

# RS8953B/8953SPB

## *HDSL Channel Unit*

The RS8953B is a High-Bit-Rate Digital Subscriber Line (HDSL) channel unit designed to perform data, clock, and format conversions necessary to construct a Pulse Code Multiplexed (PCM) channel from one, two, or three HDSL channels. The PCM channel consists of transmit and receive data, clock and frame sync signals configured for standard T1 (1544 kbps), standard E1 (2048 kbps), or custom (Nx64 kbps) formats. The PCM channel connects directly to a Bt8370 T1/E1 Controller or similar T1/E1 device. Connection to other network/subscriber physical layer devices is supported by the custom PCM frame format. Three identical HDSL channel interfaces consist of serial data and clock connected to a Bt8970 HDSL Transceiver or similar 2B1Q bit pump device. The RS8953SPB contains one HDSL channel interface.

Control and status registers are accessed via the Microprocessor Unit (MPU) interface. One common register group configures the PCM interface formatter, Pseudo-Random Bit Sequence (PRBS) generator, Bit Error Rate (BER) meter, timeslot router, Digital Phase Lock Loop (DPLL) clock recovery, and PCM Loopbacks (LB). Three groups of HDSL channel registers configure the elastic store FIFOs, overhead MUXes, receive framers, payload mappers, and HDSL loopbacks. Status registers monitor received overhead, DPLL, FIFO, and framer operations, including CRC and FEBE error counts.

The RS8953B adheres to Bellcore TA-NWT-001210 and FA-NWT-001211 and the latest ETSI RTR/TM-03036 standards. C-language software for all standard T1/E1 configuration and startup procedures is implemented on Conexant's HDSL Evaluation Module (Bt8973EVM) and is available under a no-fee license agreement. RS8953B software can also be developed for non-standard HDSL applications or to interoperate with existing HDSL equipment.

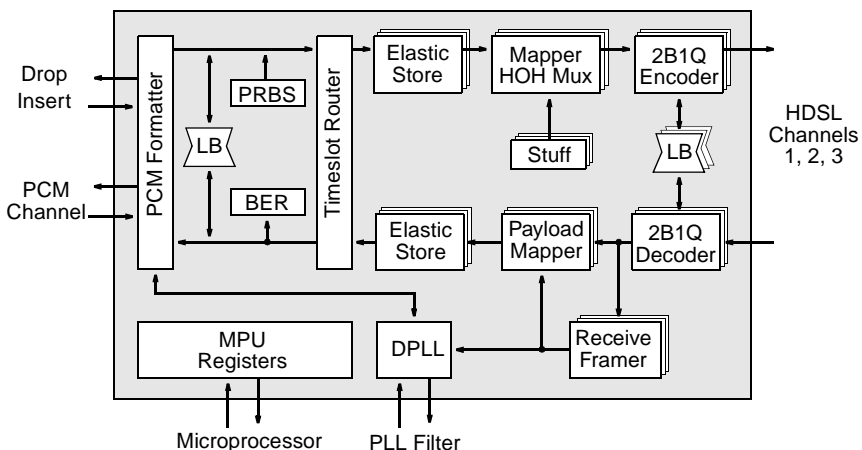
### Distinguishing Features

- Supports All HDSL Bit Rates
  - 2 pair T1 standard (784 kbps)
  - 2 pair E1 standard (1168 kbps)
  - 3 pair E1 standard (784 kbps)
  - 1/2/3 pair custom (Nx64 kbps, N=2-36)
- T1/E1 Primary Rate (PCM) Channel
  - Connects to Conexant E1/T1 Framers
  - Framed or unframed mode
  - Sync/Async payload mapping
  - Clock recovery/jitter attenuation
  - PRBS/fixed test patterns
  - BER measurement
- HDSL Channels
  - Connects to Conexant ZipWire Transceivers
  - Three independent serial channels
  - Central, remote, or repeater
  - Overhead (HOH) management
  - Programmable path delays
  - Error performance monitoring
  - Software controlled EOC and IND
  - Auxiliary payload/Z-bit data link
  - Master loop ID and interchange
  - Auto tip/ring reversal
- Programmable Data Routing
  - PCM timeslots – HDSL payload
  - Drop/Insert – HDSL payload
  - Auxiliary – HDSL payload
  - PRBS/Fixed – PCM or HDSL
  - PCM and HDSL loopbacks
- Intel® or Motorola® MPU interface
- CMOS technology, 3.3 V operation
- 68-pin PLCC or 80-pin PQFP

