Inserting E3 Overhead Bytes via the "Transmit Payload Data Input Interface" Block

6.2.2 TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

The Transmit Overhead Data Input Interface block is the second functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3 Clear-Channel Framer Applications. **Figure 220** presents an

illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3 Clear-Channel Framer Mode, with the Transmit Overhead Data Input Interface block highlighted.

FIGURE 220. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK HIGHLIGHTED)



Some Background Information

In order to fully understand the role of the Transmit Overhead Data Input Interface,some background information needs to be discussed first.

As mentioned in SEE"DESCRIPTION OF THE E3, ITU-T G.832 FRAME STRUCTURE AND OVERHEAD BITS" ON PAGE 421., the E3, ITU-T G.832 frame consists of 537 bytes. Of these bytes, 530 bytes are payload bytes and the remaining 7 bytes are overhead bytes. The XRT79L71 has been designed to handle and process both the payload type and overhead bytes of bits/bytes for each E3 frame. Within the XRT79L71, the Transmit Payload Data Input Interface Block (which was discussed in considerable detail in SEE"TRANSMIT PAYLOAD DATA INPUT INTERFACE BLOCK" ON PAGE 432.) has been designed to accept the payload data from the System-Side Terminal Equipment. Likewise, the Transmit Overhead Data Input Interface block has been designed to handle and process the overhead bits.

Accepting and Inserting E3 Overhead Bits via the Transmit Overhead Data Input Interface

By default, the Transmit E3 Framer block will be configured to internally generate and insert all of the overhead bytes within its outbound E3 data-stream. More specifically, the Transmit E3 Framer block will internally generate the E3 overhead bytes by doing all of the following, as presented below in Table 58.

TABLE 58: HOW THE TRANSMIT E3 FRAMER BLOCK INTERNALLY GENERATES EACH OF THE OVERHEAD BITS/BYTES - E3, ITU-T G.832 APPLICATIONS

BIT NAME	BIT/BYTE DESCRIPTION	HOW OVERHEAD BIT IS INTERNALLY GENERATED BY THE TRANSMIT E3 FRAMER BLOCK
FA1 Byte[7:0]	Framing Alignment Byte # 1	This byte is set to the value "0xF6"
FA2 Byte[7:0]	Framing Alignment Byte # 2	This byte is set to the value "0x28"
EM Byte[7:0]	Error Monitor Byte	The Transmit E3 Framer will perform a BIP-8 calculation over an entire E3 frame. The results of this calculation will be inserted into the EM byte within the very next frame.
TR Byte[7:0]	Trail Trace Byte	These bits carry the Trail Trace Message that is generated by the Trail Trace Message Controller within the Transmit Section of the XRT79L71.
FERF/RDI - MA Byte[1]	Far-End Receive Failure/ Remote Defect Indicator	Either Software Controlled or automatically set to "1" whenever the corresponding Receive E3 Framer block declare the LOS, LOF/OOF or AIS defect condition.
FEBE/REI - MA Byte[2]	Far-End Block Error/Remote Error Indicator	Either Software Controller or automatically set to "1" whenever the corresponding Receive E3 Framer block detects an EM byte error within its incoming E3 data-stream.
Payload_Type[2:0] - MA Byte[3:5]	Payload Type Indicator	These bits are Software-Controlled via an on-chip Register.
SSM Multi-Frame Indicator[1:0] - MA Byte[6:7]	SSM Multi-Frame Indicator	
SSM Bit - MA Byte[8]	SSM Channel	This bit carries the Synchronization Status Message that is generated by the SSM Controller within the Transmit Section of the XRT79L71
NR Byte[7:0]	Network Operator Byte	These bits are either Software-Controlled via an on-chip Register or they can be configured to carry the PMDL/LAPD Message that is generated by the Transmit LAPD Controller within the Transmit Section of the XRT79L71.
GC Byte[7:0]	General Purpose Communication Byte	These bits are either Software-Controlled via an on-chip Register or they can be configured to carry the PMDL/LAPD Message that is generated by the Transmit LAPD Controller within the Transmit Section of the XRT79L71.

However, the Transmit Section of the XRT79L71 can also be configured to externally accept values via a certain input port and to insert this data into certain user specified overhead bits, within the outbound E3 data-stream. The XRT79L71 permits the user to implement this overhead bit insertion by either of the following methods.

- By configuring the Transmit Section of the XRT79L71 to accept E3 overhead data via the Transmit Payload Data Input Interface block (as was discussed in **SEE"ACCEPTING AND INSERTING E3 OVERHEAD BYTES VIA THE "TRANSMIT PAYLOAD DATA INPUT INTERFACE" BLOCK" ON PAGE 457.**).
- By configuring the Transmit Section of the XRT79L71 to accept DS3 overhead data via the Transmit Overhead Data Input Interface.

The purpose of the Transmit Overhead Data Input Interface block is to accept overhead data from some system-side or up-stream source and to overwrite the contents of the overhead bits/bytes, within the outbound E3 data-stream with these particular overhead bit/byte values.

In order to accomplish this, the Transmit Overhead Data Input Interface block has numerous input and output pins. Table 59 presents a list and a brief definition of each of these pins.

TABLE 59: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
TxOHEnable	A12	O	**Transmit Overhead Data - Enable Output Pin:** The Transmit Overhead Data Input Interface will assert this signal (e.g., pulse it "High") for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface block is processing an overhead bit. This output pin will remain "Low" at all other times.
TxInClk	C10	I	**Transmit Section - Timing Reference Clock Input Pin:** If the XRT79L71 has been configured to operate in the Local-Timing Mode, then this input pin will function as the timing source for the Transmit Circuitry within the XRT79L71.
TxOHFrame	B11	O	**Transmit Overhead Input Interface Enable Input Pin:** This output pin pulses "High" whenever the Transmit Overhead Data Input Port is processing the last bit within a given E3 frame.
TxOHIns	D10	I	**Transmit Overhead Data - Insert Enable Input Pin:** Asserting this input pin (e.g., setting it "High") enables the Transmit Overhead Data Input Interface to accept overhead data from the System-Side Terminal Equipment. In other words, while this input pin is "High", the Transmit Overhead Data Input Interface will sample the data on the TxOH input pin, upon the falling edge of either the TxInClk or the TxOHClk clock signals. Setting this input pin "Low" configures the Transmit Overhead Data Input Interface block to NOT sample (e.g., ignore) the data on the TxOH input pin upon the falling edge of either the TxInClk or the TxOHClk clock signals. ***NOTE:*** *If the System-Side Terminal Equipment attempts to insert an overhead bit that cannot be accepted by the Transmit Overhead Data Input Interface block (e.g., if the System-Side Terminal Equipment asserts the TxOHIns input pin at a time whenever one of the non-insertable overhead bits are being processed), then this particular insertion effort will be ignored.*
TxOH	C11	I	**Transmit Overhead Data Input Interface - Overhead Data Input Pin:** The Transmit Overhead Data Input Interface block accepts the overhead data via this input pin and inserts this data into the appropriate overhead bit-position within the very next outbound E3 frame. If the TxOHIns input pin is pulled "High", then the Transmit Overhead Data Input Interface block will sample the data, residing on this input pin, upon either the rising edge of TxInClk or the falling edge of TxOHClk depending upon the Insertion Method used. If the TxOHIns input pin is pulled "Low", then the Transmit Overhead Data Input Interface block will NOT sample the data, residing on this input pin, upon the rising edge of TxInClk nor on the falling edge of TxOHClk.
TxOHClk	B12	O	**Transmit Overhead Clock Output:** This output pin functions as the "Transmit Overhead Data Input Interface" clock signal. If the Transmit Overhead Data Input Interface block is enabled by asserting the "TxOHIns" input pin, then the Transmit Overhead Data Input Interface block will sample and latch the data (residing on the "TxOH" input pin) upon the falling edge of this signal.

The Two Methods of Inserting Overhead Data Into the Transmit Overhead Data Input Interface Block

There are two methods that can be used to insert the overhead data into the Transmit Overhead Data Input Interface Block. One method is referred to as Method 1 or the TxOHClk Method, and the other method is referred to as Method 2 or the TxInClk Method. Each of these methods is described in considerable detail below.

6.2.2.1 Operating the Transmit Overhead Data Input Interface block using Method 1 - The TxOHClk Method

This particular method is referred to as the TxOHClk method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHClk clock output signal from the Transmit Overhead Data Input Interface block to clock overhead data onto the TxOH input pin.

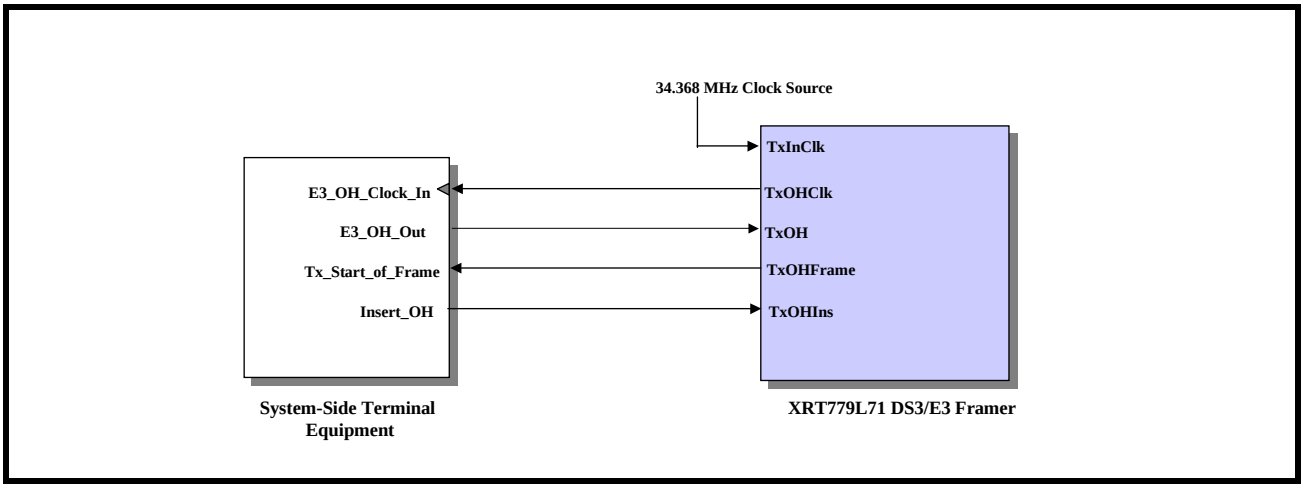
b. The Transmit Overhead Data Input Interface block will use the falling edge of the TxOHClk clock output signal to sample and latch the data residing on the TxOH input pin.

To use Method 1, the System-Side Terminal Equipment will need to interface to the following Transmit Overheads Data Input Interface pins.

- TxOH
- TxOHClk
- TxOHFrame
- TxOHIns

If Method 1 is used, the user must design their system such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in Figure 221.

FIGURE 221. ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 1



Method 1 Operation of the Transmit Overhead Data Input Interface Block

To operate the Transmit Overhead Data Input Interface block, design/configure the System-Side Terminal Equipment to continuously and repeatedly execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the state of the TxOHFrame output pin from the XRT79L71 upon the rising edge of the TxOHClk clock signal which is also output from the XRT79L71. Whenever the System-Side Terminal Equipment samples the TxOHFrame output pin "High" then it will know that the Transmit Overhead Data Input Interface block is ready to accept overhead bits for a new E3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHFrame signal, it must also keep track of the number of rising edges within the TxOHClk signal that have occurred since the last time TxOHFrame was sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which

overhead bit is currently being processed by the Transmit Overhead Data Input Interface block at any given TxOHClk clock period. When the System-Side Terminal Equipment knows which overhead bit is being processed within a given TxOHClk clock period, it can decide when to insert the appropriate bit-value into the Transmit Overhead Data Input Interface block and in-turn, force the Transmit DS3/E3 Framer block to insert this same bit value into the appropriate overhead bit-position within the outbound E3 data-stream. From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin of the XRT79L71.

Table 60 relates the number of rising clock edge, within the TxOHClk output signal, since the TxOHFrame output signal was sampled "High" to the E3 Overhead Bit being processed by the Transmit Overhead Data Input Interface block. The user can use this table as a guide for inserting the appropriate overhead bits, within the outbound E3 data-stream, for Method 1.

TABLE 60: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
0 (TxOHClk, Clock Edge is coincident with the TxOHFrame signal being sampled "High")	FA1 Byte, Bit 1 (MSB)	NO
1	FA1 Byte, Bit 2	NO
2	FA1 Byte, Bit 3	NO
3	FA1 Byte, Bit 4	NO
4	FA1 Byte, Bit 5	NO
5	FA1 Byte, Bit 6	NO
6	FA1 Byte, Bit 7	NO
7	FA1 Byte, Bit 8 (LSB)	NO
8	FA2 Byte, Bit 1 (MSB)	NO
9	FA2 Byte, Bit 2	NO
10	FA2 Byte, Bit 3	NO
11	FA2 Byte, Bit 4	NO
12	FA2 Byte, Bit 5	NO
13	FA2 Byte, Bit 6	NO
14	FA2 Byte, Bit 7	NO
15	FA2 Byte, Bit 8 (LSB)	NO
16	EM Byte, Bit 1 (MSB)	NO
17	EM Byte, Bit 2	NO
18	EM Byte, Bit 3	NO
19	EM Byte, Bit 4	NO

TABLE 60: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
20	EM Byte, Bit 5	NO
21	EM Byte, Bit 6	NO
22	EM Byte, Bit 7	NO
23	EM Byte, Bit 8 (LSB)	NO
24	TR Byte, Bit 1 (MSB)	YES
25	TR Byte, Bit 2	YES
26	TR Byte, Bit 3	YES
27	TR Byte, Bit 4	YES
28	TR Byte, Bit 5	YES
29	TR Byte, Bit 6	YES
30	TR Byte, Bit 7	YES
31	TR Byte, Bit 8 (LSB)	YES
32	MA Byte, Bit 1 (FERF/RDI)	YES
33	MA Byte, Bit 2 (FEBE/REI)	YES
34	MA Byte, Bit 3	YES
35	MA Byte, Bit 4	YES
36	MA Byte, Bit 5	YES
37	MA Byte, Bit 6	YES
38	MA Byte, Bit 7	YES
39	MA Byte, Bit 8 (LSB)	YES
40	NR Byte, Bit 1 (MSB)	YES
41	NR Byte, Bit 2	YES
42	NR Byte, Bit 3	YES
43	NR Byte, Bit 4	YES
44	NR Byte, Bit 5	YES
45	NR Byte, Bit 6	YES
46	NR Byte, Bit 7	YES
47	NR Byte, Bit 8 (LSB)	YES
48	GC Byte, Bit 1 (MSB)	YES
49	GC Byte, Bit 2	YES

TABLE 60: THE RELATIONSHIP BETWEEN THE NUMBER OF RISING CLOCK EDGES IN THE TXOHCLK SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN TXOHCLK SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
50	GC Byte, Bit 3	YES
51	GC Byte, Bit 4	YES
52	GC Byte, Bit 5	YES
53	GC Byte, Bit 6	YES
54	GC Byte, Bit 7	YES
55	GC Byte, Bit 8 (LSB)	YES

NOTE: *The shaded rows designate those Overhead bits that the XRT79L71 can be used to insert into the outbound E3 data-stream via the Transmit Overhead data Input Interface block. The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound E3 data-stream.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of clock edges from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal by pulling it "High". Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit to be inserted into the outbound E3 data-stream onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data residing on the TxOH input pin upon the very next falling edge of the TxOHClk output signal.

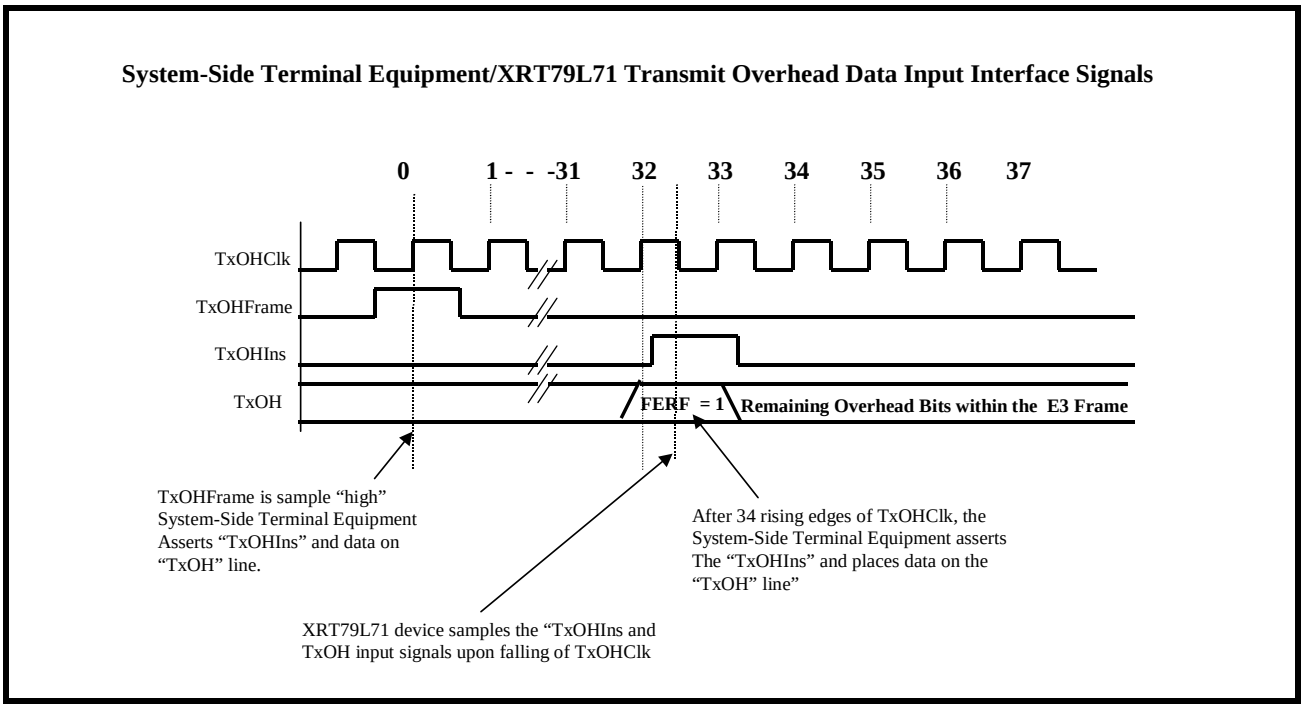
TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until the next rising edge of TxOHClk is detected. Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHClk period come up, for inserting an overhead bit into the outbound E3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 1 in order to transmit the FERF/RDI indicator to the remote terminal equipment.

For E3, ITU-T G.82 applications, the FERF/RDI indication is transmitted by setting the FERF/RDI bit-field (Bit 1 within the MA-byte, of each outbound E3 frame) to "1".

If we were to assume that the connection between the System-Side Terminal Equipment and the XRT79L71 Transmit Overhead Data Input Interface block is as illustrated in **Figure 222**, then **Figure 223** presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface, when using Method 1.

FIGURE 222. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 1)



In Figure 222, the System-Side Terminal Equipment samples the TxOHFrame signal being "High" during Rising Edge Clock Edge # 0 with the TxOHClk signal. At this point, the System-Side Terminal Equipment knows that the XRT79L71 is just about to process the very first overhead bit within a given E3 frame. According to Table 60, the FERF/RDI bit will be processed at TxOHClk clock edge # 32. In order to facilitate the transmission of the FERF/RDI indicator, the System-Side Terminal Equipment must (1) wait for 32 TxOHClk clock periods, and (2) then set this particular bit-field to "1". Once the System-Side Terminal Equipment has waited through the 32 TxOHClk clock periods, it sets the FERF/RDI bit-field to "1" by implementing the following two tasks concurrently.

TASK 1 - The System Side Terminal Equipment asserts the TxOHIns input pin by setting it "High".

TASK 2 - The System-Side Terminal Equipment sets the TxOH input pin to "1".

After the System-Side Terminal Equipment has executed these two tasks, the XRT79L71 will sample both the TxOHIns and the TxOH input pins being "High" during the very next falling edge of TxOHClk. Once the XRT79L71 has sampled these two signals, it will insert a "1" into the FERF/RDI bit-position within the outbound E3 frame.

Upon detection of the very next rising edge of the TxOHClk clock signal (designated as Clock Edge # 33, in Figure 222), the System-Side Terminal Equipment will negate or de-assert the TxOHIns input signal (e.g., toggle it "Low") and cease inserting data into the Transmit Overhead Data Input Interface for the remainder of this particular E3 frame period. Afterwards, the System-Side Terminal Equipment will repeat all of the steps that have been outlined in this case study.

6.2.2.2 Operating the Transmit Overhead Data Input Interface block using Method 2 - The TxInClk/TxOHEnable Method

This particular method is referred to as the TxInClk/TxOHEnable method for the following reasons.

a. The System-Side Terminal Equipment will use the TxOHEnable output pin from the Transmit Overhead Data Input Interface block to keep track of which overhead bit is being processed by the Transmit Overhead Data Input Interface block at any given time.

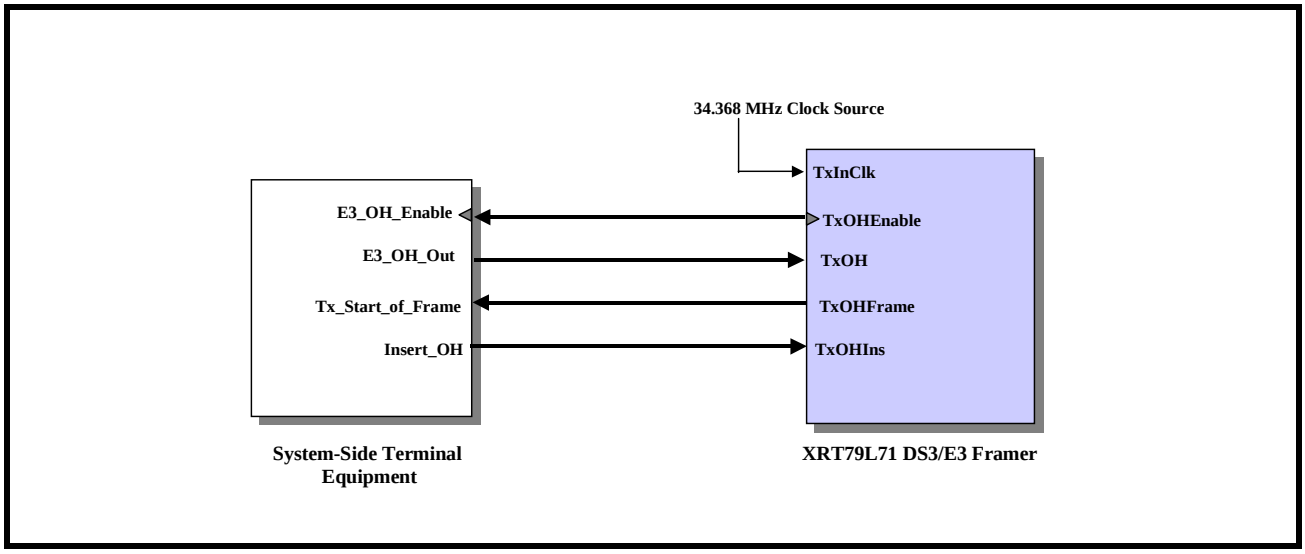
b. The Transmit Overhead Data Input Interface block will use the rising edge of TxInClk in order to sample and latch the data residing on the TxOH input pin.

If Method 2 is used, the System-Side Terminal Equipment will need to interface to the following Transmit Overhead Data Input Interface pins.

- TxOH
- TxOHFrame
- TxInClk
- TxOHEnable
- TxOHIns

To use Method 2 the user must design their system such that the System-Side Terminal Equipment will be interfaced to the Transmit Overhead Data Input Interface block, in a manner as presented below in **Figure 223**.

FIGURE 223. ILLUSTRATION OF HOW TO INTERFACE THE SYSTEM-SIDE TERMINAL EQUIPMENT TO THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK WHEN USING METHOD 2



Method 2 Operation of the Transmit Overhead Data Input Interface Block

Tto operate the Transmit Overhead Data Input Interface block per Method 2, design/configure the System-Side Terminal Equipment to continuously and repeatedly execute the following tasks.

TASK # 1: The System-Side Terminal Equipment must sample the states of both the TxOHFrame and the TxOHEnable output pins from the XRT79L71 upon the falling edge of the TxInClk clock input signal. The XRT79L71 will pulse the TxOHEnable output pin "High" for one TxInClk period, just prior to the instant that the Transmit Overhead Data Input Interface block is processing an overhead bit. Therefore, if the System-Side Terminal Equipment samples the TxOHEnable output pin "High", then it knows that an overhead bit insertion opportunity via the Transmit Overhead Data Input Interface block is just about to occur. If the System-Side Terminal Equipment samples both the TxOHEnable and the TxOHFrame output pins "High" at the same time, then it knows that the Transmit Overhead Data Input Interface block is just about to process the very first overhead bit within a new E3 frame.

TASK # 2: As the System-Side Terminal Equipment samples the TxOHEnable and TxOHFrame signals, it must also keep track of the number of times that the TxOHEnable output pin has been sampled "High" since the last time both the TxOHEnable and the TxOHFrame output pins have been sampled "High". By doing this, the System-Side Terminal Equipment will be able to keep track of which overhead bits are being processed by the Transmit Overhead Data Input Interface block at any give TxOHEnable assertion. When the System-Side Terminal Equipment knows which overhead bit is currently being processed within a given TxOHEnable

assertion period, it can decide when to insert the appropriate bit-value to the Transmit Overhead Data Input Interface block and in-turn, force the Transmit DS3/E3 Framer block to insert this bit into the appropriate overhead bit-position within the outbound E3 data-stream. From all of this, the System-Side Terminal Equipment will know when it should assert the TxOHIns input pin and place the appropriate value on the TxOH input pin of the XRT79L71.

Table 61 relates the number of TxOHEnable output pulses that have occurred since both the TxOHFrame and the TxOHEnable pins were both sampled "High", to the E3 Overhead bit being processed by the Transmit Overhead Data Input Interface block. The user can use this table as a guide for inserting the appropriate overhead bits, within the outbound E3 data-stream for Method 2.

TABLE 61: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
0 (TxOHEnable and TxOHFrame are sampled "High" simultaneously)	FA1 Byte, Bit 1 (MSB)	NO
1	FA1 Byte, Bit 2	NO
2	FA1 Byte, Bit 3	NO
3	FA1 Byte, Bit 4	NO
4	FA1 Byte, Bit 5	NO
5	FA1 Byte, Bit 6	NO
6	FA1 Byte, Bit 7	NO
7	FA1 Byte, Bit 8 (LSB)	NO
8	FA2 Byte, Bit 1 (MSB)	NO
9	FA2 Byte, Bit 2	NO
10	FA2 Byte, Bit 3	NO
11	FA2 Byte, Bit 4	NO
12	FA2 Byte, Bit 5	NO
13	FA2 Byte, Bit 6	NO
14	FA2 Byte, Bit 7	NO
15	FA2 Byte, Bit 8 (LSB)	NO
16	EM Byte, Bit 1 (MSB)	NO
17	EM Byte, Bit 2	NO
18	EM Byte, Bit 3	NO
19	EM Byte, Bit 4	NO
20	EM Byte, Bit 5	NO
21	EM Byte, Bit 6	NO
22	EM Byte, Bit 7	NO

TABLE 61: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
23	EM Byte, Bit 8 (LSB)	NO
24	TR Byte, Bit 1 (MSB)	YES
25	TR Byte, Bit 2	YES
26	TR Byte, Bit 3	YES
27	TR Byte, Bit 4	YES
28	TR Byte, Bit 5	YES
29	TR Byte, Bit 6	YES
30	TR Byte, Bit 7	YES
31	TR Byte, Bit 8 (LSB)	YES
32	MA Byte, Bit 1 (FERF/RDI)	YES
33	MA Byte, Bit 2 (FEBE/REI)	YES
34	MA Byte, Bit 3	YES
35	MA Byte, Bit 4	YES
36	MA Byte, Bit 5	YES
37	MA Byte, Bit 6	YES
38	MA Byte, Bit 7	YES
39	MA Byte, Bit 8 (LSB)	YES
40	NR Byte, Bit 1 (MSB)	YES
41	NR Byte, Bit 2	YES
42	NR Byte, Bit 3	YES
43	NR Byte, Bit 4	YES
44	NR Byte, Bit 5	YES
45	NR Byte, Bit 6	YES
46	NR Byte, Bit 7	YES
47	NR Byte, Bit 8 (LSB)	YES
48	GC Byte, Bit 1 (MSB)	YES
49	GC Byte, Bit 2	YES
50	GC Byte, Bit 3	YES
51	GC Byte, Bit 4	YES
52	GC Byte, Bit 5	YES

TABLE 61: THE RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE TXOHENABLE SIGNAL, SINCE THE TXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS CURRENTLY BEING PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK

NUMBER OF PULSES IN TXOHENABLE, SINCE TXOHFRAME BEING SAMPLED "HIGH"	THE OVERHEAD BIT TO BE PROCESSED BY THE TRANSMIT OVERHEAD DATA INPUT INTERFACE BLOCK	CAN THIS OVERHEAD BIT BE ACCEPTED BY THE XRT79L71, AND INSERTED INTO THE OUTBOUND E3 DATA-STREAM?
53	GC Byte, Bit 6	YES
54	GC Byte, Bit 7	YES
55	GC Byte, Bit 8 (LSB)	YES

NOTE: *The shaded rows designate those Overhead bits that the XRT79L71 can be inserted into the outbound E3 data-stream via the Transmit Overhead Data Input Interface block. The un-shaded rows designate those Overhead bits that the XRT79L71 CANNOT insert into the outbound E3 data-stream.*

TASK # 3: After the System-Side Terminal Equipment has waited the appropriate number of TxOHEnable pulses from the TxOHFrame signal being sampled "High", it should assert the TxOHIns input signal by pulling it "High". Concurrently, the System-Side Terminal Equipment should also place the appropriate value of the overhead bit to be inserted into the outbound E3 data-stream onto the TxOH input signal. The Transmit Overhead Data Input Interface block will sample and latch the data residing on the TxOH input pin upon the second rising edge of TxInClk after TxOHEnable was sampled "High".

TASK # 4: The System-Side Terminal Equipment should hold the TxOHIns input pin "High" and also hold the value of the TxOH signal stable until the next time the TxOHEnable outout pin is sampled "High". Afterwards, the System-Side Terminal Equipment should toggle the TxOHIns input pin "Low" and wait until another appropriate TxOHEnable pulsing period comes up, for inserting an overhead bit into the outbound E3 data-stream.

CASE STUDY: The System-Side Terminal Equipment intends to insert the appropriate overhead bits into the Transmit Overhead Data Input Interface using Method 2 in order to transmit the FERF/RDI indicator to the remote terminal equipment.

For E3, ITU-T G.832 applications, the FERF/RDI indication is transmitted by setting the FERF/RDI bit-field (Bit 1 within the MA-byte, of each outbound E3 frame) to "1".

If we were to assume that the connection between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface block of the XRT79L71 is as illustrated in **Figure 223**, then **Figure 224** presents an illustration of the signaling that must go on between the System-Side Terminal Equipment and the Transmit Overhead Data Input Interface when using Method 2.

FIGURE 224. ILLUSTRATION OF THE SIGNALING THAT MUST OCCUR BETWEEN THE SYSTEM-SIDE TERMINAL EQUIPMENT AND THE TRANSMIT OVERHEAD DATA INPUT INTERFACE OF THE XRT79L71, IN ORDER TO CONFIGURE THE XRT79L71 TO TRANSMIT THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT (USING METHOD 2)

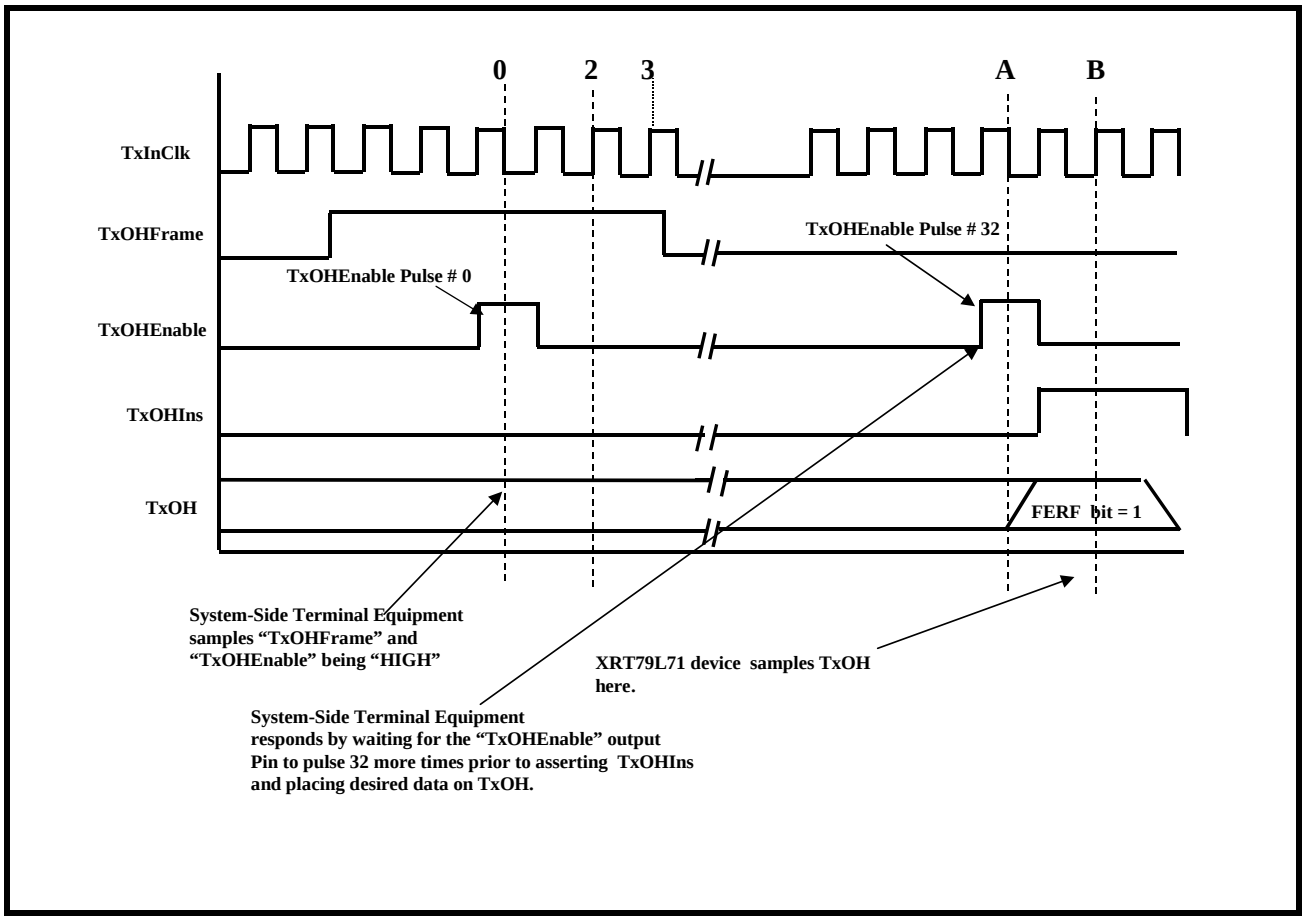


Figure 224, the System-Side Terminal Equipment samples the TxOHFrame and TxOHEnable output pins being "High" during Falling Clock Edge # 0 within the TxInClk signal. At this point, the System-Side Terminal Equipment know that the XRT79L71 is just about to process the very first overhead bit within a given outbound E3 frame. In order to facilitate the transmission of the FERF/RDI indicator, the System-Side Terminal Equipment must set the FERF/RDI bit-field within the outbound E3 frame to "1". According to Table 61, the FERF/RDI bit-field will be ready for processing coincident to the 32nd assertion of the TxOHEnable output signal, following the most recent assertion of TxOHFrame.

The System-Side Terminal Equipment can externally set the FERF/RDI bit-field within each outbound E3 frame to "1" by implementing the following tasks in the sequence as depicted below.

TASK # 1 - Continuously sample both the TxOHFrame and the TxOHEnable output pins with each falling edge of TxInClk. Whenever the System-Side Terminal Equipment samples both of these input pin "High", then it should move on to TASK # 2.

TASK # 2- Continue to sample both the TxOHEnable and TxOHFrame output pins with each falling edge of TxInClk. The System-Side Terminal Equipment should increment an internal counter each time it samples the TxOHEnable output pin "High". The System-Side Terminal Equipment should execute this task until the counter has reach the value "32". Once this counter has reached the value "32" then the System-Side Terminal Equipment should move on to TASK # 3.

TASK # 3 - The System-Side Terminal Equipment MUST execute all of the following two sub-tasks simultaneously.

TASK # 3a - The System-Side Terminal Equipment must assert the TxOHIns input pin by setting it "High".

TASK # 3b - The System-Side Terminal Equipment must set the TxOH input pin to "1".

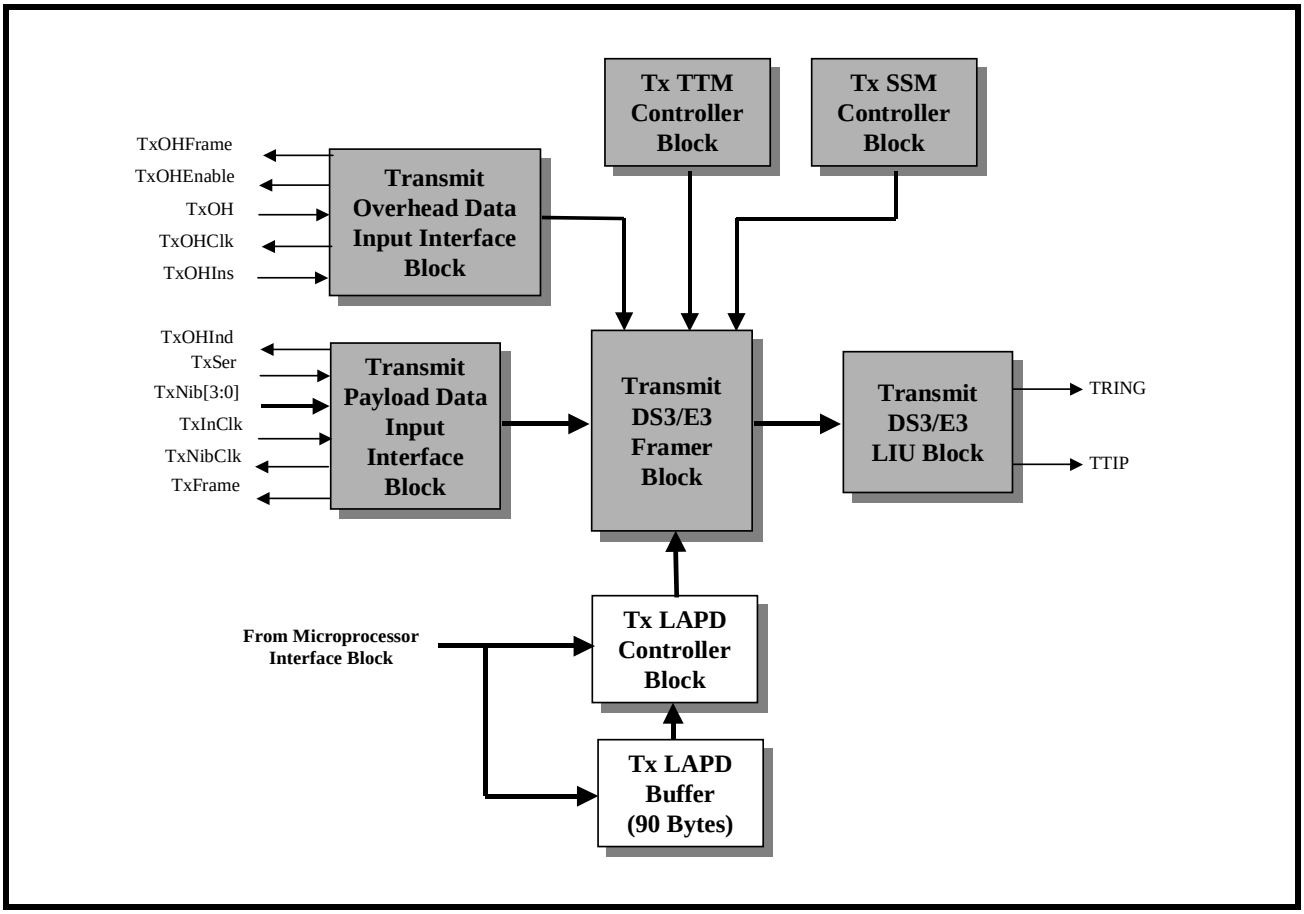
After the System-Side Terminal Equipment has executed these two sub-tasks, it must now move on to TASK # 4.

TASK # 4 - The System-Side Terminal Equipment must continue to sample the TxOHEnable output pin from the XRT79L71. Whenever the System-Side Terminal Equipment samples the TxOHEnable low then it must negate the TxOHIns input pin (e.g., by setting it to "0"). Afterwards, the System-Side Terminal Equipment should then reset the internal TxOHEnable counter to "0", and return to TASK # 1.

6.2.3 TRANSMIT LAPD CONTROLLER BLOCK

The Transmit LAPD Controller block is the third functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. Figure 225 presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Transmit LAPD Controller block highlighted.

FIGURE 225. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3 , ITU-T G.751 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT LAPD CONTROLLER BLOCK HIGHLIGHTED)



The Transmit LAPD Controller block consists of the following sections.

- The Transmit LAPD Message Buffer
- The Transmit LAPD Controller

Each of these sections is described in some detail below.

The Transmit LAPD Message Buffer

The purpose of the Transmit LAPD Message Buffer is to permit the user to write and temporarily store the contents of the very next outbound LAPD Message that is to be transmitted. The Transmit LAPD Message Buffer is actually a 90 byte FIFO that is located at Address location 0x11B0 within the XRT79L71 address space.

The Transmit LAPD Controller

The Transmit LAPD Controller permits the user to transmit path maintenance data link (PMDL) messages to the remote terminal equipment via the outbound E3 Frames. In this case the message bits are inserted into and carried by either the NR or the GC byte within each outbound E3 frame. The on-chip Transmit LAPD Controller permits the user to transmit both standard and non-standard PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the Transmit LAPD Message buffer), to store the contents of the outbound PMDL message to be transmitted. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in **Figure 226**.

FIGURE 226. LAPD MESSAGE FRAME FORMAT

Flag Sequence (8 bits)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		
76 or 82 or Any-size Bytes of Information (Payload)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

For standard Bellcore GR-499-CORE applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the Transmit LAPD Controller is not currently transmitting a LAPD Message, it will instead be transmitting a continuous stream of Flag Sequence bytes via either the NR or GC byte, depending upon user selection, within each outbound E3 frame.

SAPI - Service Access Point Identifier

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for standard Bellcore GR-499-CORE applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3 and E3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For standard Bellcore GR-499-CORE applications the user must use the Control byte the value 0x03. Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The Information Payload is the 76 bytes, 82 bytes or any number of 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 Address 0x11B0.

A Special Note about the Information Payload when transmitting standard Bellcore GR-499-CORE type LAPD Messages

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 0x11B0 within the XRT79L71. The value of this octet depends upon the type of LAPD Message frame/PMDL Message that the user wishes to transmit. **Table 62** presents a list of the various types of LAPD Message frames/PMDL Messages that are supported by the XRT79L71 and the corresponding octet value that the user must write into the first octet position within the Transmit LAPD Message buffer.

TABLE 62: THE LAPD MESSAGE TYPE AND THE CORRESPONDING VALUE OF THE FIRST BYTE, WITHIN THE INFORMATION PAYLOAD FOR STANDARD 76 OR 82 BYTE MESSAGES

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

NOTE: *If a LAPD Message and information payload is transmitted that is of a size other than 76 or 82 bytes, then there are no restrictions on the value that should be written to the first byte within the information payload.*

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, x16 + x12 + x5 + 1. Afterwards, this FCS value is inserted into the two-octet FCS value position, within the LAPD Message frame. The LAPD Receiver at the remote terminal will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see **SEE"RECEIVE LAPD CONTROLLER BLOCK" ON PAGE 549.** on how the Receive LAPD Controller block within the handles and processes incoming LAPD Message frames.

Operation of the Transmit LAPD Controller

As mentioned earlier, the Transmit LAPD Controller permits the user to transmit either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

The procedure for transmitting these types of LAPD Messages is presented below.

6.2.3.1 Transmitting Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) write the contents of an outbound PMDL Message, into the Transmit LAPD Message Buffer, and (2) command the Transmit LAPD Controller to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1", as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Select either the NR or the GC byte as the LAPD Channel.

The user can accomplish this by writing the appropriate value into Bit 4 (Transmit LAPD in NR Byte), within the Transmit E3 Configuration Register, as depicted below.

Transmit E3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Transmit LAPD in NR Byte	Unused	Transmit AIS Enable	Transmit LOS Enable	Transmit MA based on Receiver
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	X	0	0	0	X

Setting this bit-field to "1" will configure the Transmit LAPD Controller block to use the NR byte as the LAPD Channel. In this setting, the Transmit LAPD Controller block will fragment the contents of the outbound LAPD/PMDL Message into bytes and it will insert each of these bytes into the NR byte-field position within each outbound E3 frame. Conversely, setting this bit-field to "0" will configure the Transmit LAPD Controller block to use the GC byte as the LAPD Channel. In this setting, the Transmit LAPD Controller block will fragment the contents of each outbound LAPD/PMDL Message into bytes, and it will insert these bytes into the GC byte-position within each outbound E3 frame.

STEP 3 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit E3 LAPD Configuration Register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	X	X	1

NOTES:

1. *For normal operation, it is imperative that the user also make sure that bit 4 (Reserved) and bit 7 (LAPD Any) within this register are both set to "0".*
2. *Once the user executes the above mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle (Flag) Sequence (e.g., a continuous, repeating string of octets, of the value 0x7E) via either the NR or GC bytes within each outbound E3 frame.*

STEP 4 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a LAPD/PMDL Message at one-second intervals. To configure the Transmit LAPD Controller block to do this, write a "1" into Bit 3 (Auto Retransmit) within the Transmit E3 LAPD Configuration Register. This action is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	1	0	X	1

STEP 5 - Specify the type of LAPD Message to be transmitted.

At this stage it is necessary to specify the type of LAPD Message, which is to be transmitted. The Transmit and Receive LAPD Controller blocks within the XRT79L71 are capable of supporting the following standard types of LAPD Messages.

- Test Signal
- Idle Signal
- CL Path Identification
- ITU Path Identification

The ITU Path Identification type of PMDL Message consists of an 82-byte information payload. All of the remaining standard types of PMDL/LAPD Messages contain 76 byte information payloads. To transmit an 82 byte message (e.g., the ITU Path Identification type of LAPD Message), set Bit 1 (Transmit LAPD Message Length), within the Transmit E3 LAPD Configuration Register to "1". For all of the remaining types of LAPD Messages which are 76 bytes in length, set Bit 1 (Transmit LAPD Message Length) to "0". This operation is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	0	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle LAPD/PMDL messages that contain 76-byte sized Information Payloads.*

If an ITU Path Identification type of PMDL Message is to be transmitted, write the value "1" into the Transmit LAPD Message Length bit-field, as depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	0	1	1

NOTE: *This setting will configure the Transmit LAPD Controller to handle 82-byte sized PMDL Messages.*

STEP 6 - Load the Transmit LAPD Message Buffer

The next step is to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents of the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in **SEE"RECEIVING STANDARD-TYPE (76 OR 82 BYTE SIZE) LAPD MESSAGES" ON PAGE 551.**.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 6a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer

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

Sub-STEP 6a.1 - At Address Location 0x11C0 (the LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

Sub-STEP 6a.2 - At Address Location 0x11C1 (the LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 6a.1, above) this will be the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

STEP 6b - Write the value for SAPI, C/R and EA0 into the second byte-position, within the Transmit LAPD Message Buffer

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

- Bellcore mandates that the user use the values "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.

Sub-STEP 6b.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x01".

This step will cause the internal LAPD Message Buffer pointer to point to the second byte of Indirect Address 0x01 within the Transmit LAPD Message Buffer.

Sub-STEP 6b.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write in the value "0x3C" or "0x3E" for the value of SAPI, C/R and EA0.

This step will cause the value "0x3C" or "0x3E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6b.1, above) this will be the second byte of Indirect Address 0x01 within the Transmit LAPD Message Buffer

STEP 6c - Write the value for TEI and EA1 into the third byte-position, within the Transmit LAPD Message Buffer

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

Bellcore mandates that the user use the value of "0x01" for this byte.

Sub-STEP 6c.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x02".

This step will cause the internal LAPD Message Buffer pointer to point to the third byte of Indirect Address 0x02 within the Transmit LAPD Message Buffer.

Sub-STEP 6c.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x01".

This step will cause the value "0x01" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6c.1, above) this will be the third byte of Indirect Address 0x02 within the Transmit LAPD Message Buffer.

STEP 6d - Write in the CONTROL BYTE value of 0x03 into the fourth byte-position, within the Transmit LAPD Message buffer

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

Sub-STEP 6d.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x03".

This step will cause the internal LAPD Message Buffer pointer to point to the fourth byte of Indirect Address 0x03 within the Transmit LAPD Message Buffer.

Sub-STEP 6d.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the value "0x03"

This step will cause the value "0x03" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case (per Sub-STEP 6d.1, above) this will be the fourth byte of Indirect Address 0x03 within the Transmit LAPD Message Buffer.

STEP 6e - Write in a specific value, which the remote terminal equipment can use to identify the PMDL Message type.

Therefore, the exact value that the user must write in depends upon the type of LAPD Message being transmitted to the Remote Terminal Equipment. Table 63 presents a mapping of the value to be written into this byte, with the corresponding PMDL Message type, to be transmitted.

TABLE 63: A MAPPING OF THE VALUE TO BE WRITTEN INTO INDIRECT ADDRESS LOCATION 0X11B0 AND THE CORRESPONDING PMDL MESSAGE

PMDL MESSAGE TYPE	VALUE TO BE WRITTEN INTO ADDRESS LOCATION 0X11B0 DURING STEP 5E
Test Signal	0x32
Idle Signal	0x34
CL Path Identification	0x38
ITU Path Identification	0x3F

The user can accomplish all of this by executing the following two sub-steps.

Sub-STEP 6e.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write the value "0x04".

This step will cause the internal LAPD Message Buffer pointer to point to the fifth byte of Indirect Address 0x04 within the Transmit LAPD Message Buffer.

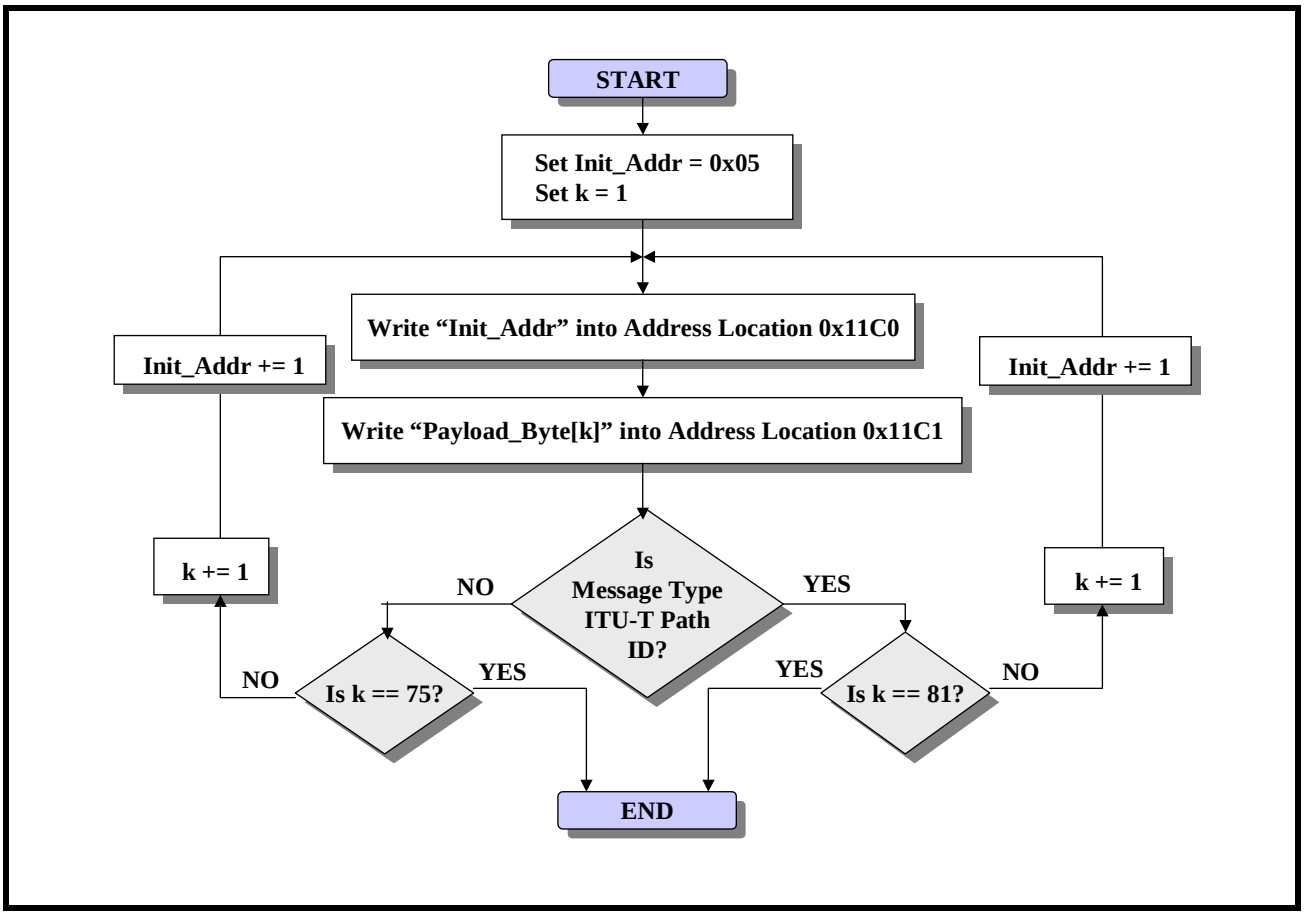
Sub-STEP 6e.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write the appropriate value (per Table 63) into the location that is being identified by the LAPD Message Buffer pointer.

In this case (per Sub-STEP 6e.1, above) this will be the fifth byte of Indirect Address 0x04 within the Transmit LAPD Message Buffer.

STEP 6f - Write in the remaining 75 or 81 byte of the Information Payload within the PMDL Message into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented in the following flow-chart.

FIGURE 227. -FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING THE PAYLOAD PORTION OF THE LAPD/PMDL MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



STEP 7 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller has completed its transmission of a given PMDL Message. The purpose of this interrupt is to alert the system µC/µP that the Transmit LAPD Controller has completed transmitting its most recent LAPD/PMDL message, and that it is now available to transmit a different LAPD Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 7a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 7b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

This step enables the Transmit DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 7c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit E3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 8 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit E3 LAPD Status/ Interrupt Register, as depicted below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0->1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will "zero"-stuff the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin (in a bit-by-bit manner) inserting the resulting PMDL Message into the N bit-fields , within the outbound E3 frames.

NOTES:

1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised (at some later time) to execute another write operation to this register that sets the Initiate Transmission of LAPD/PMDL Message bit back to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field within the Transmit E3 LAPD Status/Interrupt register to "1", as illustrated below.*

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	TransmitLAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	R/W	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will then toggle back to "0".

6.2.3.2 Transmitting Non-Standard Variable Length (e.g., up to 82 bytes) LAPD Messages

The user can (1) write the contents of the outbound PMDL Message into the Transmit LAPD Message buffer and (2) command the Transmit LAPD Controller to begin the transmission of this PMDL Message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame Format	Timing Reference Select [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

STEP 2 - Select either the NR or GC byte as the LAPD Channel

The user can accomplish this by setting Bit 4 The user can accomplish this by writing the appropriate value into Bit 4 (Transmit LAPD in NR Byte), within the "Transmit E3 Configuration" Register, as depicted below.

Transmit E3 Configuration Register (Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			Transmit LAPD in NR Byte	Unused	Transmit AIS Enable	Transmit LOS Enable	Transmit MA based on Receiver
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	X	0	0	0	X

Setting this bit-field to "1" will configure the Transmit LAPD Controller block to use the "NR" byte as the "LAPD Channel". In this setting, the Transmit LAPD Controller block will fragment the contents of the "outbound" LAPD/PMDL Message into "bytes" and it will insert each of these bytes into the "NR" byte-field position within each "outbound" E3 frame. Conversely, setting this bit-field to "0" will configure the Transmit LAPD Controller block to use the GC byte as the "LAPD Channel". In this setting, the Transmit LAPD Controller block will fragment the contents of each outbound LAPD/PMDL Message into bytes, and it will insert these bytes into the "GC" byte-position within each outbound E3 frame.

STEP 3 - Enable the Transmit LAPD Controller

This is accomplished by setting Bit 0 (Transmit LAPD Enable) within the Transmit E3 LAPD Configuration register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
0	0	0	0	X	X	X	1

NOTE: *Once the user executes the above-mentioned step, then the Transmit LAPD Controller will begin to transmit the Idle Flag Sequence (e.g., a repeating string of 0x7E) via either the N bit within each outbound E3 frame.*

STEP 4 - Configure the Transmit LAPD Controller to transmit a non-standard size LAPD Message.

This is accomplished by setting Bit 7 (LAPD Any) within the Transmit E3 LAPD Configuration Register to "1", as illustrated below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	X	X	X	1

STEP 5 - Configure the Transmit LAPD Controller to Auto-Retransmit (Optional)

In some applications, it may be desirable to configure the Transmit LAPD Controller to repeatedly transmit a PMDL Message at one-second intervals. To configure the Transmit LAPD Controller to do this, write a "1" into Bit 3 (Auto-Retransmit) within the Transmit E3 LAPD Configuration register. This operation is depicted below.

Transmit E3 LAPD Configuration Register (Address = 0x1133)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LAPD Any	Unused			Auto Retransmit	Unused	Transmit LAPD Message Length	Transmit LAPD Enable
R/W	RO	RO	R/O	R/W	R/O	R/W	R/W
1	0	0	0	1	0	X	1

STEP 6 - Specify the size of the outbound LAPD Message

This is accomplished by writing the size of the information payload in terms of number of bytes into the Transmit LAPD Byte Count Register as depicted below.

Transmit LAPD Byte Count Register (Address = 0x1183)

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

STEP 7 - Load the Transmit LAPD Message Buffer

The next step is to load into the contents of the outbound LAPD/PMDL Message into the Transmit LAPD Message Buffer. Whenever the user wishes to load in the contents o the outbound PMDL/LAPD Message into the Transmit LAPD Message buffer, then the user MUST employ an Indirect Addressing scheme that will be presented below.

In order to begin the process of loading in the contents of the outbound PMDL Message, the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Transmit LAPD Message Buffer. The approach that one should use to access the Receive LAPD Message buffer will be presented in **SEE"RECEIVING NON-STANDARD VARIABLE LENGTH (E.G., UP TO 82 BYTES) LAPD MESSAGES)" ON PAGE 557.**.

The exact approach that one should use, when loading the contents of their PMDL Message into the Transmit LAPD Message buffer is presented below.

STEP 7a - Write the value "0x7E" into the very first byte-position, within the Transmit LAPD Message Buffer.

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

Sub-STEP 7a.1 - At Address Location 0x11C0 (The LAPD Message Buffer - Indirect Address Register) write in the value "0x00".

This step will cause an internal LAPD Message Buffer pointer to point to the very first byte of Indirect Address = 0x00 within the Transmit LAPD Message Buffer.

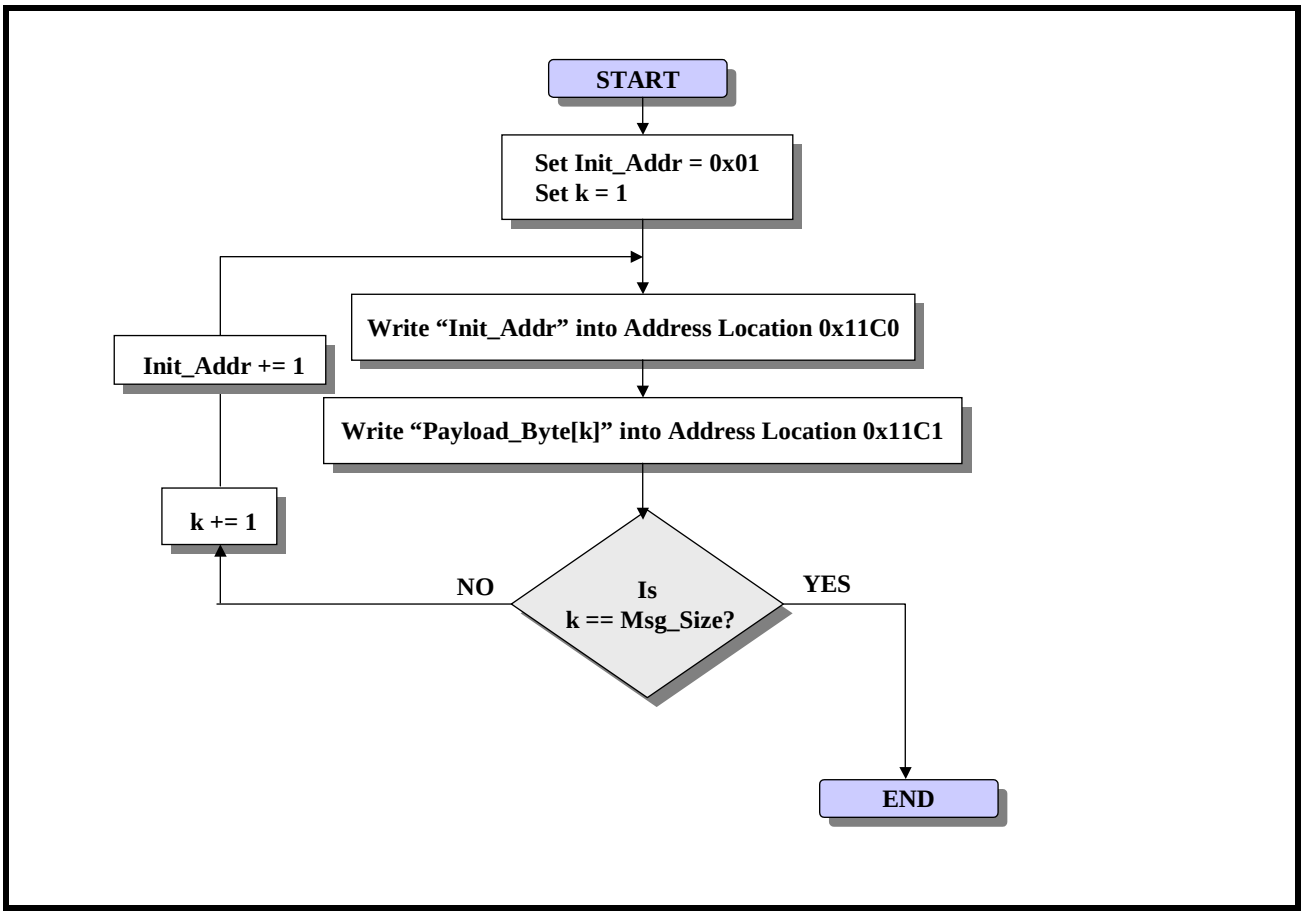
Sub-STEP 7a.2 - At Address Location 0x11C1 (The LAPD Message Buffer - Indirect Data Register) write in the value "0x7E".

This step will cause the value "0x7E" to be written into the location that is being identified by the LAPD Message Buffer pointer. In this case, (per Sub-STEP 7a.1, above) this will be the very first byte of Indirect Address 0x00 within the Transmit LAPD Message Buffer.

STEP 7b - Write in the remaining bytes of this outbound message into the Transmit LAPD Message Buffer.

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 228. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO WRITING IN THE REMAINING BYTES OF THE OUTBOUND MESSAGE INTO THE TRANSMIT LAPD MESSAGE BUFFER



NOTE: *about* **Figure 228***:*

The value of the variable Msg_Size within the Decision Diamond is the value that the user writes into the Transmit LAPD Byte Count Register, during STEP 6.

STEP 8 - Enable the Transmit LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, the XRT79L71 will generate an interrupt to the Microprocessor, anytime the Transmit LAPD Controller has completed its transmission of a given PMDL Message.

The procedure for enabling the Transmit LAPD Interrupt is actually a three-step process.

STEP 8a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

STEP 8b - Enable the Transmit DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 1 (Transmit DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/ E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
X	0	0	0	0	0	1	X

STEP 8c - Enable the Transmit LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Transmit LAPD Interrupt Enable), within the Transmit E3 LAPD Status/Interrupt Register to "1", as illustrated below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0	0	1	0

STEP 9 - Command the Transmit LAPD Controller to begin its transmission.

At this point, it is finally time to command the Transmit LAPD Controller to do its job and transmit the loaded PMDL Message to the remote terminal equipment. The user accomplishes this by inducing a "0" to "1" transition, within Bit 3 (Initiate Transmission of LAPD/PMDL Message) in the Transmit E3 LAPD Status/ Interrupt Register, as depicted below.

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	0->1	0	1	0

Once the user executes this step, then the Transmit LAPD Controller will proceed to do the following.

a. It will parse through the contents of the Transmit LAPD Message buffer and will zero-stuff the payload portion of the outbound PMDL Message.

b. It will compute and append the Frame Check Sequence (FCS) value, to the back end of the outbound PMDL Message.

c. It will begin in a byte-by-byte manner inserting the resulting PMDL Message into either the NR or the GC byte-fields, depending upon user selection during STEP 2, within the outbound E3 frames.

NOTES:

1. *After the user has set the Initiate Transmission of LAPD/PMDL Message bit to "1", the user is advised to (some time later) to execute another write operation to this register that sets the Initiate Transmission of LAPD/PMDL Message bit back to "0".*
2. *Once the Transmit LAPD Controller has started to transmit the PMDL Message to the remote terminal, it will denote this by setting the Transmit LAPD Controller Busy bit-field within the Transmit E3 LAPD Status/Interrupt register to "1", as illustrated below.*

Transmit E3 LAPD Status/Interrupt Register (Address = 0x1134)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused				Initiate Transmission of LAPD/PMDL Message	Transmit LAPD Controller Busy	Transmit LAPD Interrupt Enable	Transmit LAPD Interrupt Status
RO	RO	RO	RO	R/W	RO	R/W	RUR
0	0	0	0	1	1	1	0

This bit-field permits the user to poll the status of the Transmit LAPD Controller. Once the Transmit LAPD Controller has completed the transmission of the LAPD Message frame, this bit-field will toggle back to "0".

The Mechanics of Transmitting a New LAPD Message, if the Transmit LAPD Controller has been configured to retransmit the LAPD Message frame, repeatedly at one-second intervals

If the Transmit LAPD Controller has been configured to retransmit the LAPD Message repeatedly at one-second intervals, then it will do the following at one-second intervals.

- Parse through the contents of the Transmit LAPD Message buffer and Zero-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.

To transmit another (e.g., different) PMDL message to the remote Receive LAPD Controller, the user will have to write this new message into the Transmit LAPD Message buffer, via the Microprocessor Interface section of the channel. However, the user must be careful when writing in this new message. If the user writes this message into the Transmit LAPD Message buffer at the wrong time with respect to these one-second LAPD Message frame transmissions, the user's action could interfere with these transmissions thereby causing the Transmit LAPD Controller to transmit a corrupted message to the remote Receive LAPD Controller. 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 DS3/E3 Framer Block to automatically reset activated interrupts

The user can do this by writing a "1" into Bit 1 (Enable Interrupt Auto-Clear) within the Operating Mode Register, as depicted below.

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	1	0

This action will prevent the Transmit LAPD Controller from generating its own one-second interrupts.

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 = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/ E3 Framer Block Interrupt Enable	One Second Interrupt
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
0	0	0	0	0	0	0	1

3. Write the new message into the Transmit LAPD Message buffer immediately after the occurrence of the One-Second interrupt.

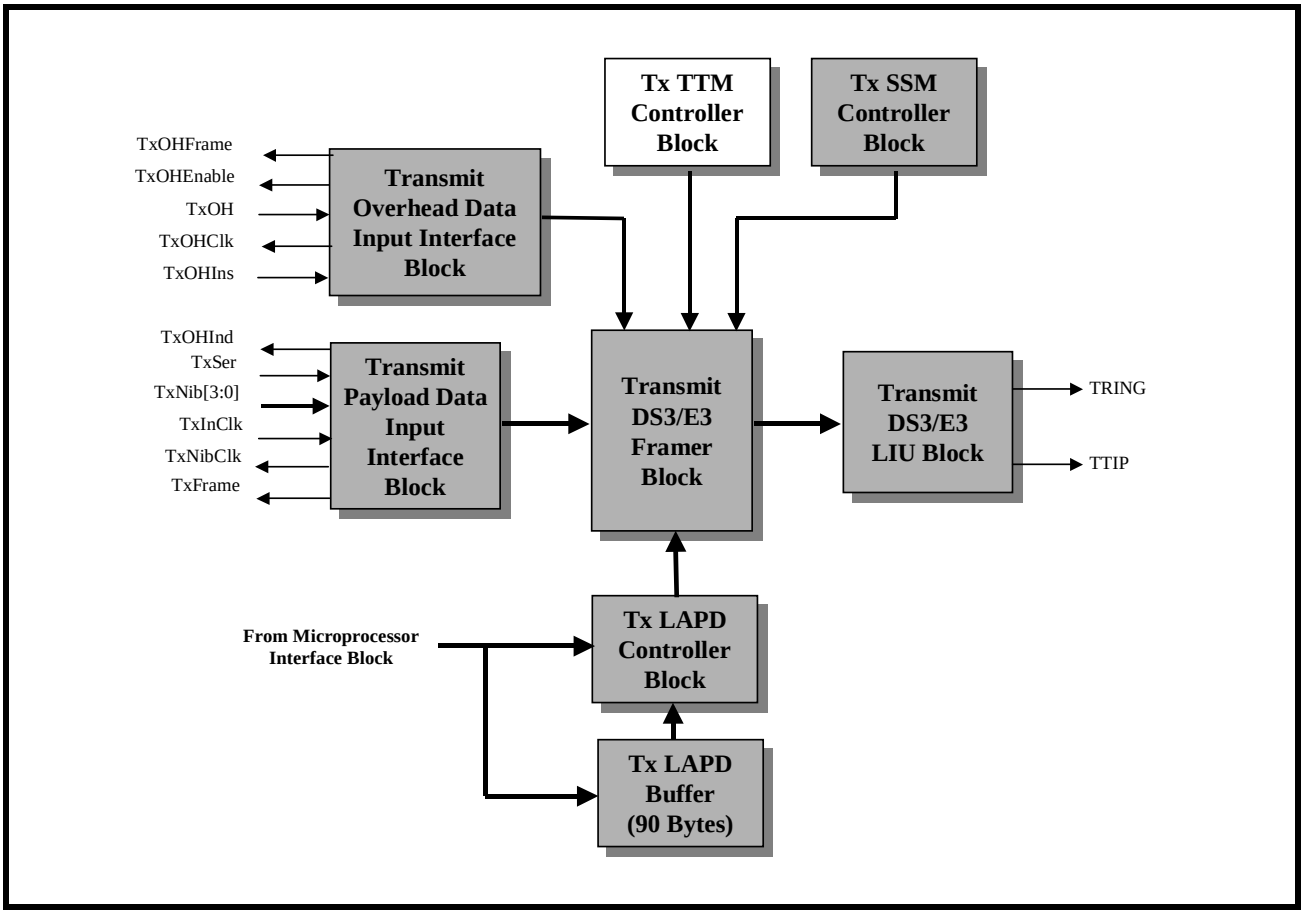
By timing the writes to the Transmit LAPD Message buffer to occur immediately after the occurrence of the One-Second interrupt, the user avoids conflicting with the one-second transmissions of the LAPD Message frame, and will transmit the correct messages to the remote Receive LAPD Controller.

6.2.3.3 Transmit LAPD Controller Block Interrupt

6.2.4 TRANSMIT TRAIL-TRACE MESSAGE CONTROLLER BLOCK

The Transmit Trail-Trace Message Controller Block is the fourth functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3 Clear-Channel Framer Applications. **Figure 229** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Transmit Trail Trace Message Controller block highlighted.

FIGURE 229. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT TRAIL-TRACE MESSAGE CONTROLLER BLOCK HIGHLIGHTED)



6.2.4.1 AN INTRODUCTION TO TRAIL-TRACE MESSAGES

If the XRT79L71 is operating in the E3, ITU-T G.832 Frame Format, then the TR (Trail-Trace) Byte can be used to transmit Trail-Trace Messages also known as a Trail Access Point Identifier. The purpose of the Trail Access Point Identifier Message is to permit a given terminal to repetitively identify itself, thereby permitting the Receiving Terminal to verify its continued connection to the correct terminal, via the E3 Transport Medium.

The Trail-Trace or Trail Access Point Identifier Message consists of a 16-byte string that is repeatedly transmitted from one terminal to the remote terminal equipment. The byte-format of the Trail-Trace Message is presented below in Table 64.

TABLE 64: THE BYTE-FORMAT OF THE TRAIL-TRACE MESSAGE THAT THIS BEING TRANSPORTED VIA AN E3 DATA-STREAM VIA THE TR BYTE

BYTE NUMBER		BIT 1 (MSB)	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8 (LSB)
1 (FrameStart Marker)		1	CRC_Value[6:0]						
2		0	TTM_Byte_1[6:0]						
*		0	TTM_Byte_N[6:0]						
15		0	TTM_Byte_15[6:0]						
16		0	TTM_Byte_16[6:0]						

The very first byte of this 16-byte string is referred to as the Frame Start Marker byte. The Frame Start Marker" byte is typically of the form [1, C6, C5, C4, C3, C2, C1, C0]. This "1" in the MSB (most significant bit) of this first byte is used by a Receiving Terminal to identify this byte as the Frame Start Marker byte. All of the remaining bytes within a given Trail-Trace message will contain a "0" in their MSB. The bits C6 through C0 typically contains the results of a CRC-7 calculation that was performed over the previous 16-byte Trail-Trace Message frame.

NOTE: *The XRT79L71 will NOT compute nor insert this CRC-7 value into these bit-fields. The user will need to externally compute the CRC-7 value over a given outbound Trail-Trace Message and through the XRT79L71 insert this CRC-7 into these bit-fields within the Frame Start Marker byte.*

The remaining 15 bytes are used for the transport of 15 ASCII characters which transport the actual Trail-Trace message. Typically, these Trail-Trace Messages are used to transport messages that are of content/protocol that are compliant to some of the requirements as listed in ITU-T G.831 and E.164.

NOTE: *As mentioned earlier, the MSBs (most significant bits) within the each of the 15 remaining (e.g., non-Frame Start Marker byte) MUST be set to "0".*

6.2.4.2 CONFIGURING THE XRT79L71 TO TRANSMIT TRAIL-TRACE MESSAGES

The XRT79L71 contains a total of 16 Transmit Trail-Trace Message Registers, and 16 Receive Trail-Trace Message Registers. The role of the Receive Trail-Trace Message Registers is described in **SEE"RECEIVE TRAIL-TRACE MESSAGE CONTROLLER BLOCK" ON PAGE 534.**.

As mentioned earlier, the XRT79L71 contains 16 Transmit Trail-Trace Message Registers (e.g., Transmit Trail-Trace Message Register - Byte 1 through Transmit Trail-Trace Message Register - Byte 16). The purpose of these registers is to permit the user to specify the contents of each of the 16-bytes within the outbound Trail-Trace Message.

For Trail-Trace Message purposes, the Transmit Trail-Trace Message Controller block along with the Transmit DS3/E3 Framer block will group 16 consecutive outbound E3 frames, into a Trail-Trace Message Super-Frame. As the Transmit Section of the XRT79L71 is generating the very first E3 frame within this Trail-Trace Message Super-Frame, it will read out the contents of the Transmit Trail-Trace Message Register - Byte 1, and it will write that value into the TR byte-field, within this particular outbound E3 frame. When the Transmit Section of the XRT79L71 is generating the second E3 frame within this Trail-Trace Message Super-Frame, then it will read out the contents of the Transmit Trail-Trace Message Register - Byte 2 and it will write that value into the TR byte-field, within this second outbound E3 frame. As the Transmit Section of the XRT79L71 is creating each subsequent E3 frame, within this particular Trail-Trace Message Super-Frame, it will continue to increment to the very next Transmit Trail-Trace Message Register, read out the contents of this register, and write this value into the TR byte-field within the current outbound E3 frame. After the Transmit Section of the XRT79L71 has created the 16th outbound E3 frame, within this Trail-Trace Message Super-Frame (e.g., it has read out the contents within the Transmit Trail-Trace Message Register - Byte 16 and has written this value into the TR byte-field within this outbound E3 frame), it will begin to create and transmit a new Trail-Trace Message Super-Frame by reading out the contents within the Transmit Trail-Trace Message Register - Byte 1 and repeating the above-mentioned procedure.

Steps to configure the Transmit Trail-Trace Message Controller block to transmit a given Trail-Trace Message

To transmit a given Trail-Trace Message via the E3 data-stream, execute the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing format.

The user can accomplish this by setting Bit 6 (IsDS3) to "0" and by setting Bit 2 (Frame Format) to "1" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	1	1

STEP 2 - Write in the appropriate value into the Transmit Trail-Trace Message Register - Byte 1, as depicted below.

Transmit E3 Trail-Trace Message Byte 1 Register - G.832 (Address = 0x1138)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	C6	C5	C4	C3	C2	C1	C0

The contents within the Transmit Trail-Trace Message Register - Byte 1 will typically be of the form [1, C6, C5, C4, C3, C2, C1, C0]. The "1" within the MSB (most significant bit) position of this byte is used to designate that this octet is the Frame-Start Marker byte (e.g., the very first of the 16 TR bytes, within a Trail-Trace Message Super-Frame) within the outbound Trail-Trace Message. The remaining seven bits (e.g., C6 through C0) is typically the result of a CRC-7 calculation that was computed over the previous Trail-Trace Message that was transmitted to the remote terminal equipment.

NOTES:

1. *It is imperative that as the user writes in a value into the Transmit E3 Trail-Trace Message Byte 1 Register that the user set the MSB (most significant bit) within this register to "1" as depicted above.*
2. *The circuitry within the XRT79L71 will NOT compute this CRC-7 value. To support this CRC-7 feature, the user will need to externally compute the CRC-7 value over a given outbound Trail-Trace Message and will need to write this value into Bits 6 through 0 within the Transmit Trail-Trace Message Register - Byte 1, as depicted above.*

STEP 3 - Write in the value of the remaining 15 ASCII characters within the outbound Trail-Trace Message into the Transmit Trail-Trace Message Register - Bytes 2 through 16, as depicted below.

Transmit E3 Trail-Trace Message Byte 2 Register - G.832 (Address = 0x1139)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	A6	A5	A4	A3	A2	A1	A0

Transmit E3 Trail-Trace Message Byte 3 Register - G.832 (Address = 0x113A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	B6	B5	B4	B3	B2	B1	B0

Transmit E3 Trail-Trace Message Byte 4 Register - G.832 (Address = 0x113B)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	C6	C5	C4	C3	C2	C1	C0

Transmit E3 Trail-Trace Message Byte 5 Register - G.832 (Address = 0x113C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_5							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	D6	D5	D4	D3	D2	D1	D0

Transmit E3 Trail-Trace Message Byte 6 Register - G.832 (Address = 0x113D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_6							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	E6	E5	E4	E3	E2	E1	E0

Transmit E3 Trail-Trace Message Byte 7 Register - G.832 (Address = 0x113E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_7							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	F6	F5	F4	F3	F2	F1	F0

Transmit E3 Trail-Trace Message Byte 8 Register - G.832 (Address = 0x113F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_8							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	G6	G5	G4	G3	G2	G1	G0

Transmit E3 Trail-Trace Message Byte 9 Register - G.832 (Address = 0x1140)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_9							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	H6	H5	H4	H3	H2	H1	H0

Transmit E3 Trail-Trace Message Byte 10 Register - G.832 (Address = 0x1141)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_10							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	I6	I5	I4	I3	I2	I1	I0

Transmit E3 Trail-Trace Message Byte 11 Register - G.832 (Address = 0x1142)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_11							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	J6	J5	J4	J3	J2	J1	J0

Transmit E3 Trail-Trace Message Byte 12 Register - G.832 (Address = 0x1143)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_12							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	K6	K5	K4	K3	K2	K1	K0

Transmit E3 Trail-Trace Message Byte 13 Register - G.832 (Address = 0x1144)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_13							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	L6	L5	L4	L3	L2	L1	L0

Transmit E3 Trail-Trace Message Byte 14 Register - G.832 (Address = 0x1145)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_14							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	M6	M5	M4	M3	M2	M1	M0

Transmit E3 Trail-Trace Message Byte 15 Register - G.832 (Address = 0x1146)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_15							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	N6	N5	N4	N3	N2	N1	N0

Transmit E3 Trail-Trace Message Byte 16 Register - G.832 (Address = 0x1147)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxTTB_Byte_16							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	P6	P5	P4	P3	P2	P1	P0

NOTE: *As the user writes in the values for these remaining fifteen (15) registers, it is imperative that the user make sure that the MSB (Most Significant Bit) position within each of these registers is set to "0". If the user sets any of the MSB's within these remaining 15 registers to "1" then the remote terminal equipment will not be able to properly receive this particular Trail-Trace Message.*

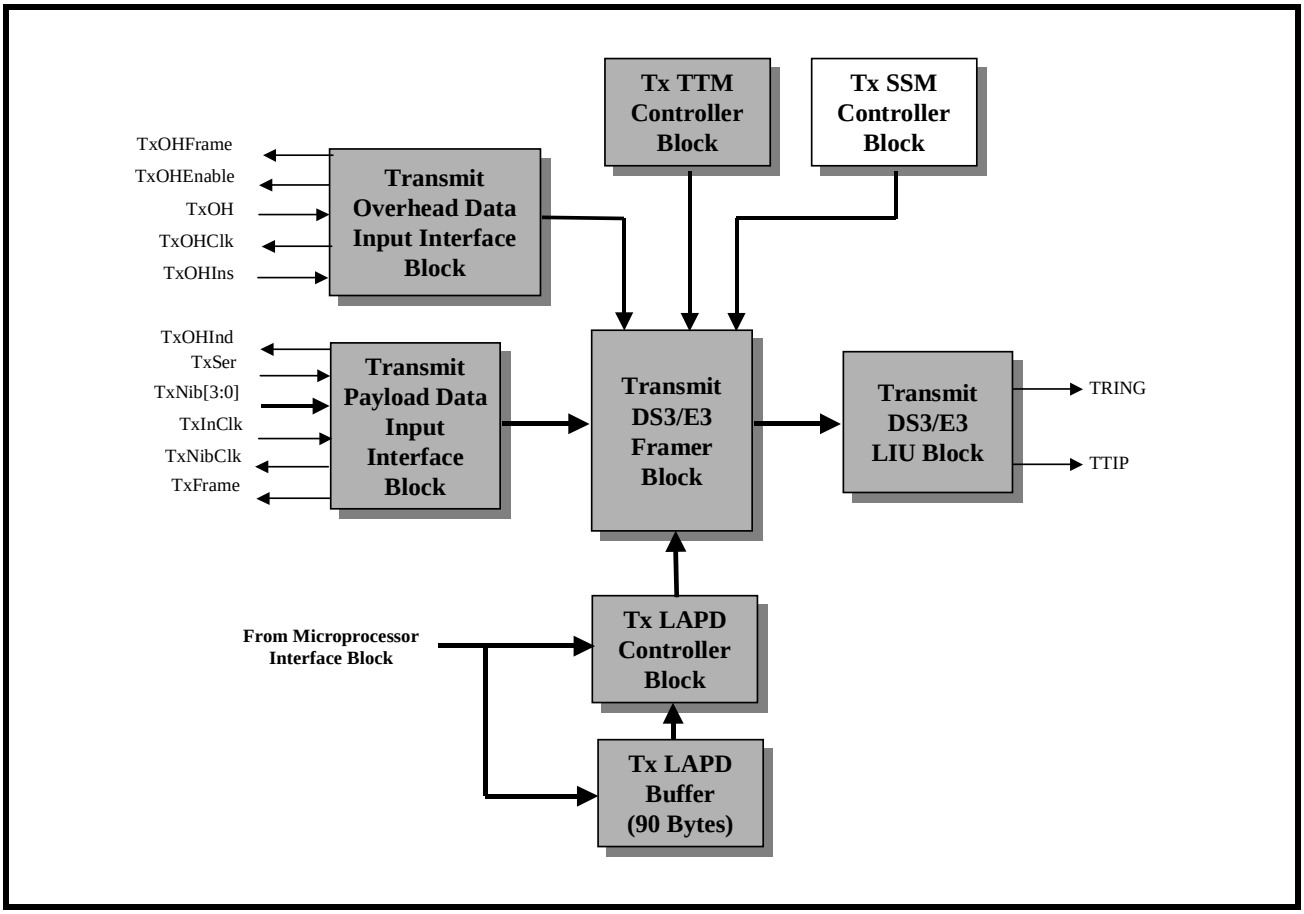
Once the user has executed all of these steps, then the Transmit Trail-Trace Message Controller will immediately begin to do its job, by writing the appropriate values into the TR byte-positions, within each outbound E3 frame.

For information on how the Receive Trail-Trace Message Controller handles incoming Trail-Trace Messages, as well has how to use the Receive Trail-Trace Message Controller, please see **SEE"RECEIVE TRAIL-TRACE MESSAGE CONTROLLER BLOCK" ON PAGE 534.**.

6.2.5 TRANSMIT SSM CONTROLLER BLOCK

The Transmit SSM Controller block is the fifth functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3, Clear-Channel Framer Applications. **Figure 230** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Transmit Trail Trace Message Controller block highlighted.

FIGURE 230. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT SSM CONTROLLER BLOCK HIGHLIGHTED)



6.2.5.1 AN INTRODUCTION TO SSM (SYNCHRONIZATION STATUS MESSAGES)

If the XRT79L71 is operating in the E3, ITU-T G.832 Frame Format, then Bits 6, 7 and 8 (the three least significant bit-fields) within the MA byte can be used to transport the Synchronization Status Message, referred to as SSM from this point on, from one terminal equipment to another. The bit-format of the MA Byte, with these bit-fields shaded is presented below.

FIGURE 231. THE BIT-FORMAT OF THE MA-BYTE WITHIN THE E3, ITU-T G.832 FRAMING FORMAT

BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8
FERF/RDI	FEBE/REI	Payload_Type[2:0]			SSM Multi-Frame Indicator[1:0]		SSM Bit

According to ITU-T G.707, the SSM is a four-bit value that is used to identify the quality-level of the synchronization/timing that the Transmitting E3 Terminal Equipment is currently operating at.

Whenever a given Transmitting E3 Terminal transmits the SSM to the remote terminal, it does so by repeatedly transmitting this four-bit SSM message, one bit at a time or one bit per E3 frame period via Bit 8 (the least significant bit-field), within the MA byte. As this Transmitting E3 Terminal repeatedly transmits this byte to the remote terminal equipment it will use Bits 6 and 7 (within the MA Byte) to indicate which SSM Message bit is being transported via Bit 8, within the current MA byte. Depending upon which of the four bits (within the SSM) that is being transported via the MA byte, within a given E3 frame, the Transmitting E3 Terminal will set Bits 6 and 7 to the appropriate value, in order to identify this SSM bit, as depicted below in **Table 65**.

TABLE 65: THE RELATIONSHIP BETWEEN THE STATES OF BITS 6 AND 7 (WITHIN THE MA BYTE) AND THE EXACT SSM BIT THAT IS BEING TRANSPORTED VIA BIT 8, WITHIN THE CURRENT MA BYTE

MA BYTE, BIT 6	MA BYTE, BIT 7	THE SSM BIT BEING TRANSPORTED IN BIT 8
0	0	SSM Bit 1 (The MSB)
0	1	SSM Bit 2
1	0	SSM Bit 3
1	1	SSM Bit 4 (The LSB)

6.2.5.2 CONFIGURING THE XRT79L71 TO TRANSMIT SYNCHRONIZATION STATUS MESSAGES

The user can enable the Transmit SSM Controller Block within the XRT79L71 and configure it to repeatedly transmit the SSM to the remote terminal equipment, by executing the following three steps.

STEP 1 - Specify the SSM Message that is to be transmitted.

This is accomplished by writing the four-bit SSM value into Bits 3 through 0, within the Transmit E3 SSM Register, as depicted below.

Transmit E3 SSM Register - G.832 (Address = 0x114B)

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

STEP 2 - Configure the Transmit E3 Framer block to transmit the MA Byte via each outbound E3 Frame based upon conditions as detected by the Receive DS3/E3 Framer block.

This is accomplished by setting Bit 0 (Transmit MA Byte based upon Receive Conditions), within the Transmit E3 Configuration Register, to "1" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	1

STEP 3 - Enable the Transmit SSM Controller block.

This is accomplished by setting Bit 7 (TxSSM Enable), within the Transmit E3 SSM Register, to "1" as depicted below.