### Applications

- Full, Fractional or Multipoint T1/E1
- Single and Multichannel Repeaters
- Voice Pair Gain Systems
- Wireless LAN/PBX
- PCS, Cellular Base Station
- Fiber Access/Distribution
- Loop Carrier, Remote Switches
- Subscriber Line Modem

### Functional Block Diagram



## Ordering Information

Order Number	Package	Number of HDSL Channels	Operating Temperature Range
RS8953BEPF	80-Pin Plastic Quad Flat Pack (PQFP)	3	–40°C to +85°C
RS8953BEPJ	68-Pin Plastic Leaded Chip Carrier (PLCC)	3	–40°C to +85°C
RS8953SPB EPF	80-Pin Plastic Quad Flat Pack (PQFP)	1	–40°C to +85°C
RS8953SPB EPJ	68-Pin Plastic Leaded Chip Carrier (PLCC)	1	–40°C to +85°C

Information provided by Conexant Systems, Inc. (Conexant) is believed to be accurate and reliable. However, no responsibility is assumed by Conexant for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Conexant other than for circuitry embodied in Conexant products. Conexant reserves the right to change circuitry at any time without notice. This document is subject to change without notice.

Conexant and “What’s Next in Communications Technologies” are trademarks of Conexant Systems, Inc.

Product names or services listed in this publication are for identification purposes only, and may be trademarks or registered trademarks of their respective companies. All other marks mentioned herein are the property of their respective holders.

© 1999 Conexant Systems, Inc.  
Printed in U.S.A.  
All Rights Reserved

**Reader Response:** To improve the quality of our publications, we welcome your feedback. Please send comments or suggestions via e-mail to [Conexant Reader Response@conexant.com](mailto:Conexant.Reader.Response@conexant.com). Sorry, we can't answer your technical questions at this address. Please contact your local Conexant [sales office](#) or local field applications engineer if you have technical questions.

# Table of Contents

---

<b>List of Figures</b> .....	xi
<b>List of Tables</b> .....	xiii
<b>1.0 HDSL Systems</b> .....	1-1
<b>1.1 HTU Applications</b> .....	1-1
1.1.1 Repeaters .....	1-2
1.1.2 Fractional Transport .....	1-3
1.1.3 Switching Systems .....	1-4
1.1.4 Loop Carrier/Pair Gain .....	1-5
1.1.5 Point-to-Multipoint .....	1-6
1.1.6 Subscriber Modem .....	1-7
<b>1.2 System Interfaces</b> .....	1-8
<b>2.0 Pin Descriptions</b> .....	2-1
<b>2.1 Pin Assignments</b> .....	2-1
<b>2.2 Signal Definitions</b> .....	2-7
<b>3.0 Circuit Descriptions</b> .....	3-1
<b>3.1 MPU Interface</b> .....	3-1
3.1.1 Address/Data Bus .....	3-2
3.1.2 Bus Controls .....	3-2
3.1.3 Interrupt Request .....	3-3
3.1.4 Hardware Reset .....	3-3
<b>3.2 PCM Channel</b> .....	3-4
3.2.1 PCM Transmit .....	3-5
3.2.1.1 Transmit Synchronization .....	3-6
3.2.1.2 Transmit Routing Table .....	3-7
3.2.1.3 PRBS Generator .....	3-7
3.2.1.4 Drop/Insert Channel .....	3-8
3.2.1.5 TFIFO Water Levels .....	3-9
3.2.2 PCM Receive .....	3-10