Transmit E3 SSM Register - G.832 (Address = 0x114B)

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

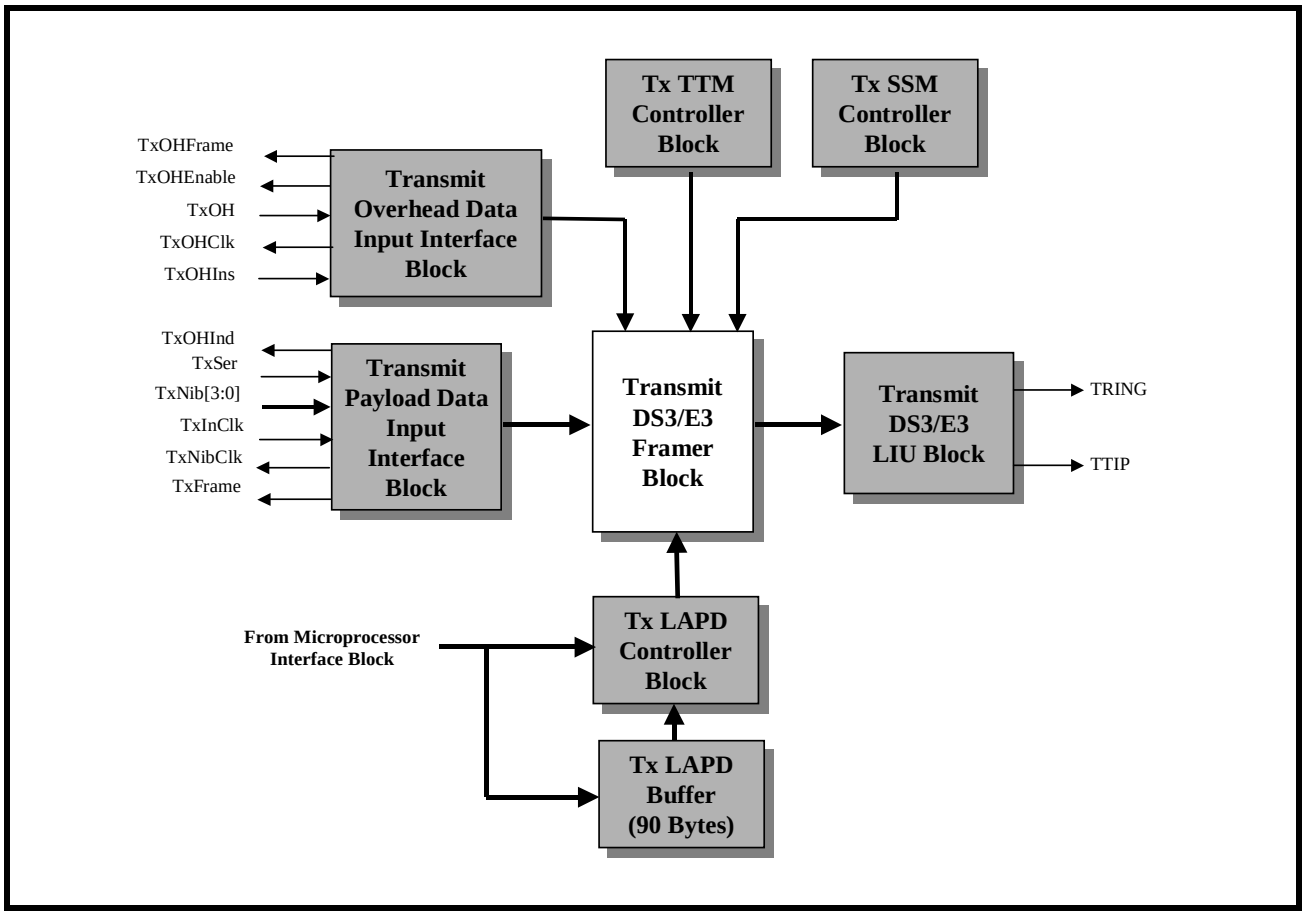
Once the user has executed these two steps, then the Transmit SSM Controller block will be enabled, and will now be repeatedly transmitting the SSM Message, that has been loaded into the Transmit E3 SSM Register, during STEP 1. As the Transmit SSM Controller block transmits this SSM to the remote terminal equipment, it will set Bits 6 and 7 (within the MA byte) to the appropriate value as to identify the SSM bit that is being transported within Bit 8 of the current MA byte.

For information on how the Receive SSM Controller handles incoming Synchronization Status Messages, as well as how to use the Receive SSM Controller, please see **SEE"RECEIVE SSM CONTROLLER BLOCK" ON PAGE 539.**.

6.2.6 TRANSMIT E3 FRAMER BLOCK

The Transmit DS3/E3 Framer block is the sixth functional block within the Transmit Direction of the XRT79L71 that we will discuss for E3, Clear-Channel Framer Applications. **Figure 232** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Transmit DS3/E3 Framer Block highlighted.

FIGURE 232. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT DS3/E3 FRAMER BLOCK HIGHLIGHTED)



The purpose of the Transmit E3 Framer block is to accept data from the Transmit Payload Data Input Interface block, the Transmit LAPD Controller, the Transmit Trail-Trace Message Controller block, and the Transmit SSM Controller block, and to create an E3 data-stream. Afterwards, the Transmit E3 Framer block will route this E3 data-stream to the Transmit E3 LIU Block for transmission to the remote terminal equipment. The Transmit E3 Framer block also supports the following functions.

- Transmitting the LOS Pattern (under Software control)
- Transmitting the AIS Pattern (under Software control)
- Transmitting the FERF (RDI) indicator (automatically and under software control)
- Transmitting the FEBE (REI) indicator (automatically and under software control).

6.2.6.1 TRANSMITTING THE LOS PATTERN

The Transmit DS3/E3 Framer block within the XRT79L71 permits the user to transmit the LOS (Loss of Signal) pattern to the remote terminal equipment. In this case, the Transmit E3 Framer block will generate and transmit an "All Zeros" pattern to the remote terminal equipment. The procedure for transmitting the LOS Pattern is presented below.

Transmitting the LOS Pattern

The user can configure the Transmit DS3/E3 Framer block to transmit an LOS pattern by transmitting an "All Zeros" pattern. This can be accomplished by setting Bit 1 (TxLOS Enable) within the Transmit E3 Configuration Register - G.832" to "1" as depicted below.

STEP 1 - Set Bit 1 (TxLOS Enable) within the Transmit E3 Configuration Register, to "1" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	1	0

Once the user executes this step, then the Transmit DS3/E3 Framer block will override all of the outbound E3 data, payload and overhead bits, with an All Zeros pattern.

6.2.6.2 TRANSMITTING THE E3 AIS PATTERN

The Transmit DS3/E3 Framer block permits the user upon Software Control to transmit the E3 AIS indicator which is an Unframed All Ones Pattern to the remote terminal equipment. The user can invoke this feature by setting Bit 2 (TxAIS Enable) within the Transmit E3 Configuration Register, to "1" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	1	0	0

Once the user executes this step, then the Transmit DS3/E3 Framer block will override the contents of all outbound E3 data, with this E3 AIS (e.g., an Unframed, All Ones) Pattern.

6.2.6.3 TRANSMITTING THE FERF/RDI INDICATOR

The Transmit E3 Framer block provides the user with two options associated with transmitting the FERF (Far-End Receive Failure) condition, within the outbound E3 data-stream.

- Forced transmission of the FERF indicator (e.g., under Software control)
- Automatic transmission of the FERF indicator

6.2.6.3.1 Forced Transmission of the FERF/RDI Indicator

The XRT79L71 permits the user to force the Transmit E3 Framer block to transmit the FERF indicator to the remote terminal equipment. The user can accomplish this by executing the following steps.

STEP 1 - Write the value "0x80" into the Transmit E3 MA Byte Register, as depicted below.

Transmit E3 MA Byte Register - G.832 (Direct Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxMA Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	0	0	0	0	0	0

STEP 2 - Configure the Transmit DS3/E3 Framer block to use the Transmit E3 MA Byte Register as the source of content for the MA byte, within each outbound E3 frame.

This is accomplished by setting Bit 0 (Transmit MA Byte based upon Receive Conditions), within the Transmit E3 Configuration Register to "0" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user executes these two steps, then the Transmit DS3/E3 Framer block will begin to read out the contents of the Transmit E3 MA Byte Register, and will be writing this particular value into the MA Byte, within each outbound E3 frame. Bit 7, within the Transmit E3 MA Byte Register, corresponds to Bit 1 (FERF/RDI) within the MA byte. Therefore, by setting Bit 7 within the Transmit E3 MA Byte Register to "1" the Transmit DS3/E3 Framer block will set the FERF/RDI bit-field within the MA byte-field of each outbound E3 frame to "1". This will result in the transmission of the FERF/RDI indicator to the remote terminal equipment.

6.2.6.3.2 Automatic Transmission of the FERF/RDI Indicator

The XRT79L71 permits the user to configure the Transmit E3 Framer block to automatically transmit the FERF/RDI indicator, in response to any of the following conditions.

- Whenever the corresponding near-end Receive DS3/E3 Framer block is declaring the LOS (Loss of Signal) defect condition
- Whenever the corresponding near-end Receive DS3/E3 Framer block is declaring the OOF (Out-of-Frame) defect condition
- Whenever the corresponding near-end Receive DS3/E3 Framer block is declaring the AIS defect condition.

The user can configure the Transmit DS3/E3 Framer block to automatically transmit the FERF/RDI indicator to the remote terminal equipment, by setting Bit 0 (Transmit MA Byte based upon Receive Conditions) to "1" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	1

In summary, the XRT79L71 can be configured to transmit the FERF indicator to the remote terminal equipment anytime its Receive DS3/E3 Framer block detects either an LOS, OOF/LOF or AIS defect condition in the E3 data that it is receiving from the remote terminal equipment. In the case of E3 applications, the Transmit DS3/E3 Framer block will transmit the FERF indicator to the remote terminal by setting the FERF/RDI bit within the MA byte of each E3 frame to "1".

Conversely, if the Receive DS3/E3 Framer block within the XRT79L71 does not declare either an LOS, OOF or AIS defect condition, then the corresponding Transmit DS3/E3 Framer block will respond by NOT sending the FERF indicator to the remote terminal equipment. In this case, the Transmit DS3/E3 Framer block will NOT transmit the FERF indicator, by setting the FERF/RDI bit within the MA byte of each E3 frame to "0".

Figure 233 through **Figure 236** illustrate this phenomenon.

Figure 233 illustrates the Near-End Receive DS3/E3 Framer block within the XRT79L71 declaring the LOS defect condition within the E3 data-stream that it is receiving from the remote terminal equipment. **Figure 233** illustrates the subsequent action of the corresponding Near-End Transmit DS3/E3 Framer block. In this case, the Transmit DS3/E3 Framer block will set the FERF/RDI bit, within the outbound E3 frame which is destined for the remote terminal to "1".

FIGURE 233. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK DECLARING THE LOS DEFECT CONDITION

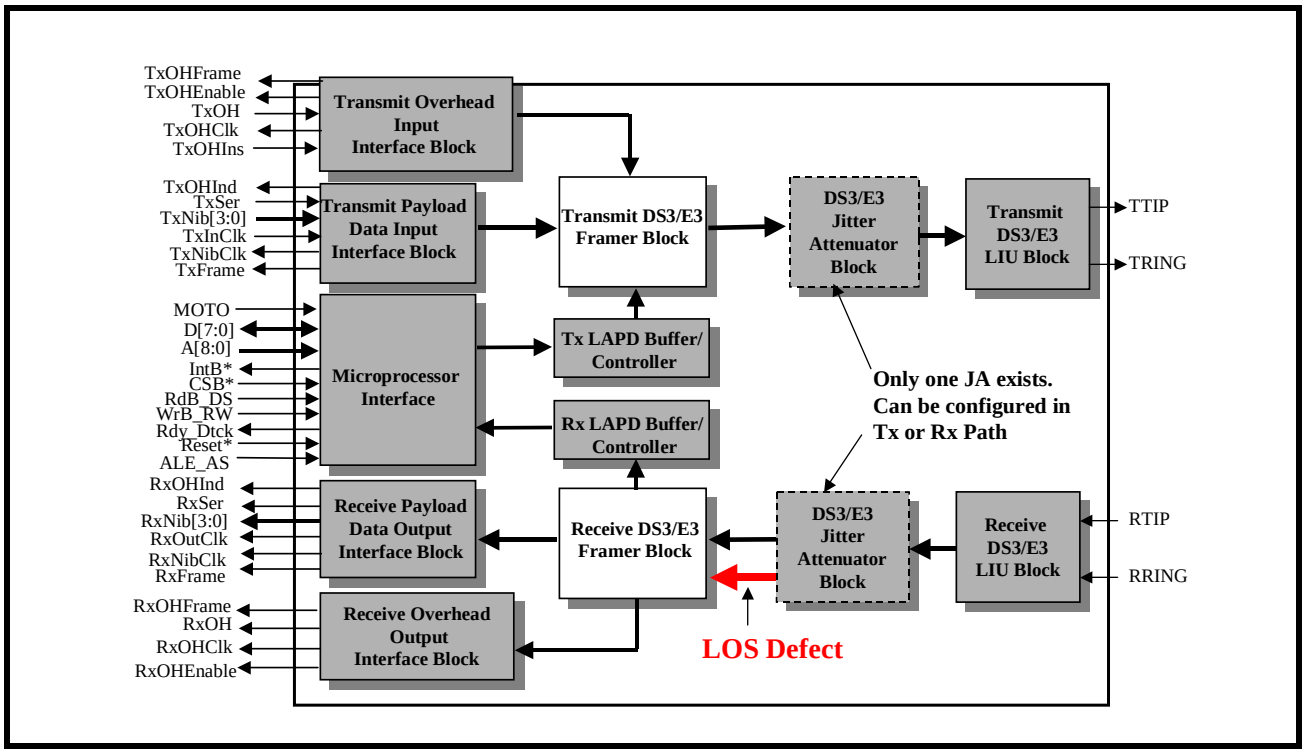


FIGURE 234. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK, TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL WITH THE FERF/RDI BIT SET TO "1"

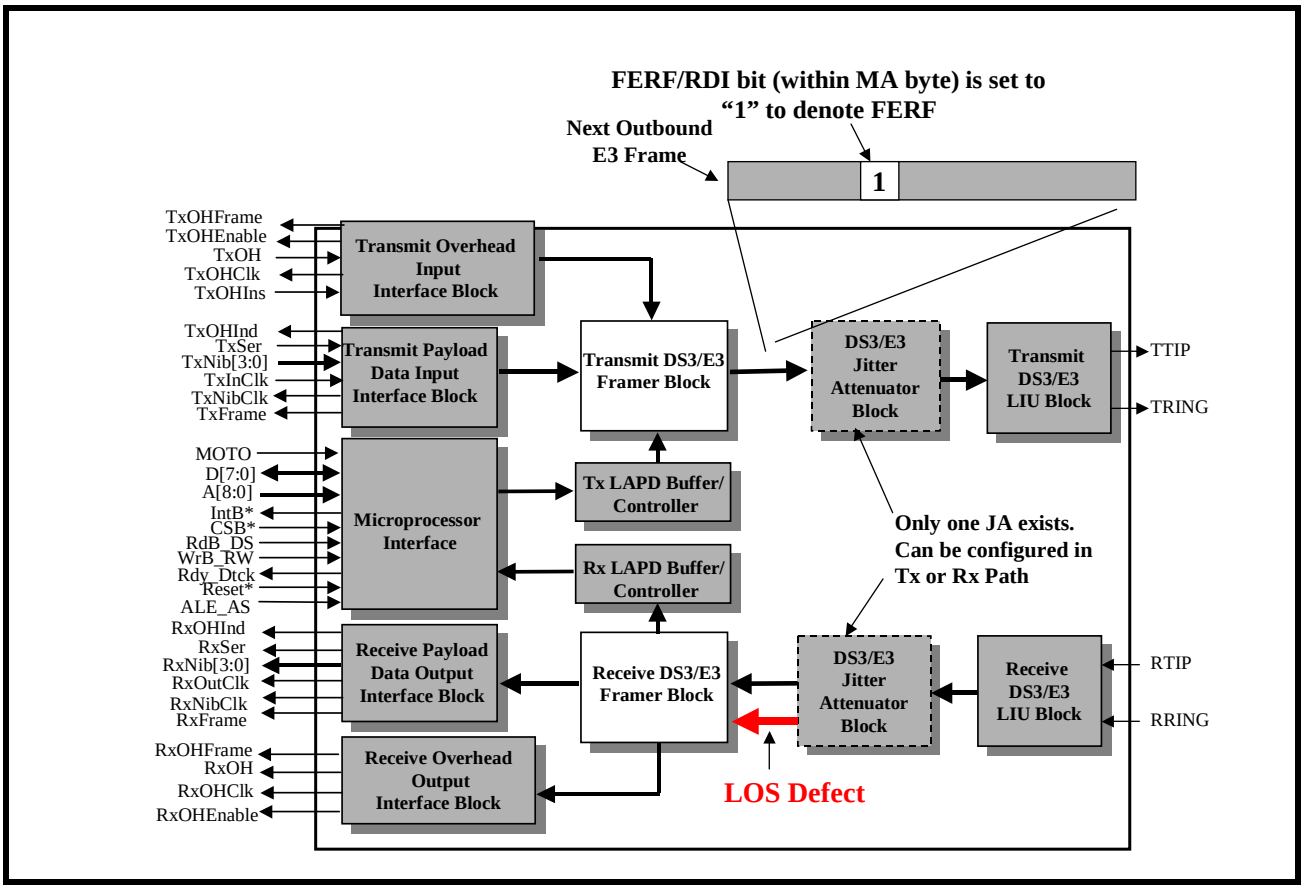


Figure 235 illustrates the Near-End Receive DS3/E3 Framer block receiving a normal E3 data-stream from the remote terminal equipment. Figure 236 illustrates the subsequent action of the Near-End Transmit DS3/E3 Framer block. In this case, the Transmit DS3/E3 Framer block will set the FERF/RDI bit, within the outbound E3 frame which is destined for the remote terminal to "0".

FIGURE 235. ILLUSTRATION OF THE NEAR-END RECEIVE DS3/E3 FRAMER BLOCK RECEIVING A PROPER E3 SIGNAL FROM THE REMOTE TERMINAL EQUIPMENT (E.G., THE LOS DEFECT CONDITION IS CLEARED)

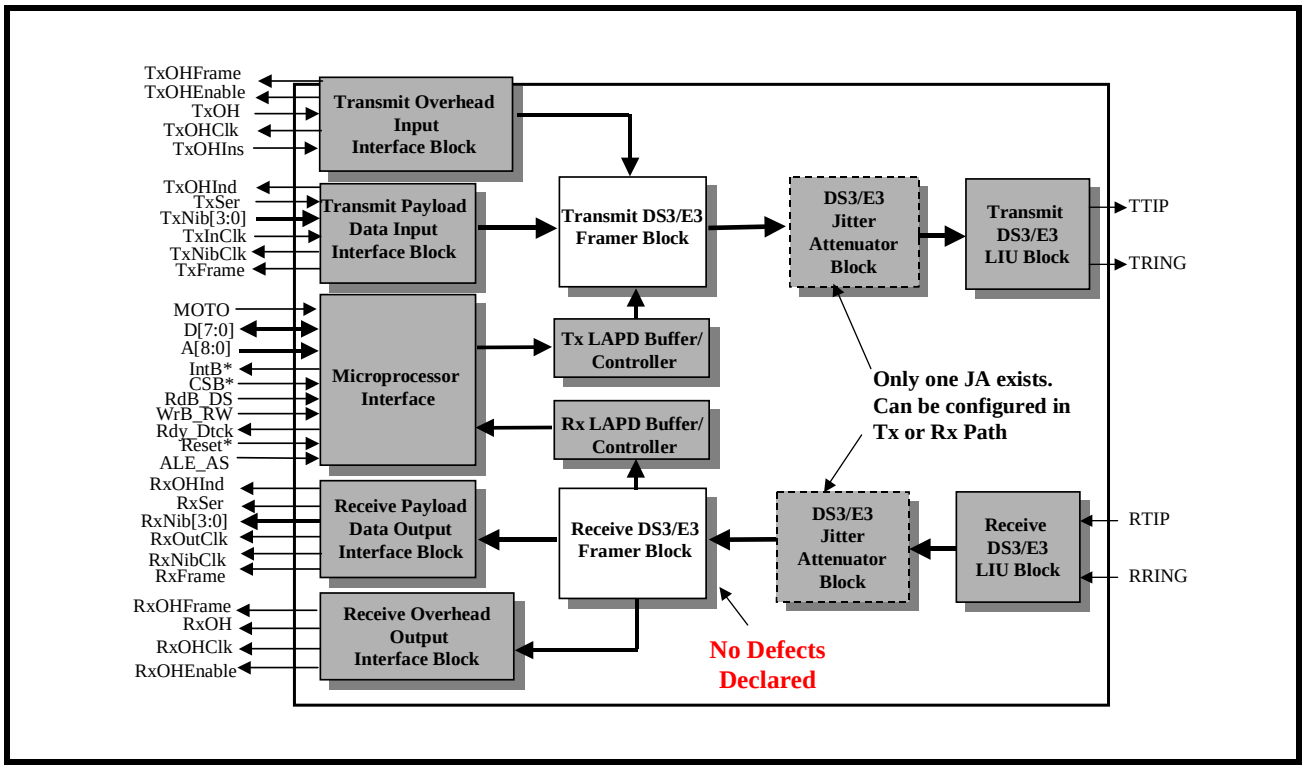
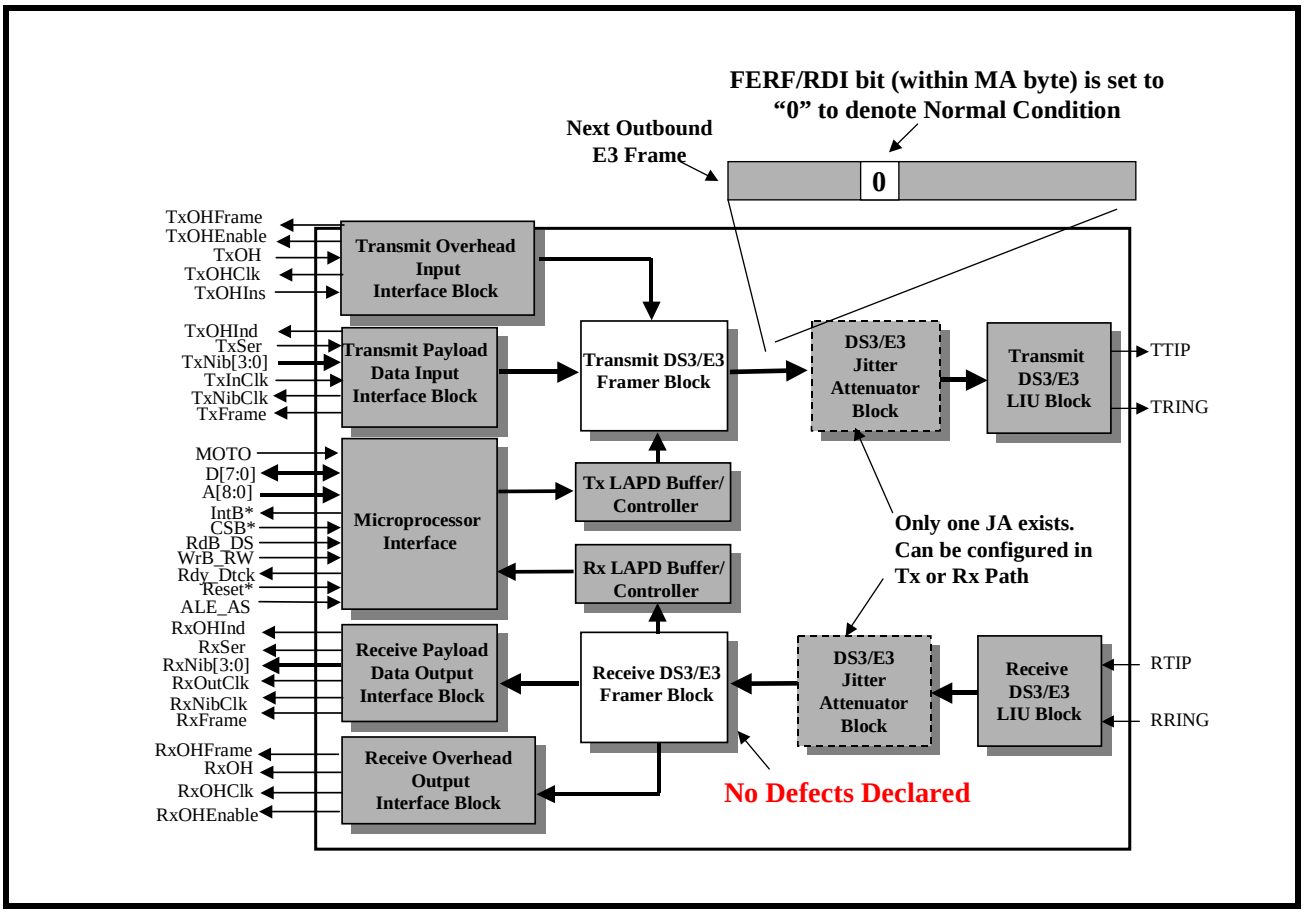


FIGURE 236. ILLUSTRATION OF THE NEAR-END TRANSMIT DS3/E3 FRAMER BLOCK TRANSMITTING AN E3 FRAME TO THE REMOTE TERMINAL EQUIPMENT WITH THE FERF/RDI BIT-FIELD SET TO "0"



6.2.6.4 TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR

If the Transmit DS3/E3 Framer block is configured to support the E3, ITU-T G.832 framing format, then it will be capable of transmitting the FEBE/REI indicator to the remote terminal equipment.

The purpose of the FEBE/REI bit-field, within the MA byte of an E3 frame, is two-fold.

1. It permits a Terminal which is transmitting an E3 data-stream to the remote terminal to determine whether or not this remote terminal is receiving its E3 data, in an error-free manner.
2. It permits a Terminal which is receiving an E3 data-stream from a remote terminal to inform this remote terminal when it is receiving erred E3 frames.

The role of the FEBE/REI bit-field within the MA byte is best presented in the practical example below.

Example:

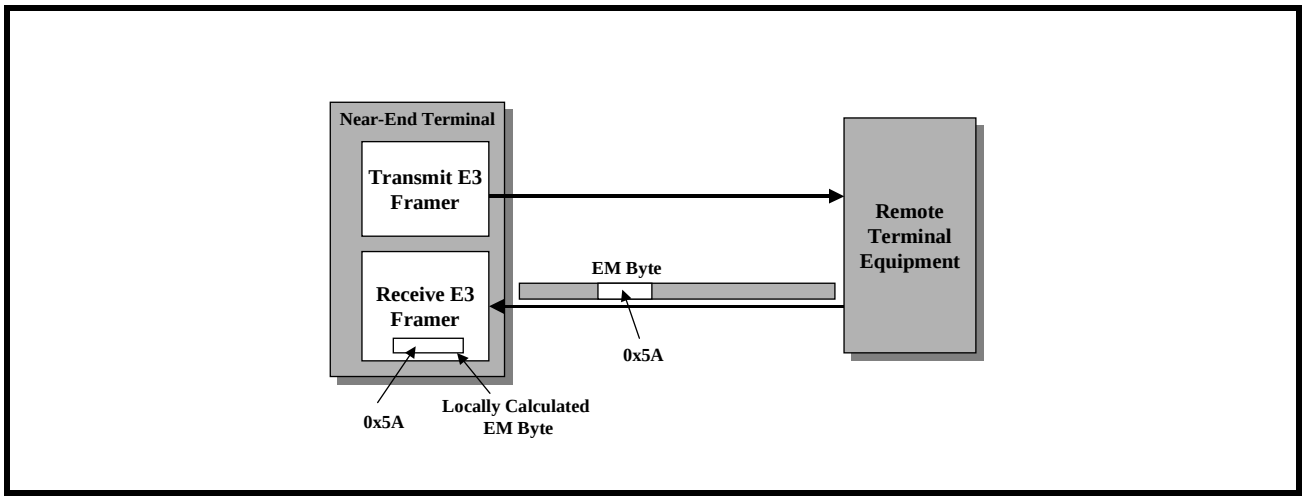
Consider a Near-End terminal that is communicating with a remote terminal. This Near-End terminal consists of the Transmit DS3/E3 Framer block and the Receive DS3/E3 Framer block within the XRT79L71, as depicted below in **Figure 237**.

The Transmit DS3/E3 Framer block will generate and transmit E3 frames to the remote terminal. Likewise, the Receive DS3/E3 Framer block will receive and process E3 frames, originating from the remote terminal. The Near-End Receive DS3/E3 Framer block (e.g., the Receive DS3/E3 Framer block within this particular device) is going to verify the values of the EM (Error Monitor) byte within the incoming E3 frames from the remote terminal equipment. If the Near-End Receive DS3/E3 Framer block detects no EM byte errors in the incoming E3 frame, then it will notify the remote terminal of this fact by forcing the Near-End Transmit DS3/E3 Framer

block, to set the FEBE/REI bit, within the very next outbound E3 frame which is destined for the remote terminal to "0". This phenomenon is illustrated in Figure 237 and Figure 238 below.

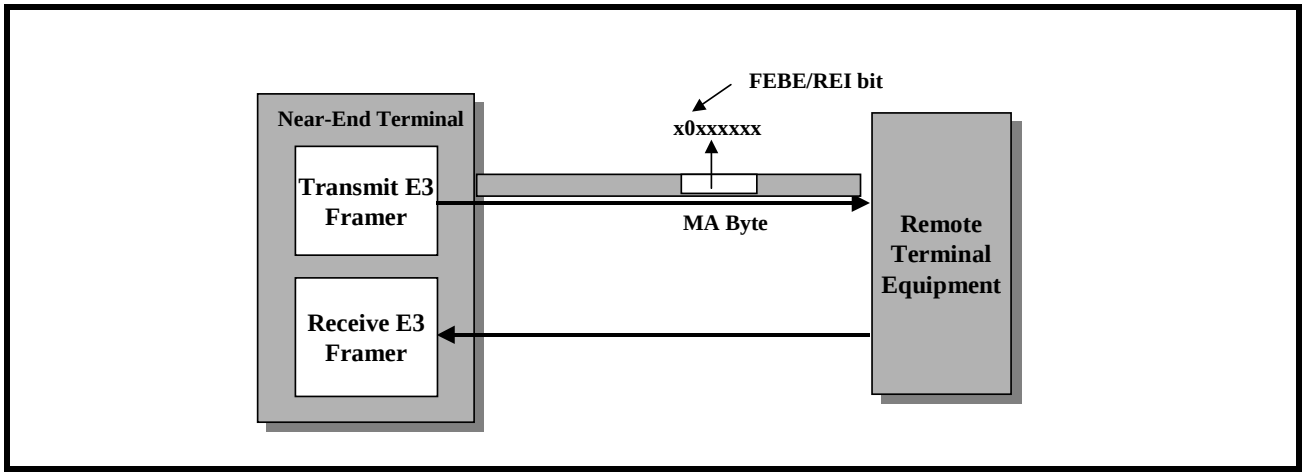
Figure 237 illustrates the Near-End Receive DS3/E3 Framer block receiving an error-free E3 frame. In this figure, the Receive DS3/E3 Framer block has locally computed an EM byte value of "0x5A" that was computed over a given incoming E3 frame. Figure 237 indicates that the EM byte value which resides within the very next incoming E3 frame is of the value "0x5A". As a consequence there is no EM byte errors being detected as this time.

FIGURE 237. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL EXCHANGING E3 DATA WITH A REMOTE TERMINAL, IN AN UN-ERRED MANNER



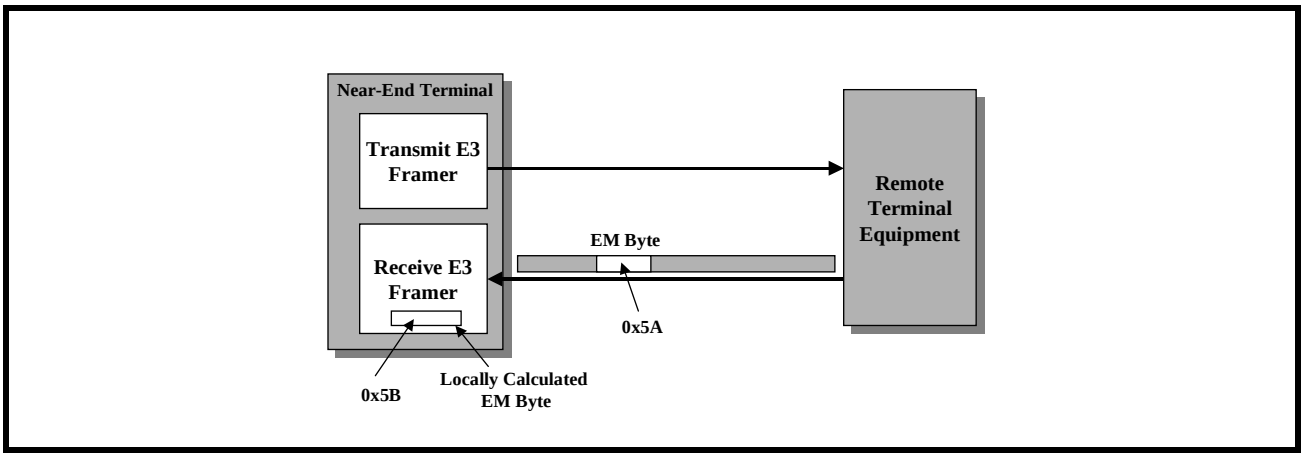
In response to this un-erred condition, the Transmit DS3/E3 Framer block within the Near-End Terminal will respond by setting the FEBE/REI bit-field to "0". Figure 238 presents an illustration of the Transmit DS3/E3 Framer block sending this un-erred indicator to the Remote Terminal Equipment.

FIGURE 238. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL TRANSMITTING THE UN-ERRED INDICATION TO THE REMOTE TERMINAL EQUIPMENT



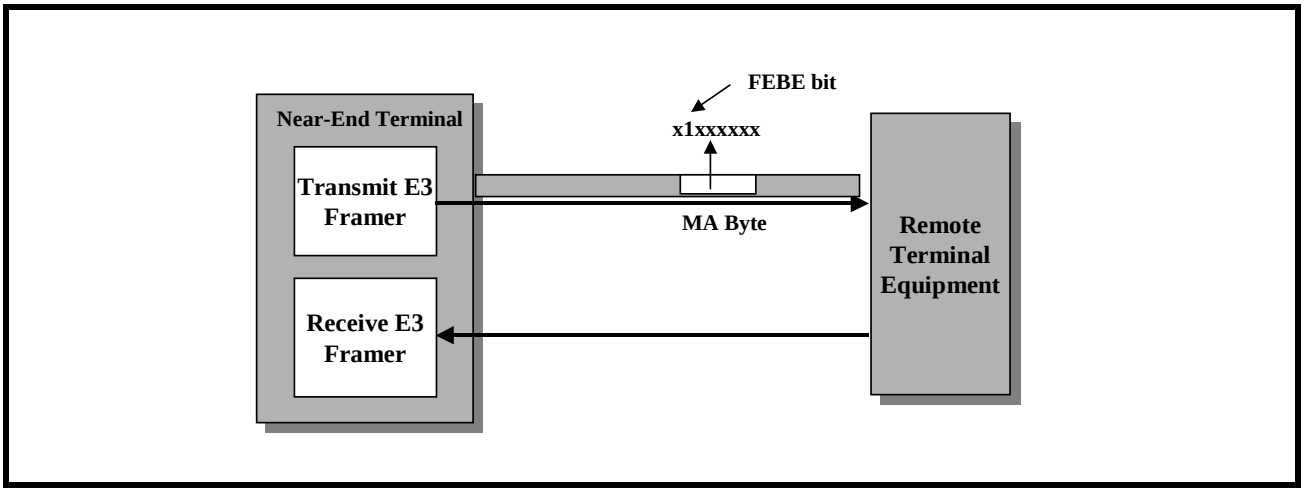
Next, Figure 239 presents an illustration of a given Near-End Terminal that is detecting an EM byte error, within its incoming E3 signal. More specifically, the Receive DS3/E3 Framer block has locally computed an EM byte value of "0x5B" that was computed over a given incoming E3 frame. Figure 239 indicates that the EM byte value which resides within the very next incoming E3 frame is of the value "0x5A". As a consequence, an EM byte error is being detected as this time.

FIGURE 239. A SIMPLE ILLUSTRATION OF A NEAR-END TERMINAL DETECTING EM BYTE ERRORS WITHIN ITS INCOMING E3 SIGNAL



In response to this erred condition, the Transmit DS3/E3 Framer block within the Near-End Terminal will respond by setting the FEBE/REI bit, within the very next outbound E3 frame, to the value "1" in order to denote a FEBE/REI event. More specifically, the Transmit DS3/E3 Framer block within the Near-End Terminal will transmit an E3 frame with the FEBE/REI bit-field set to "1" each time the corresponding Receive DS3/E3 Framer block also within the Near-End Terminal detects an EM byte error within an E3 frame. **Figure 240** presents an illustration of the Transmit DS3/E3 Framer block sending the FEBE/REI indicator to the remote terminal equipment.

FIGURE 240. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FEBE/REI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



For information on how the Receive DS3/E3 Framer block processes the FEBE bit-fields, within each incoming E3 frame, please see **SEE"DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS" ON PAGE 531.**.

The Transmit DS3/E3 Framer block provides the user with two options associated with transmitting the FEBE indicator.

- Forced transmission of the FEBE/REI indicator (e.g., under Software Control)
- Automatic transmission of the FEBE/REI indicator in response to the Receive DS3/E3 Framer block detecting EM byte errors

6.2.6.4.1 Forced Transmission of the FEBE/REI Indicator

The XRT79L71 permits the user to force the Transmit DS3/E3 Framer block to force the transmission of the FEBE/REI indicator upon software control to the remote terminal equipment. The user can accomplish this by executing the following two-step procedure.

STEP 1 - Write the value "0x40" into the Transmit E3 MA Byte Register, as depicted below.

Transmit E3 MA Byte Register - G.832 (Direct Address = 0x1136)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxMA Byte[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	0	0

STEP 2 - Configure the Transmit DS3/E3 Framer block to use the Transmit E3 MA Byte Register as the source of content for the MA byte, within each outbound E3 frame.

This is accomplished by setting Bit 0 (Transmit MA Byte based upon Receive Conditions), within the Transmit E3 Configuration Register to "0" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Once the user executes these two steps, then the Transmit DS3/E3 Framer block will begin to read out the contents of the Transmit E3 MA Byte Register, and will be writing this particular value into the MA Byte, within each outbound E3 frame. Bit 6, within the Transmit MA Byte Register, corresponds to Bit 2 (FEBE/REI) within the MA byte. Therefore, by setting Bit 6 (within the Transmit E3 MA Byte Register to "1" the Transmit DS3/E3 Framer block will set the FEBE/REI bit-field within the MA byte-field of each outbound E3 frame to "1". This will result in the transmission of the FEBE/REI indicator to the remote terminal equipment.

NOTE: *In this configuration, the value of the FEBE/REI bit-field, that is being generated by the Transmit DS3/E3 Framer block does not reflect the health of the E3 signal that is being received by the corresponding Near-End Receive DS3/E3 Framer block.*

6.2.6.4.2 Automatic Transmission of the FEBE/REI Indicator

The XRT79L71 permits the user to configure the Transmit E3 Framer block to automatically transmit the FEBE/REI indicator, in response to whenever the Receive DS3/E3 Framer block detects EM byte errors.

The user can configure the Transmit DS3/E3 Framer block to automatically transmit the FEBE/REI indicator to the remote terminal equipment, by setting Bit 0 (Transmit MA Byte based upon Receive Conditions) to "1" as depicted below.

Transmit E3 Configuration Register - G.832 (Direct Address = 0x1130)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused			TxDL in NR	Reserved	TxAIS Enable	TxLOS Enable	Transmit MA Byte based upon Receive Conditions
R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
0	0	0	0	0	0	0	1

This setting will configure the Transmit DS3/E3 Framer block to respond to occurrence of EM byte errors as detected by the corresponding, Near-End Receive DS3/E3 Framer block as depicted above in **Figure 239** and **Figure 240**.

6.2.6.5 Setting the Payload-Type Bit-fields within the outbound E3 Data-stream

6.2.6.6 User Control over the GC Byte within the outbound E3 data-stream

6.2.6.7 User Control over the NR Byte within the outbound E3 data-stream

6.2.6.8 Setting the Transmit E3 Framer Block Timing Reference

As a user designs their system, they ultimately have to decide whether their system is going to be configured to operate in either the Local-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from a "local" [or in-system] clock source) or in the Loop-Timing Mode (e.g., where the timing source for the Transmit [or outbound] Direction traffic is derived from the "remote terminal equipment's" clock source. The XRT79L71 can be configured to support either of these applications.

In all, the XRT79L71 supports the following three (3) different Timing Reference Modes.

- Local-Timing/Asynchronous
- Local-Timing/TxFrameRef
- Loop-Timing

6.2.6.8.1 Local-Timing/Asynchronous Mode

If the XRT79L71 is configured to operate in the Local-Timing/Asynchronous Mode, then the following are true.

- The Transmit E3 Framer block will use the clock signal that is applied to the TxInClk input pin as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.
- E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT79L71.

Configuring the XRT79L71 to operate in the Local-Timing/Asynchronous Mode

The XRT 79L71 can be configured to operate in the "Local-Timing/Asynchronous" Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[1, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT79L71 is configured to operate in the Local-Timing/Asynchronous Mode, do the following.

- Apply a 34.368MHz clock signal to the TxInClk input pin (Ball C10).
- Tie the TxFrameRef input pin (Ball A11) to GND.

6.2.6.8.2 Local-Timing/TxFrameRef Mode

If the XRT79L71 is configured to operate in the Local-Timing/TxFrameRef Mode, then the following are true.

- The Transmit E3 Framer block will use the clock signal that is applied to the "TxInClk" input pin as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.
- The Transmit E3 Framer block will initiate the generation (and transmission) of a new E3 frame, anytime it detects a rising edge at the TxFrameRef input pin. In this case E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) will be synchronized to the signal that is applied to the TxFrameRef input pin.

Configuring the XRT 79L71 to operate in the Local-Timing/TxFrameRef Mode

The XRT 79L71 can be configured to operate in the Local-Timing/TxFrameRef Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 1]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

Requirements Associated with the "Local-Timing/Asynchronous" Mode

If the XRT 79L71 is configured to operate in the "Local-Timing/TxFrameRef" Mode, do the following.

- Apply a 34.368MHz clock signal to the TxInClk input pin (Ball C10).
- Apply an 8kHz clock signal to the TxFrameRef input pin (Ball A11).
- The 8kHz clock signal being applied to the TxFrameRef input pin must be synchronous with the 34.368MHz clock signal (that is being applied to the TxInClk input pin).

6.2.6.8.3 Loop-Timing Mode

If the XRT 79L71 is configured to operate in the Loop-Timing Mode, then the following are true.

- The Transmit E3 Framer block will use the "Recovered" clock signal (from the Receive DS3/E3 LIU Block) as its timing source, for generating and transmitting the outbound E3 traffic to the remote terminal equipment.
- E3 Frame Generation (e.g., the instant whenever the Transmit E3 Framer block begins to generate and transmit a new E3 frame) is asynchronous with respect to any externally supplied clock signal to the XRT 79L71.

Configuring the XRT 79L71 to operate in the Loop-Timing Mode

The XRT 79L71 can be configured to operate in the Local-Timing/TxFrameRef Mode by setting Bits 1 and 0 (TimRefSel[1:0]) within the Framer Operating Mode Register to "[0, 0]" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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

- **Requirements Associated with the "Loop-Timing" Mode**

If the XRT 79L71 is configured to operate in the "Loop-Timing" Mode, do the following.

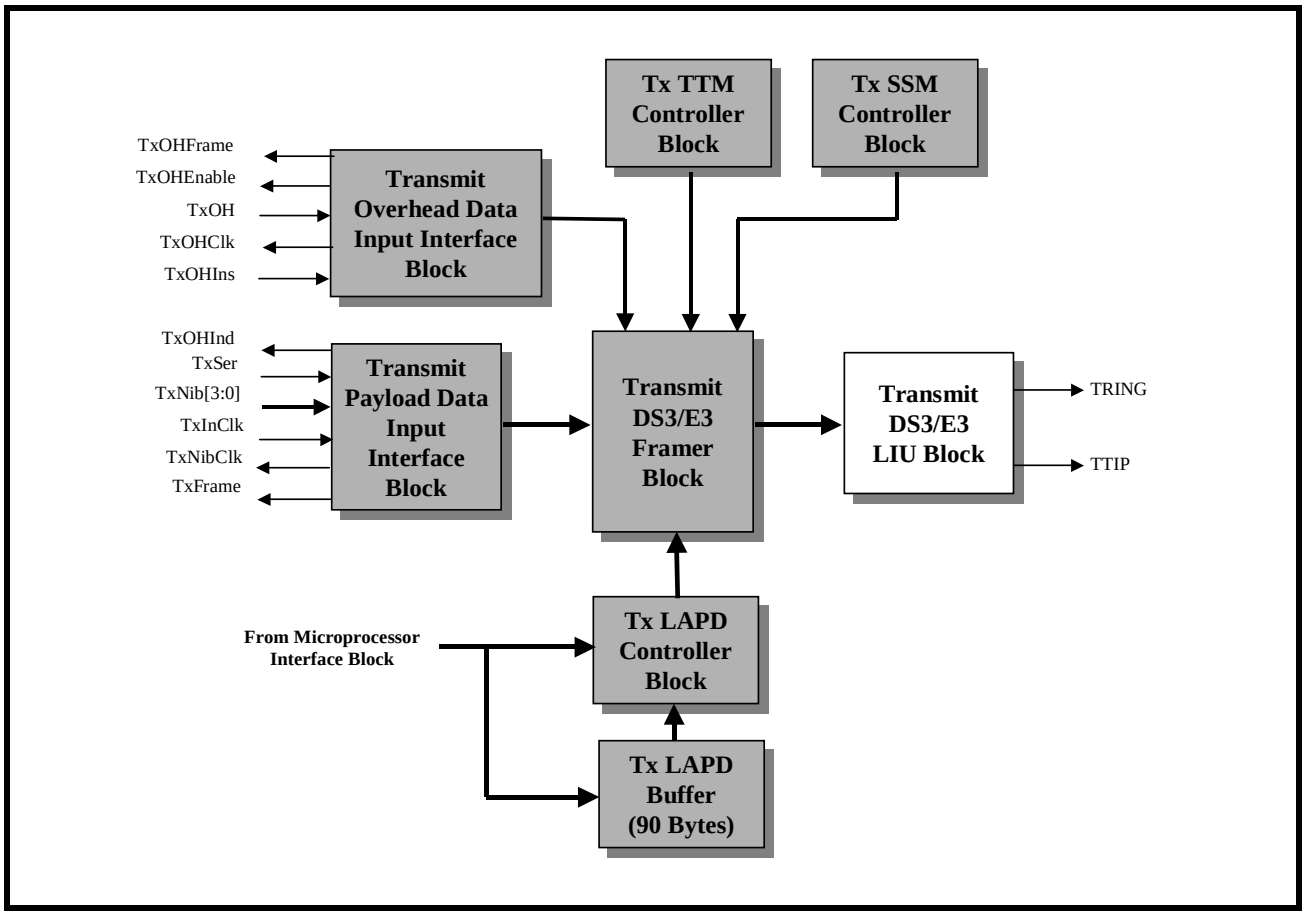
- Insure that either one of the following conditions are true.
  - **a.** That the Receive DS3/E3 LIU Block is receive a proper E3 line signal from the remote terminal equipment, or
  - **b.** That the SFM Synthesizer block (within the Receive DS3/E3 LIU Block) is configured to generate a 34.368MHz clock (to the remainder of the Receive DS3/E3 LIU Block circuitry) from either an externally supplied 12.288MHz or a 34.368MHz clock signal. (Please see **SEE"THE SFM (SINGLE-FREQUENCY MODE) SYNTHESIZER BLOCK" ON PAGE 361.** for details on how to accomplish this)
- Tie the TxFrameRef input pin (Ball A11) to GND.

NOTE: *In order to permit the Microprocessor Interface to function (for Revision A silicon) the user is still required to supply a sufficiently high frequency clock signal to the "TxInClk" input pin, even if the XRT79L71 is configured to operate in the "Loop-Timing" Mode.*

6.2.7 TRANSMIT DS3/E3 LIU INTERFACE BLOCK - E3 APPLICATIONS

The Transmit DS3/E3 LIU Block is the seventh and last functional block within the Transmit Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 241** presents an illustration of the Transmit Direction circuitry whenever the XRT79L71 has been configured to operate in the E3 Clear-Channel Framer Mode, with the Transmit DS3/E3 LIU block highlighted.

FIGURE 241. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE TRANSMIT DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE TRANSMIT DS3/E3 LIU BLOCK HIGHLIGHTED)



Please see SEE"TRANSMIT DS3/E3 LIU BLOCK - E3 APPLICATIONS" ON PAGE 331. for a detailed description of the Transmit DS3/E3 LIU Block - for E3 applications.

6.3 THE RECEIVE DIRECTION - E3, ITU-T G.832 CLEAR-CHANNEL FRAMER APPLICATIONS

The next several sections will present an in-depth functional description of all of the blocks that are operating in the Receive Direction, within the XRT79L71, when configured to operate in the Clear-Channel E3 Framer Mode. **Figure 242** presents a functional block diagram of the Receive Direction circuitry within the XRT79L71.

FIGURE 242. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3 CLEAR-CHANNEL FRAMER MODE

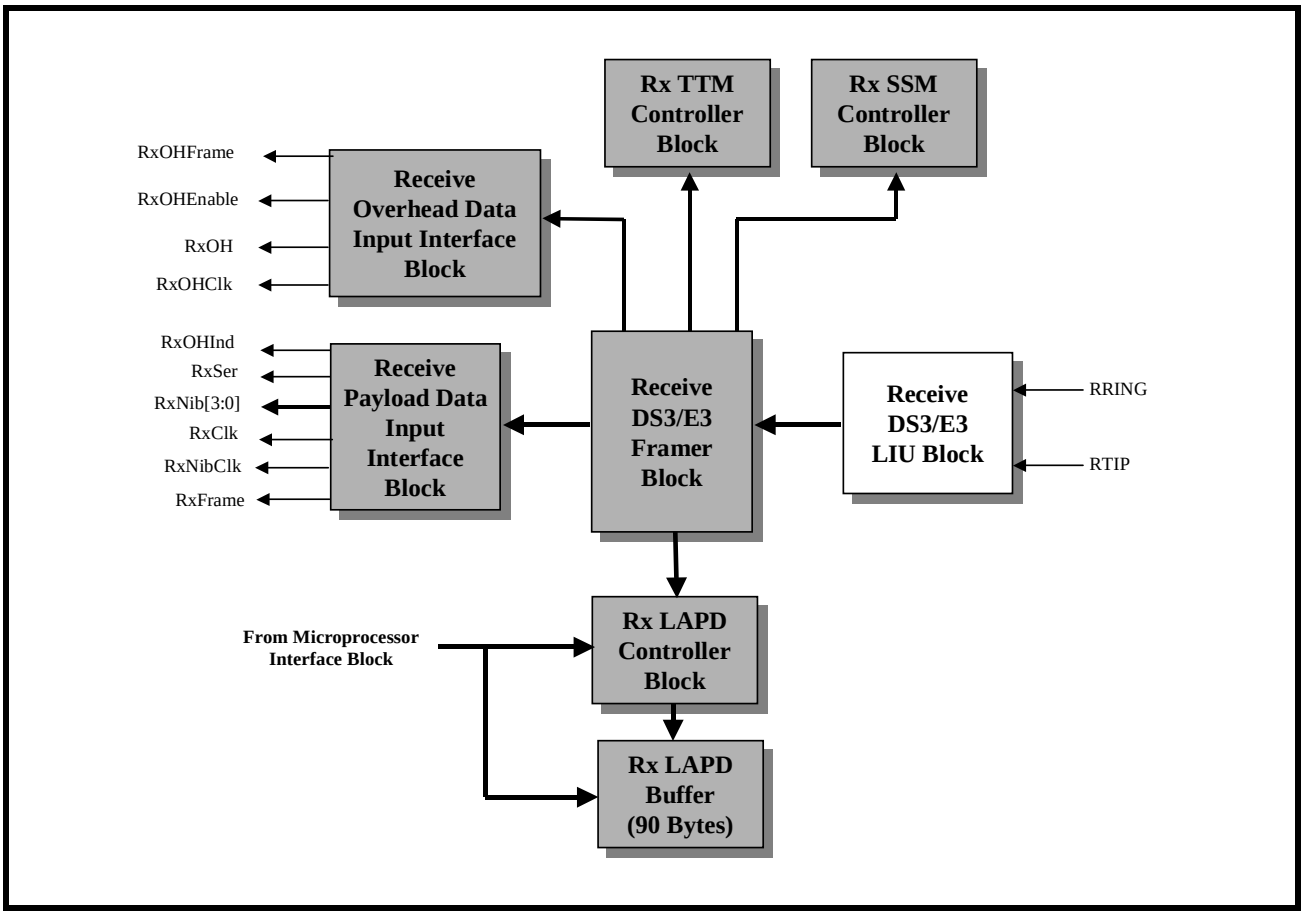


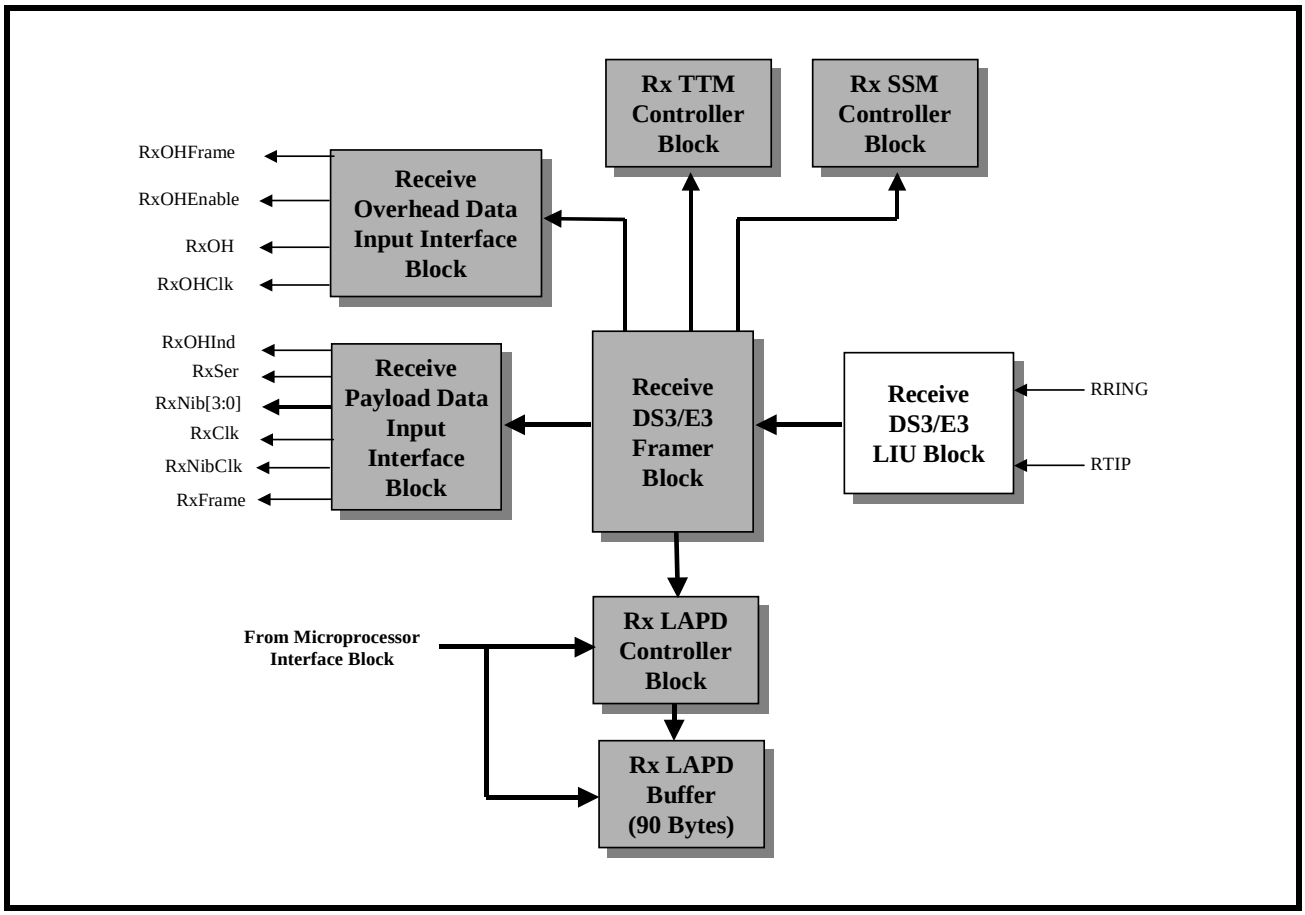
Figure 242 indicates that the Receive Direction circuitry consists of the following functional blocks.

- The Receive E3 LIU Block
- The Receive E3 Framer Block
- The Receive Trail-Trace Message Controller Block
- The Receive SSM Controller Block
- The Receive LAPD Controller Block
- The Receive Payload Data Output Interface Block
- The Receive Overhead Data Output Interface Block

6.3.1 THE RECEIVE E3 LIU BLOCK

The Receive E3 LIU Block is the first functiona block (within the Receive Direction) of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 241 presents an illustration of the "Receive Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive E3 LIU Block highlighted.

FIGURE 243. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE DS3/E3 LIU" BLOCK HIGHLIGHTED)

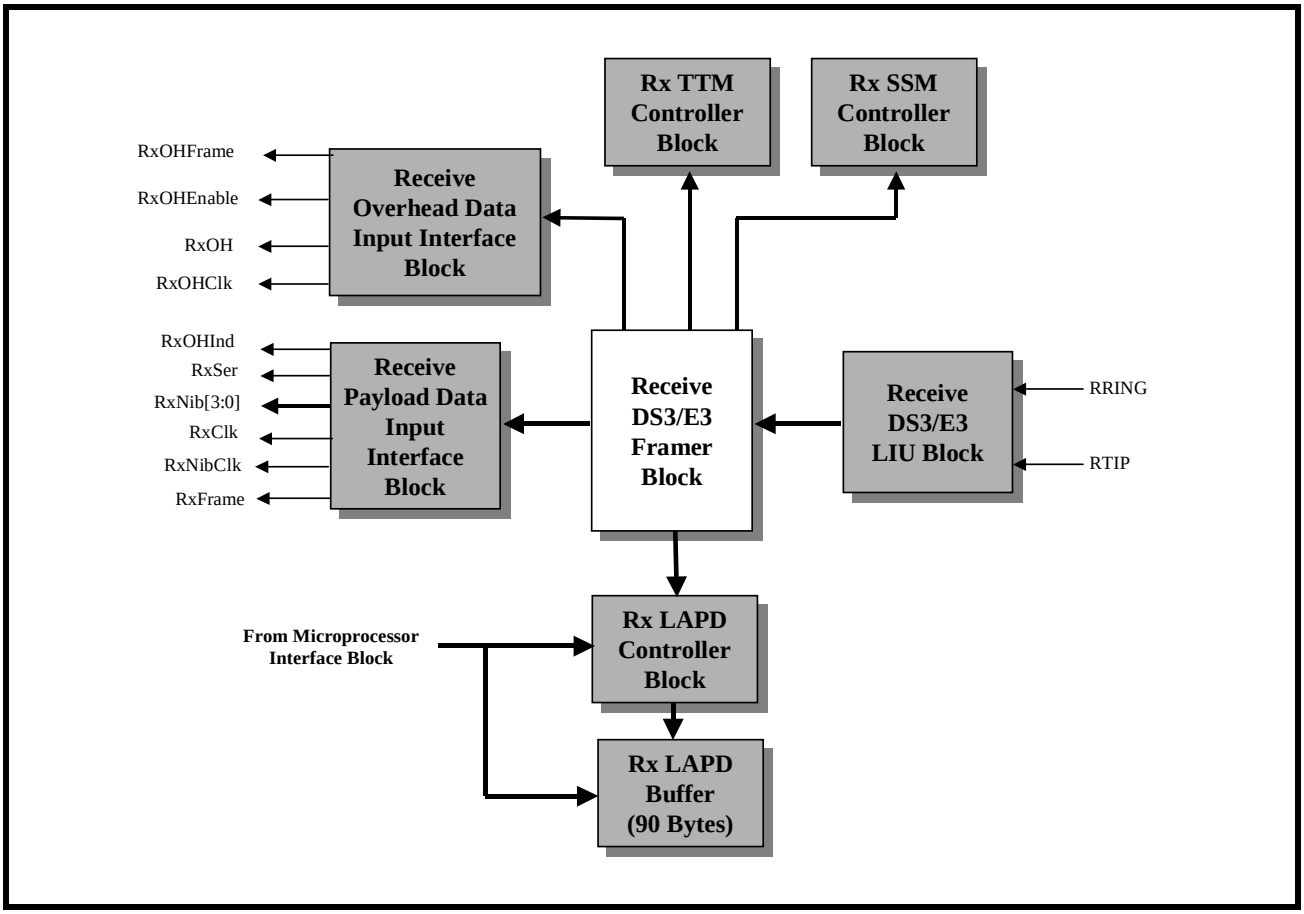


Please see SEE"THE RECEIVE E3 LIU BLOCK" ON PAGE 354. for a detailed description of the Receive DS3/E3 LIU Block, for E3 Applications.

6.3.2 THE RECEIVE E3 FRAMER BLOCK

The Receive DS3/E3 Framer block is the second functional block within the Receive Direction of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. **Figure 244** presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive DS3/E3 Framer block highlighted.

FIGURE 244. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE DS3/E3 FRAMER BLOCK HIGHLIGHTED)



The purpose of the Receive E3 Framer block within the XRT79L71 is to accomplish the following.

- To acquire and maintain frame synchronization with the incoming E3 data-stream
- To detect and declare all of the following defect/error conditions.
  - **a.** LOS (Loss of Signal)
  - **b.** AIS (Alarm Indication Signal)
  - **c.** LOF (Loss of Frame)
  - **d.** FERF/RDI (Far-End Receive Failure or Remote Defect Indicator)
  - **e.** FEBE/REI (Far-End Block Error or Remote Error Indicator) Events
  - **f.** EM Byte Errors

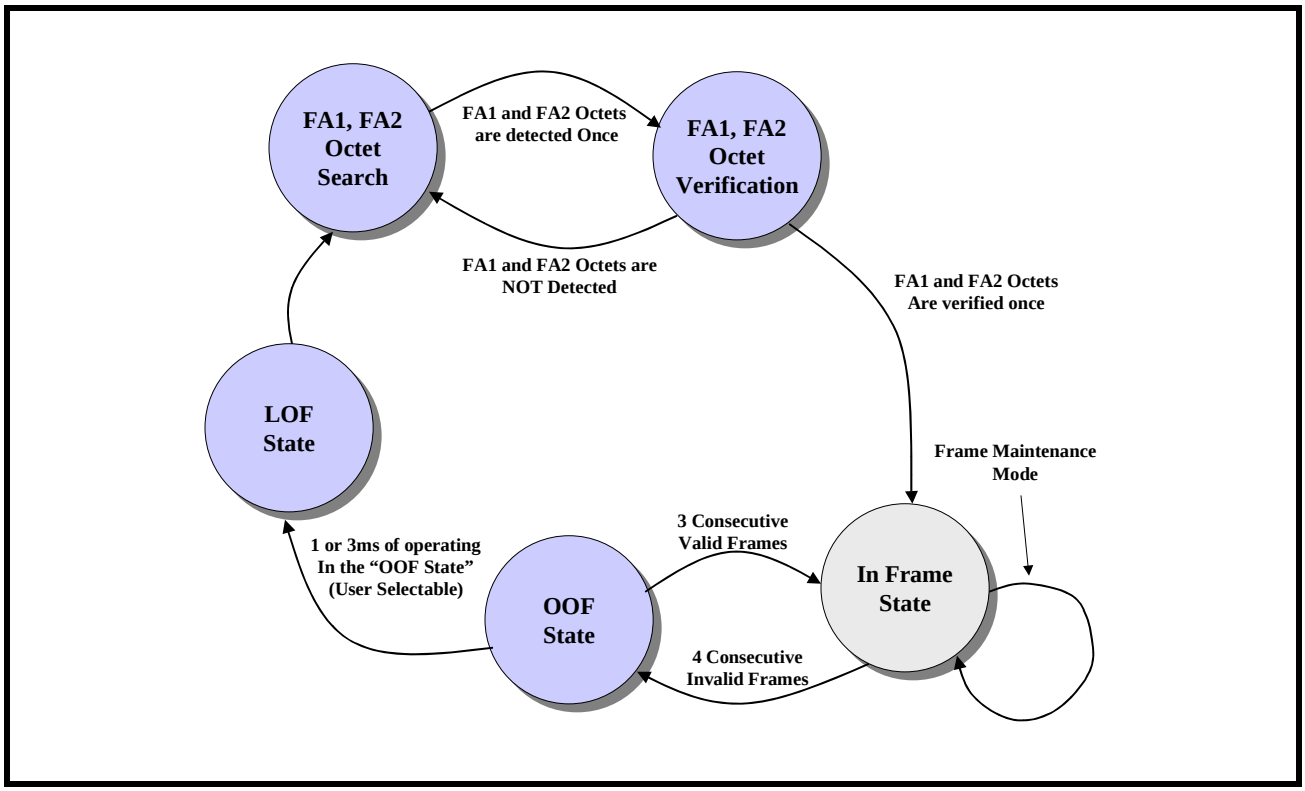
At any given time, the Receive E3 Framer block will be operating in one of two modes.

- **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.

6.3.2.1 THE FRAME-ACQUISITION MODES

The operation of the Receive E3 Framer block, while in the Frame Acquisition mode, is best understood by reviewing the Receive E3 Framer block's Frame Acquisition/Maintenance Algorithm state machine diagram, presented below in Figure 245.

FIGURE 245. THE RECEIVE E3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM - E3, ITU-T G.832 APPLICATIONS



The Receive E3 Framer block will be performing Frame Acquisition operation while it is operating in the following states per the E3, ITU-T G.832 Frame Acquisition/Maintenance algorithm State Machine diagram, as depicted in Figure 248.

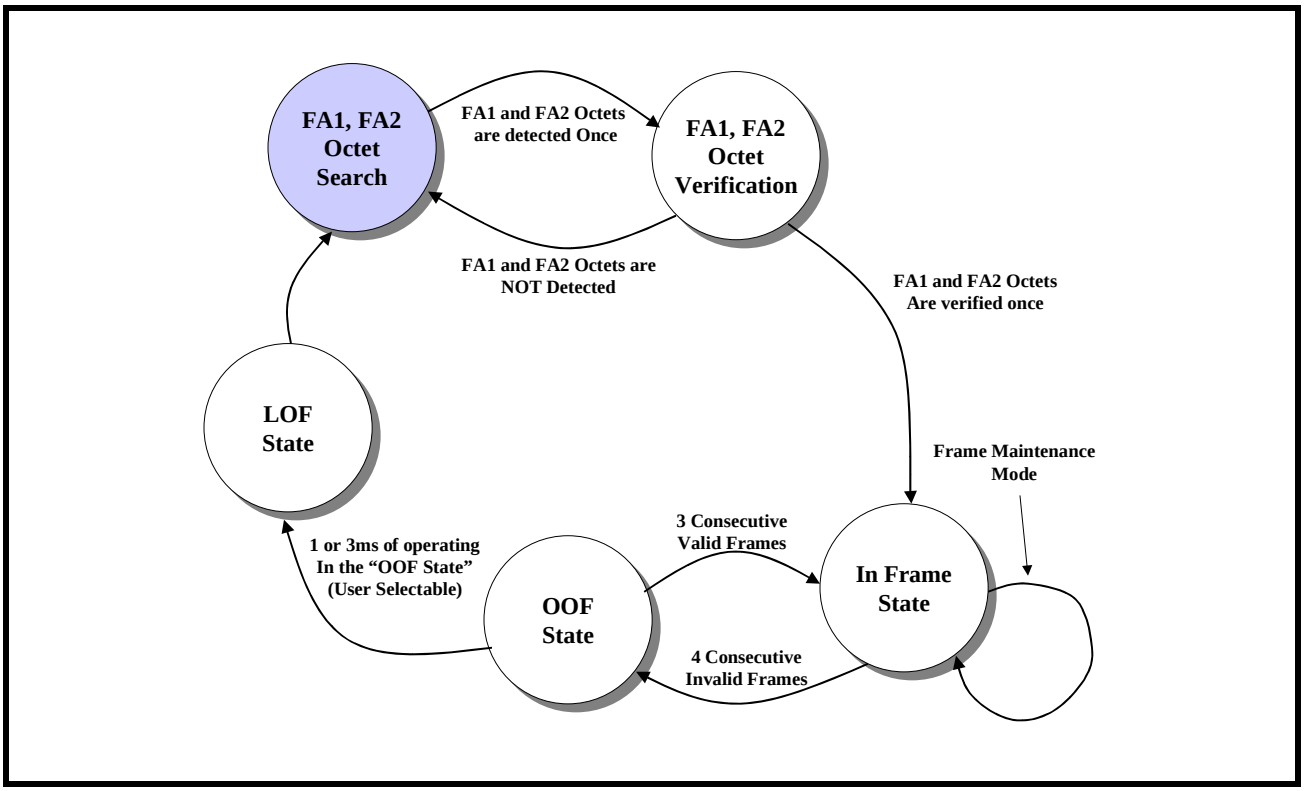
- FA1, FA2 Octet Search
- FA1, FA2 Octet Verification
- OOF State
- LOF State

Once the Receive E3 Framer block enters the In-Frame state (per Figure 248), then it will begin Frame Maintenance operation. The Receive E3 Framer block's operation in each of the Frame Acquisition states is presented below.

The FA1, FA2 Octet Search State

When the XRT79L71 is first powered up, or exits a Hardware RESET event, the Receive E3 Framer block will be operating in the Frame Acquisition Mode. At this point, the very first thing that the Receive E3 Framer block will do is to begin to look for valid E3 frames within the incoming E3 data-stream by first searching for the FA1 and FA2 octets. At this initial point the Receive E3 Framer block will be operating in the FA1 and FA2 Octet Search state within the E3 Frame Acquisition/Maintenance Algorithm state machine diagram. In order to clearly convey the mode that the Receive E3 Framer block is currently operating in, Figure 245 has been repeated below, with the FA1 and FA2 Octet Search state shaded.

FIGURE 246. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM (WITH THE FA1 AND FA2 OCTET SEARCH STATE SHADED)

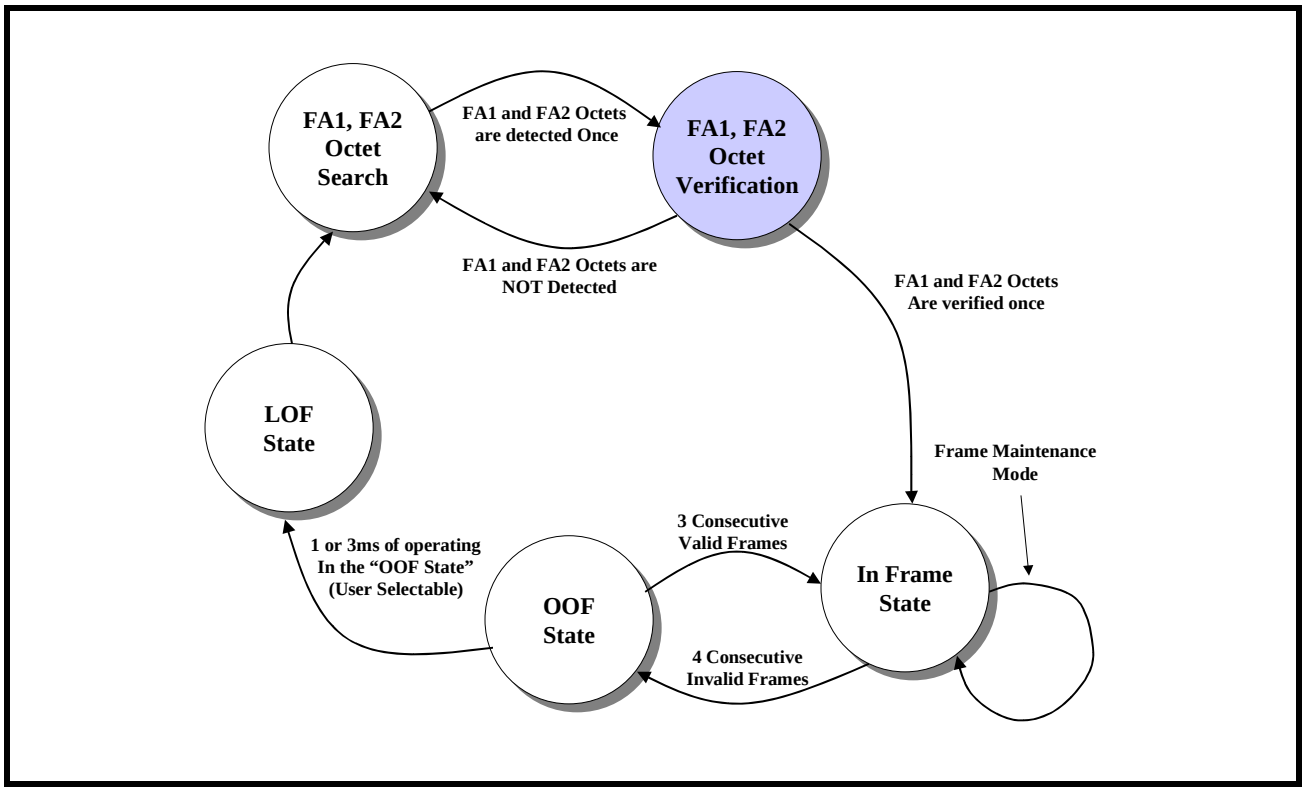


Recall from the discussion in SEE"DESCRIPTION OF THE E3, ITU-T G.832 FRAME STRUCTURE AND OVERHEAD BITS" ON PAGE 421., that each E3, ITU-T G.832 frame consists of one FA1 byte which is of the value "0xF6" that is immediately followed by the FA2 byte which is of the value "0x28". In fact, the FA1 and FA2 bytes are the very first two bytes to appear within each E3, ITU-T G.832 frame. The Receive E3 Framer block will attempt to locate this 0xF628 pattern by performing five (5) different searches in parallel. The FA1 and FA2 Octet Search will be declared successful if the Receive E3 Framer block can locate a single instantiation of the 0xF628 pattern. Once the Receive E3 Framer block detects this particular pattern, then it will declare this instantiation of the 0xF628 pattern within the incoming E3 data-stream as a possible or candidate FA1 and FA2 Octet within the incoming E3 data-stream. Additionally, the Receive E3 Framer block will now transition to the FA1, FA2 Octet Verification State, within the E3 Frame Acquisition/Maintenance algorithm (per **Figure 245**).

The FA1, FA2 Octet Verification State

Once the Receive E3 Framer block has detected the 0xF628 pattern (e.g., the concatenation of the FA1 and FA2 bytes), it must now verify that this pattern is indeed the FA1 and FA2 octets, and not some other set of bytes within the incoming E3 data-stream that are mimicking the FA1 and FA2 bytes. Hence, the purpose of the FA1, FA2 Octet Verification State is to implement this additional check on the Candidate FA1, FA2 Byte. In order to clearly convey the mode that the Receive E3 Framer block is currently operating in, **Figure 245** has been repeated below, with the FA1 and FA2 Octet Verification state shaded.

FIGURE 247. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM (WITH THE FA1 AND FA2 OCTET VERIFICATION STATE SHADED)



When the Receive E3 Framer block enters this state, it will then quit performing the bit-by-bit search within the incoming E3 data-stream for the FA1 and FA2 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 FA1 and FA2 bytes were first detected. If these two bytes exhibit the 0xF628 pattern, then the Receive E3 Framer block will conclude that it has TRULY found the Framing Alignment FA1 and FA2 bytes and will transition into the In-Frame state. However, if these two bytes do not exhibit the 0xF628 pattern, then the Receive E3 Framer block will conclude that it has been fooled by data within the incoming E3 data-stream mimicking the FA1 and FA2 bytes, and will transition back into the FA1, FA2 Octet Search state.

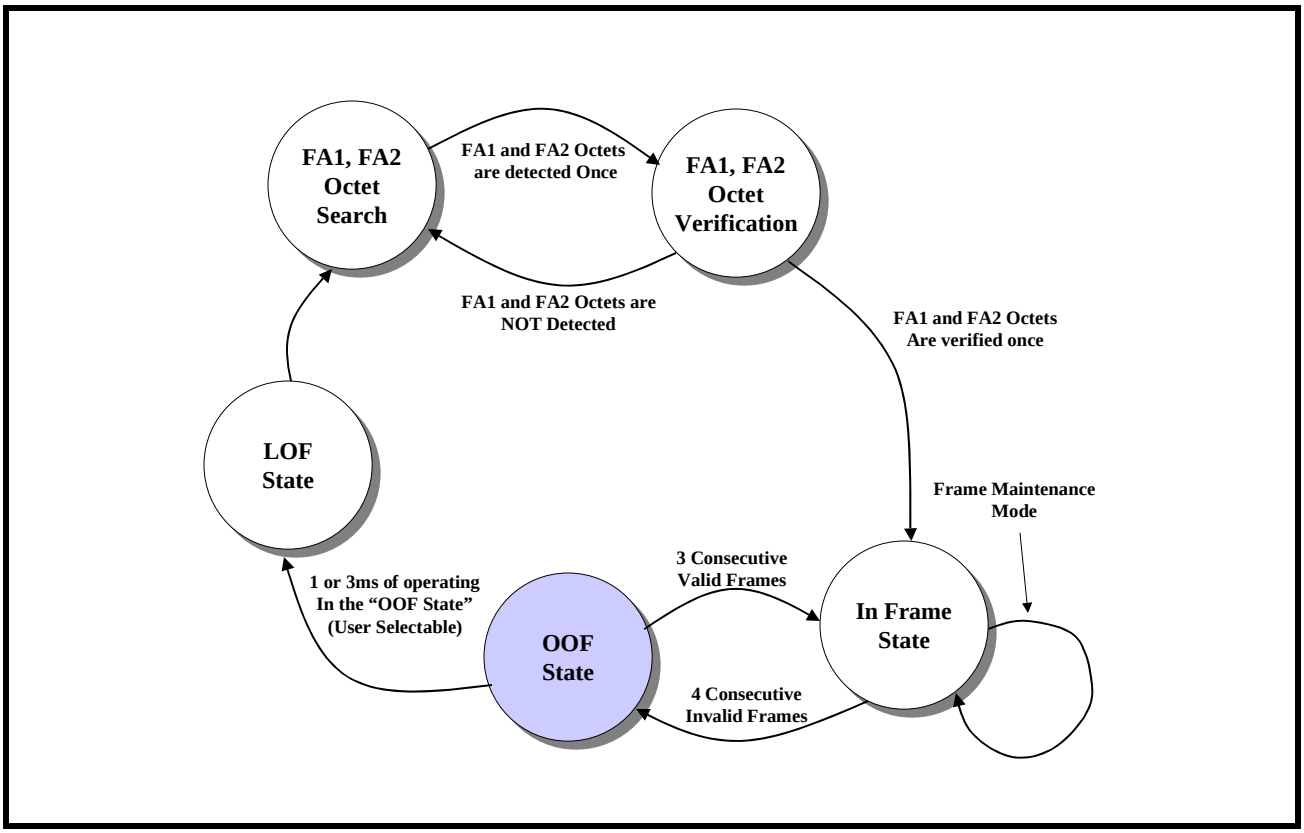
The In-Frame State

Once the Receive E3 Framer block while operating in the FA1, FA2 Octet Verification State detects the 0xF628 pattern, in the correct location (e.g., exactly one E3 frame period, after the Candidate FA1 and FA2 bytes were first detected), then it will conclude that it has correctly located the FA1 and FA2 bytes, and it will transition into the In-Frame state. Once the Receive E3 Framer block enters the In-Frame state, it will cease performing Frame Acquisition functions, and it will begin to perform Frame Maintenance functions. A detailed description of the Receive E3 Framer block, operating in the In-Frame state can be found in **SEE"THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DEFECT DECLARATION CRITERIA" ON PAGE 522.**.

The OOF (Out of Frame) State

If the Receive E3 Framer block, while operating in the In-Frame state, receives four (4) consecutive incoming E3 frames, in which the FA1 or FA2 bytes are erred, then the Receive E3 Framer block will transition into the OOF State. In order to clearly convey the mode that the Receive E3 Framer block is currently operating in, **Figure 245** has been repeated below, with the OOF State shaded.

FIGURE 248. THE STATE MACHINE DIAGRAM FOR THE RECEIVE E3 FRAMER BLOCK'S FRAME ACQUISITION/MAINTENANCE ALGORITHM (WITH THE OOF STATE SHADED)



The Receive E3 Framer block's operation, while in the OOF State is a unique mix of the Framing Maintenance and Frame Acquisition operations. In the OOF State the Receive E3 Framer block will exhibit some Frame Acquisition operational characteristics by attempting to locate the FA1 and FA2 bytes. However, the Receive E3 Framer block will also exhibit some Frame Maintenance operational behavior by still using the most recent frame synchronization for its overhead and payload byte processing.

What happens when the Receive E3 Framer block transitions into the OOF State?

As the Receive E3 Framer block transitions from the In-Frame into the OOF State, it will inform the Microprocessor of this fact by doing all of the following.

- **Declaring the OOF Defect Condition**

The Receive E3 Framer block will indicate that it is declaring the OOF Defect Condition by setting bit 5 (OOF Defect Declared), within the Receive E3 Configuration and Status Register # 2, to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	1	0	0	0	0	0

- **Generating the Change in OOF Defect Condition Interrupt**

The Receive E3 Framer block will indicate that it is generating the Change in OOF Defect Condition Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "Low".

b. Setting Bit 3 (Change in OOF Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

What happens if the Receive E3 Framer block transitions back into the In-Frame State?

If the Receive E3 Framer block while operating in the OOF State is able to find/locate the FA1 and FA2 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 inform the Microprocessor of this occurrence by doing all of the following.

- **Clearing the OOF Defect Condition**

The Receive E3 Framer block will indicate that it is clearing the OOF Defect Condition by setting Bit 5 (OOF Defect Declared), within the Receive E3 Configuration and Status Register # 2, to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- **Generating the Change in OOF Defect Condition Interrupt**

The Receive E3 Framer block will indicate that it is declaring the Change in OOF Defect Condition Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "Low".

b. Setting Bit 3 (Change in OOF Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

Transitioning into the LOF State

However, if the Receive E3 Framer block resides in the OOF State for more than the User-Selectable number of E3 frame periods without being able to locate the FA1 and FA2 bytes, then the Receive E3 Framer block will automatically transition into the LOF State.

Selecting the number E3 Frame Periods, before transitioning into the LOF State

The user can configure the Receive E3 Framer block to reside in the OOF State and search the incoming E3 data-stream for either 1 or 3ms, prior to transitioning into the LOF State. The user can accomplish this configuration setting by writing the appropriate value into Bit 7 (Receive LOF Algo), within the Receive E3 Configuration and Status Register # 2 as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
X	0	0	0	0	0	0	0

Setting this bit-field to "0" configures the Receive E3 Framer block to reside within the OOF State for up to 3ms (24 E3 frame periods) before transitioning into the LOF State. Conversely, setting this bit-field to "1" configures the Receive E3 Framer block to reside within the OOF State for up to 1ms (8 E3 frame periods) before transitioning into the LOF State.

The LOF (Loss of Frame) 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 that it had, and
- The Receive E3 Framer block will make an unconditional transition to the FA1, FA2 Octet Search State, within the Receive E3 Framer Block's - Frame Acquisition/Maintenance Algorithm.

Additionally, the Receive E3 Framer block will notify the Microprocessor/Microcontroller of this transition into the LOF State, by doing the following.

- **Declaring the LOF Defect Condition**

The Receive E3 Framer block will indicate that it is declaring the LOF Defect Condition by setting bit 6 (LOF Defect Declared), within the Receive E3 Configuration and Status Register # 2 to "1", as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	1	1	0	0	0	0	0

- **Generating the Change in LOF Defect Condition Interrupt**

The Receive E3 Framer block will indicate that it is declaring the Change in LOF Defect Condition Interrupt by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "Low".

b. Setting Bit 2 (Change in LOF Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

6.3.2.2 THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DEFECT DECLARATION CRITERIA

Once the Receive E3 Framer block has entered the In-Frame state, then it is considered to be operating in the Frame Maintenance Mode. As the Receive E3 Framer block transitions into the In-Frame State, it will notify the Microprocessor/Microcontroller of this fact by doing all of the following.

- **Clearing the OOF (Out-of-Frame) and LOF (Loss of Frame) Defect Conditions.**

The Receive E3 Framer block will indicate that it is clearing both the OOF and LOF Defect Conditions by setting Bits 5 (OOF Defect Declared) and 6 (LOF Defect Declared) within the Receive E3 Configuration and Status Register # 2 to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- **Generating the Change in OOF Defect Condition and Change in LOF Defect Condition Interrupts**

The Receive E3 Framer block will indicate that it is generating the Change in OOF Defect Condition and the Change in LOF Defect Condition Interrupts, by doing all of the following.

a. Asserting the Interrupt Request output pin (INT*), by pulling it "Low".

b. Setting Bits 2 (Change in LOF Defect Condition Interrupt Status) and 3 (Change in OOF Defect Condition Interrupt Status) within the Receive E3 Interrupt Status Register # 1 to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	1	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 Framing Alignment Octets FA1 and FA2 are present at their proper locations, within the incoming E3 data-stream. In general, as long as the FA1 and FA2 bytes are present at their proper locations, with a small number of errors the Receive E3 Framer block will continue to operate in the Frame Maintenance Mode. However, if this Receive E3 Framer block begins to detect a large number of FA1 and FA2 byte errors within the incoming E3 data-stream, then it will exit the In-Frame state and will then declare the OOF defect condition whenever the Receive E3 Framer block receives at least four (4) consecutive E3 frames, in which the FA1 or FA2 bytes were erred.

Forcing a Reframe via Software Command

The Receive DS3/E3 Framer block permits the user to command a reframe procedure with the Receive E3 Framer block via software command. The user can accomplish this by inducing a 0 to 1 transition in Bit 0 (Reframe), within the I/O Control Register as depicted below. Once the user executes this step, then the Receive E3 Framer block will be forced into the Frame Acquisition Mode or more specifically, in the FA1, FA2 Octet Search State, per Figure 248, and will begin to search for the FA1 and FA2 bytes. The XRT79L71 will also respond to this command by declaring both the OOF and LOF Defect Conditions, and by generating the Change in OOF Defect Condition and the Change in the LOF Defect Condition interrupts.

I/O Control Register (Address = 0x1101)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Disable TxLOC	LOC	Disable RxLOC	Unused				Reframe
R/W	R/O	R/W	R/O	R/O	R/O	R/O	R/W
1	0	1	0	1	0	0	0 -> 1

NOTE: *After the user has implemented the 0 to 1 transition within Bit 0 (Reframe), the user should also go back and induce a 1 to 0 transition within this bit-field.*

6.3.2.3 DECLARING AND CLEARING THE LOS DEFECT CONDITION

The Receive E3 Framer block has the responsibility for declaring and clearing the LOS (Loss of Signal) defect condition, within the incoming E3 data-stream, as described below.

6.3.2.3.1 Declaring the LOS Defect Condition

The Receive E3 Framer block will declare the Loss of Signal (LOS) Defect condition when it detects at least 32 consecutive "0s" within the incoming E3 data-stream, or if the Receive DS3/E3 LIU Block declares the LOS defect condition. The Receive E3 Framer block will indicate that it is declaring the LOS defect condition by doing all of the following.

- Setting Bit 4 (LOS Defect Declared) within the Receive E3 Configuration and Status Register # 2 to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	1	0	0	0	0

- The Receive E3 Framer block will also generate the Change of LOS Defect Condition Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 1 (Change in LOS Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

NOTE: *The LOS Defect Declaration Criteria for the Receive DS3/E3 LIU Block will be discussed in* ***SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.****.*

6.3.2.3.2 Clearing the LOS Defect Condition

The Receive E3 Framer block will clear the LOS Defect condition when both of the following conditions are met.

a. When the Receive E3 Framer block while declaring the LOS Defect condition receive a stream of 32 consecutive E3 bits, which does not contain a string of four (4) consecutive "0s".

b. When the Receive DS3/E3 LIU Block clear the LOS defect condition.

The Receive E3 Framer block will indicate that it is clearing the LOS defect condition by:

- Setting Bit 4 (LOS Defect Declared) within the Receive E3 Configuration and Status Register # 2 to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Direct Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

The Receive E3 Framer block will also generate the Change of LOS Defect Condition interrupt by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 1 (Change in LOS Defect Condition Interrupt Status), within the Receive E3 Configuration and Status Register # 2 to "1" as illustrated below.