3.2.2.1	Receive Synchronization . . . . .	3-12
3.2.2.2	Receive Combination Table . . . . .	3-12
3.2.2.3	BER Meter . . . . .	3-13
3.2.2.4	RFIFO Water Level. . . . .	3-14
<b>3.3</b>	<b>Clock Recovery DPLL . . . . .</b>	<b>3-15</b>
<b>3.4</b>	<b>Loopbacks. . . . .</b>	<b>3-17</b>
<b>3.5</b>	<b>HDSL Channel. . . . .</b>	<b>3-19</b>
3.5.1	HDSL Transmit. . . . .	3-24
3.5.1.1	Transmit Payload Mapper . . . . .	3-24
3.5.1.2	HOH Multiplexer . . . . .	3-24
3.5.1.3	CRC Calculation . . . . .	3-25
3.5.1.4	Scrambler. . . . .	3-25
3.5.1.5	STUFF Generator. . . . .	3-25
3.5.1.6	2B1Q Encoder. . . . .	3-26
3.5.1.7	HDSL Auxiliary Transmit . . . . .	3-27
3.5.2	HDSL Receive . . . . .	3-28
3.5.2.1	2B1Q Decoder. . . . .	3-28
3.5.2.2	HDSL Receive Framer . . . . .	3-29
3.5.2.3	Descrambler . . . . .	3-31
3.5.2.4	CRC Checking . . . . .	3-32
3.5.2.5	HOH Demux . . . . .	3-32
3.5.2.6	Receive Payload Mapper . . . . .	3-32
3.5.2.7	HDSL Auxiliary Receive . . . . .	3-32
<b>3.6</b>	<b>PRA Function . . . . .</b>	<b>3-34</b>
3.6.1	Transferring Data from HDSL to RSER . . . . .	3-34
3.6.2	Definitions of Detection Algorithms. . . . .	3-35
3.6.3	Inserting Data Transferred from HDSL to RSER. . . . .	3-35
3.6.4	Transferring Data from TSER to HDSL. . . . .	3-35
3.6.5	Inserting Data Transferred from TSER to HDSL . . . . .	3-36
<b>4.0</b>	<b>Registers . . . . .</b>	<b>5-1</b>
<b>4.1</b>	<b>Address Map . . . . .</b>	<b>5-3</b>
<b>4.2</b>	<b>HDSL Transmit . . . . .</b>	<b>5-9</b>

0x00—Transmit Embedded Operations Channel (TEOC_LO) . . . . .	5-10
0x01—Transmit Embedded Operations Channel (TEOC_HI) . . . . .	5-10
0x02—Transmit Indicator Bits (TIND_LO) . . . . .	5-10
0x03—Transmit Indicator Bits (TIND_HI) . . . . .	5-10
0x04—Transmit Z-Bits (TZBIT_1) . . . . .	5-10
0xDF—Transmit Z-Bits (TZBIT_2) . . . . .	5-11
0xE0—Transmit Z-Bits (TZBIT_3) . . . . .	5-11
0xE1—Transmit Z-Bits (TZBIT_4) . . . . .	5-11
0xE2—Transmit Z-Bits (TZBIT_5) . . . . .	5-11
0xE3—Transmit Z-Bits (TZBIT_6) . . . . .	5-11
0x05—Transmit FIFO Water Level (TFIFO_WL) . . . . .	5-12
0x06—Transmit Command Register 1 (TCMD_1) . . . . .	5-12
0x07—Transmit Command Register 2 (TCMD_2) . . . . .	5-13
<b>4.3 Transmit Payload Mapper . . . . .</b>	<b>5-15</b>
0x08—Transmit Payload Map (TMAP_1) . . . . .	5-15
0x09—Transmit Payload Map (TMAP_2) . . . . .	5-15
0x0A—Transmit Payload Map (TMAP_3) . . . . .	5-15
0x0B—Transmit Payload Map (TMAP_4) . . . . .	5-15
0x0C—Transmit Payload Map (TMAP_5) . . . . .	5-15
0x0F—Transmit Payload Map (TMAP_6) . . . . .	5-16
0x10—Transmit Payload Map (TMAP_7) . . . . .	5-16
0x11—Transmit Payload Map (TMAP_8) . . . . .	5-16
0x12—Transmit Payload Map (TMAP_9) . . . . .	5-16
0x0D—Transmit FIFO Reset (TFIFO_RST) . . . . .	5-17
0x0E—Scrambler Reset (SCR_RST) . . . . .	5-17
<b>4.4 HDSL Receive . . . . .</b>	<b>5-18</b>
0x60—Receive Command Register 1 (RCMD_1) . . . . .	5-19
0x61—Receive Command Register 2 (RCMD_2) . . . . .	5-20
0x62—Receive Elastic Store FIFO Reset (RFIFO_RST) . . . . .	5-21
0x63—Receive Framing Synchronization Reset (SYNC_RST) . . . . .	5-21
<b>4.5 Receive Payload Mapper . . . . .</b>	<b>5-22</b>
0x64—Receive Payload Map (RMAP_1) . . . . .	5-22
0x65—Receive Payload Map (RMAP_2) . . . . .	5-22
0x66—Receive Payload Map (RMAP_3) . . . . .	5-22
0x69—Receive Payload Map (RMAP_4) . . . . .	5-22
0x6A—Receive Payload Map (RMAP_5) . . . . .	5-22
0x6B—Receive Payload Map (RMAP_6) . . . . .	5-23
0x67—Error Count Reset (ERR_RST) . . . . .	5-23
0x68—Receive Signaling Location (RSIG_LOC) . . . . .	5-23
<b>4.6 PCM Formatter . . . . .</b>	<b>5-25</b>
0xC0—TSER Frame Bit Location (TFRAME_LOC_LO) . . . . .	5-26
0xC1—TSER Frame Bit Location (TFRAME_LOC_HI) . . . . .	5-26