Receive E3 Interrupt Status Register # 1 - G.832 (Direct Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

NOTE: *The LOS Clearance criteria for the Receive DS3/E3 LIU Block will be discussed in* ***SEE"THE LOS DECLARATION AND CLEARANCE CRITERIA FOR E3 APPLICATIONS" ON PAGE 365.****.*

Configuration Options for the LOS Declaration/Clearance Criteria

The Receive DS3/E3 Framer block within the XRT79L71 permits the user to change the LOS Declaration criteria such that the LOS defect condition is declared only if the Receive DS3/E3 LIU Block Interface declares the LOS defect condition. If this configuration selection is implemented, then the internally-generated LOS criteria of 32 consecutive 0s will be disabled. This configuration selection can be accomplished by writing a "0" to bit 3 (Internal LOS Enable) within the Operating Mode Register as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	1

6.3.2.3.3 The Relationship between the LOS Defect Condition being declared or cleared in the Receive DS3/E3 LIU Block and in the Receive DS3/E3 Framer Block

6.3.2.4 DECLARING AND CLEARING THE AIS DEFECT CONDITION

The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the AIS (Alarm Indication Signal) defect, as described below.

If the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing format, then the Receive E3 Framer block will declare the AIS defect condition anytime it receives an E3 data-stream that contains 7 of less "0s" within two consecutive E3 frame periods.

If the Receive E3 Framer block declares the AIS defect condition, then it will do all of the following.

- It will set Bit 3 (AIS Defect Declared) within the Receive E3 Configuration and Status Register # 2 to "1", as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxT Mark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	1	0	0	0

- The Receive E3 Framer block will also generate the Change in AIS Defect Condition Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 0 (Change in AIS Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

Clearing the AIS Defect Condition

The Receive E3 Framer block will clear the AIS Defect condition whenever it receives two consecutive E3 frames that contain 8 or more "0s", within the incoming E3 data-stream. The Receive E3 Framer block will indicate that it is clearing the AIS defect by:

- It will set Bit 3 (AIS Defect Declared) within the Receive E3 Configuration and Status Register # 2 to "0", as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxT Mark	FERF Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

- The Receive E3 Framer block will also generate the Change in AIS Defect Condition Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low"), and setting Bit 0 (Change in AIS Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 1 to "1" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Status	Change in SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF Defect Condition Interrupt Status	Change in LOF Defect Condition Interrupt Status	Change in LOS Defect Condition Interrupt Status	Change in AIS Defect Condition Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	1

6.3.2.5 DECLARING AND CLEARING THE FERF/RDI DEFECT CONDITION

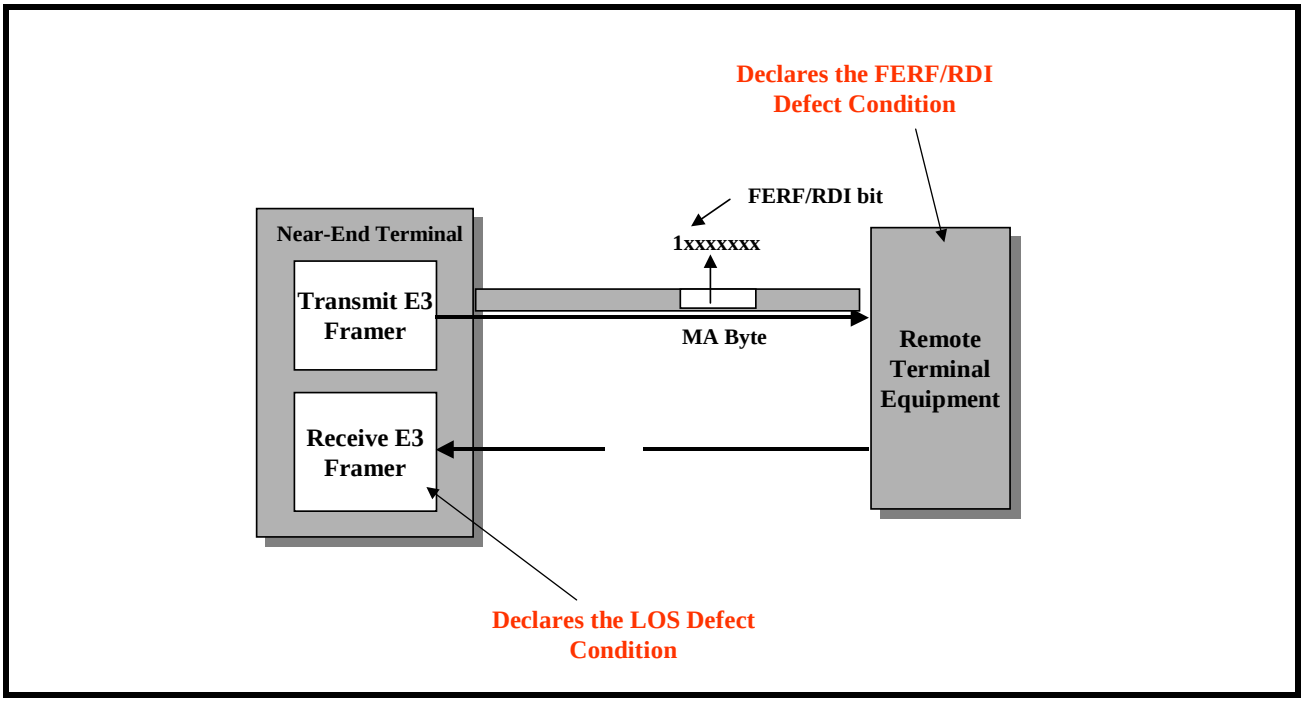
The Receive DS3/E3 Framer block has the responsibility for declaring and clearing the FERF/RDI defect condition, as described below.

6.3.2.5.1 Declaring the FERF/RDI (Far-End Receive Failure/Remote Defect Indicator) Defect Condition

The Receive E3 Framer block will declare the FERF or RDI defect condition, if it starts to receive E3 frames in which the FERF/RDI bit-field is set to "1".

Recall, that back in SEE"TRANSMITTING THE FERF/RDI INDICATOR" ON PAGE 499., we described how one can configure the Transmit DS3/E3 Framer block to automatically transmit the FERF indicator to the remote terminal, anytime and for the duration that the Near-End Corresponding Receive DS3/E3 Framer Block declares either the LOS, LOF/OOF or AIS defect condition. Figure 249 recaps some of this discussion by presenting a figure that depicts the Transmit DS3/E3 Framer block automatically transmitting the FERF indicator to the remote terminal equipment by setting the FERF/RDI bit within each outbound E3 frame to "1" because the Near-End Corresponding Receive DS3/E3 Framer block was declaring the LOS defect condition.

FIGURE 249. A SIMPLE ILLUSTRATION OF THE NEAR-END TERMINAL EQUIPMENT TRANSMITTING THE FERF/RDI INDICATOR TO THE REMOTE TERMINAL EQUIPMENT



In SEE"TRANSMITTING THE FERF/RDI INDICATOR" ON PAGE 499., we described how a given Terminal will generate the FERF/RDI indicator. In this Section, we will describe how a given Terminal that contains the XRT79L71 will respond to receiving the FERF/RDI indicator from the remote terminal equipment.

The FERF/RDI Defect Declaration Criteria

The Receive E3 Framer block will declare the FERF/RDI defect condition, if it receives a user-selectable number of consecutive E3 frames, in which the FERF/RDI bit-field is set to "1".

The user can select the appropriate number of consecutive incoming E3 frames in which the FERF/RDI bit-field are set to "1" prior to the Receive E3 Framer block declaring the FERF/RDI Defect condition by writing the appropriate value into Bit 4 (RxFERF Algo) within the Receive E3 Configuration and Status Register # 1 as depicted below.

Receive E3 Configuration and Status Register # 1 - G.832 (Direct Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxPLDType[2:0]			RxFERF Algo.	RxTMark Algo	RxPLDTypeExp[2:0]		
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	1	0	X	0	0	1	0

Setting this bit-field to "0" configures the Receive E3 Framer block to declare the FERF/RDI defect condition, if it has received at least three (3) consecutive E3 frames, in which the FERF/RDI bit-field has been set to "1". Conversely, setting this bit-field to "1" configures the Receive E3 Framer block to declare the FERF/RDI defect condition, if it has received at least five (5) consecutive E3 frames, in which the FERF/RDI bit-field has been set to "1".

When the Receive E3 Framer block declares the FERF or RDI defect condition in the incoming E3 frame, then it will then do the following.

- It will set Bit 0 (FERF/RDI Defect Declared), within the Receive E3 Configuration and Status Register # 2, to "1" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF/RDI Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

This bit-field will remain asserted for the duration that the Receive E3 Framer block declares the FERF/RDI defect condition.

- The Receive E3 Framer block will also generate the Change in FERF/RDI Defect Condition Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 3 (Change in FERF/RDI Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 2 to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

6.3.2.5.2 Clearing the FERF/RDI Defect Condition

The Receive E3 Framer block will clear the FERF/RDI Defect Condition whenever it receives a User-Selectable number of E3 frames, in which the FERF/RDI bit-field is set to "0".

Setting the FERF/RDI Defect Clearance Criteria

The user can specify the FERF/RDI Defect Clearance Criteria, by writing the appropriate value into Bit 4 (RxFERF Algo), as depicted below.

Receive E3 Configuration and Status Register # 1 - G.832 (Direct Address = 0x1110)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxPLDType[2:0]			RxFERF Algo.	RxTMark Algo	RxPLDTypeExp[2:0]		
R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W
0	1	0	X	0	0	1	0

If this bit-field is set to "0", then the Receive E3 Framer block will clear the FERF/RDI defect condition anytime it receives at least three (3) consecutive E3 frames, in which the FERF/RDI bit-field has been set to "0". Conversely, if this bit-field is set to "1", then the Receive E3 Framer block will clear the FERF/RDI defect condition anytime it receives at least five (5) consecutive E3 frames, in which the FERF/RDI bit-field has been set to "0".

Whenever the Receive E3 Framer block clears the FERF/RDI defect condition, it will do so, by doing all of the following.

- Setting Bit 0 (FERF/RD Defect DeclaredI) within the Receive E3 Configuration and Status Register # 2 to "0" as depicted below.

Receive E3 Configuration and Status Register # 2 - G.832 (Address = 0x1111)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive LOF Algo	LOF Defect Declared	OOF Defect Declared	LOS Defect Declared	AIS Defect Declared	RxPLD Unstab	RxTMark	FERF/RDI Defect Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	0

The Receive E3 Framer block will also generate the Change in FERF/RDI Defect Condition Interrupt, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 3 (Change in FERF/RDI Defect Condition Interrupt Status), within the Receive E3 Interrupt Status Register # 2 to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	1	0	0	0

6.3.2.6 DETECTING EM BYTE ERRORS

The Receive E3 Framer block has the responsibility for detecting and flagging the occurrence of EM byte errors, as described below.

Processing at the Remote Terminal Equipment

As the remote terminal equipment is generating and transmitting the E3 data-stream that the local terminal equipment will ultimately receive and process, it will compute the BIP-8 value over an entire E3 frame. The results of this BIP-8 calculation will be inserted into the EM byte-position, within the very next outbound E3 frame. The purpose of the EM byte, is to support Performance Monitoring and Error Detection within the E3 data-stream, as it is transported from one terminal equipment to another.

Processing at the Local Terminal Equipment

As the Receive E3 Framer block receives a given E3 frame that was generated by the remote terminal equipment will locally-compute its own BIP-8 value for this incoming E3 frame. Afterwards, the Receive E3 Framer block will compare its locally-computed BIP-8 value with the contents of the EM byte, within the very next incoming E3 frame. If these two values match, then the Receive E3 Framer block will conclude that it has received this particular E3 frame, in an un-erred manner. Conversely, if these two values DO NOT match, then the Receive E3 Framer block will conclude that is has received this particular E3 frame, in an erred manner.

If the Receive E3 Framer block determines that the EM byte within a given E3 frame are erred, then it will do the following.

- It will generate the Detection of BIP-8 Error Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 2 (Detection of BIP-8 Error Interrupt Status), within the Receive E3 Interrupt Status Register # 2 to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	1	0	0

- It will increment the PMON P-bit/Parity Error Count Register once for each E3 frame that is determined to have an erred EM byte. The PMON P-bit/Parity Error Count Register is located at Address = 0x1154 and 0x1155. The bit-format for each of these registers is presented below.

PMON Parity/P-Bit Error Count Register - MSB (Address = 0x1154)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON Parity/P-Bit Error Count Register - LSB (Address = 0x1155)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Parity_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these Performance Monitor Register, please see Section 1.4.*

- It will also increment the One Second - P-bit/Parity Error Count - Accumulator Register once for each incoming E3 frame that is determined to have an erred EM byte. The One Second - P-Bit/Parity Error Count - Accumulator Register is located at Address = 0x1170 and 0x1171. The bit-format for this 16-bit register is presented below.

One Second - Parity Error Accumulator Register - MSB (Address = 0x1170)

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

One Second - Parity Error Accumulator Register - LSB (Address = 0x1171)

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

NOTE: *The Near-End Transmit DS3/E3 Framer block also within this particular XRT79L71 will automatically be configured to set the FEBE/REI bit-field within the MA byte of a given outbound E3 frame to "1" for each time in which the Receive E3 Framer block receives an E3 frame with an erred EM byte. Please see* ***SEE"TRANSMITTING THE FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR) INDICATOR" ON PAGE 505.*** *for more information on this transmission of the FEBE/REI indicator.*

6.3.2.7 DETECTING FEBE/REI (FAR-END BLOCK ERROR/REMOTE ERROR INDICATOR) EVENTS

The Receive DS3/E3 Framer block has the responsibility for detecting and tallying the number of times that it receives disturbed FEBE/REI indicators from the remote terminal equipment, as described below.

Each E3 frame consists of a single FEBE/REI bit-field. The remote terminal equipment which is generating the incoming E3 data-stream will set the FEBE/REI bit-field to the appropriate value that indicates whether or not the remote terminal is experiencing any EM byte errors. If the remote terminal equipment is currently not detecting any EM byte errors, then it will set the FEBE/REI bit-field within the MA byte of each outbound E3 frame to "0". Hence, the FEBE/REI value for an un-erred condition is "0".

Conversely, if the remote terminal equipment is detecting EM byte errors, then it will proceed to set the FEBE/REI bit-field within the MA byte of the very next outbound E3 frame to "1".

NOTES:

1. *The remote terminal equipment will set the FEBE/REI bit-field to "1", within a given outbound E3 frame, each time the Near-End Corresponding Receive E3 Framer block has received an E3 frame with an erred EM byte.*
2. *The FEBE/REI value within the incoming E3 data-stream is a reflection of the receive conditions of the remote terminal equipment, not the local terminal equipment.*

If the Receive E3 Framer block receives any E3 frames, in which the FEBE/REI bit-field is set to "1", then it will do the following.

- It will generate the Detection of FEBE Event Interrupt request, by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 4 (Detection of FEBE Event Interrupt Status), within the Receive E3 Interrupt Status Register # 2 to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	1	0	0	0	0

- It will increment the PMON FEBE Event Count Register once, for each E3 frame that it receives, in which the FEBE/REI bit-field was set to "1". The bit-format and address locations of these registers are presented below.

PMON FEBE Event Count Register - MSB (Address = 0x1156)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PMON FEBE Event Count Register - LSB (Address = 0x1157)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_FEBE_Event_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *For instructions on how to read out these "Performance Monitor" Registers, please see Section 1.4.*

6.3.2.8 DETECTING FRAMING BYTE ERRORS

Recall in SEE"THE FRAME-MAINTENANCE MODE - THE OOF AND LOF DEFECT DECLARATION CRITERIA" ON PAGE 522., we mentioned that in order to verify that the Receive E3 Framer is maintaining proper Frame Synchronization with the incoming E3 data-stream, it will continuously check and verify that the FA1 and FA2 bytes (1) can be found in their proper locations, and (2) that they are of the correct value. This Section went on to state that if the Receive E3 Framer block were to detect errors in the FA1 or FA2 bytes, within four (4) consecutive E3 frames, then it (the Receive E3 Framer block) would transition over to the "OOF State" and would declare the "OOF Defect" condition.

In addition to checking and determining whether or not to declare the "OOF" or "LOF" defect condition, the Receive DS3/E3 Framer block will also flag and tally the occurrences of any Framing (FA1 or FA2) Byte Errors (that are detected within the incoming E3 data-stream), as described below.

While the Receive E3 Framer block is operating in the Frame Maintenance Mode, it will continue to check for valid FA1 and FA2 bytes. If the Receive E3 Framer block detects any errors in the FA1 or FA2 bytes, then it will do the following.

- It will generate the Detection of Framing Byte Error Interrupt request by asserting the Interrupt Output pin (e.g., by pulling it "Low") and setting Bit 1 (Detection of Framing Byte Error Interrupt Status), within the Receive E3 Interrupt Status Register # 2 as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	1	0

- It will increment the PMON Framing Bit/Byte Error Count Registers, once for each E3 frame that is determined to contain an erred FA1 or FA2 byte. The PMON Framing Bit/Byte Error Count Register is located at Address = 0x1152/1153. The bit-format of these registers is presented below.

PMON Framing Bit/Byte Error Count Register - MSB (Address = 0x1152)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Upper_Byte[7:0]							
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 = 0x1153)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PMON_Framing_Bit/Byte_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTES:

1. *1.These "Performance Monitor" Registers will cease to increment anytime the Receive E3 Framer block is declaring the "OOF" or "LOF" defect condition.*
2. *For instructions on how to read out these "Performance Monitor" Registers, please see Section 1.4.*

6.3.2.9 Declaring and Clearing the Payload-Type Mismatch Defect Condition

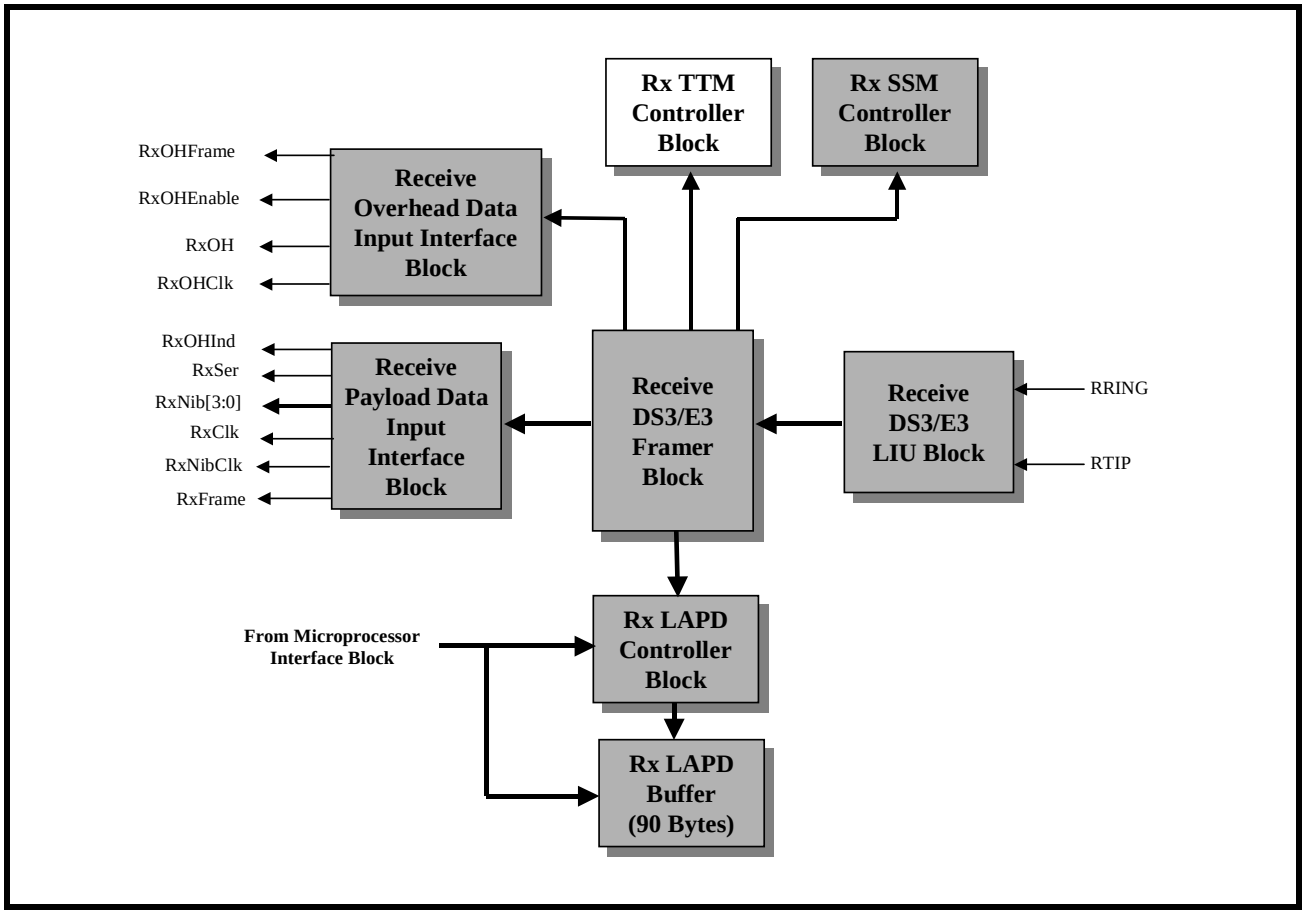
6.3.2.10 Monitoring the GC Byte within the incoming E3 data-stream

6.3.2.11 Monitoring the NR Byte within the incoming E3 data-stream

6.3.3 RECEIVE TRAIL-TRACE MESSAGE CONTROLLER BLOCK

The Receive Trail-Trace Message Controller block is the third functional block within the Receive Direction of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. **Figure 250** presents an illustration of the Receive Direction circuitry, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive Trail-Trace Message Controller block highlighted.

FIGURE 250. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE TRAIL-TRACE MESSAGE CONTROLLER BLOCK HIGHLIGHTED)



6.3.3.1 AN INTRODUCTION TO TRAIL-TRACE MESSAGES

If the XRT79L71 is operating in the E3, ITU-T G.832 Frame Format, then the TR (Trail-Trace) Byte can be used to receive Trail-Trace Messages, also known as a Trail Access Point Identifier. The purpose of the Trail Access Point Identifier Message is to permit a given terminal to repetitively identify itself, thereby permitting the Receiving Terminal to verify its continued connection to the correct terminal, via the E3 Transport Medium.

The Trail-Trace Message consists of a 16-byte string that is repeatedly transmitted from one terminal to the remote terminal equipment. The byte-format of the Trail-Trace Message is presented below in Table 66.

TABLE 66: THE BYTE-FORMAT OF THE TRAIL-TRACE MESSAGE THAT THIS BEING TRANSPORTED VIA AN E3 DATA-STREAM VIA THE TR BYTE

BYTE NUMBER		BIT 1 (MSB)	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8 (LSB)
1 (FrameStart Marker)		1	CRC_Value[6:0]						
2		0	TTM_Byte_1[6:0]						
*		0	TTM_Byte_N[6:0]						
15		0	TTM_Byte_15[6:0]						
16		0	TTM_Byte_16[6:0]						

The very first byte of this 16-byte string is referred to as the Frame Start Marker byte. The Frame Start Marker byte is typically of the form [1, C6, C5, C4, C3, C2, C1, C0]. This "1" in the MSB (most significant bit) of this first byte is used by a Receiving Terminal to identify this byte as the Frame Start Marker byte. All of the remaining bytes within a given Trail-Trace message will contain a "0" in their MSB. The bits C6 through C0 typically contains the results of a CRC-7 calculation that was performed over the previous 16-byte Trail-Trace Message frame.

NOTE: *The XRT79L71 will NOT compute nor verify this CRC-7 value that is residing within these bit-fields. The user will need to externally compute and verify the CRC-7 value over a given inbound Trail-Trace Message.*

The remaining 15 bytes are used for the transport of 15 ASCII characters which transport the actual Trail-Trace message. Typically, these Trail-Trace Messages are used to transport messages that are of content/protocol that are compliant to some of the requirements as listed in ITU-T G.831 and E.164.

NOTE: *As mentioned earlier, the MSBs (most significant bits) within the each of the 15 remaining (e.g., non-Frame Start Marker byte) MUST be set to "0".*

6.3.3.2 CONFIGURING THE XRT79L71 TO RECEIVE TRAIL-TRACE MESSAGES

The XRT79L71 contains a total of 16 Receive Trail-Trace Message Registers, and 16 Transmit Trail-Trace Message Registers. The role of the Transmit Trail-Trace Message Registers is described in SEE"CONFIGURING THE XRT79L71 TO TRANSMIT TRAIL-TRACE MESSAGES" ON PAGE 490..

As mentioned earlier, the XRT79L71 contains 16 Receive Trail-Trace Message Registers (e.g., Receive Trail-Trace Message Register - Byte 1 through Receive Trail-Trace Message Register - Byte 16). The purpose of these registers is to permit the user to read out and process the contents of the incoming Trail-Trace Message.

For Trail-Trace Message purposes, the Receive Trail-Trace Message Controller block along with the Receive DS3/E3 Framer block will group 16 consecutive inbound E3 frames, into a Trail-Trace Message Super-Frame. As the Receive Section of the XRT79L71 is receiving the very first E3 frame within this Trail-Trace Message Super-Frame, it will read out the contents of the TR byte within this particular E3 frame, and it will write that value into the Receive Trail-Trace Message Register - Byte 1. When the Receive Section of the XRT79L71 is receiving the second E3 frame within this Trail-Trace Message Super-Frame, then it will read out the contents of the TR byte within this second E3 frame and it will write that value into the Receive Trail-Trace Message Register - Byte 2. As the Receive Section of the XRT79L71 is receiving each subsequent E3 frame, within this particular Trail-Trace Message Super Frame, it will continue to read out the contents of the TR byte-fields within these incoming E3 frames, and it will continue to increment to, and write the contents of these TR bytes into the very next Receive Trail-Trace Message Register. After the Receive Section of the XRT79L71 has received the 16th inbound E3 frame, within this Trail-Trace Message Super-Frame (e.g., it has read out the contents of the TR byte-field within this 16th E3 frame, and has written the contents of this TR byte into the Receive Trail-Trace Message Register - Byte 16), it will begin to receive and process the TR bytes within a new incoming Trail-Trace Message Super-Frame. As the Receive Section of the XRT79L71 does this, it will repeat

the above-mentioned process by reading out the contents of the TR byte, within this first E3 frame and writing the contents of this TR byte into the Receive Trail-Trace Message Register - Byte 1, as so on.

How the Receive Trail-Trace Message Controller Identifies the very first incoming E3 frame, within a Trail-Trace Message Super-Frame

Earlier in this section, we mentioned that of the 16 TR bytes within a given Trail-Trace Message Super-Frame only the Frame-Start Marker byte or the TR byte within the very first E3 frame within this Trail-Trace Message Super-Frame has its MSB (most significant bit) set to "1". The MSB's within all of the remaining 15 TR bytes will be set to "0".

As the Receive Trail-Trace Message Controller block receives each of the TR bytes, within the incoming E3 data-stream it will search for the Frame Start Marker TR byte. Once the Receive Trail-Trace Message Controller block locates this Frame Start Marker byte, then it will (1) read out the contents of this Frame Start Marker byte, and (2) it will write this data into the Receive Trail-Trace Message Register - Byte 1. At this point, the Receive Trail-Trace Message Controller block has acquired framing alignment with the incoming Trail-Trace Buffer Super Frame, and at this point will proceed to read out the contents of the TR within all subsequent E3 frames, and write the contents of these TR bytes into the appropriate one of 16 Receive Trail-Trace Message Registers (e.g., Receive Trail-Trace Message Register - Byte 1 through Receive Trail-Trace Message Register - Byte 16). As the Receive Trail-Trace Message Controller block receives and processes the contents of these TR bytes, it will continue to check and make sure that it has proper framing alignment with the incoming Trail-Trace Message Super Frame. In other words, the Receive Trail-Trace Message Controller block will continue to verify that it is properly receiving the Frame-Start Marker byte every 16 frames, and that the Frame-Start Marker is consistently located in the correct incoming E3 frame. Finally, the Receive Trail-Trace Message Controller block will also parse through and check for any changes in content within the incoming Trail-Trace Message.

What happens if there is a change in the incoming Trail-Trace Message?

If the Receive Trail-Trace Message Controller block detects a change in the content of the incoming Trail-Trace Message, or if the Receive Trail-Trace Message Controller detects a change in Trail-Trace Message Super-Frame alignment, then it will do the following.

1. It will generate the Change in Receive Trail-Trace Message Interrupt. The Receive Trail-Trace Message Controller will indicate that it is generating this interrupt request by asserting the Interrupt Request output pin by pulling it "Low" and by setting Bit 6 (Change in Receive Trail-Trace Message Interrupt Status) within the Receive E3 Interrupt Status Register # 2 to "1" as depicted below.

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in RxTTB Message Interrupt Status	Reserved	Detection of FEBE Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrupt Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

2. The Receive Trail-Trace Message Controller will proceed to receive and write the contents of this newly received Trail-Trace Message into the Receive Trail-Trace Message Registers.

For completeness, the address locations and bit-formats of each of the Receive Trail-Trace Message registers are presented below.

Receive E3 Trail-Trace Message Byte 1 Register - G.832 (Address = 0x111C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	C6	C5	C4	C3	C2	C1	C0

NOTES:

1. *The contents within the "Receive Trail-Trace Message Register - Byte 1" will typically be of the [1, C6, C5, C4, C3, C2, C1, C0]. The "1" within the MSB (most significant bit) position of this byte is used to designate that this octet is the "Frame-Start Marker" byte (e.g., the very first of 16 TR byte, within a "Trail-Trace Message" Super-Frame) within the "incoming" Trail-Trace Message. The remaining seven bits (e.g., C6 through C0) is typically the result of a CRC-7 calculation that was computed over the previous "Trail-Trace Message" Super-Frame) from the remote terminal equipment.*
2. *The circuitry within the XRT79L71 will NOT compute and verify this CRC-7 value. To support this feature, then do so externally.*

Receive E3 Trail-Trace Message Byte 2 Register - G.832 (Address = 0x111D)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	A6	A5	A4	A3	A2	A1	A0

Receive E3 Trail-Trace Message Byte 3 Register - G.832 (Address = 0x111E)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	B6	B5	B4	B3	B2	B1	B0

Receive E3 Trail-Trace Message Byte 4 Register - G.832 (Address = 0x111F)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_4							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	C6	C5	C4	C3	C2	C1	C0

Receive E3 Trail-Trace Message Byte 5 Register - G.832 (Address = 0x1120)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_5							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	D6	D5	D4	D3	D2	D1	D0

Receive E3 Trail-Trace Message Byte 6 Register - G.832 (Address = 0x1121)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_6							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	E6	E5	E4	E3	E2	E1	E0

Receive E3 Trail-Trace Message Byte 7 Register - G.832 (Address = 0x1122)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_7							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	F6	F5	F4	F3	F2	F1	F0

Receive E3 Trail-Trace Message Byte 8 Register - G.832 (Address = 0x1123)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_8							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	G6	G5	G4	G3	G2	G1	G0

Receive E3 Trail-Trace Message Byte 9 Register - G.832 (Address = 0x1124)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_9							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	H6	H5	H4	H3	H2	H1	H0

Receive E3 Trail-Trace Message Byte 10 Register - G.832 (Address = 0x1125)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_10							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	I6	I5	I4	I3	I2	I1	I0

Receive E3 Trail-Trace Message Byte 11 Register - G.832 (Address = 0x1126)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_11							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	J6	J5	J4	J3	J2	J1	J0

Receive E3 Trail-Trace Message Byte 12 Register - G.832 (Address = 0x1127)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_12							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	K6	K5	K4	K3	K2	K1	K0

Receive E3 Trail-Trace Message Byte 13 Register - G.832 (Address = 0x1128)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_13							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	L6	L5	L4	L3	L2	L1	L0

Receive E3 Trail-Trace Message Byte 14 Register - G.832 (Address = 0x1129)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_14							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	M6	M5	M4	M3	M2	M1	M0

Receive E3 Trail-Trace Message Byte 15 Register - G.832 (Address = 0x112A)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_15							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	N6	N5	N4	N3	N2	N1	N0

Receive E3 Trail-Trace Message Byte 16 Register - G.832 (Address = 0x112B)

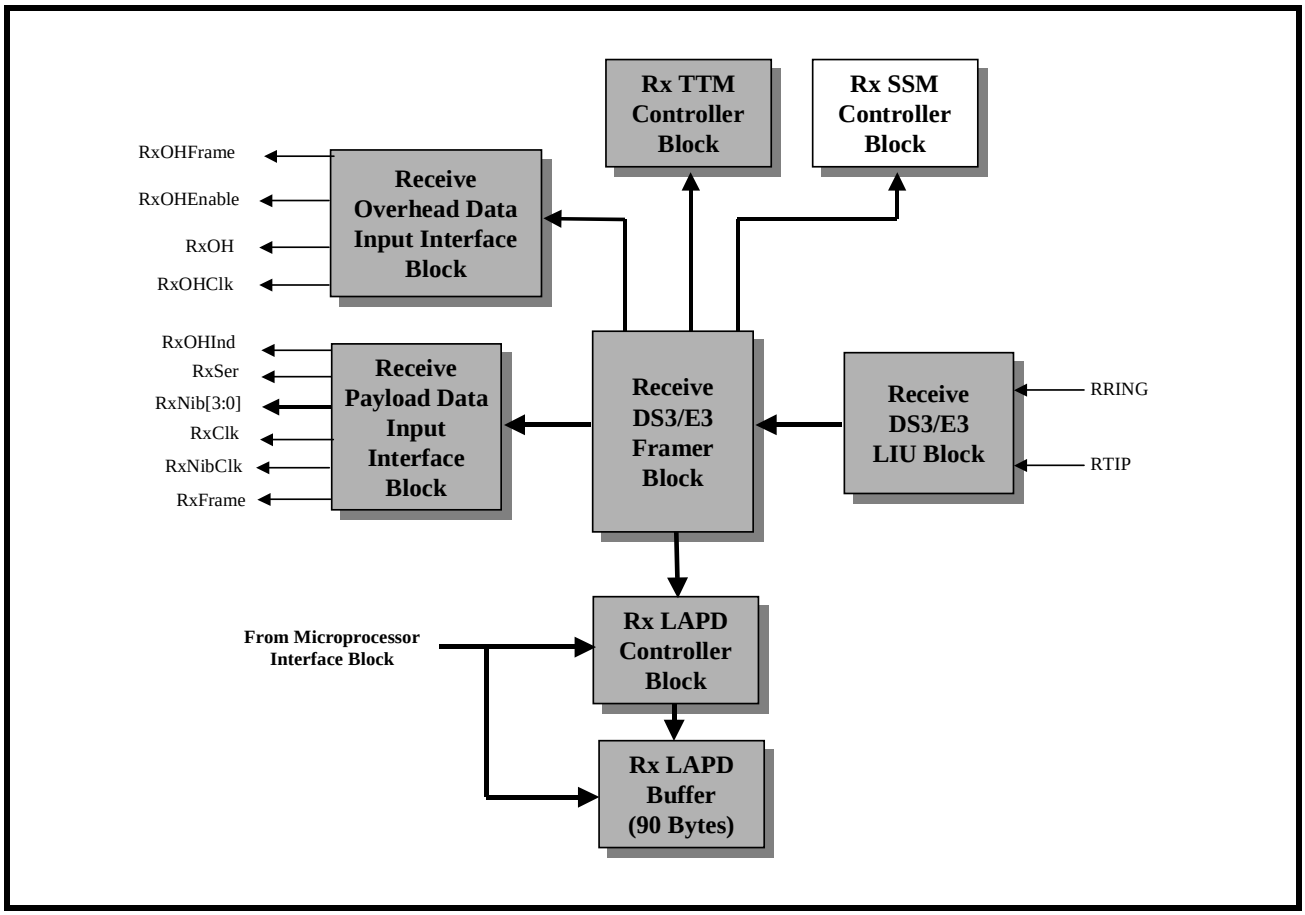
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxTTB_Byte_16							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	P6	P5	P4	P3	P2	P1	P0

6.3.3.3 Receive Trail-Trace Message Controller Block Interrupt

6.3.4 RECEIVE SSM CONTROLLER BLOCK

The Receive SSM Controller block is the fourth functional block within the Receive Direction of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. **Figure 251** presents an illustration of the Receive Direction circuitry, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive SSM Controller block highlighted.

FIGURE 251. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE SSM CONTROLLER BLOCK HIGHLIGHTED)



6.3.4.1 AN INTRODUCTION TO SSM (SYNCHRONIZATION STATUS MESSAGES)

If the XRT79L71 is operating in the E3, ITU-T G.832 Frame Format, then Bits 6, 7 and 8 (the three least significant bit-fields) within the MA byte can be used to transport the Synchronization Status Message (referred to as "SSM" from this point on) from one terminal equipment to another. The bit-format of the MA byte, with these bit-fields shaded is presented below.

FIGURE 252. THE BIT-FORMAT OF THE MA-BYTE WITHIN THE E3, ITU-T G.832 FRAMING FORMAT

BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8
FERF/RDI	FEBE/REI	Payload_Type[2:0]			SSM Multi-Frame Indicator[1:0]		SSM Bit

According to ITU-T G.707, the SSM is a four-bit value that is used to identify the quality-level of the synchronization/timing that the "Transmitting" E3 Terminal Equipment is currently operating at.

Whenever a given "Transmitting" E3 Terminal transmits the SSM to the remote terminal, it does so by repeatedly transmitting this four-bit SSM message, one bit at a time (or one bit per E3 frame period) via Bit 8 (the least significant bit-field) within the MA byte. As this "Transmitting" E3 Terminal repeatedly transmits this byte (to the remote terminal equipment) it will use Bits 6 and 7 (within the MA Byte) to indiciate which SSM Message bit is being transported via Bit 8, within the current MA byte. Depending upon which of the four bits (within the SSM) that is being transported via the MA byte, within a given E3 frame, the "Transmitting" E3

Terminal will set Bits 6 and 7 to the appropriate value, in order to "identify" this SSM bit, as depicted below in Table 67.

TABLE 67: THE RELATIONSHIP BETWEEN THE STATES OF BITS 6 AND 7 (WITHIN THE MA BYTE) AND THE EXACT SSM BIT THAT IS BEING TRANSPORTED VIA BIT 8, WITHIN THE CURRENT MA BYTE

MA BYTE, BIT 6	MA BYTE, BIT 7	THE SSM BIT BEING TRANSPORTED IN BIT 8
0	**0**	SSM Bit 1 (The MSB)
0	**1**	SSM Bit 2
1	**0**	SSM Bit 3
1	**1**	SSM Bit 4 (The LSB)

6.3.4.2 CONFIGURING THE XRT79L71 TO RECEIVE SYNCHRONIZATION STATUS MESSAGES

To enable the Receive SSM Controller block (within the XRT79L71) and configure it to repeatedly receive the SSM from the remote terminal equipment, execute the following two steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing Format.

This can be accomplish this by setting Bit 6 (IsDS3) to "0" and by setting Bit 2 (Frame Format) to "1" as depicted below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Mapped ATM	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	1	1	1

STEP 2 - Enable the Receive SSM Controller Block

This is accomplished by setting Bit 7 (RxSSM Enable) to "1" as depicted below.

Receive E3 SSM Register - G.832 (Address = 0x112C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxSSM Enable	MF[1:0]		Reserved	RxSSM[3:0]			
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
1	0	0	0	0	0	0	0

Once these two steps have been executed, then the Receive SSM Controller block will be enabled, and will now be capable of receiving SSM Messages from the remote terminal equipment.

6.3.4.3 READING OUT THE SSM FROM THE XRT79L71

The user can read out the contents of the most recently received SSM (by the Receive SSM Controller block) by reading out the contents of Bits 3 through 0 (RxSSM[3:0]), within the "Receive E3 SSM Register - G.832" as depicted below.

Receive E3 SSM Register - G.832 (Address = 0x112C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxSSM Enable	MF[1:0]		Reserved	RxSSM[3:0]			
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
1	X	X	0	X	X	X	X

6.3.4.4 RECEIVE SSM CONTROLLER BLOCK INTERRUPTS

Once the Receive SSM Controller block has been enabled, then it can be configured to generate either of the following two interrupts.

- The "Change in SSM Message" Interrupt
- The "SSM Out-of-Sequence" Interrupt

The mechanisms causing these interrupts to occur, the procedures for enabling and servicing these two interrupts is described below.

6.3.4.4.1 THE CHANGE IN SSM MESSAGE INTERRUPT

As mentioned earlier, once the Receive SSM Controller block has been enabled it will begin to receive and extract out the SSM from the incoming E3 data-stream. Further, we indicated that the Receive SSM Controller block will continuously write the contents of each incoming SSM into the "RxSSM[3:0]" bit-fields.

In order to alleviate the user from having to use precious "processor overhead" to continuously read out and poll the contents of the "RxSSM[3:0]" bit-fields, the XRT79L71 includes the "Change in SSM Message" Interrupt.

Enabling the Change in SSM Message Interrupt

The user can enable the "Change in SSM Message" Interrupt, by executing the following steps.

STEP 1 - Enable the DS3/E3 Framer Block for Interrupt Generation, at the Operational Block Level.

This can be accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) within the "Operation Interrupt Enable Register - Byte 1" to "1" as depicted below.

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	1	0	0

STEP 2 - Enable the Receive DS3/E3 Framer Block for Interrupt Generation.

This can be accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable) within the "Framer Block Interrupt Enable Register" to "1", as depicted below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt
R/W	R/O	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	0	0

STEP 3 - Enable the Change in SSM Message Interrupt, at the "Source Level".

This is accomplished by setting Bit 6 (Change in SSM Message Interrupt Enable) within the "Receive E3 Interrupt Enable Register # 1 - G.832" to "1" as depicted below.

Receive E3 Interrupt Enable Register # 1 - G.832 (Address = 0x1112)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Enable	SSM OOS Interrupt Enable	COFA Interrupt Enable	Change in OOF State Interrupt Enable	Change in LOF State Interrupt Enable	Change in LOS State Interrupt Enable	Change in AIS State Interrupt Enable
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	1	0	0	0	0	0	0

Once the user has executed these three steps, then the Receive SSM Controller block will be configured to generate an interrupt request, anytime the Receive SSM Controller block detects a change in the SSM that it receives and extracts out of the incoming E3 data-stream.

Servicing the Change of SSM Interrupt

If the "Change of SSM" Interrupt has been enabled, and if the XRT79L71 does generate the "Change of SSM" Interrupt, then the user must service this particular interrupt by executing the following steps.

STEP 1 - Determine which kind of interrupt has been requested as the "Operational Block" Level

This is accomplished by reading out the contents of both the "Operation Interrupt Status Registers - Byte 1" and "Byte 0" registers. The bit-format for each of these registers is presented below.

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	1	0	0

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 Inter-rupt Status	Transmit UTOPIA/ POS-PHY Interface Block nterrupt 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

NOTE: *Since this particular interrupt is the "Change in SSM" Interrupt, then Bit 2 (DS3/E3 Framer Block Interrupt Status) within the "Operation Interrupt Status Register - Byte 1" Register will be set to "1" as indicated above.*

STEP 2 - Determine whether this particular interrupt occurred within the Transmit or Receive DS3/E3 Framer block.

This can be accomplished by reading out the contents of the "Framer Block Interrupt Status" Register. Since we are discussing the "Change in SSM Message" Interrupt, then Bit 7 (Receive DS3/E3 Framer Block Interrupt Status) should be set to "1" as depicted below.

Framer Block Interrupt Status Register (Address = 0x1105)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Status	Receive PLCP Processor Block nterrupt Status	Unused				Transmit DS3/E3 Framer Block Interrupt Status	One Second Interrupt
R/O	R/O	R/O	R/O	R/O	R/O	R/O	RUR
1	0	0	0	0	0	0	0

STEP 3 - Determine which Receive DS3/E3 Framer block interrupt has been requested.

This is accomplished by reading out the contents of both the "Receive E3 Interrupt Status Register # 1 and # 2" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSGI nterrupt Status	SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF State Interrupt Status	Change in LOF State Interrupt Status	Change in LOS State Interrupt Status	Change in AIS State Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	1	0	0	0	0	0	0

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in Receive Trail-Trace Message Interrupt Status	Reserved	Detection of FEBE/REI Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrup Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTE: *Since we are discussing the "Change in SSM" Interrupt, Bit 6 (Change in SSM Message Interrupt Status), within the "Receive E3 Interrupt Status Register # 1 - G.832" has been set to "1" as depicted above.*

STEP 4 - At a minimum, the user should read out the contents of the new SSM.

This is accomplished by reading out Bits 3 through 0 (RxSSM[3:0]) within the "Receive E3 SSM Register - G.832" as depicted below.