0xC2—TSER Multiframe Bit Location (TMF_LOC) . . . . .	5-26
0xC3—RSER Frame Bit Location (RFRAME_LOC_LO) . . . . .	5-26
0xC4—RSER Frame Bit Location (RFRAME_LOC_HI) . . . . .	5-27
0xC5—RSER Multiframe Bit Location (RMF_LOC) . . . . .	5-27
0xC6—PCM Multiframe Length (MF_LEN) . . . . .	5-27
0xC7—PCM Multiframes per HDSL Frame (MF_CNT) . . . . .	5-28
0xC8—PCM Frame Length (FRAME_LEN_LO) . . . . .	5-28
0xC9—PCM Frame Length (FRAME_LEN_HI) . . . . .	5-28
<b>4.7 HDSL Channel Configuration . . . . .</b>	<b>5-29</b>
0xCA—HDSL Frame Length (HFRAME_LEN_LO) . . . . .	5-29
0xF5—HDSL Frame Length (HFRAME_LEN_HI) . . . . .	5-30
0xF8—HDSL Frame Length (HFRAME2_LEN_LO) . . . . .	5-30
0xF9—HDSL Frame Length (HFRAME2_LEN_HI) . . . . .	5-30
0xFA—HDSL Frame Length (HFRAME3_LEN_LO) . . . . .	5-30
0xFB—HDSL Frame Length (HFRAME3_LEN_HI) . . . . .	5-31
0xCB—SYNC Word A (SYNC_WORD_A) . . . . .	5-31
0xCC—SYNC Word B (SYNC_WORD_B) . . . . .	5-31
0xCD—RX FIFO Water Level (RFIFO_WL_LO) . . . . .	5-31
0xCE—RX FIFO Water Level (RFIFO_WL_HI) . . . . .	5-32
<b>4.8 Transmit Bit Stuffing Thresholds . . . . .</b>	<b>5-32</b>
0xCF—Bit Stuffing Threshold A (STF_THRESH_A_LO) . . . . .	5-33
0xD0—Bit Stuffing Threshold A (STF_THRESH_A_HI) . . . . .	5-33
0xD1—Bit Stuffing Threshold B (STF_THRESH_B_LO) . . . . .	5-33
0xD2—Bit Stuffing Threshold B (STF_THRESH_B_HI) . . . . .	5-33
0xD3—Bit Stuffing Threshold C (STF_THRESH_C_LO) . . . . .	5-33
0xD4—Bit Stuffing Threshold C (STF_THRESH_C_HI) . . . . .	5-34
<b>4.9 DPLL Configuration . . . . .</b>	<b>5-35</b>
0xD5—DPLL Residual (DPLL_RESID_LO) . . . . .	5-35
0xD6—DPLL Residual (DPLL_RESID_HI) . . . . .	5-36
0xD7—DPLL Factor (DPLL_FACTOR) . . . . .	5-36
0xD8—DPLL Gain (DPLL_GAIN) . . . . .	5-37
0xDB—DPLL Phase Detector Init (DPLL_PINI) . . . . .	5-38
0xF6—Reset DPLL Phase Detector (DPLL_RST) . . . . .	5-38
<b>4.10 Data Path Options . . . . .</b>	<b>5-39</b>
0xDC—Data Bank Pattern 1 (DBANK_1) . . . . .	5-39
0xDD—Data Bank Pattern 2 (DBANK_2) . . . . .	5-39
0xDE—Data Bank Pattern 3 (DBANK_3) . . . . .	5-40
0xEA—Fill Pattern (FILL_PATT) . . . . .	5-40
0xE4—Transmit Stuff Bit Value (TSTUFF) . . . . .	5-40
0xED—Transmit Routing Table (ROUTE_TBL) . . . . .	5-41
0xEE—Receive Combination Table (COMBINE_TBL) . . . . .	5-42
0xF2—Receive Signaling Table (RSIG_TBL) . . . . .	5-43
<b>4.11 Common Command . . . . .</b>	<b>5-44</b>
0xE5—Command Register 1 (CMD_1) . . . . .	5-44
0xE6—Command Register 2 (CMD_2) . . . . .	5-45