Receive E3 SSM Register - G.832 (Address = 0x112C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxSSM Enable	MF[1:0]		Reserved	RxSSM[3:0]			
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
1	X	X	0	X	X	X	X

NOTE: *The user will likely have some additional (system-related) task that must be performed in response to the "Change in SSM" Interrupt.*

6.3.4.4.2 THE SSM OUT-OF-SEQUENCE INTERRUPT

As mentioned earlier, whenever a "transmitting" E3 terminal transmits the SSM to the remote terminal equipment, it does so by repeatedly transmitting this four-bit SSM message, one bit at a time (or one bit per E3 frame period) via Bit 8 (the least significant bit-field) within the MA byte. As this "Transmitting" E3 Terminal repeatedly transmits this byte (to the remote terminal equipment) it will use Bits 6 and 7 (within the MA byte) to indicate which SSM Message bit is being transported via Bit 8, within the current MA byte. Depending upon which of the four bits (within the SSM) that is being transported via the MA byte, within a given E3 frame, the "Transmitting" E3 Terminal will set Bits 6 anid 7 to the appropriate value, in order to "identify" this SSM bit, as depicted below in **Table 68**.

TABLE 68: THE RELATIONSHIP BETWEEN THE STATES OF BITS 6 AND 7 (WITHIN THE MA BYTE) AND THE EXACT SSM BIT THAT IS BEING TRANSPORTED VIA BIT 8, WITHIN THE CURRENT MA BYTE

MA BYTE, BIT 6	MA BYTE, BIT 7	THE SSM BIT BEING TRANSPORTED IN BIT 8
0	0	SSM Bit 1 (The MSB)
0	1	SSM Bit 2
1	0	SSM Bit 3
1	1	SSM Bit 4 (The LSB)

NOTE: Bits 6 and 7 (within the MA byte) are often referred to as carrying the "SSM Multi-Frame" indicator.

The Receive SSM Controller block contains circuitry that reads in and processes Bits 6 and 7 (within each incoming MA byte) in order properly extract out the SSM message from the incoming E3 data-stream. As the Receive SSM Controller block receives and processes the MA bytes of each incoming E3 frame, it expects Bits 6 and 7 to exhibit the repetitive sequence, as depicted above in Table 68. Anytime the Receive SSM Controller block detects something other than this sequence (via Bits 6 and 7, within each incoming MA byte), then it will generate the "SSM Out-of-Sequence" Interrupt. The purpose of the "SSM Out-of-Sequence" Interrupt is to inform the user that the Receive SSM Controller block is (1) having difficulty locating the "Multi-Frame" pattern within Bits 6 and 7 (of each incoming MA byte), and (2) is therefore having difficulty extracting out an SSM message from the incoming E3 data-stream.

Enabling the SSM Out-of-Sequence Interrupt

To enable the SSM Out-of-Sequence Interrup, execute the following steps.

STEP 1 - Enable the "DS3/E3 Framer Block" for Interrupt Generation, at the "Operational Block" Level.

This can be accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) within the "Operation Interrupt Enable Register - Byte 1" to "1" as depicted below.

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	1	0	0

STEP 2 - Enable the "Receive DS3/E3 Framer Block" for Interrupt Generation.

This can be accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable) within the "Framer Block Interrupt Enable Register" to "1", as depicted below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt
R/W	R/O	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	0	0

STEP 3 - Enable the "SSM Out-of-Sequence" Interrupt, at the "Source Level".

This is accomplished by setting Bit 5 (SSM OOS Interrupt Enable) within the "Receive E3 Interrupt Enable Register # 1 - G.832" to "1" as depicted below.

Receive E3 Interrupt Enable Register # 1 - G.832 (Address = 0x1112)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSG Interrupt Enable	SSM OOS Interrupt Enable	COFA Interrupt Enable	Change in OOF State Interrupt Enable	Change in LOF State Interrupt Enable	Change in LOS State Interrupt Enable	Change in AIS State Interrupt Enable
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	1	0	0	0	0	0

Once the user has executed these three steps, then the Receive SSM Controller block will be configured to generate an interrupt request, anytime the Receive SSM Controller block declares the "SSM Out-of-Sequence" condition within the incoming E3 data-stream.

Servicing the "SSM Out-of-Sequence" Interrupt

If the "SSM Out-of-Sequence" Interrupt has been enabled, and if the XRT79L71 does generate the "SSM Out-of Sequence" Interrupt, then the user must service this particular interrupt by executing the following steps.

STEP 1 - Determine which kind of interrupt has been requested as the "Operational Block" Level

This is accomplished by reading out the contents of both the "Operation Interrupt Status Registers - Byte 1" and "Byte 0" registers. The bit-format for each of these registers is presented below.

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	1	0	0

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 Inter-rupt Status	Transmit UTOPIA/ POS-PHY Interface Block nterrupt 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

NOTE: *Since this particular interrupt is the "SSM Out-of-Sequence" Interrupt, then Bit 2 (DS3/E3 Framer Block Interrupt Status) within the "Operation Interrupt Status Register - Byte 1" Register will be set to "1" as indicated above.*

STEP 2 - Determine whether this particular interrupt occurred within the "Transmit" or "Receive DS3/E3 Framer" block.

This can be accomplished by reading out the contents of the "Framer Block Interrupt Status" Register. Since we are discussing the "SSM Out-of-Sequence" Interrupt, then Bit 7 (Receive DS3/E3 Framer Block Interrupt Status) should be set to "1" as depicted below.

Framer Block Interrupt Status Register (Address = 0x1105)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Status	Receive PLCP Processor Block nterrupt Status	Unused				Transmit DS3/E3 Framer Block Interrupt Status	One Second Interrupt
R/O	R/O	R/O	R/O	R/O	R/O	R/O	RUR
1	0	0	0	0	0	0	0

STEP 3 - Determine which "Receive DS3/E3 Framer block" interrupt has been requested.

This is accomplished by reading out the contents of both the "Receive E3 Interrupt Status Register # 1 and # 2" as depicted below.

Receive E3 Interrupt Status Register # 1 - G.832 (Address = 0x1114)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in SSM MSGI nterrupt Status	SSM OOS Interrupt Status	COFA Interrupt Status	Change in OOF State Interrupt Status	Change in LOF State Interrupt Status	Change in LOS State Interrupt Status	Change in AIS State Interrupt Status
R/O	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	1	0	0	0	0	0

Receive E3 Interrupt Status Register # 2 - G.832 (Address = 0x1115)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	Change in Receive Trail-Trace Message Interrupt Status	Reserved	Detection of FEBE/REI Event Interrupt Status	Change in FERF/RDI Defect Condition Interrupt Status	Detection of BIP-8 Error Interrupt Status	Detection of Framing Byte Error Interrupt Status	RxPLD Mismatch Interrup Status
R/O	RUR	R/O	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

NOTES:

1. *Since we are discussing the "SSM Out-of-Sequence" Interrupt, Bit 5 (SSM OOS Interrupt Status), within the "Receive E3 Interrupt Status Register # 1 - G.832" has been set to "1" as depicted above.*

2. *As mentioned earlier, the Receive SSM Controller block will generate this particular interrupt whenever it cannot identify the "SSM Multi-Frame" pattern (within Bits 6 and 7 of the MA byte) within the incoming E3 frame. If the Receive SSM Controller block is generating this interrupt, then either of the following phenomenon is occurring*

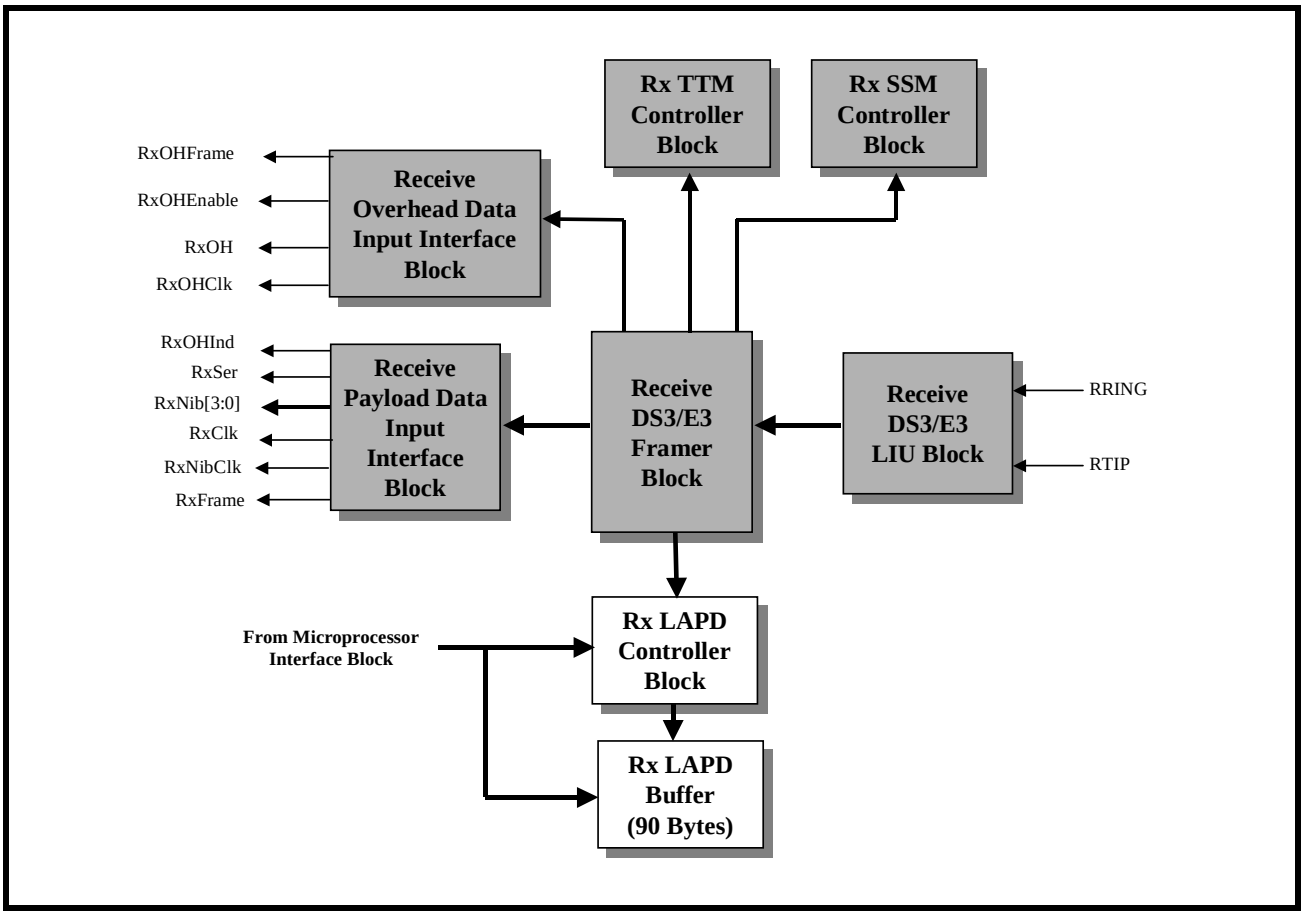
a. The user has (in programmable logic) incorrectly implemented the generation of the "SSM Multi-Frame" pattern within the "Transmitting" Terminal (which the Receive SSM Controller block circuitry does not recognize), or

b. The Receive Section of the XRT79L71 is currently receiving an erred E3 signal. In this case, the user is advised to check and verify that the Receive E3 Framer block is NOT declaring the LOS or LOF/OOF defect condition. Further, the user should verify that the Receive E3 Framer block is not flagging EM Byte or Framing Alignment byte errors as well.

6.3.5 RECEIVE LAPD CONTROLLER BLOCK

The Receive LAPD Controller block is the fifth functional block within the Receive Direction of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. **Figure 253** presents an illustration of the Receive Direction circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive LAPD Controller block highlighted.

FIGURE 253. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE RECEIVE LAPD CONTROLLER BLOCK HIGHLIGHTED)



The Receive LAPD Controller block consists of the following sections.

- The Receive LAPD Message Buffer
- The Receive LAPD Controller

Each of these sections is described in some detail below.

The Receive LAPD Message Buffer

The purpose of the Receive LAPD Message Buffer is to store the contents of LAPD/PMDL Messages that have been received by the Receive LAPD Controller block. The Receive LAPD Message Buffer will serve as a temporary storage location of these incoming LAPD/PMDL Messages until they can be read out by the Microprocessor. the Receive LAPD Message Buffer is actually a 90 byte FIFO that is located at Address Location 0x11C0 within the XRT79L71 address space.

The Receive LAPD Controller

The Receive LAPD Controller permits the user to receive path maintenance data link (PMDL) message from the remote terminal equipment via the "inbound" DS3 frames. In this case, the PMDL Message is extracted out of either the "NR" or the "GC" bytes (depending upon user configuration) within each incoming E3 frame. The on-chip Receive LAPD Controller is capable of receiving both standard and non-standard PMDL Messages of any length up to 82 bytes. The XRT79L71 allocates a block of 90 bytes of on-chip RAM (e.g., the Receive LAPD Message buffer), to store the contents of newly received PMDL Messages. The message format complies with ITU-T Q.921 (LAPD) protocol with different addresses and is presented below in Figure 254.

FIGURE 254. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		
76 or 82 or Any-size Bytes of Information (Payload)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

For standard Bellcore GR-499-CORE applications:

Flag Sequence = 0x7E

SAPI + CR + EA = 0x3C or 0x3E

TEI + EA = 0x01

Control = 0x03

The following text 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. Additionally, whenever the Receive LAPD Controller is not currently receiving a LAPD Message, it will instead be receiving a continuous stream of Flag Sequence bytes via either the "NR" or the "GC" Byte (depending upon which byte was configured to function as the LAPD/PMDL Channel) within each "inbound" E3 frame.

SAPI - Service Access Point Identifier

Traditionally, for N-ISDN applications, the SAPI field typically indicates the type of data or service being supported by the LAPD Message. However, for standard Bellcore GR-499-CORE applications, this parameter has no meaning and is assigned the value "001111b" or 1510 per Bellcore GR-499-CORE

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 terminals. However, since DS3/E3 data is transmitted in a point-to-point manner, the TEI value is unimportant in this application.

Control

The Control byte-field identifies the type of frame being transmitted. There are three general types of frame formats: Information, Supervisory, and Unnumbered. For standard Bellcore GR-499-CORE applications the user must use the Control byte the value 0x03. Hence, the XRT79L71 will be transmitting and receiving Unnumbered LAPD Message frames.

Information Payload

The Information Payload is the 76 bytes, 82 bytes or any number of bytes of data (e.g., the PMDL Message) that the user has written into the on-chip Receive LAPD Message buffer which is located at Address 0x>>>>.

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, x16 + x12 + x5 + 1. Afterwards, this FCS value is inserted into the two-octet FCS value position, within the LAPD Message frame. The Receive LAPD Controller block will use the FCS bytes in order to verify that it has received a given LAPD Message in an un-erred manner. Please see **SEE"TRANSMIT LAPD CONTROLLER BLOCK" ON PAGE 471.** on how the Transmit LAPD Controller block computes and inserts the FCS values into its outbound LAPD Message frames.

Operation of the Receive LAPD Controller

As mentioned earlier, the Receive LAPD Controller permits the user to receive either of the following basic types of LAPD Messages.

- Standard (e.g., 76 or 82 byte size) LAPD Messages
- Variable Length (e.g., up to 82 byte size) LAPD Messages

The procedure for receiving these types of LAPD Messages is presented below.

6.3.5.1 Receiving Standard-type (76 or 82 byte size) LAPD Messages

The user can (1) configure the Receive LAPD Controller block to extract out the contents of the incoming PMDL Messages from the incoming DS3 data-stream, and (2) to properly read out the contents of a newly received message which is being stored in the Receive LAPD Message buffer, by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing format.

This is accomplished by reading out the contents of the Framer Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	Timing Reference Select [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

STEP 2 - Select either the NR or the GC byte as the LAPD Channel.

The user can accomplish this by writing the appropriate value into Bit 3 (Receive LAPD from NR Byte), within the Receive E3 LAPD Control Register, as depicted below.

Receive E3 LAPD Control Register - G.832 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused			Receive LAPD from NR Byte	Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	X	0	0	0

Setting this bit-field to "1" will configure the Receive LAPD Controller block to extract out the incoming LAPD/PMDL Message from the NR bytes within the incoming E3 data-stream. In this setting, the Receive LAPD Controller block will accept LAPD/PMDL Message bytes from the NR byte within each incoming E3 frame, and it will re-assemble these bytes into a newly received LAPD/PMDL Message. Conversely, setting this bit-field to "0" will configure the Receive LAPD Controller block to extract out the incoming LAPD/PMDL Message from the GC bytes within the incoming E3 data-stream.

STEP 3 - Enable the Receive LAPD Controller

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive E3 LAPD Control Register to "1" as depicted below.

Receive E3 LAPD Control Register - G.832 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused			Receive LAPD from NR Byte	Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	X	1	0	0

NOTES:

1. *For normal operation, it is imperative that the user also make sure that Bit 7 (Receive LAPD Any) within this register is set to "0".*
2. *Once the user executes the above-mentioned step, then the Receive LAPD Controller will begin to extract out the contents of any incoming LAPD/PMDL Message that is being transported via either the NR or GC byte, depending upon user configuration, during STEP 2 within the incoming E3 data-stream. In most cases, the Receive LAPD Controller block will simply begin to receive the Flag Sequence octet which is originating from the remote terminal.*

STEP 4 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming E3 data-stream, then it will assert Bit 0 (Flag Present) within the Receive E3 LAPD Status Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 5 - Enable the Receive LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the Receive LAPD Interrupt is actually a three-step process.

STEP 5a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 5b - Enable the Receive DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the Block Interrupt Enable Register, to "1", as illustrated below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor-Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the Receive DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 5c - Enable the Receive LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the Receive DS3 LAPD Control Register, as depicted below.

Receive E3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 6 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 7 - Service the Receive LAPD Interrupt

Please see**SEE"RECEIVE LAPD CONTROLLER BLOCK INTERRUPT" ON PAGE 562.** of how to service the Receive LAPD Interrupt.

STEP 8 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive E3 LAPD Status Register as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an un-erred manner (e.g., there are no FCS errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an erred manner.

NOTE: *The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the RxFCS Error bit-field, prior to processing a given message.*

STEP 9 - Identify the Type and Size of Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of Bits 4 and 5 (RxLAPDType[1:0]) within the Receive E3 LAPD Status Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCR Type	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	0	1	0

Table 69 presents the relationship between the contents within Bits 4 and 5 (RxLAPDType[1:0]) and the corresponding LAPD Message that is residing within the Receive LAPD Message Buffer.

TABLE 69: THE RELATIONSHIP BETWEEN THE CONTENTS WITHIN BITS 4 AND 5 (RXLAPDTYPE[1:0]) AND THE TYPE OF LAPD/PMDL MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER

RXLAPDTYPE[1:0]	TYPE OF LAPD/PMDL MESSAGE RESIDING IN THE RECEIVE LAPD MESSAGE BUFFER	SIZE OF MESSAGE RESIDING WITHIN THE RECEIVE LAPD MESSAGE BUFFER (INFORMATION PAYLOAD/TOTAL MESSAGE SIZE)
00	CL Path Identification Message	76 Bytes/82 Bytes
01	Idle Signal Identification	76 Bytes/82 Bytes
10	Test Signal Identification	76 Bytes/82 Bytes
11	ITU-T Path Identification	82 Bytes/88 Bytes

Table 69 indicates that if the value within the RxLAPDType[1:0] bit-fields are set to [1, 1], then the size of total message residing within the Receive LAPD Message Buffer is 88 bytes, and that the size of the Information Payload within these 88 bytes of data is 82 bytes. Likewise, this table also indicates that if the value within the RxLAPDType[1:0] bit-fields are set to some value other than [1, 1], then the size of the total message residing within the Receive LAPD Message Buffer is 82 bytes, and that the size of the Information Payload within these 82 bytes of data is 76 bytes.

The Relationship between the Total LAPD/PMDL Message Size and Information Payload Size

The Relationship between the size of the total LAPD/PMDL Message, and the size of the Information Payload, is best explained by presenting the byte-format of the LAPD Message below in Figure 255.

FIGURE 255. LAPD MESSAGE FRAME FORMAT

FLAG SEQUENCE (8 BITS)		
SAPI (6-bits)	C/R	EA
TEI (7 bits)		EA
Control (8-bits)		
76 OR 82 OR ANY-SIZE BYTES OF INFORMATION (PAYLOAD)		
FCS - MSB		
FCS - LSB		
Flag Sequence (8-bits)		

Figure 255 indicates that the complete LAPD/PMDL Message will consist of an information payload of either 76 or 82 bytes, along with a total of six overhead bytes consisting of the Flag Sequence, SAPI, TEI, Control and the two FCS bytes.

NOTE: *When receiving and processing the Standard 76 or 82-byte type of LAPD Messages, then the data residing within the Receive LAPD Message will be exactly of the byte-format, as presented above in* Figure 255*.*

Why determining the Size of the Incoming LAPD Message is important

If the Microprocessor has learned based upon reading out the values of the RxLAPDType[1:0] bit-fields that the size of the total LAPD/PMDL Message is 88 bytes, then the Microprocessor knows when it comes time to read out the contents of the Message, residing within the Receive LAPD Message Buffer that it can read out and process the contents of 88-bytes out of 90 bytes within this Buffer. Conversely, if the Microprocessor has learned that the size of the total LAPD/PMDL Message is only 82 bytes, then the Microprocessor must know that it can only read out and process the first 82 bytes of data within the Receive LAPD Message Buffer. The last eight bytes within the Buffer are simply junk bytes and have no value.

STEP 10 - Read out the contents of the Receive LAPD Message Buffer

The instructions that follow were written with the assumption that the user only wishes to extract out the Information Payload bytes, from the complete LAPD Message that is residing within the Receive LAPD Message Buffer. Whenever the user wishes to read out the contents of newly received PMDL/LAPD Messages from the Receive LAPD Message buffer, then the user MUST employ the Indirect Addressing scheme that will be presented below.

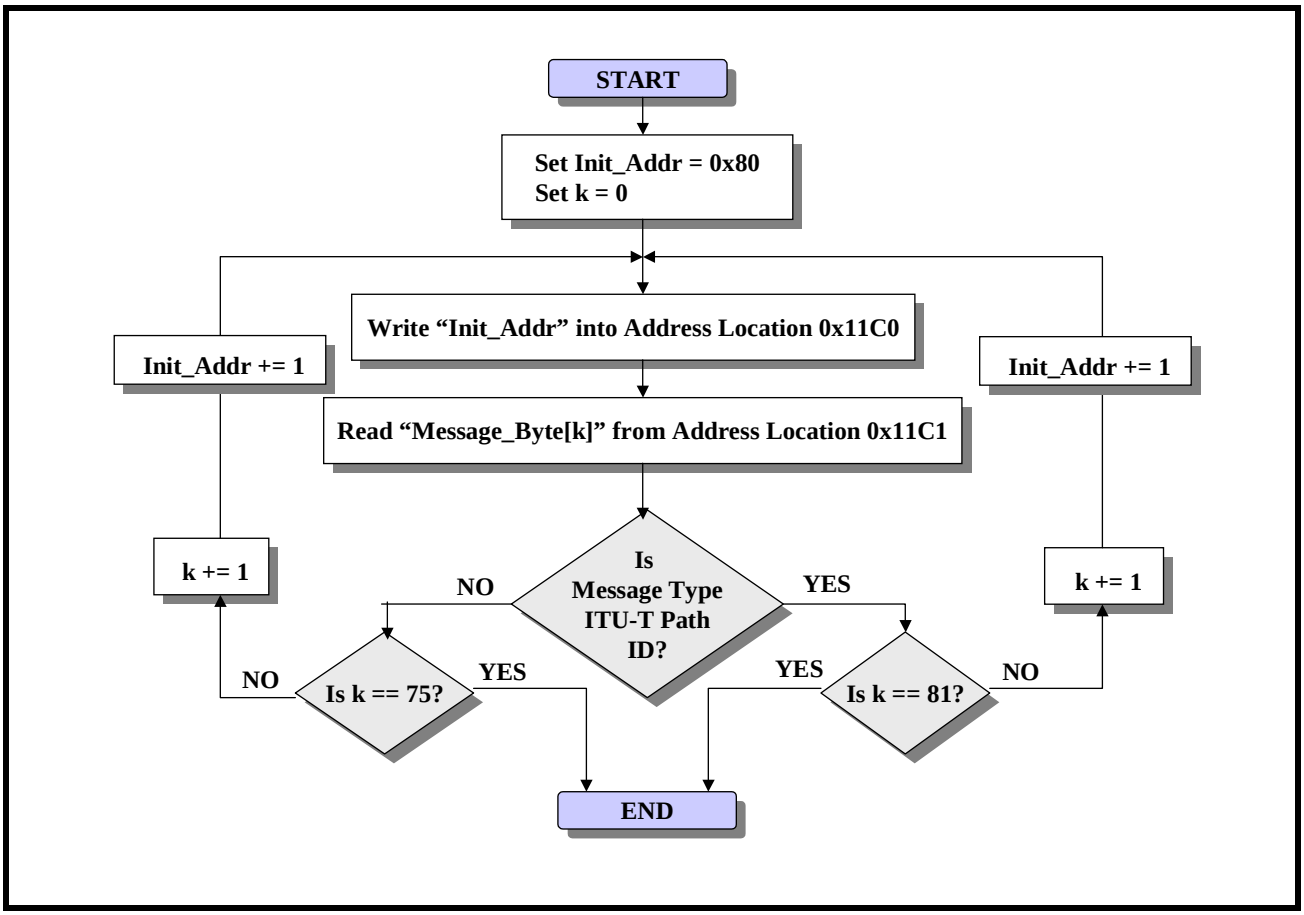
In order to begin the process of reading out the contents of the incoming PMDL Message, then the user must be aware of the following important Address Locations, within the XRT79L71 Address space.

1. The LAPD Message Buffer - Indirect Address Register - Address = 0x11C0
2. The LAPD Message Buffer - Indirect Data Register - Address = 0x11C1

By performing READ or WRITE operations to these two registers, the user can actually obtain READ/WRITE access to both the Transmit LAPD Message Buffer and the Receive LAPD Message Buffer. This section will describe the approach that one should use to access the Receive LAPD Message Buffer. The approach that one should use to access the Transmit LAPD Message Buffer is presented in **SEE"RECEIVING NON-STANDARD VARIABLE LENGTH (E.G., UP TO 82 BYTES) LAPD MESSAGES)" ON PAGE 557.**.

The exact approach that one should use, when reading out the contents of the newly received PMDL Message from the Receive LAPD Message buffer is presented in the flow-chart below.

FIGURE 256. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE FOR READING OUT THE CONTENTS OF A NEWLY RECEIVED LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER



NOTE: About Figure 256:

The answer to the very Is the Message Type ITU-T Path ID Decision Diamond of the flow-chart was obtained during STEP 9 (see above).

STEP 11 - Remove Header and FCS Bytes from this newly received (and read out) PMDL/LAPD Message

As mentioned above, the byte of the data, residing within the Receive LAPD Message buffer, will be as is presented within Figure 255. Therefore, if the user is only interested in processing the Information Payload portion of this LAPD/PMDL Message, then the user must remove these additional bytes from this block of data, before further processing.

6.3.5.2 Receiving Non-Standard Variable Length (e.g., up to 82 bytes) LAPD Messages)

The user can (1) enable the Receive LAPD Controller, and (2) read out a non-Standard incoming PMDL/LAPD message by executing the following steps.

STEP 1 - Make sure that the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Framing Format.

This is accomplished by reading out the contents of the Frame Operating Mode Register (Address = 0x1100) and verifying that Bit 6 (DS3/E3*) is set to "0" and that Bit 2 (Frame Format) is set to "1" as illustrated below.

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Local Loop-back	DS3/E3*	Internal LOS Enable	RESET	Direct Mapped ATM	Frame For-mat	Timing Reference Select [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

STEP 2 - Select either the NR or the GC byte as the LAPD Channel.

The user can accomplish this by writing the appropriate value into Bit 3 (Receive LAPD from NR Byte), within the Receive E3 LAPD Control Register, as depicted below.

Receive E3 LAPD Control Register - G.832 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused			Receive LAPD from NR Byte	Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	X	0	0	0

Setting this bit-field to "1" will configure the Receive LAPD Controller block to extract out the incoming LAPD/PMDL Message from the NR bytes within the incoming E3 data-stream. In this setting, the Receive LAPD Controller block will accept LAPD/PMDL Message bytes from the NR byte within each incoming E3 frame, and it will re-assemble these bytes into a newly received LAPD/PMDL Message. Conversely, setting this bit-field to "0" will configure the Receive LAPD Controller block to extract out the incoming LAPD/PMDL Message from the GC bytes within the incoming E3 data-stream.

STEP 3 - Enable the Receive LAPD Controller

This is accomplished by setting Bit 2 (Receive LAPD Enable) within the Receive E3 LAPD Control Register to "1" as depicted below.

Receive E3 LAPD Control Register - G.832 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused			Receive LAPD from NR Byte	Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	X	1	0	0

STEP 4 - Configure the Receive LAPD Controller to receive a non-standard size LAPD Message

This is accomplished by setting Bit 7 (RxLAPD Any) within the Receive E3 LAPD Control Register, to "1", as depicted below.

Receive E3 LAPD Control Register - G.832 (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused			Receive LAPD from NR Byte	Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/W	R/W	R/W	RUR
0	0	0	0	X	1	0	0

STEP 5 - Check and verify that the Receive LAPD Controller is receiving the Flag Sequence Octets

If the Receive LAPD Controller block is currently receiving the Flag Sequence octets within the incoming E3 data-stream, then it will assert Bit 0 (Flag Present) within the Receive E3 LAPD Status Register, as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCRType	RxFCSError	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	0	0	0	0	0	1

STEP 6 - Enable the Receive LAPD Interrupt (Optional).

This step is optional. However, if this step is executed, the XRT79L71 will generate an interrupt to the Microprocessor anytime the Receive LAPD Controller block has completed its reception of a new PMDL Message. The purpose of this interrupt is to notify the Microprocessor that the Receive LAPD Message buffer contains a newly received LAPD/PMDL Message that needs to be read.

The procedure for enabling the Receive LAPD Interrupt is actually a three-step process.

STEP 6a - Enable the DS3/E3 Framer block interrupts - At the Operational Block Level.

This step is accomplished by setting Bit 2 (DS3/E3 Framer Block Interrupt Enable) to "1" as illustrated below.

Operation Block 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	1	0	0

This step enables the DS3/E3 Framer block for interrupt generation at the Operational Block Level.

STEP 6b - Enable the Receive DS3/E3 Framer block Interrupts - At the Block Level.

This step is accomplished by setting Bit 7 (Receive DS3/E3 Framer Block Interrupt Enable), within the Framer Block Interrupt Enable Register, to "1", as illustrated below.

Framer Block Interrupt Enable Register (Address = 0x1104)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Receive DS3/E3 Framer Block Interrupt Enable	Receive PLCP Processor Block Interrupt Enable	Unused				Transmit DS3/E3 Framer Block Interrupt Enable	One Second Interrupt Enable
R/W	R/W	R/O	R/O	R/O	R/O	R/W	R/W
1	0	0	0	0	0	X	X

This step enables the Receive DS3/E3 Framer block for interrupt generation, at the Block Level.

STEP 6c - Enable the Receive LAPD Interrupt at the Source Level.

This step is accomplished by setting Bit 1 (Receive LAPD Interrupt Enable), within the Receive E3 LAPD Control Register, as depicted below.

Receive E3 LAPD Control Register (Address = 0x1118)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RxLAPD Any	Unused				Receive LAPD Enable	Receive LAPD Interrupt Enable	Receive LAPD Interrupt Status
R/W	R/O	R/O	R/O	R/O	R/W	R/W	RUR
0	0	0	0	0	1	1	0

STEP 7 - Wait for the occurrence of the Receive LAPD Interrupt

STEP 8 - Service the Receive LAPD Interrupt

Please see **SEE"RECEIVE LAPD CONTROLLER BLOCK INTERRUPT" ON PAGE 562.** of how to service the Receive LAPD Interrupt.

STEP 9 - Check and verify that there are no FCS (Frame Check Sequence) Errors within the LAPD/ PMDL Message that is residing within the Receive LAPD Message Buffer.

This can be accomplished by reading out and testing the state of Bit 2 (RxFCS Error) within the Receive E3 LAPD Status Register as depicted below.

Receive E3 LAPD Status Register (Address = 0x1119)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	RxABORT	RxLAPDType[1:0]		RxCRType	RxFCS Error	End of Message	Flag Present
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	0	X	X	0	X	0	0

If this bit-field is set to "0", then the Receive LAPD Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an un-erred manner (e.g., there are no FCS errors within this particular LAPD message). Conversely, if this bit-field is set to "1", then the Receive LAPD

Controller block has received this particular LAPD Message that is residing within the Receive LAPD Message Buffer in an erred manner.

NOTE: *The Receive LAPD Controller block will not generate any interrupt in response to it detecting any FCS Errors within an incoming LAPD Message. The user is expected to validate each incoming LAPD Message, by testing the state of the RxFCS Error bit-field, prior to processing a given message.*

STEP 10 - Determine the Size of the Message that the Receive LAPD Controller has just received.

This can be accomplished by reading out the contents of the Receive LAPD Byte Count Register, as depicted below.

Receive LAPD Byte Count Register (Address = 0x1184)

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

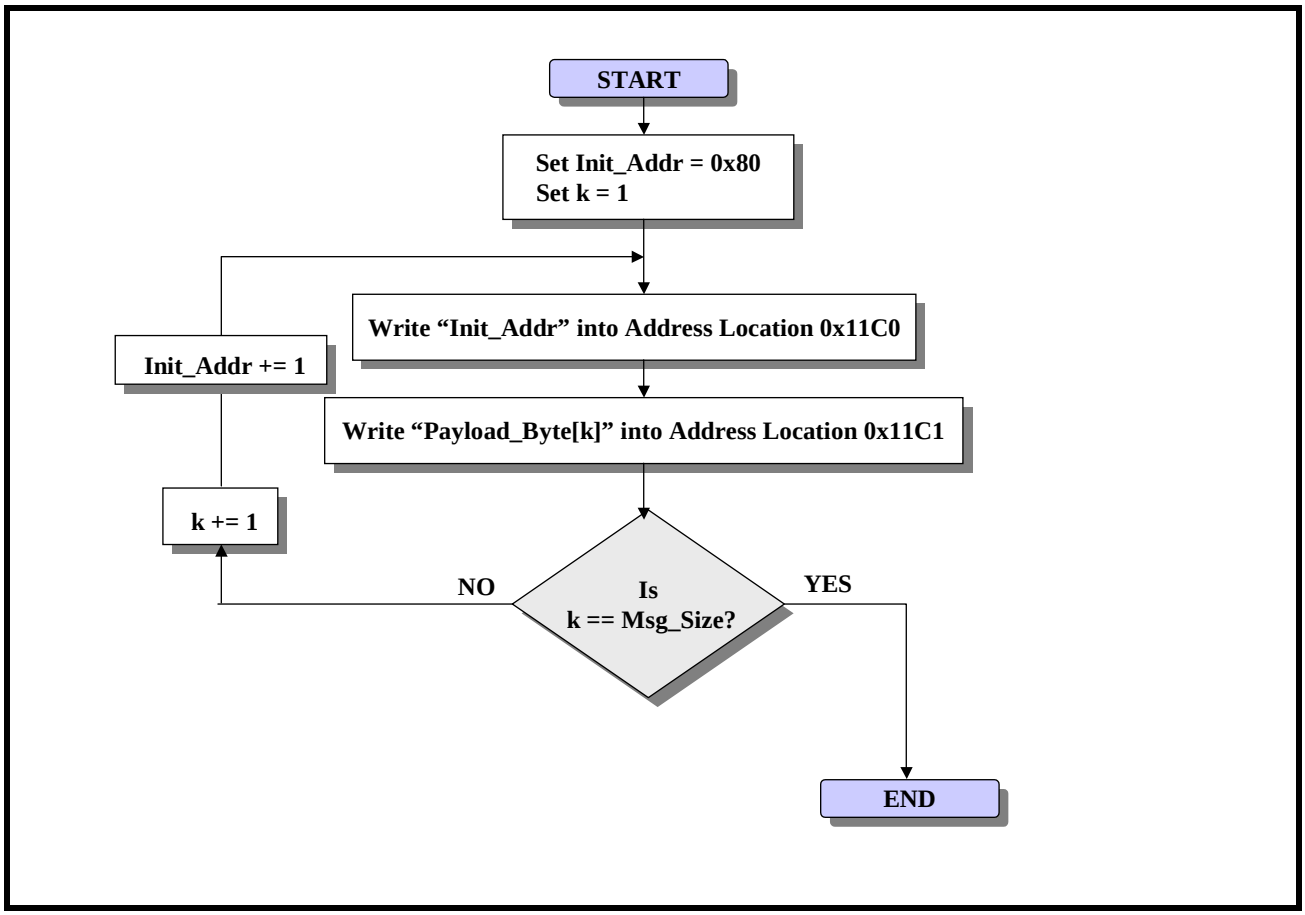
NOTES:

1. *The Receive LAPD Byte Count Register will have the size of the newly received LAPD/PMDL Message, in terms of bytes.*
2. *This register is only active if the Receive LAPD Controller is receiving a non-standard LAPD/PMDL Message.*

STEP 11 - Read out the contents of the Receive LAPD Message Buffer

This is accomplished by executing the procedure that is defined and presented within the following flow-chart.

FIGURE 257. FLOW-CHART DEPICTING AN APPROACH THAT ONE CAN USE TO READING OUT THE CONTENTS OF THE NEWLY RECEIVE LAPD/PMDL MESSAGE FROM THE RECEIVE LAPD MESSAGE BUFFER



NOTE: *About* Figure 257*:*

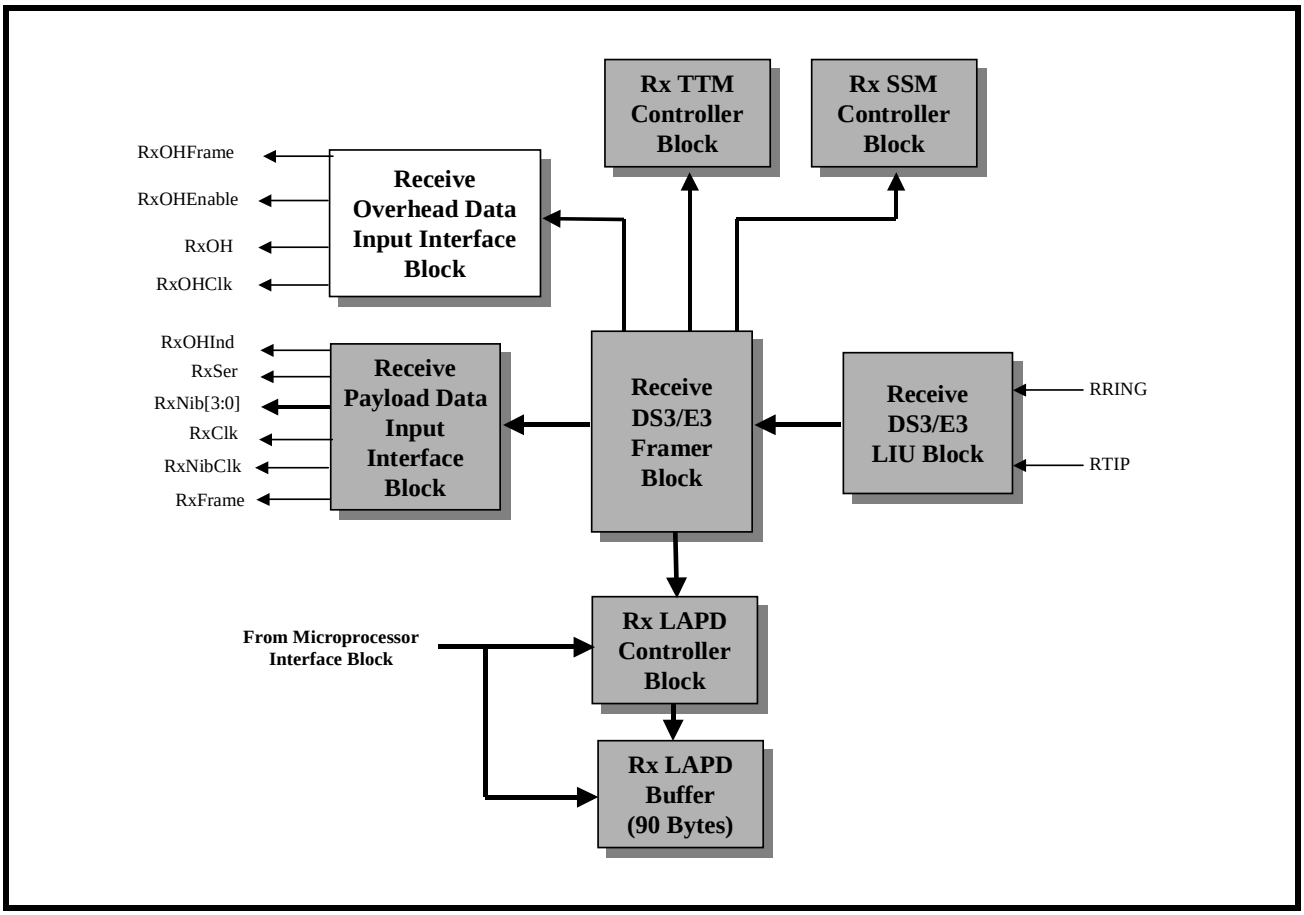
The answer to the Is k = Msg_Size? Decision Diamond in the flow-chart was obtained during STEP 10 (see above).

6.3.5.3 Receive LAPD Controller Block Interrupt

6.3.6 RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

The Receive Overhead Data Output Interface block is the sixth functional block (within the Receive Direction) of the XRT79L71 that we will discuss for E3, ITU-T G.832 Clear-Channel Framer Applications. Figure 258 presents an illustration of the "Receive Direction" circuitry, whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the Receive Overhead Data Output Interface block highlighted.

FIGURE 258. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK HIGHLIGHTED)



Some Background Information

In order to fully understand the role of the Receive Overhead Data Output Interface, some background information needs to be discussed first.

As mentioned in SEE"DESCRIPTION OF THE E3, ITU-T G.832 FRAME STRUCTURE AND OVERHEAD BITS" ON PAGE 421., the E3, ITU-T G.832 frame consists of 537 bytes. Of these bytes, 530 bytes are payload bytes and the remaining 7 bytes are overhead bytes. The XRT79L71 has been designed to handle and process both the payload type and overhead type of bits/bytes for each E3 frame. Within the XRT79L71, the Receive Payload Data Output Interface Block (which is discussed in considerable detail in SEE"RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK" ON PAGE 571.) has been designed to output the payload data (that has been extracted from the incoming E3 data-stream) to the "System-Side" Terminal Equipment. Likewise, the Receive Overhead Data Output Interface block has been designed to handle and processo the overhead bits.

The Receive Overhead Data Output Interface block unconditionally outputs the contents of all overhead bits within the incoming E3 data-stream. The XRT79L71 does not offer the user a means to shut off this transmission of overhead data. However, the Receive Overhead Data Output Interface block does not provide the user with the appropriate output signals for an external Data-Link Layer (or System-Side Terminal Equipment) to sample and process these overhead bits.

In order to accomplish this, the Receive Overhead Data Output Interface block has numerous output pins. Table 70 presents a list and a brief definition of each of these pins.