0xE7—Command Register 3 (CMD_3) .....	5-46
0xE8—Command Register 4 (CMD_4) .....	5-47
0xE9—Command Register 5 (CMD_5) .....	5-47
0xF3—Command Register 6 (CMD_6) .....	5-48
0xF4—Command Register 7 (CMD_7) .....	5-49
<b>4.12 Interrupt and Reset</b> .....	5-51
0xEB—Interrupt Mask Register (IMR) .....	5-51
0xEC—Interrupt Clear Register (ICR) .....	5-51
0xEF—Reset BER Meter/Start BER Measurement (BER_RST) .....	5-52
0xF0—Reset PRBS Generator (PRBS_RST) .....	5-52
0xF1—Reset Receiver (RX_RST) .....	5-52
<b>4.13 Receive/Transmit Status</b> .....	5-53
0x00—Receive Embedded Operations Channel (REOC_LO) .....	5-53
0x01—Receive Embedded Operations Channel (REOC_HI) .....	5-54
0x02—Receive Indicator Bits (RIND_LO) .....	5-54
0x03—Receive Indicator Bits (RIND_HI) .....	5-54
0x04—Receive Z-Bits (RZBIT_1) .....	5-54
0x18—Receive Z-Bits (RZBIT_2) .....	5-55
0x19—Receive Z-Bits (RZBIT_3) .....	5-55
0x1A—Receive Z-Bits (RZBIT_4) .....	5-55
0x1B—Receive Z-Bits (RZBIT_5) .....	5-55
0x1C—Receive Z-Bits (RZBIT_6) .....	5-55
0x05—Receive Status 1 (STATUS_1) .....	5-55
0x06—Receive Status 2 (STATUS_2) .....	5-57
0x07—Transmit Status (STATUS_3) .....	5-58
0x21—CRC Error Count (CRC_CNT) .....	5-58
0x22—Far End Block Error Count (FEBE_CNT) .....	5-59
<b>4.14 Common Status</b> .....	5-60
0x1D—Bit Error Rate Meter (BER_METER) .....	5-60
0x1E—BER Status (BER_STATUS) .....	5-61
0x1F—Interrupt Request Register (IRR) .....	5-61
0x28—DPLL Residual Output (RESID_OUT_LO) .....	5-62
0x20—DPLL Residual Output (RESID_OUT_HI) .....	5-62
0x30—Interrupt Mask Register (IMR) .....	5-62
0x38—DPLL Phase Error (PHS_ERR) .....	5-63
0x39—Multiframe Sync Phase Low (MSYNC_PHS_LO) .....	5-63
0x3A—Multiframe Sync Phase High (MSYNC_PHS_HI) .....	5-63
0x3B—Shadow Write (SHADOW_WR) .....	5-64
0x3C—Error Status (ERR_STATUS) .....	5-64
<b>4.15 PRA Transmit Read</b> .....	5-65
0x40—PRA Transmit Control Register 0 (TX_PRA_CTRL0) .....	5-65
0x41—PRA Transmit Control Register 1 (TX_PRA_CTRL1) .....	5-66

0x42—PRA Transmit Monitor Register 1 (TX_PRA_MON1) .....	5-67
0x43—PRA Transmit E-Bits Counter (TX_PRA_E_CNT) .....	5-67
0x45—PRA Transmit In-Band Code (TX_PRA_CODE) .....	5-67
0x46—PRA Transmit Monitor Register 0 (TX_PRA_MON0) .....	5-68
0x47—PRA Transmit Monitor Register 2 (TX_PRA_MON2) .....	5-68
<b>4.16 PRA Transmit Write</b> .....	5-69
0x70—PRA Transmit Control Register 0 (TX_PRA_CTRL0) .....	5-69
0x71—PRA Transmit Control Register 1 (TX_PRA_CTRL1) .....	5-70
0x72—PRA Transmit Bits Buffer 1 (TX_BITS_BUFF1) .....	5-71
0x73—PRA Transmit TMSYNC offset Register (TX_PRA_TMSYNC_OFFSET) .....	5-71
0x74—PRA Transmit Bits Buffer 0 (TX_BITS_BUFF0) .....	5-72
<b>4.17 PRA Receive Read</b> .....	5-73
0x80—PRA Receive Control Register 0 (RX_PRA_CTRL0) .....	5-73
0x81—PRA Receive Control Register 1 (RX_PRA_CTRL1) .....	5-74
0x82—PRA Receive Monitor Register 1 (RX_PRA_MON1) .....	5-75
0x83—PRA Receive E bits Counter (RX_PRA_E_CNT) .....	5-75
0x84—PRA Receive CRC4 Errors Counter (RX_PRA_CRC_CNT) .....	5-75
0x85—PRA Receive In-Band Code (RX_PRA_CODE) .....	5-76
0x86—PRA Receive Monitor Register 0 (RX_PRA_MON0) .....	5-76
0x87—PRA Receive Monitor Register 2 (RX_PRA_MON2) .....	5-76
<b>4.18 PRA Receive Write</b> .....	5-77
0xB0—PRA Receive Control Register 0 (RX_PRA_CTRL0) .....	5-77
0xB1—PRA Receive Control Register 1 (RX_PRA_CTRL1) .....	5-78
0xB2—PRA Receive Bits Buffer 1 (RX_BITS_BUFF1) .....	5-79
0xB4—PRA Receive Bits Buffer 0 (RX_BITS_BUFF0) .....	5-79
<b>5.0 Applications</b> .....	6-1
5.1 Interfacing to a Conexant HDSL Transceiver .....	6-2
5.2 Interfacing to the Bt8370 E1/T1 Framer .....	6-3
5.3 Interfacing to the 68302 Processor .....	6-4
5.4 Interfacing to the 8051 Controller .....	6-5
5.5 References .....	6-6
<b>6.0 Electrical and Timing Specifications</b> .....	7-1
6.1 Absolute Maximum Ratings .....	7-1
6.1.1 Recommended Operating Conditions .....	7-1
6.1.2 Electrical Characteristics .....	7-2
6.1.3 Timing Requirements .....	7-3
6.1.4 Switching Characteristics .....	7-5
6.1.5 MPU Interface Timing .....	7-7
6.2 Mechanical Specifications .....	7-10
<b>7.0 Acronyms, Abbreviations and Notation</b> .....	8-1
7.1 Arithmetic Notation .....	8-1



*HDSL Channel Unit*

7.1.1	Bit Numbering .....	8-1
7.1.2	Acronyms and Abbreviations .....	8-1
<b>Appendix A .....</b>		<b>A-1</b>
A.1	<b>Differences Between Bt8953A and RS8953B .....</b>	<b>A-1</b>
<b>Appendix B: Bt8953A/RS8953B Product Bulletin .....</b>		<b>B-1</b>
B.1	<b>BCLK Phase Constraints In Repeater Mode; Non-Conformance Product Affected: Bt8953A and RS8953B .....</b>	<b>B-1</b>