TABLE 70: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOH	C7	O	**Receive Overhead Data Output Interface block - Data Output pin:** How to sample this output pin depends upon whether "Method 1" (see SEE"METHOD 1 - OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING THE "RXOHCLK" METHOD" ON PAGE 565., below) or "Method 2" (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/RXOHENABLE" METHOD" ON PAGE 568., below) is used for extracting data from the "Receive Overhead Data Output Interface block. **If Method 1 is used:** The XRT79L71 outputs the overhead bits, within the incoming E3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will output a given bit, upon the falling edge of RxOHClk. Hence, the "System-Side Terminal Equipment" should be designed (or configured) to sample the data, at this pin, upon the rising edge of RxOHClk. **If Method 2 is used:** The XRT79L71 outputs the overhead bits, within the incoming E3 data-stream, via this output pin. The Receive Overhead Data Output Interface block will assert the "RxOHEnable" output pin (for one "RxClk" period) whenever the data, residing on the "RxOH" output pin has become stable and is safe for "sampling". In this case, the user should design (or configure) the System-Side Terminal Equipment to sample and latch the "RxOH" data upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high". The XRT79L71 will always output the E3 overhead bits via this output pin. There are no external input pins or register bits settings available that will disable this output pin.
RxOHClk	A7	O	**Receive Overhead Data Output Interface block - Clock Output pin:** This output pin is only used if "Method 1" is employed to extract overhead data from the "Receive Overhead Data Output Interface" port (see-SEE"METHOD 1 - OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING THE "RXOHCLK" METHOD" ON PAGE 565., below). The XRT79L71 will output the overhead bits (within the incoming E3 data-stream), via the "RxOH" output pin, upon the falling edge of this particular clock signal. As a consequence, the "System-Side Terminal Equipment" should be designed (or configured) to sample the data, at this pin, upon the rising edge of RxOHClk. ***NOTE:*** *This output clock signal is always active.*

TABLE 70: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOHFrame	A8	O	**Receive Overhead Data Output Interface block - Start of Frame Indicator Output pin:** The way that this output pin is sampled depends upon whether "Method 1" (see SEE"METHOD 1 - OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING THE "RXOHCLK" METHOD" ON PAGE 565., below) or "Method 2" (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/RXOHENABLE" METHOD" ON PAGE 568., below) is used for extracting data from the "Receive Overhead Data Output Interface block. **If Method 1 is used:** The XRT79L71 will pulse this output pin "high" (for one period of RxOHClk) coincident to whenever the Receive Overhead Data Output Interface block outputs the very first overhead bit of the most recently received E3 frame. This output pin will be "low" at all other times. The Receive Overhead Data Output Interface block will update this output pin, upon the falling edge of "RxOHClk". Hence, the System-Side Terminal Equipment should be designed (or configured) to sample the data, at this pin, upon the rising edge of "RxOHClk". **If Method 2 is used:** The Receive Overhead Data Output Interface block will assert the "RxOHEnable" output pin (for one "RxClk" period) whenever the data, residing on the "RxOH" output pin has become stable and is safe for "sampling". In this case, the user should design (or configure) the System-Side Terminal Equipment to sample and latch both the "RxOH" and the "RxOHFrame" output pins upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high".
RxOHEnable	B7	O	**Receive Overhead Data - Enable Output Pin:** This particular output pin is only used if Method 2 is employed (see SEE"OPERATING THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK USING METHOD 2 - THE "RXCLK/RXOHENABLE" METHOD" ON PAGE 568., below). The Receive Overhead Data Output Interface will assert this signal (e.g., pulse it "high") for one "RxClk" period, coincident to whenever it is safe for the System-Side Terminal Equipment to sample the overhead bit that is being output via the "RxOH" output pin. This output pin will remain "low" at all other times. If "Method 2" is to be used, then design (or configure) the System-Side Terminal Equipment such that it will sample and latch the "RxOH" output (from the XRT79L71), upon the falling edge of "RxClk" coincident to whenever the "RxOHEnable" output pin is sampled "high".

The Two Methods of Extracting out Overhead Data from the Receive Overhead Data Output Interface Block

There are two methods that can be used to extract the overhead data from the Receive Overhead Data Output Interface block. One "Method" is referred to as "Method 1" or the "RxOHClk" Method, and the other "method" is referred to as "Method 2" or the "RxClk/RxOHEnable" Method. Each of these methods is described in considerable detail below.

6.3.6.1 Method 1 - Operating the Receive Overhead Data Output Interface block using The "RxOHClk" Method

This particular method is referred to as the "RxOHClk" method for the following reasons.

a. The System-Side Terminal Equipment will use the RxOHClk clock output signal (from the Receive Overhead Data Output Interface block) to sample and latch the "overhead" data via the "RxOH" output pin.

b. The "Receive Overhead Data Output Interface" block will update the data (via the RxOH output pin) upon the falling edge of the "RxOHClk" clock output signal.

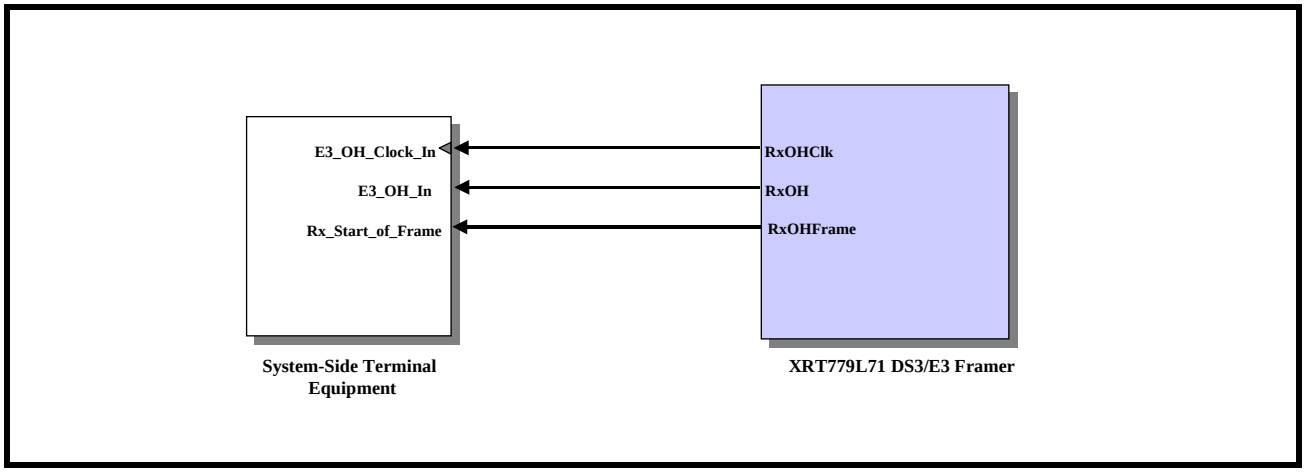
If

"Method 1" is used then the "System-Side Terminal Equipment" will need to interface to the following "Receive Overhead Data Output Interface" pins.

- RxOH
- RxOHClk
- RxOHFrame

If "Method 1" is used then the users system must be designed such that the "System-Side" Terminal Equipment will be interfaced to the "Receive Overhead Data Output Interface" block in the manner presented below in **Figure 259**.

FIGURE 259. ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK WHEN USING "METHOD 1"



Method 1 Operation of the Receive Overhead Data Output Interface Block

IWhen operating the Receive Overhead Data Output Interface block, the "System-Side" Terminal Equipment must designed/configured to continuously execute the following tasks.

TASK # 1: The "System-Side" Terminal Equipment must sample the state of the "RxOHFrame" output pin (from the XRT79L71) upon the rising edge of the "RxOHClk" clock signal (which is also output from the XRT79L71). Whenever the "System-Side Terminal Equipment" samples the "RxOHFrame" output pin "high", then it will "know" that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit (within the most recently received E3 frame) via the "RxOH" output pin.

TASK # 2: As the "System-Side Terminal Equipment' samples the "RxOHFrame" output signal, it must also keep track of the number of rising edges (within the RxOHClk signal) that have occurred since the last time "RxOHFrame" was sampled "high". By doing this, the "System-Side" Terminal Equipment will be able to keep track of which overhead bit is being output via the "RxOH" output pin, at any given "RxOHClk" period. When the System-Side Terminal Equipment "knows" which overhead bit is being output (via the RxOH output pin) during a particular RxOHClk clock period, then it can "make sense" of these overhead bits and process these overhead bits as appropriate.

Table 71 relates the number of rising clock edges, within the "RxOHClk" output signal, since the "RxOHFrame" output signal was sampled "high" to the DS3 Overhead Bit being output via the RxOH output pin (of the Receive Overhead Data Output Interface block), for Method 1.

TABLE 71: RELATIONSHIP BETWEEN 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 PROCESSED BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN RXOHCLK SINCE RXOHFRAME BEING SAMPLED "HIGH"	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
0 (RxOHClk, Clock Edge is coincident with the "RxOHFrame" signal being sampled "high")	FA1 Byte, Bit 1 (MSB)
1	FA1 Byte, Bit 2
2	FA1 Byte, Bit 3
3	FA1 Byte, Bit 4
4	FA1 Byte, Bit 5
5	FA1 Byte, Bit 6
6	FA1 Byte, Bit 7
7	FA1 Byte, Bit 8 (LSB)
8	FA2 Byte, Bit 1 (MSB)
9	FA2 Byte, Bit 2
10	FA2 Byte, Bit 3
11	FA2 Byte, Bit 4
12	FA2 Byte, Bit 5
13	FA2 Byte, Bit 6
14	FA2 Byte, Bit 7
15	FA2 Byte, Bit 8 (LSB)
16	EM Byte, Bit 1 (MSB)
17	EM Byte, Bit 2
18	EM Byte, Bit 3
19	EM Byte, Bit 4
20	EM Byte, Bit 5
21	EM Byte, Bit 6
22	EM Byte, Bit 7
23	EM Byte, Bit 8 (LSB)
24	TR Byte, Bit 1 (MSB)
25	TR Byte, Bit 2
26	TR Byte, Bit 3

TABLE 71: RELATIONSHIP BETWEEN 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 PROCESSED BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF RISING EDGES IN RXOHCLK SINCE RXOHFRAME BEING SAMPLED "HIGH	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
27	TR Byte, Bit 4
28	TR Byte, Bit 5
29	TR Byte, Bit 6
30	TR Byte, Bit 7
31	TR Byte, Bit 8 (LSB)
32	MA Byte, Bit 1 (FERF/RDI)
33	MA Byte, Bit 2 (FEBE/REI)
34	MA Byte, Bit 3
35	MA Byte, Bit 4
36	MA Byte, Bit 5
37	MA Byte, Bit 6
38	MA Byte, Bit 7
39	MA Byte, Bit 8 (LSB)
40	NR Byte, Bit 1 (MSB)
41	NR Byte, Bit 2
42	NR Byte, Bit 3
43	NR Byte, Bit 4
44	NR Byte, Bit 5
45	NR Byte, Bit 6
46	NR Byte, Bit 7
47	NR Byte, Bit 8 (LSB)
48	GC Byte, Bit 1 (MSB)
49	GC Byte, Bit 2
50	GC Byte, Bit 3
51	GC Byte, Bit 4
52	GC Byte, Bit 5
53	GC Byte, Bit 6
54	GC Byte, Bit 7
55	GC Byte, Bit 8 (LSB)

6.3.6.2 Operating the Receive Overhead Data Output Interface block using Method 2 - The "RxClk/RxOHEnable" Method

This particular method is referred to as the "RxClk/RxOHEnable" method for the following reasons.

a. The System-Side Terminal Equipment will use the "RxOHEnable" output pin (form the Receive Overhead Data Output Interface block) to keep track of which overhead bit is being processed by the Receive Overhead Data Output Interface (via the RxOH output pin) as any given time.

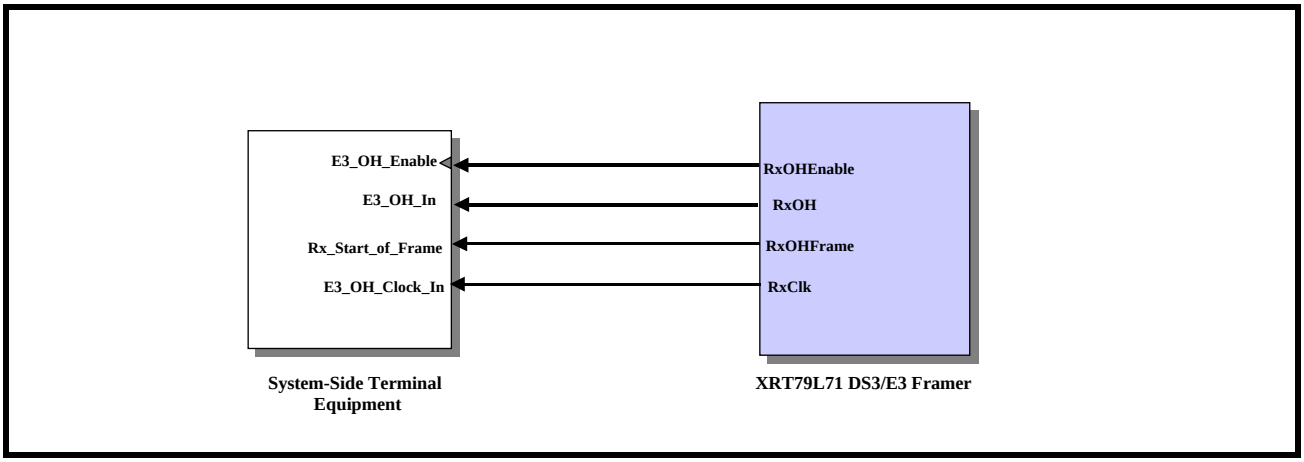
b. The System-Side Terminal Equipment must use the falling edge of "RxClk" in order to sample and latch the data residing on the "RxOH" output pin.

If "Method 2" is used then the "System-Side" Terminal Equipment will need to interface to the following "Receive Overhead Data Output Interface" pins.

- RxOH
- RxOHFrame
- RxOHEnable
- RxClk

If "Method 2" is used then design your system such that the "System-Side" Terminal Equipment will be interfaced to the "Receive Overhead Data Output Interface" block as shown below in **Figure 260**.

FIGURE 260. ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE OVERHEAD DATA OUTPUT INTERFACE" BLOCK WHEN USING "METHOD 2"



Method 2 Operation of the Receive Overhead Data Output Interface Block

If the Receive Overhead Data Output Interface block is operated per Method 2", design/configure the "System-Side" Terminal Equipment to continuously execute the following tasks.

TASK # 1: The "System-Side" Terminal Equipment must sample the states of the "RxOHFrame", "RxOH" and the "RxOHEnable" output pins (from the XRT79L71) upon the falling edge of "RxClk" clock output signal. The XRT79L71 will pulse the "RxOHEnable" output pin "high" for one "RxClk" period, coincident to whenever it is "safe" (or OK) to sample the states of the "RxOH" and the "RxOHFrame" output pins. If the System-Side Terminal Equipment samples both the "RxOHEnable" and the "RxOHFrame" output pin "high", then it "knows" that the Receive Overhead Data Output Interface block is currently placing the very first overhead bit (within the most recently received DS3 frame) via the "RxOH" output pin.

TASK # 2: As the "System-Side" Terminal Equipment samples the "RxOHEnable" and "RxOHFrame" output signals, it must also keep track of the number of times that the "RxOHEnable" output pin has been sampled "high" since the last time that both the "RxOHEnable" and "RxOHFrame" output pins have been sampled "high". By doing this, the "System-Side" Terminal Equipment will be able to keep track of which overhead bits are being output via the "RxOH" output at any time. This will permit the "System-Side Terminal Equipment to properly handle these overhead bits as appropriate.

Table 72 relates the number of "RxOHEnable" output pulses that have occurred since both the "RxOHFrame" and the "RxOHEnable" output pins were sampled "high", to the E3 Overhead Bit that is being output via the "RxOH" output pin. The user can use this table as a guide for extracting the appropriate overhead bits, via the "Receive Overhead Data Output Interface" block, whenever "Method 2" is used.

TABLE 72: RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE "RXOHENABLE" SIGNAL, SINCE THE RXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING OUTPUT (VIA THE RXOH OUTPUT PIN) BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF PULSES IN RXOHENABLE SINCE RXOHFRAME BEING SAMPLED "HIGH	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
0 (RxOHEnable and RxOHFrame are sampled "high" simultaneously)	FA1 Byte, Bit 1 (MSB)
1	FA1 Byte, Bit 2
2	FA1 Byte, Bit 3
3	FA1 Byte, Bit 4
4	FA1 Byte, Bit 5
5	FA1 Byte, Bit 6
6	FA1 Byte, Bit 7
7	FA1 Byte, Bit 8 (LSB)
8	FA2 Byte, Bit 1 (MSB)
9	FA2 Byte, Bit 2
10	FA2 Byte, Bit 3
11	FA2 Byte, Bit 4
12	FA2 Byte, Bit 5
13	FA2 Byte, Bit 6
14	FA2 Byte, Bit 7
15	FA2 Byte, Bit 8 (LSB)
16	EM Byte, Bit 1 (MSB)
17	EM Byte, Bit 2
18	EM Byte, Bit 3
19	EM Byte, Bit 4
20	EM Byte, Bit 5
21	EM Byte, Bit 6
22	EM Byte, Bit 7
23	EM Byte, Bit 8 (LSB)
24	TR Byte, Bit 1 (MSB)
25	TR Byte, Bit 2
26	TR Byte, Bit 3

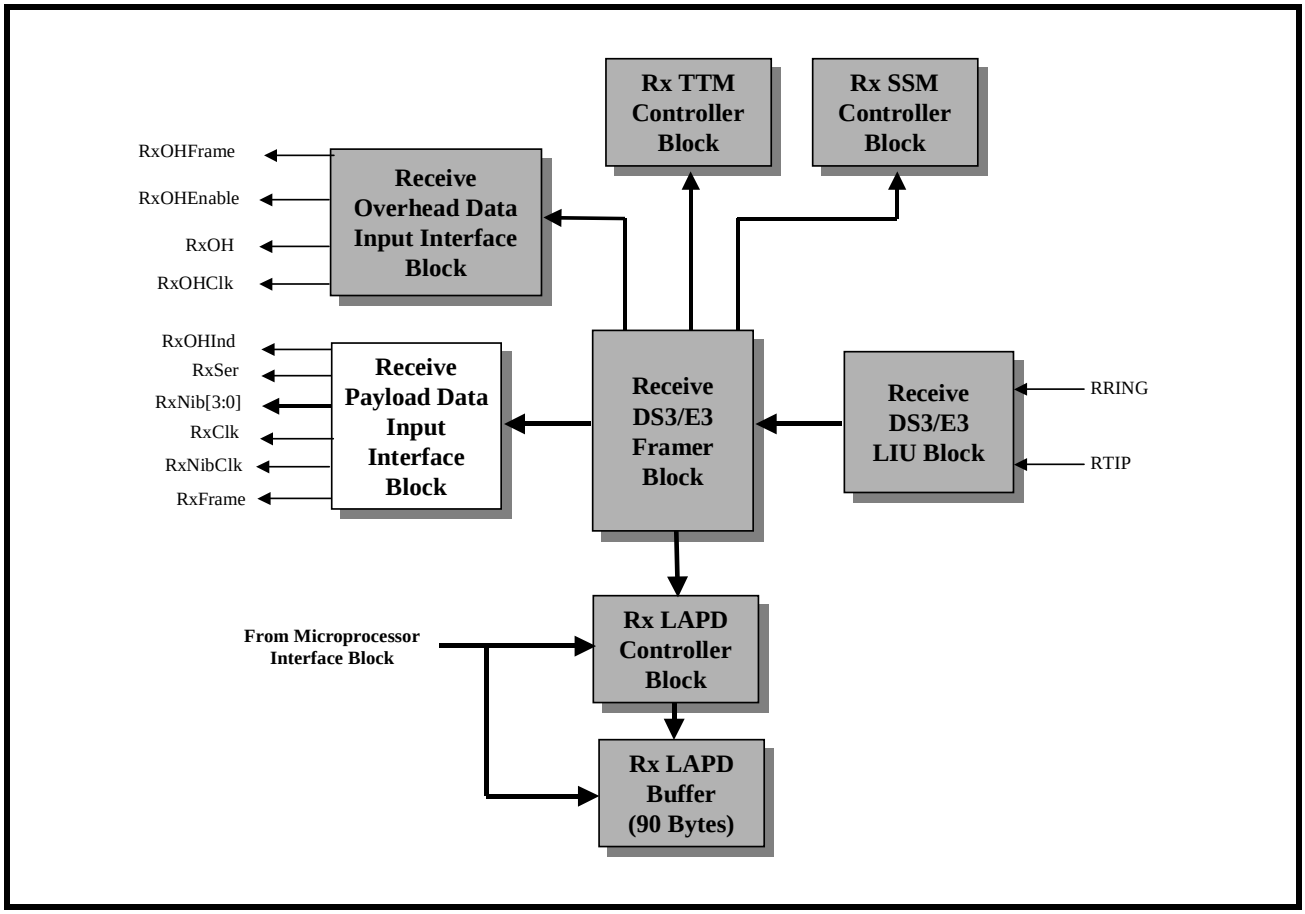
TABLE 72: RELATIONSHIP BETWEEN THE NUMBER OF PULSES IN THE "RXOHENABLE" SIGNAL, SINCE THE RXOHFRAME SIGNAL WAS LAST SAMPLED "HIGH" TO THE E3 OVERHEAD BIT THAT IS BEING OUTPUT (VIA THE RXOH OUTPUT PIN) BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK

NUMBER OF PULSES IN RXOHENABLE SINCE RXOHFRAME BEING SAMPLED "HIGH	"THE OVERHEAD BIT TO BE OUTPUT BY THE RECEIVE OVERHEAD DATA OUTPUT INTERFACE BLOCK
27	TR Byte, Bit 4
28	TR Byte, Bit 5
29	TR Byte, Bit 6
30	TR Byte, Bit 7
31	TR Byte, Bit 8 (LSB)
32	MA Byte, Bit 1 (FERF/RDI)
33	MA Byte, Bit 2 (FEBE/REI)
34	MA Byte, Bit 3
35	MA Byte, Bit 4
36	MA Byte, Bit 5
37	MA Byte, Bit 6
38	MA Byte, Bit 7
39	MA Byte, Bit 8 (LSB)
40	NR Byte, Bit 1 (MSB)
41	NR Byte, Bit 2
42	NR Byte, Bit 3
43	NR Byte, Bit 4
44	NR Byte, Bit 5
45	NR Byte, Bit 6
46	NR Byte, Bit 7
47	NR Byte, Bit 8 (LSB)
48	GC Byte, Bit 1 (MSB)
49	GC Byte, Bit 2
50	GC Byte, Bit 3
51	GC Byte, Bit 4
52	GC Byte, Bit 5
53	GC Byte, Bit 6
54	GC Byte, Bit 7
55	GC Byte, Bit 8 (LSB)

6.3.7 RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

The Receive Payload Data Output Interface block is the seventh (and final) functional block (within the Receive Direction) of the XRT79L71 that we will discuss for Clear-Channel Framer Applications. Figure 261 presents an illustration of the "Receie Direction" circuitry whenever the XRT79L71 has been configured to operate in the E3, ITU-T G.832 Clear-Channel Framer Mode, with the "Receive Payload Data Output Interface" block highlighted.

FIGURE 261. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE RECEIVE DIRECTION CIRCUITRY, WHENEVER THE XRT79L71 HAS BEEN CONFIGURED TO OPERATE IN THE E3, ITU-T G.832 CLEAR-CHANNEL FRAMER MODE (WITH THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK HIGHLIGHTED)



The purpose of the "Receive Payload Data Output Interface" block is to output payload data that has been extracted from the incoming E3 data-stream that has been received and processed by the "Receive Direction" circuitry within the XRT79L71.

In order to accomplish this, the Receive Payload Data Output Interface block has numerous output pins. Table 73 presents a list and a brief definition of each of these pins.

TABLE 73: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxSer	A5	O	**Receive Serial Payload Data Output pin:** If the Receive Payload Data Output Interface block is operated in the "Serial" Mode, then the XRT79L71 will output the payload data (that has been extracted from the incoming E3 data-stream), via this output pin. The XRT79L71 will update the data (on this pin) upon the rising edge of the RxCLK output clock signal.T Design (or configure) the System-Side Terminal Equipment such that it will sample the data that is output via this output pin, upon the falling edge of RxCLK. ***NOTES:*** *1. This signal is only active if the "NibIntf" input pin is pulled "low".* *2. In reality, for Serial Mode operation, the entire incoming E3 data-stream (payload bits and overhead bits) will be output via the "RxSer" output pin. The user will need to use the "RxOHInd/RxGapClk" signals in order to distinguish the "payload bits" from the "overhead bits" as they are output via the "RxSer" output pin.*
RxNib[3:0]	B4 A4 D6 C5	O	**Receive Nibble-Parallel Payload Data Output Pin:** If the Receive Payload Data Output Interface block is operated in the "Nibble-Parallel" Mode, then the XRT79L71 will output the payload data (that has been extracted from the incoming E3 data-stream), via these output pins, in a "Nibble-Parallel" manner. The XRT79L71 will update the data (via these four output pins) upon the falling edge of the RxCLK output signal. Design (or configure) the System-Side Terminal Equipment such that it will sample this data upon the rising edge of RxCLK. ***NOTE:*** *These pins are only active if the "NibIntf" input pin is pulled "high".*

TABLE 73: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxCLK/ RxNibClk	A6	O	**Receive Payload Data Output Interface -Clock Output Pin:** The behavior of this signal depends upon whether the XRT79L71 has been configured to operate in the "Serial Mode" or in the "Nibble-Parallel Mode". **Serial Mode Operation - RxCLK** If the Receive Payload Data Output Interface block has been configured to operate in the Serial Mode, then this signal will be a 34.368MHz clock output signal. The Receive Payload Data Output Interface block will update the data (via the RxSer output pin) upon the rising edge of this clock signal. For "Serial Mode" operation design (or configure) the System-Side Terminal Equipment to sample the data on the "RxSer" output pin, upon the falling edge of this clock signal. **Nibble-Parallel Mode Operation - RxNibClk** If the Receive Payload Data Output Interface block has been configured to operate in the Nibble-Parallel Mode, then the XRT79L71 will pulse this output pin 1074 times for each inbound E3 frame. The Receive Payload Data Output Interface block will update the data (via the "RxNib[3:0]" output pins upon the falling edge of this clock signal. For "Nibble-Parallel Mode" operation design (or configure) the System-Side Terminal Equipment to sample the data on the "RxNib[3:0]" output pins, upon the rising edge of this clock signal. ***NOTE:*** *This output clock signal is ultimately derived from the Recovered Clock signal (via the Receive DS3/E3 LIU Block).*

TABLE 73: LIST AND DESCRIPTION OF THE PINS ASSOCIATED WITH THE RECEIVE PAYLOAD DATA OUTPUT INTERFACE BLOCK

SIGNAL NAME	PIN/BALL NUMBER	TYPE	DESCRIPTION
RxOHInd/ RxGapClk	C6	O	**Receive Overhead Bit Indicator Output/Receive Gap-Clock Output:** The function of this output pin depends upon whether the XRT79L71 has been configured to operate in the Non-Gapped Clock or the Gapped-Clock Mode. **Non-Gapped Clock Mode - RxOHInd:** This output pin will pulse "high", for one "RxClk" period, coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the "RxSer" output pin. This output pin will be held "low" at all other times. The purpose of this output pin is to alert the System-Side Terminal Equipment that the current bit (e.g., the one that is currently residing on the RxSer output pin) is an overhead bit and should not be processed by the System-Side Terminal Equipment. The XRT79L71 will update this output signal upon the rising edge of RxClk. Therefore, design (or configure) the System-Side 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 active in the "RxOHInd" role, if the Receive Payload Data Output Interface block has been configured to operate in either the Serial or the Nibble-Parallel Mode. This output pin will be held "low" at all times. This is in contrast to DS3 applications, in which is pin is only active for "Serial" Mode operation.* **Gapped Clock Mode - RxGapClk:** In this mode this output pin will function as a "payload bit" output clock signal. In other words, the Receive Payload Output Interface block will only generate a clock pulse (via this output pin) coincident to when it outputs a payload bit via the RxSer output pin. The Receive Payload Data Output Interface block will NOT generate a clock edge (via this output pin) coincident to whenever it outputs an overhead bit via the "RxSer" output pin. As a consequence, there will be "gaps" within this particular clock output signal (hence the name "Gapped Clock Mode"). If the XRT79L71 is configured to operate in the Gapped Clock Mode, then design or configure the System-Side Terminal Equipment to sample and latch the "RxSer" data upon the falling edge of the "RxOHInd/ RxGapClk" clock signal.
RxFrame	B6	O	**Receive Payload Data Output Interface - Receive Start of Frame Output Indicator:** The behavior of this output pin depends upon whether the XRT79L71 has been configured to operate in the "Serial" or in the "Nibble-Parallel" Mode. **Serial Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "high" (for one RxCLK period) coincident to whenever it outputs the very first bit of a new E3 frame via the "RxSer" output pin. This output pin will remain "low" at all other times. **Nibble-Parallel Mode Operation** The Receive Payload Data Output Interface block will pulse this output pin "high" (for one RxCLK or "Nibble-Period") coincident to whenever it outputs the very first nibble of a new E3 frame via the "RxNib[3:0]" output pins. This output pin will remain "low" at all other times.

Operation of the Receive Payload Data Output Interface Block

The Receive Payload Data Output Interface block permits the user to configure it to operate in either of the following modes.

- The "Serial" Mode
- The "Nibble-Parallel" Mode

6.3.7.1 Serial Mode Operation of the Receive Payload Data Output Interface

If the Receive Payload Data Output Interface block is configured to operate in the "Serial Mode" then the Receive Payload Data Output Interface block can be further configured to operate in either the "Non-Gapped Clock" Mode or in the "Gapped Clock" Mode.

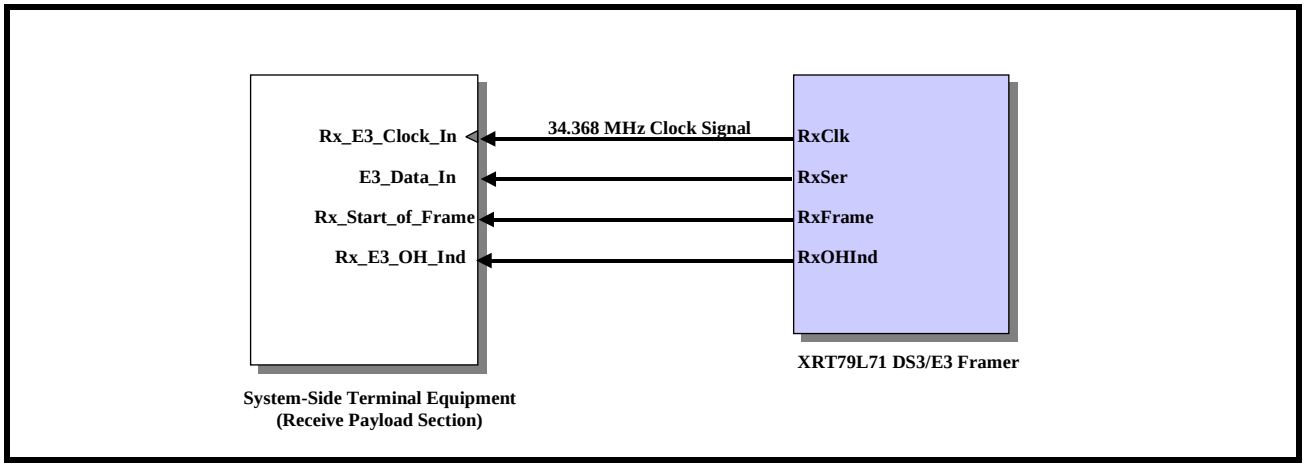
6.3.7.1.1 Operating the Receive Payload Data Output Interface block in the "Non-Gapped Clock" Mode

If the Receive Payload Data Output Interface block is configured to operate in the "Serial/Non-Gapped Clock" Mode, then all of the following are true.

- The XRT79L71 will output the entire contents of the incoming E3 data-stream (consisting of both payload and overhead bits) via the RxSer output pin, upon the rising edge of the RxCLK signal (which is a 34.368MHz clock signal).
- The user will need to rely on the "RxOHInd/RxGapClk" output pin in order to distinguish a payload bit from an overhead bit, within the data that is output via the "RxSer" output pin.
- The XRT79L71 will pulse the "RxFrame" output pin "high" for one "RxCLK" period, coincident to whenever it outputs the very first bit within a given E3 frame via the "RxSer" output pin. The "RxFrame" output pin will be held "low" at all other times.

Figure 262 presents an illustration of how to Interface the "System-Side" Terminal Equipment to the "Receive Payload Data Output Interface" block (of the XRT79L71), for Serial Mode Operation.

FIGURE 262. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "SERIAL MODE" OPERATION



Whenever the XRT79L71 has been configured to operate in this mode, then the Receive Payload Data Output Interface block will function as the source of the 34.368MHz clock signal (via the RxCLK output signal). This clock signal is used as the "System-Side Terminal Equipment" clock source by both the "Receive Payload Data Output Interface block (of the XRT79L71) and the "System-Side Terminal Equipment" device or circuitry.

The Receive Payload Data Output Interface block will serially output the entire contents of the incoming E3 data-stream via the "RxSer" output pin. As mentioned earlier, the Receive Payload Data Output Interface block will output this data upon the rising edge of the RxCLK signal. As a consequence, the user is advised to

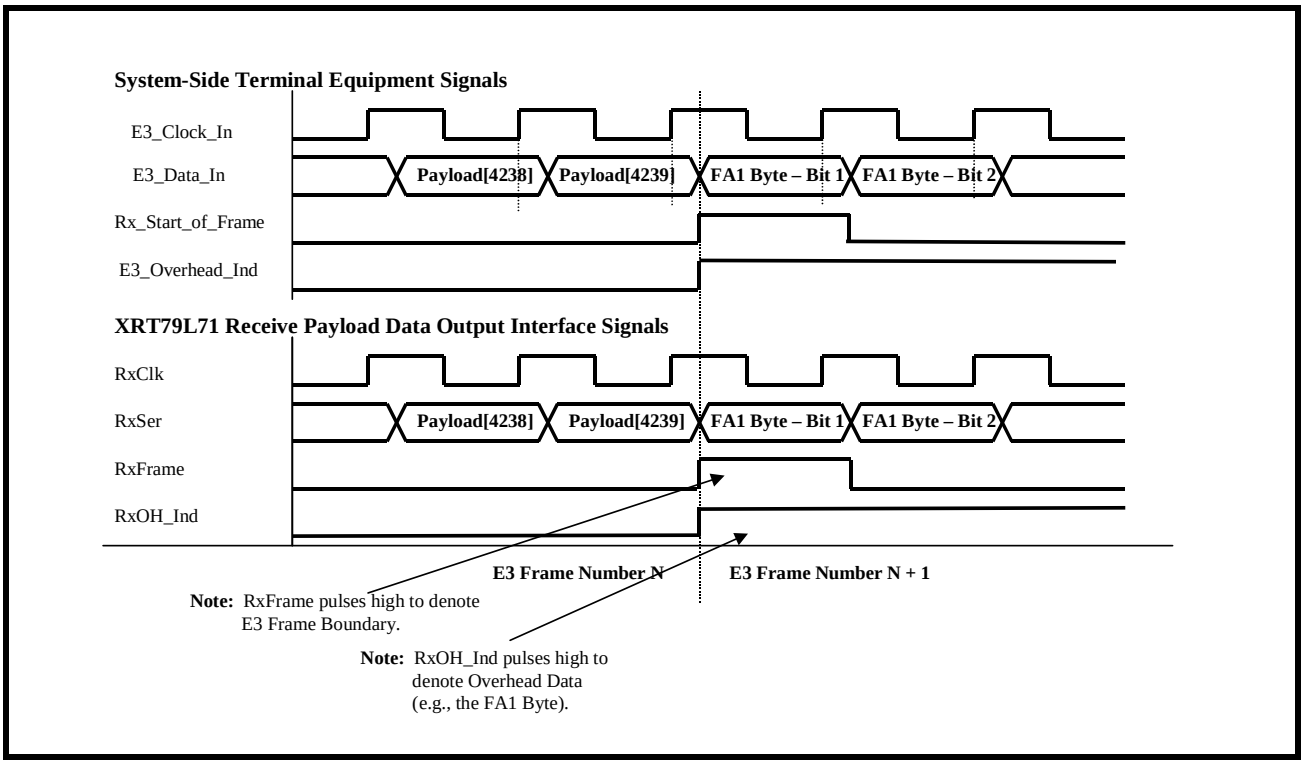
design (or configure) the "System-Side Terminal Equipment" circuitry to sample and latch this data (via the "E3_Data_In" input pin) upon the falling edge of RxCLK (Rx_E3_Clock_In), as depicted below in **Figure 263**.

The Receive Payload Data Output Interface block (within the XRT79L71) will indicate that it is processing the very first bit of a given E3 frame by pulsing the "RxFrame" output pin "HIGH" for one bit-period. The "RxFrame" output pin will be held "LOW" at all other times.

Finally, the Receive Payload Data Output Interface block (within the XRT79L71) permits the System-Side Terminal Equipment to identify a given bit (that is being output via the "RxSer" output pin) as either an "overhead" or a "payload" bit by pulsing the "RxOH_Ind" output pin "HIGH" (for one bit-period) coincident to whenever the Receive Payload Data Output Interface block outputs an overhead bit via the "RxSer" output pin. Conversely, the Receive Payload Data Output Interface block will hold the "RxOH_Ind" output pin "LOW" coincident to whenever the Receive Payload Data Output Interface block outputs a payload bit via the "RxSer" output pin.

NOTE: *Since the E3, ITU-T G.832 framing format consists of "Overhead Bytes", whenever the "RxOH_Ind" output pin pulses "high" (in order to denote an overhead bit), it will typically pulse high for at least eight (8) consecutive RxCLK periods*

FIGURE 263. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR "SERIAL MODE" OPERATION



Configuring the XRT79L71 to operate in Serial Mode.

The XRT79L71 can be configured to operate in the Serial Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in **Figure 262**.

STEP 2 - Configure the XRT79L71 to operate in the Serial Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "LOW".

NOTE: *This step also configures the "Transmit Payload Data Input Interface" block to operate in the "Serial Mode".*

Operating the Receive Payload Data Output Interface in the "Non-Gapped Clock" Mode

If the Receive Payload Data Output Interface block (within the XRT79L71) has been configured to operate in the "Serial" Mode, then we have recommended that the user design or configure their "System-Side Terminal Equipment" to do the following, when receiving/accepting E3 data via the "RxSer" output pin.

- Check the state of the "RxOH_Ind" output pin (from the XRT79L71) upon the falling edge of the "RxClk" signal.
- Perform either of the following actions, depending upon the sampled state of the "RxOH_Ind" output pin, as described below.

If RxOH_Ind is sampled "LOW"

Then the "System-Side Terminal Equipment" should accept this particular data bit (that is being sampled from the RxSer output pin) and treat it as a payload bit.

If RxOH_Ind is sampled "HIGH"

Then the "System-Side Terminal Equipment" should NOT accept this particular data bit (that is being sampled from the RxSer output pin) and should definitely NOT treat it as a payload bit.

In this particular approach, the user must "gate" the acceptance of a particular data bit (being sampled via the RxSer output pin) based upon the corresponding sampled state of the "RxOH_Ind" output pin. While implementing such a design into a CPLD or ASIC design is not very difficult, the user can take advantage of an easier approach by configuring the Receive Payload Data Output Interface block to operate in the "Gapped-Clock" Mode.

6.3.7.1.2 Operating the Receive Payload Data Output Interface block in the "Gapped-Clock" Mode

As mentioned above, in order to simplify the task of interfacing the Receive Payload Data Output Interface block to certain devices, the XRT79L71 can be to configure the Transmit Payload Data Input Interface and the Receive Payload Data Output Interface blocks to operate in the "Gapped-Clock" Mod If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then the role of the "RxOH_Ind" output pin will change from being the "Overhead Indicator" output pin, to now being a "payload data clock" output pin. In other words, If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then it (the Receive Payload Data Output Interface block) will only generate a clock pulse (via the "RxOH_Ind" output pin) coincident to whenever it outputs a payload bit via the "RxSer" output pin. Whenever the Receive Payload Data Output Interface block outputs an overhead bit (via the "RxSer" output pin) then it will NOT generate a clock pulse via the "RxOH_Ind" output pin. This action will result in the Receive Payload Data Output Interface block generating a "gapped" clock signal via the "RxOH_Ind" output pin (hence the term, "Gapped-Clock" Mode).

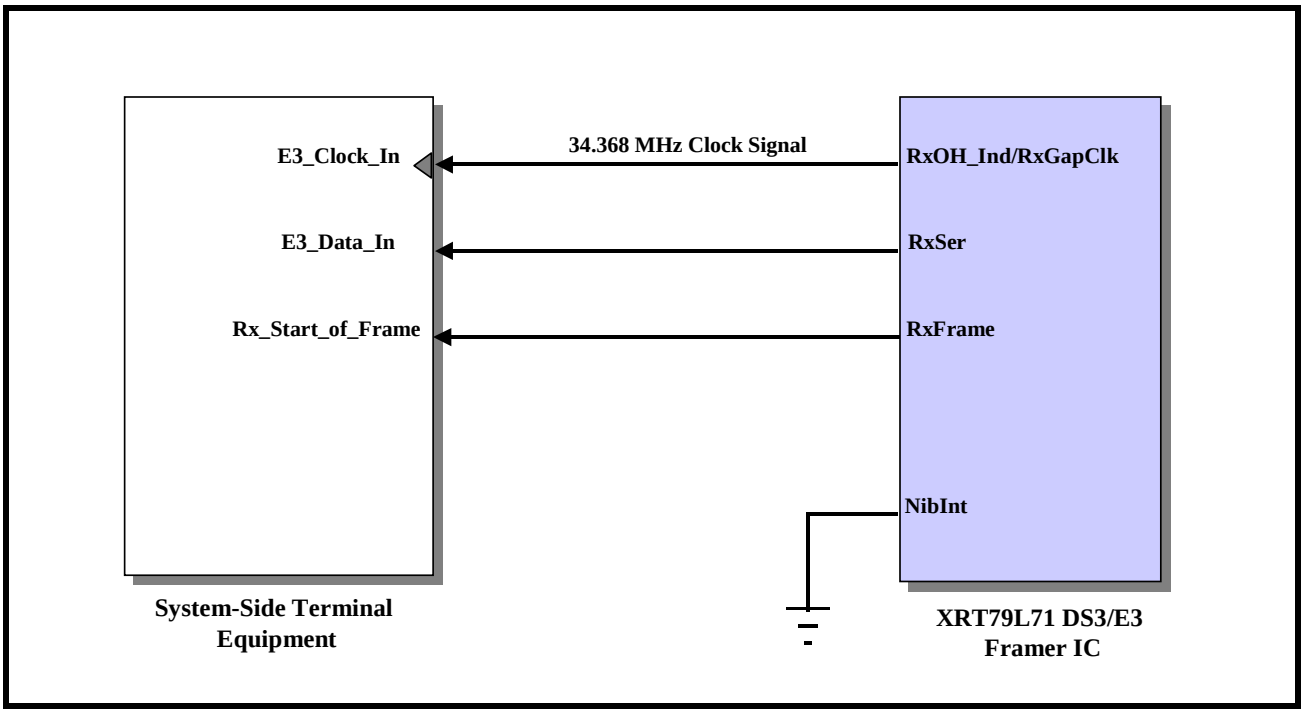
If the Receive Payload Data Output Interface block is configured to operate in the "Gapped-Clock" Mode, then the "System-Side" Terminal Equipment will be expected to sample the data (that is being output via the RxSer output pin) upon the rising edge of RxClk. In this case, there is no need to "check the state" of the certain output pin, then "gate" the acceptance/treatment of the next bit (output via the RxSer output pin) based upon the state of this particular output pin. The System-Side Terminal Equipment only needs to sample the "RxSer" output signal upon the rising edge of this particular "Gapped-Clock" signal.

Configuring the Receive Payload Data Input Interface block to operate in the Gapped-Clock Mode

If the Receive Payload Interface block is to be configured to operate in the "Gapped-Clock" Mode, do all of the following.

STEP 1 - Interface the "System-Side" Terminal Equipment to the Receive Payload Data Input Interface block as indicated below.

FIGURE 264. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE" TERMINAL EQUIPMENT TO THE "RECEIVE PAYLOAD DATA INPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "GAPPED-CLOCK" MODE OPERATIONS



STEP 2 - Set Bit 6 (RxGapped Clock Mode Enable), within the "Framer Test Register" to "1" as depicted below.

Framer Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOH Source	Receive Gapped Clock Mode Enable	Transmit Gapped Clock Mode Enable	Receive PRBS Lock	Receive PRBS Detector Enable	Transmit PRBS Generator Enable	Unused	
R/W	R/W	R/W	R/O	R/W	R/W	R/O	R/O
0	1	0	0	0	0	0	0

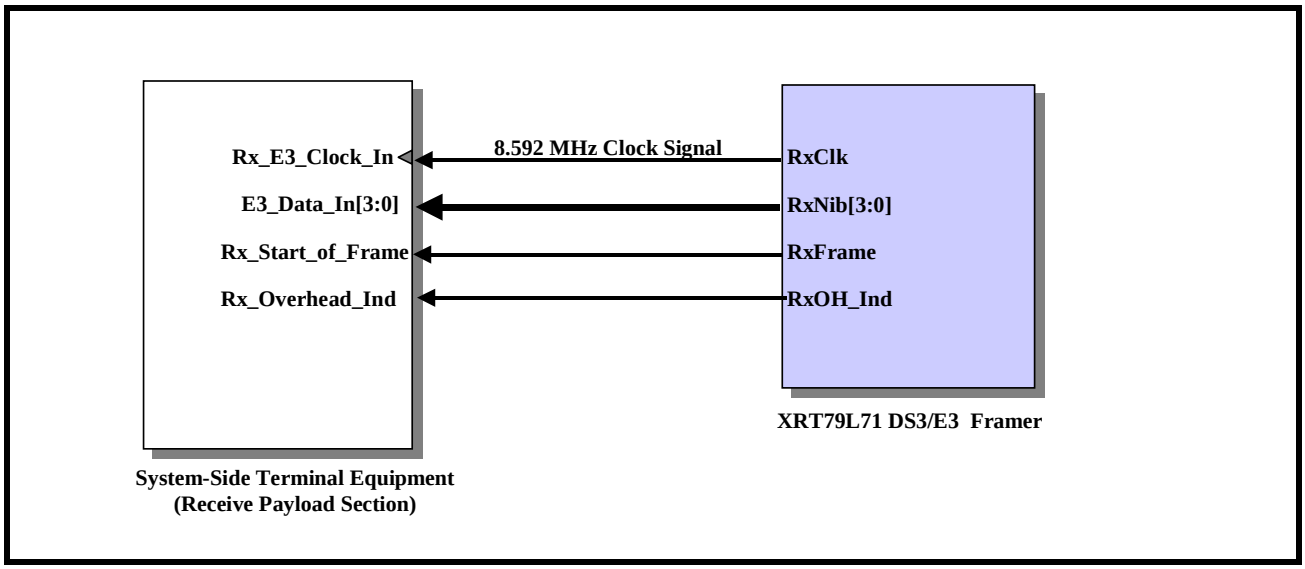
6.3.7.2 Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface

If the XRT79L71is configured to operate in the "Nibble-Parallel" Mode, then all of the following is true.

- The XRT79L71 will output the contents of both the payload bits and the overhead bits (within the incoming E3 data-stream) via the "RxNib[3:0]" output pins, upon the falling edge of the RxCLK signal.
- The XRT79L71 will pulse the "RxFrame" output pin "high" for one "RxCLK" (or Nibble) Period coincident to whenever it outputs the very first nibble within a given E3 frame via the "RxNib[3:0]" output pins. The "RxFrame" output pin will be held "low" at all other times.

Figure 265 presents an illustration of how to Interface the "System-Side Terminal Equipment" to the "Receive Payload Data Output Interface" block (of the XRT79L71), for "Nibble-Parallel Mode" Operation.

FIGURE 265. AN ILLUSTRATION OF HOW TO INTERFACE THE "SYSTEM-SIDE TERMINAL EQUIPMENT" TO THE "RECEIVE PAYLOAD DATA OUTPUT INTERFACE" BLOCK (OF THE XRT79L71) FOR "NIBBLE-PARALLEL MODE" OPERATION



Nibble-Parallel Mode Operation of the Receive Payload Data Output Interface Block

Whenever the XRT79L71 has been configured to operate in the "Nibble-Parallel" Mode, then the Receive Payload Data Output Interface block will function as the source of a Nibble Clock signal (via the "RxCLK" output signal).

The Receive Payload Data Output Interface block will output all of the payload and overhead data (that has been extracted out to the incoming E3 data-stream) in a "Nibble-Parallel" Mode via the "RxNib[3:0] output pins. As mentioned earlier, the Receive Payload Data Output Interface block will output this data upon the falling edge of the "RxCLK" signal. As a consequence, the user is advised to design (or configure) the "System-Side Terminal Equipment" circuitry to sample and latch this data (via the "E3_Data_In[3:0]" input pins) upon the rising edge of RxCLK (Rx_E3_Clock_In), as depicted below in **Figure 266**.

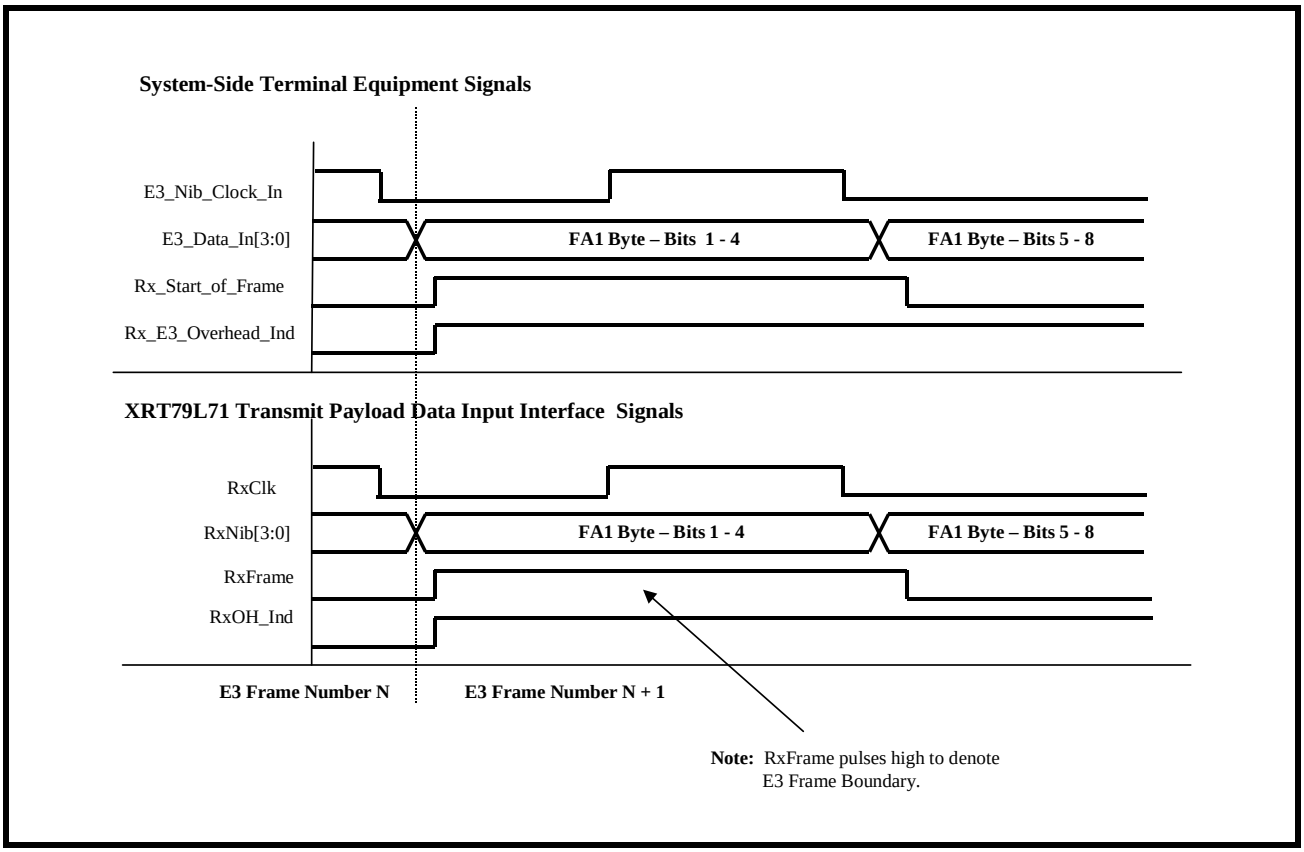
The Receive Payload Data Output Interface block (within the XRT79L71) will indicate that it is processing the very first payload nibble of a given E3 frame by pulsing the "RxFrame" output pin "HIGH" for one "nibble-period". The "RxFrame" output pin will be held "LOW" at all other times.

Finally, since (for E3 Applications) the Receive Payload Data Output Interface block (within the XRT79L71) does process and output overhead bits (in contrast to DS3, Nibble-Parallel Mode applications), then the Receive Payload Data Output Interface block will drive the "RxOH_Ind" output pin "high" coincident to whenever it outputs a nibble that contains "overhead" data. Conversely, the Receive Payload Data Output Interface block will drive the "RxOH_Ind" output pin "low" coincident to whenever it outputs a nibble that contains "payload" data.

The Frequency of the RxClk signal for E3, Nibble-Parallel Mode Operation

As mentioned above, whenever the Receive Payload Data Input Interface block has been configured to operate in the "Nibble-Parallel" Mode, it will process both the E3 payload and overhead bits. As a consequence, the frequency of the RxClk signal (for Nibble-Parallel Mode applications) will be exactly 34.368MHz/4 (or 8.592MHz).

FIGURE 266. AN ILLUSTRATION OF THE BEHAVIOR OF THE "SYSTEM-SIDE TERMINAL EQUIPMENT" SIGNALS FOR "NIBBLE-PARALLEL MODE" OPERATION



Configuring the XRT79L71 to operate in the Nibble-Parallel Mode

The XRT79L71 can be configured to operate in the Nibble-Parallel Mode by executing the following steps.

STEP 1 - Design your board such that the System-Side Terminal Equipment circuitry interfaces to the Receive Payload Data Input Interface in the manner as depicted above in Figure 268.

STEP 2 - Configure the XRT 79L71 to operate in the Nibble-Parallel Mode

This can be accomplished by setting the "NibIntf" input pin to a logic "HIGH".

NOTE: *This step also configures the "Transmit Payload Data Input Interface" block to operate in the "Nibble-Parallel Mode".*

7.0 DIAGNOSTIC OPERATION - CLEAR-CHANNEL FRAMER MODE

If the XRT79L71 has been configured to operate in the Clear-Channel Framer mode, then there are numerous "diagnostic" resources that are available to the user. These resources include (1) a variety of loop-back modes available, and (2) a PRBS Generator and Receiver.

7.1 THE LOOPBACK MODES AVAILABLE WITHIN THE XRT79L71

In all, If the XRT79L71 has been configured to operate in the Clear-Channel Framer Mode, then it will support the following loop-back modes.

- The LIU Analog Local Loop-back Mode
- The LIU Digital Local Loop-back Mode
- The LIU Remote Loop-back Mode
- The Framer Local Loop-back Mode

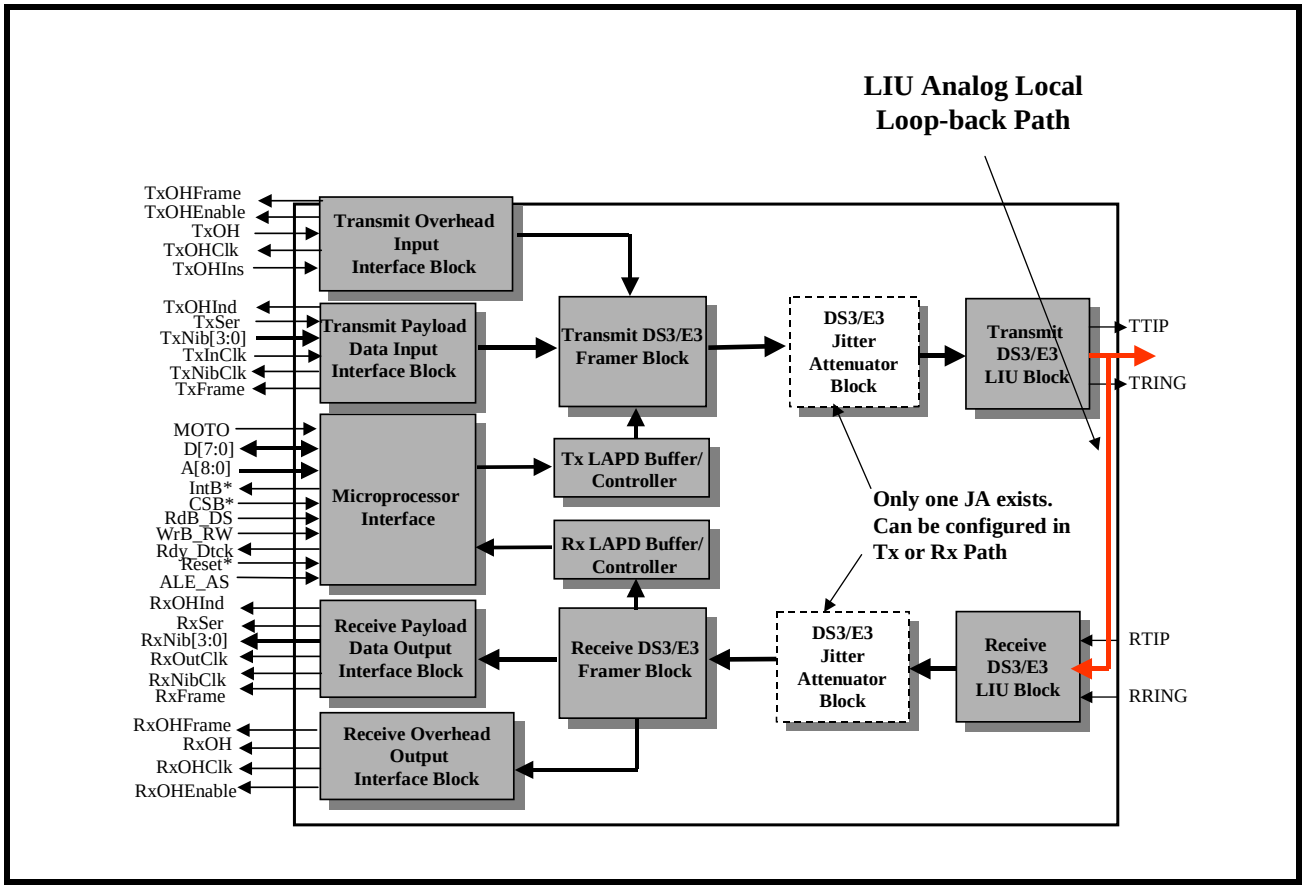
- The Framer/Internal Remote Loop-back Mode

7.1.1 The LIU Analog Local Loop-back Mode

If the XRT79L71 has been configured to operate in the "LIU Analog Local Loop-back" Mode, then (in essence) the signal that is being output via the "TTIP/TRING" output pads will be internally looped-back to the "RTIP/RRING" input pads. This type of Loop-back is referred to as the "LIU Analog Local-Loop-back" Mode, because it is the "analog" signal (that is output via the TTIP/TRING output pads) that is being looped back into the RTIP/RRING input pads.

Figure 267 presents an illustration of the Functional Block Diagram of the XRT79L71, when it is configured to operate in the "LIU Analog Local-Loop-back" Mode.

FIGURE 267. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 (WHEN CONFIGURED TO OPERATE IN THE CLEAR-CHANNEL FRAMER" MODE) WITH THE "LIU ANALOG LOCAL LOOP-BACK" PATH INDICATED



SOME THINGS TO NOTE ABOUT THE LIU ANALOG LOCAL LOOP-BACK MODE:

1. Whenever the XRT79L71 is configured to operate in the "LIU Analog Local Loop-back" Mode, the "Transmit Output" line signal will still be output to the remote terminal equipment. A replica of this "Transmit Output" line signal will also be routed to the "RTIP/RRING" input pads of the device.
2. Whenever the XRT79L71 is configured to operate in the "LIU Analog Local Loop-back" Mode, then it will ignore any data that is being applied to the RTIP/RRING input pins.
3. The LIU Analog Local Loop-back will be broken (e.g., this loop-back mode will NOT function) if the user sets Bit 0 (TxON), within the "LIU Transmit APS/Redundancy Control" Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

Configuring the XRT79L71 to operate in the "LIU Analog Local Loop-back" Mode

To configure the XRT79L71 to operate in the "LIU Analog Local Loop-back" Mode execute the following steps.

STEP 1 - Make sure that the XRT79L71 is operating in the "Local-Timing" Mode.

This is accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register is set to some value other than [0, 0].

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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	X	X

STEP 2 - Make sure that the Transmit Output Driver (of the Transmit DS3/E3 LIU Block) is turned ON

This is accomplished by making sure that Bit 0 (TxON), within the "LIU Transmit APS/Redundancy Control" Register is set to "1".

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

STEP 3 - Configure the XRT79L71 to operate in the "LIU Analog Local Loop-back" Mode

This is accomplished by setting Bits 4 (LIU Remote Loop-back Mode) and 3 (LIU Local Loop-back Mode) to the value [0, 1], as depicted below

LIU Control Register (Address = 0x1306)

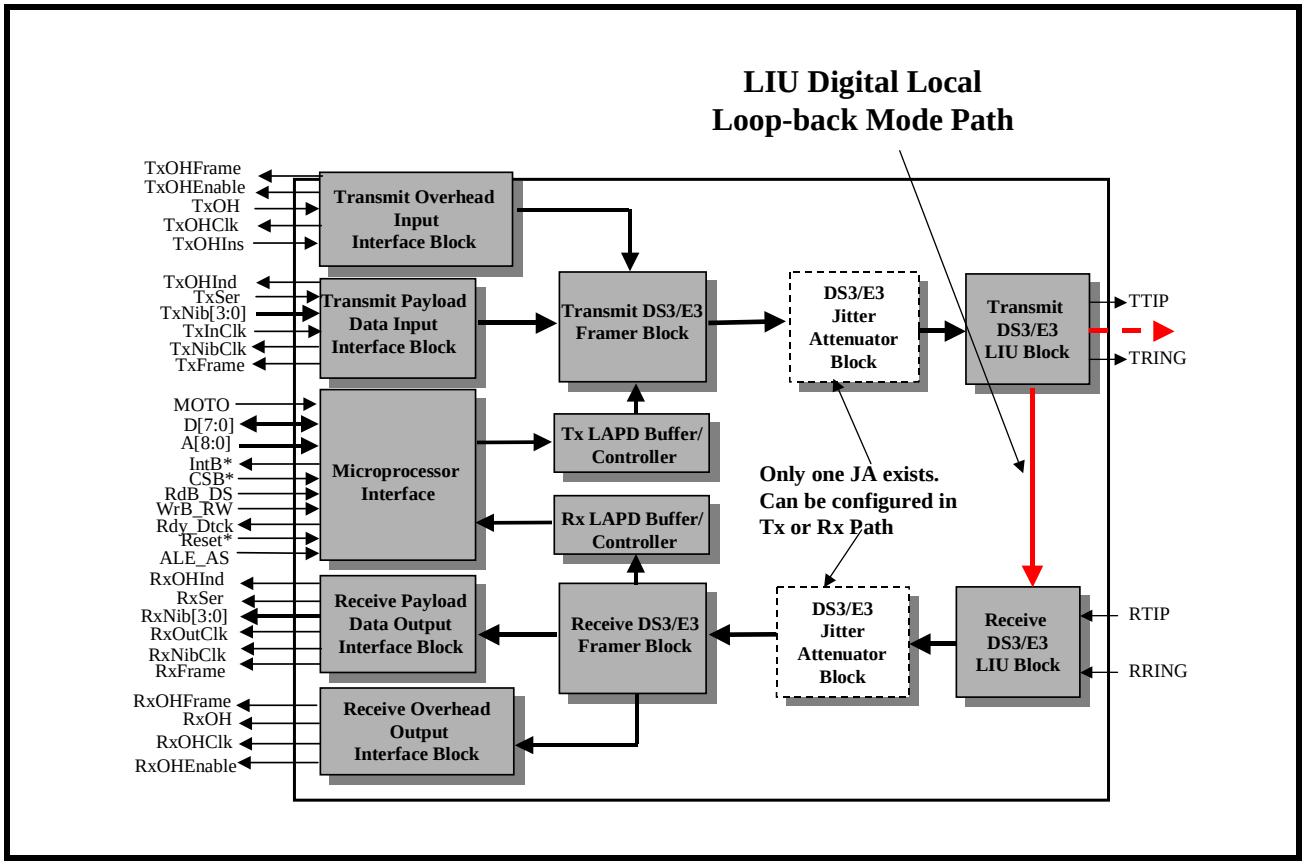
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	0	0	1	0	0	0

7.1.2 The LIU Digital Local Loop-back Mode

If the XRT79L71 has been configured to operate in the "LIU Digital Local Loop-back" Mode, then the output of the "Timing Control" block (within the Transmit DS3/E3 LIU Block) will be internally routed to the Output of the Clock and Data Recovery block (within the Receive DS3/E3 LIU Block). This type of Loop-back is referred to as the "LIU Digital Local Loop-back" Mode, because (in this case) a "Digital" Transmit Output signal is being looped back into the Receive Direction.

Figure 268 presents an illustration of the Functional Block Diagram of the XRT79L71, when it is configured to operate in the "LIU Digital Local Loop-back" Mode.

FIGURE 268. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 (WHEN CONFIGURED TO OPERATE IN THE "CLEAR-CHANNEL FRAMER" MODE) WITH THE "LIU DIGITAL LOCAL LOOP-BACK" PATH INDICATED



SOME THINGS TO NOTE ABOUT THE LIU DIGITAL LOCAL LOOP-BACK MODE:

1. Whenever the XRT79L71 is configured to operate in the "LIU Digital Local Loop-back" Mode, the "Transmit Output" line signal MAY still be output to the remote terminal equipment (depending upon the state of the "TxON" bit-field - see Note 3, below). A replica of this "Transmit Output" signal will also be routed into the Receive Path of the device.
2. Whenever the XRT79L71 is configured to operate in the "LIU Digital Local Loop-back" Mode, then it will ignore any data that is being applied to the RTIP/RRING input pins.
3. In contrast to the "LIU Analog Local Loop-back" Mode, the "LIU Digital Local Loop-back" Mode will NOT be broken (e.g., this loop-back mode will still function) even if the user sets Bit 0 (TxON), within the "LIU Transmit APS/Redudancy Control" Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

Configuring the XRT79L71 to operate in the "LIU Digital Local Loop-back" Mode

Ton configure the XRT79L71 to operate in the "LIU Digital Local Loop-back" Mode execute the following steps.

STEP 1 - Make sure that the XRT79L71 is operating in the "Local-Timing" Mode.

This is accomplished by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to some value other than [0, 0].

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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	X	X

STEP 2 - Configure the XRT79L71 to operate in the "LIU Digital Loop-back" Mode

This is accomplished by setting Bits 4 (LIU Remote Loop-back Mode) and 3 (LIU Local Loop-back Mode) within the "LIU Control" Register to the value [1, 1], as depicted below

LIU Control Register (Address = 0x1306)

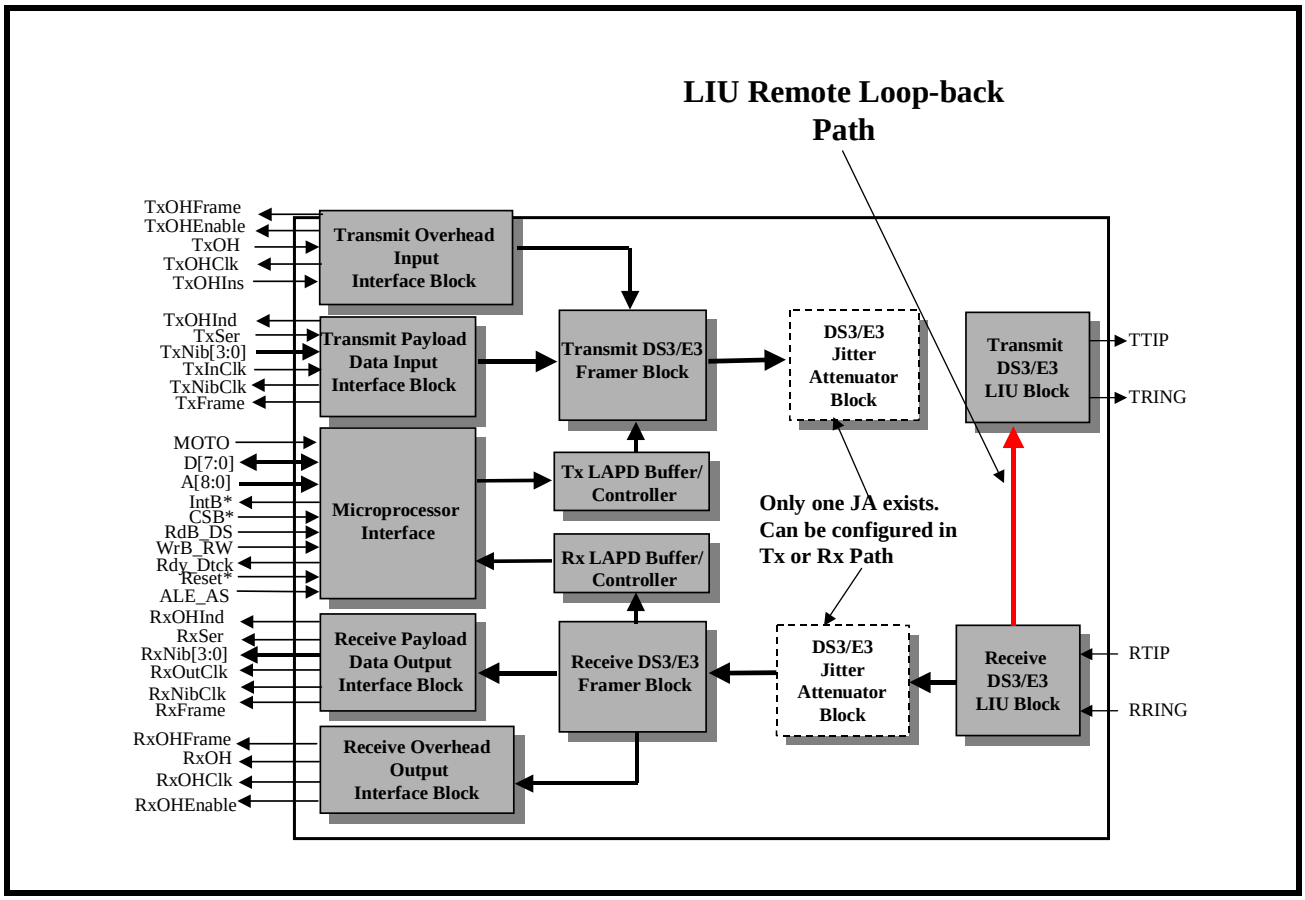
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	0	1	1	0	0	0

7.1.3 The LIU Remote Loop-back Mode

If the XRT79L71 has been configured to operate in the "LIU Remote Loop-back" Mode, then the output of the "Clock and Data Recovery" block (within the Receive DS3/E3 LIU Block) will be internally routed to the input of the "Timing Control" block (within the Transmit DS3/E3 LIU Block).

Figure 269 presents an illustration of the Functional Block Diagram of the XRT79L71, when it is configured to operate in the "LIU Remote Loop-back" Mode.

FIGURE 269. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 (WHEN CONFIGURED TO OPERATE IN THE "CLEAR-CHANNEL FRAMER" MODE) WITH THE "LIU REMOTE LOOP-BACK" PATH INDICATED



SOME THINGS TO NOTE ABOUT THE LIU REMOTE LOOP-BACK MODE:

1. Whenever the XRT79L71 is configured to operate in the "LIU Remote Loop-back" Mode, the "Receive signal will still be routed to all down-stream circuitry (e.g., such as the Receive DS3/E3 Framer block, etc.). A replica of this "Receive Input" signal will also be routed to the "Transmit Direction" within the device.
2. Whenever the XRT79L71 is configured to operate in the "LIU Remote Loop-back" Mode, then the Transmit DS3/E3 LIU Block will ignore any data that is being generated by the Transmit DS3/E3 Framer block.
3. The LIU Remote Loop-back will be broken (e.g., this loop-back mode will NOT function) if the user sets Bit 0 (TxON), within the "LIU Transmit APS/Redundancy Control" Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

Configuring the XRT79L71 to operate in the "LIU Remote Loop-back" Mode

To configure the XRT79L71 to operate in the "LIU Remote Loop-back" Mode execute the following steps.

STEP 1 - Make sure that the Transmit Output Driver (of the Transmit DS3/E3 LIU Block) is turned ON

This is accomplished by making sure that Bit 0 (TxON), within the "LIU Transmit APS/Redundancy Control" Register is set to "1" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

STEP 2 - Configure the XRT79L71 to operate in the "LIU Remote Loop-back" Mode

This is accomplished by setting Bits 4 (LIU Remote Loop-back Mode) and 3 (LIU Local Loop-back Mode), within the "LIU Control" Register to the value [1, 0], as depicted below

LIU Control Register (Address = 0x1306)

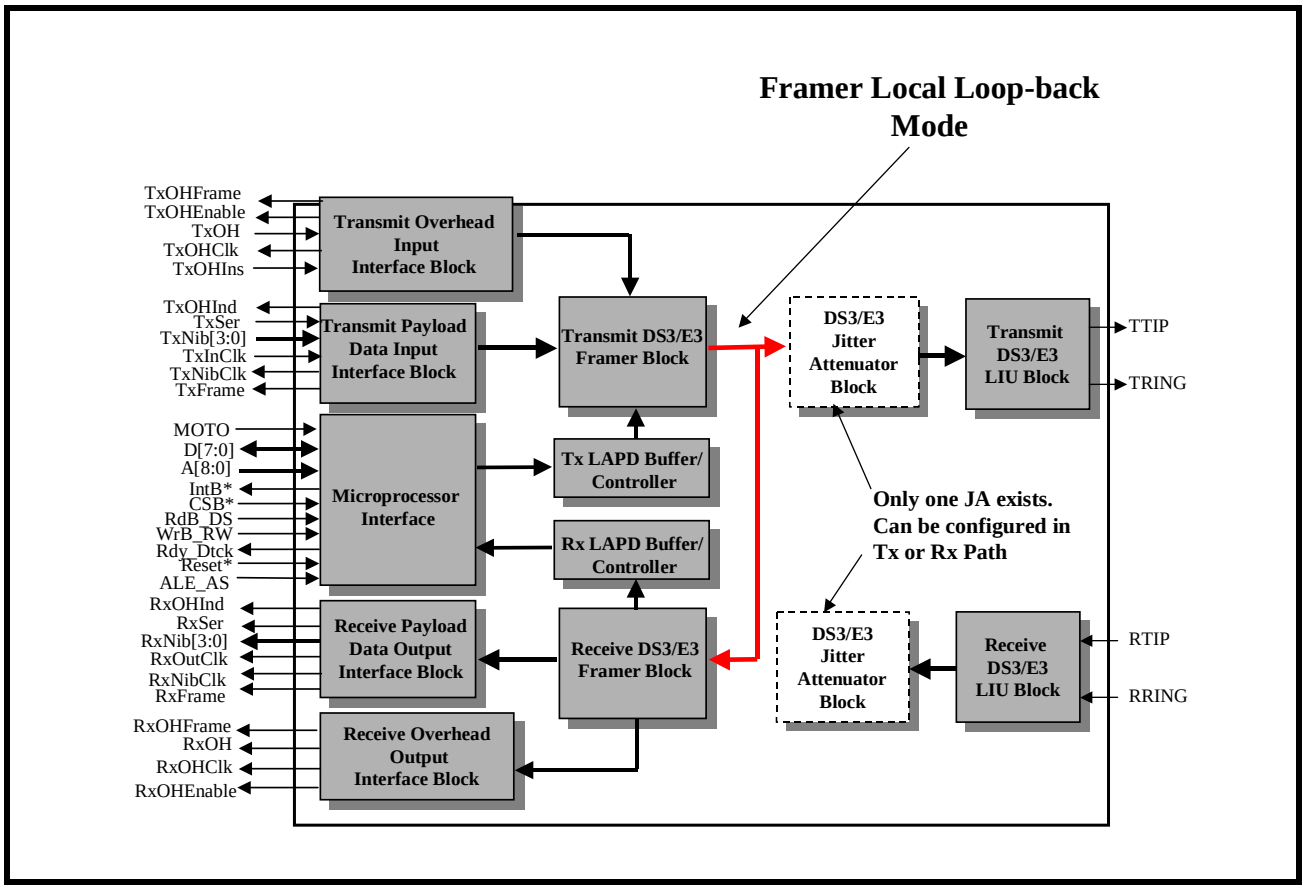
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Unused	SFM Clock Out Enable	SFM Enable	LIU Remote Loop-back Mode	LIU Local Loop-back Mode	Unused		
R/O	R/O	R/O	R/W	R/W	R/O	R/O	R/O
0	0	0	1	0	0	0	0

7.1.4 The Framer Local Loop-back Mode

If the XRT79L71 has been configured to operate in the "Framer Local Loop-back" Mode, then the output of the Transmit DS3/E3 Framer block will be internally routed to the input of the Receive DS3/E3 Framer block.

Figure 270 presents an illustration of the Functional Block Diagram of the XRT79L71, when it is configured to operate in the "Framer Local Loop-back" Mode.

FIGURE 270. ILLUSTRATION OF THE FUNCTIONAL BLOCK DIAGRAM OF THE XRT79L71 (WHEN CONFIGURED TO OPERATE IN THE "CLEAR-CHANNEL FRAMER" MODE) WITH THE "FRAMER LOCAL LOOP-BACK" PATH INDICATED



SOME THINGS TO NOTE ABOUT THE FRAMER LOCAL LOOP-BACK MODE:

1. Whenever the XRT79L71 is configured to operate in the "Framer Local Loop-back" Mode, the output of the Transmit DS3/E3 Framer block signal MAY still be output to the remote terminal equipment (depending upon the state of the "TxON" bit-field - see Note 3, below). A replica of this "Transmit Output" signal will also be routed into the input of the Receive DS3/E3 Framer block.
2. Whenever the XRT79L71 is configured to operate in the "Framer Local Loop-back" Mode, then the Receive DS3/E3 Framer block will ignore any data that it receives from the Receive DS3/E3 LIU Block.
3. In contrast to the "LIU Analog Local Loop-back" Mode, the "Framer Local Loop-back" Mode will NOT be broken (e.g., this loop-back mode will still function) even if the user sets Bit 0 (TxON), within the "LIU Transmit APS/Redudancy Control" Register to "0" as depicted below.

LIU Transmit APS/Redundancy Control Register (Address = 0x1300)

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

Configuring the XRT79L71 to operate in the "Framer Local Loop-back" Mode

To configure the XRT79L71 to operate in the "Framer Local Loop-back" Mode execute the following steps.

STEP 1 - Make sure that the XRT79L71 is operating in the "Local-Timing" Mode.

Accomplish this by setting Bits 1 and 0 (TimRefSel[1:0]) within the "Framer Operating Mode" Register to some value other than [0, 0].

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	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	X	X

STEP 2 - Configure the XRT79L71 to operate in the "Framer Loop-back" Mode

This is accomplished by setting Bit 7 (Framer Local Loop-back), within the "Framer Operating Mode" Register to the value "1", as depicted below

Framer Operating Mode Register (Address = 0x1100)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Framer Local Loop Back	IsDS3	Internal LOS Enable	RESET	Direct Map ATM	Frame Format	TimRefSel[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	0	1	0	X	X

7.2 USING THE PRBS PATTERN GENERATOR AND RECEIVER

The XRT79L71 contains a PRBS Pattern Generator and Receiver. The PRBS Pattern Generator resides within the Transmit DS3/E3 Framer block and the PRBS Pattern Receiver resides within the Receive DS3/E3 Framer block.

If the PRBS Pattern Generator is enabled, then the XRT 79L71 will then ignore any data that is being applied to the Transmit Payload Data Input Interface block. The payload bits (within each outbound DS3 or E3 frame) will now be filled by the PRBS Pattern Generator block.

For DS3 Applications, the PRBS Pattern Generator will generate a 2^{15-1} PRBS Pattern. As a consequence, whenever the PRBS Pattern Generator has been enabled, then the Transmit DS3/E3 Framer block will generate a DS3 signal that is of a "Framed" 2^{15-1} PRBS Pattern.

For E3 Applications, the PRBS Pattern Generator will generate a 2^{23-1} PRBS Pattern. As a consequence, whenever the PRBS Pattern Generator has been enabled, then the Transmit DS3/E3 Framer block will generate an E3 signal that is of a "Framed" 2^{23-1} PRBS Pattern.

Similarly, if thePRBS Pattern Receiver is enabled, then it will do the following.

- It will receive the payload DS3 or E3 data-stream from the Receive DS3/E3 Framer block
- It will attempt to acquire "PRBS Lock" with this DS3 or E3 data-stream.

For DS3 Applications, the PRBS Pattern Receiver will be expecting a 2^15-1 PRBS Pattern within the data-stream that it receives from the Receive DS3/E3 Framer block. For E3 Applications, the PRBS Pattern Receiver will be expecting a 2^23-1 PRBS Pattern within the data-stream that it receives from the Receive DS3/E3 Framer block.

The PRBS Generator and Receiver can be used in conjunction with any of the afore-mentioned "loop-back" modes.

- The LIU Analog Local Loop-back Mode
- The Framer Local Loop-back Mode
- Any external "Local Loop-back" Mode

7.2.1 Enabling the PRBS Pattern Generator and Receiver

The user can enable the PRBS Pattern Generator and Receiver by setting Bits 2 (RxPRBS Enable) and 3 (TxPRBS Enable), within the "Test Register" to "1" as depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOHSrc	Unused		RxPRBS Lock	RxPRBS Enable	TxPRBS Enable	Unused	
R/W	R/O	R/O	R/O	R/W	R/W	R/O	R/O
0	0	0	0	1	1	0	0

Once these steps have been executed, then the PRBS Pattern Generator will be enabled and will now proceed to fill the payload bits (within the outbound DS3 or E3 frames) with either a $2^{\wedge 15-1}$ PRBS Pattern (for DS3 Applications) or a $2^{\wedge 23-1}$ PRBS Pattern (for E3 Applications). Additionally, the PRBS Pattern Receiver will now begin to receive the payload data-stream (from the Receive DS3/E3 Framer block) and it will begin to do all of the following.

- Check for "PRBS Lock"
- Detect and Flag PRBS Bit Errors

7.2.2 Checking for "PRBS Lock"

As mentioned earlier, if the PRBS Pattern Receiver is enabled, then it will accept the payload data-stream from the Receive DS3/E3 Framer block and it will proceed to do the following.

- Check for "PRBS Lock"
- Detect and Flag PRBS Bit Errors

If the PRBS Pattern Receive acquires "PRBS Lock" then it will set Bit 4 (RxPRBS Lock) within the Test Register, to "1" as depicted below.

Test Register (Address = 0x110C)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TxOHSrc	Unused		RxPRBS Lock	RxPRBS Enable	TxPRBS Enable	Unused	
R/W	R/O	R/O	R/O	R/W	R/W	R/O	R/O
0	0	0	1	1	1	0	0

NOTE: *The user is also advise to validate the assertion of Bit 4 (RxPRBS Lock) with the state(s) of the appropriate "OOF Defect Declared" and "LOF Defect Declared" bit-fields. In other words, for DS3 Applications, one should validate*

Bit 4 (RxPRBS Lock), within the "Test Register" with Bit 4 (OOF Defect Condition Declared), within the "Receive DS3 Configuration and Status" Register, as depicted below.

Receive DS3 Configuration and Status Register (Address = 0x1110)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AIS Defect Condition Declared	LOS Defect Condition Declared	DS3 Idle Condition Declared	OOF Defect Condition Declared	Unused	Framing with Valid P-Bits	F-Sync Algo	M-Sync Algo
R/O	R/O	R/O	R/O	R/O	R/W	R/W	R/W
0	0	0	X	0	1	0	0

Similarly, for E3 Applications, one will need to validate Bit 4 (RxPRBS Lock) within the "Test Register" with Bits 5 (OOF Defect Condition Declared) and 6 (LOF Defect Condition Declared) within the "Receive E3 Configuration and Status Register # 2" registers, as depicted below for ITU-T G.751 and G.832, respectively.

Receive E3 Configuration and Status Register # 2 - G.751 (Address = 0x1111)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	Unused		FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	X	X	0	0	0	0	1

Receive E3 Configuration and Status Register # 2 - G.832 (Address = 0x1111)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxLOF Algo	LOF Defect Condition Declared	OOF Defect Condition Declared	LOS Defect Condition Declared	AIS Defect Condition Declared	RxPLD Unstab	RxT Mark	FERF/RDI Defect Condition Declared
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
0	X	X	0	0	1	1	1

7.2.3 Checking for PRBS Bit Errors

The XRT 79L71 permits one to check for any occurrence of PRBS Bits errors, for the duration that the PRBS Pattern Receiver is enabled. To determine if (and how many) PRBS errors have occurred during a certain period of time, read out the contents of the "PRBS Error Count" Registers. The address locations and bit-fields of these register is depicted below.

PRBS Error Count Register - MSB (Address = 0x1168)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PRBS_Error_Count_Upper_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

PRBS Error Count Register - LSB (Address = 0x1169)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PRBS_Error_Count_Lower_Byte[7:0]							
RUR	RUR	RUR	RUR	RUR	RUR	RUR	RUR
0	0	0	0	0	0	0	0

A NOTE ABOUT READING OUT THE CONTENTS OF THE PRBS ERROR COUNT REGISTERS

The PRBS Error Count Registers (below) are 16-bit "RESET-upon-READ" registers. However, the manner in which these registers are to be read is listed below.

As mentioned earlier, these Registers are 16-bits in length. More specifically these registers will consist of a "MSB" (Most Significant Byte) 8-bit register, and a "LSB" (Least Significant Byte) register. Since the Microprocessor Interface of the XRT 79L71 contains an eight-bit wide bi-directional data bus, the user will have to execute two consecutive read operations in order to obtain the full 16-bit contents of these PRBS Error Count Registers. As the user reads out the contents of these Registers, the user must be aware of the following restrictions.

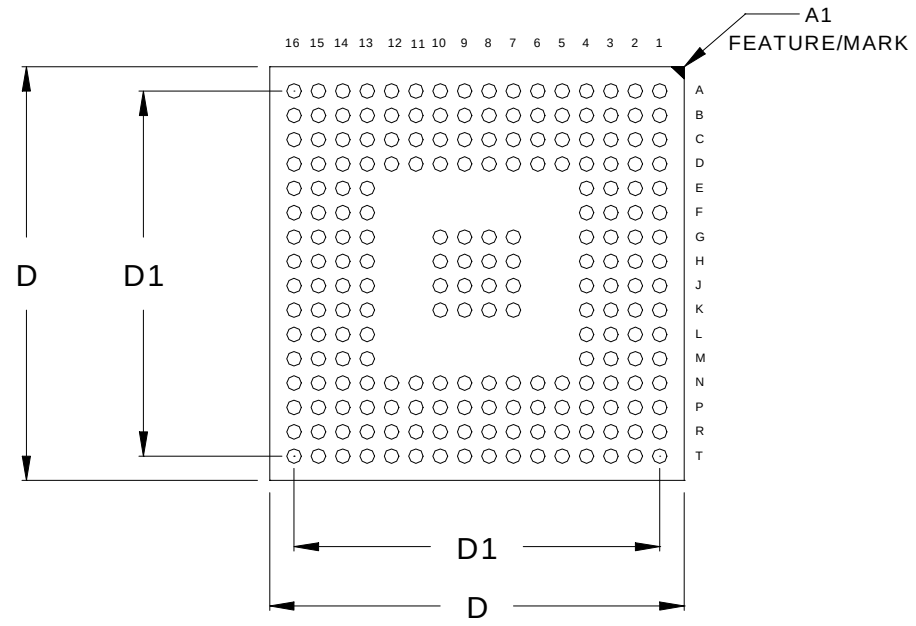
- During the first (of the two) read operations (to the PRBS Error Count Registers), the user read out either the "MSB" or the "LSB" Register.
- However, as the user executes this "first" read operation, the entire 16-bit contents of this particular PRBS Error Count Register will be cleared to "0x0000". The XRT 79L71 will store the contents of the "un-read" register into the "PMON Holding Register (Address = 0x116C).
- Therefore, during the second (of the two) read operations (to the PRBS Error Count Register), the user MUST obtain the contents of the "un-read" byte, from the PMON Holding Register.

ORDERING INFORMATION

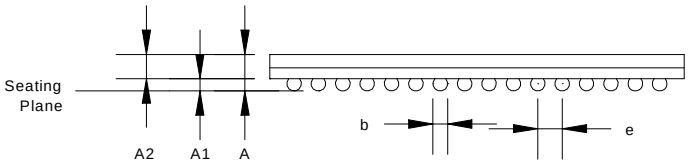
PRODUCT NUMBER	PACKAGE TYPE	OPERATING TEMPERATURE RANGE
XRT79L71IB	17X17 mm 208 Ball Shrink Thin Ball Grid Array	-40^{0}C to +85^{0}C

PACKAGE DIMENSIONS

208 SHRINK THIN BALL GRID ARRAY (17.0 MM X 17.0 MM, STBGA)



(A1 corner feature is mfger option)



Note: The control dimension is in millimeter.

	INCHES		MILLIMETERS	
SYMBOL	MIN	MAX	MIN	MAX
A	0.047	0.067	1.20	1.70
A1	0.010	0.022	0.25	0.55
A2	0.031	0.043	0.80	1.10
D	0.661	0.677	16.80	17.20
D1	0.591 BSC		15.00 BSC	
b	0.018	0.022	0.45	0.55
e	0.0394 BSC		1.00 BSC	

REVISION HISTORY

REVISION #	DATE	DESCRIPTION
P1.0.0	07/18/02	1st release of the XRT99L00 mkII.0 preliminary data sheet.
P1.0.1	02/12/03	Added package outline and pin-out diagram.
P1.0.2	05/03	Added Pin Descriptions
P1.0.3	06/03	Added Electrical Specifications and Register Information.
P1.0.4	07/03	Default Value added to Address Locations 104 and 105 in register map. Add pin TxSer (C9) to pin list. I/O Control Register (Direct Address = 0x1101, edit Bit 4 AMI/Zero Sup*.
P1.0.5	11/03	Made edits to ATM and CC sections and created PPP sections.
P1.0.6	4/04	Made edits to ATM and CC sections and created PPP sections.
P1.0.7	8/04	Made minor edits to CC section
P1.0.8	11/05	Made edits to CC section.
P2.0.0	10/10	Release of preliminary datasheet for DS3/E3 Clear-Channel Framer/LIU Combo device.

NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Datasheet October 2010.