## List of Figures

Figure 1-1.	HTU Block Diagram . . . . .	1-2
Figure 1-2.	Repeater System Block Diagram . . . . .	1-2
Figure 1-3.	Drop/Insert System Block Diagram . . . . .	1-3
Figure 1-4.	Switch/Mux System Block Diagram . . . . .	1-4
Figure 1-5.	Voice (Pairgain/Cellular/PCS) System Block Diagram . . . . .	1-5
Figure 1-6.	Point-to-Multipoint (Fractional) System Block Diagram . . . . .	1-6
Figure 1-7.	Subscriber Modem (Terminal) System Block Diagram . . . . .	1-7
Figure 1-8.	RS8953B System Interfaces . . . . .	1-8
Figure 2-1.	Three-Pair PLCC Pin Assignments . . . . .	2-2
Figure 2-2.	Single-Pair PLCC Pin Assignments . . . . .	2-3
Figure 2-3.	Three-Pair PQFP Pin Assignments . . . . .	2-4
Figure 2-4.	Single-Pair PQFP Pin Assignments . . . . .	2-5
Figure 3-1.	MPU Bus Control Logic . . . . .	3-1
Figure 3-2.	MPU Interrupt Logic . . . . .	3-2
Figure 3-3.	PCM Channel Block Diagram . . . . .	3-4
Figure 3-4.	PCM Transmit Block Diagram . . . . .	3-5
Figure 3-5.	PCM Transmit Sync Timing . . . . .	3-6
Figure 3-6.	PCM Transmit Data Timing . . . . .	3-6
Figure 3-7.	Drop/Insert Channel Timing . . . . .	3-8
Figure 3-8.	TFIFO Water Level Timing . . . . .	3-9
Figure 3-9.	PCM Receive Block Diagram . . . . .	3-10
Figure 3-10.	PCM Receive Data Timing . . . . .	3-11
Figure 3-11.	PCM Receive Sync Timing . . . . .	3-12
Figure 3-12.	PRBS/BER Measurements . . . . .	3-13
Figure 3-13.	RFIFO Water Level Timing . . . . .	3-14
Figure 3-14.	DPLL Block Diagram . . . . .	3-15
Figure 3-15.	PCM and HDSL Loopbacks . . . . .	3-17
Figure 3-16.	2T1 Frame Format . . . . .	3-21
Figure 3-17.	2E1 Frame Format . . . . .	3-22
Figure 3-18.	3E1 Frame Format . . . . .	3-22
Figure 3-19.	HDSL Transmitter Block Diagram . . . . .	3-24
Figure 3-20.	HDSL Auxiliary Channel Payload Timing . . . . .	3-27
Figure 3-21.	HDSL Auxiliary Channel Z-bit Timing . . . . .	3-27
Figure 3-22.	HDSL Receiver Block Diagram . . . . .	3-28
Figure 3-23.	HDSL Receive Framer Synchronization . . . . .	3-30
Figure 3-24.	Threshold Correlation Effect on Expected Sync Locations . . . . .	3-31
Figure 3-25.	HDSL Auxiliary Receive Payload Timing . . . . .	3-32
Figure 3-26.	HDSL Auxiliary Receive Z-bit Timing . . . . .	3-33

Figure 3-27.	An Overview of the PRA Transfer of Data . . . . .	3-34
Figure 4-1.	Transmit Routing . . . . .	5-41
Figure 5-1.	RS8953B HDSL Channel Unit to Conexant HDSL Transceiver Interconnection . . . . .	6-2
Figure 5-2.	RS8953B HDSL Channel Unit to Bt8360 DS1 Framer Interconnection . . . . .	6-3
Figure 5-3.	RS8953B to 68302 Processor Interconnection . . . . .	6-4
Figure 5-4.	RS8953B HDSL Channel Unit to 8051 Controller Interconnection . . . . .	6-5
Figure 6-1.	Input Clock Timing . . . . .	7-3
Figure 6-2.	Input Setup and Hold Timing. . . . .	7-3
Figure 6-3.	Output Clock and Data Timing. . . . .	7-5
Figure 6-4.	MPU Write Timing, MPUSEL = 1 . . . . .	7-8
Figure 6-5.	MPU Read Timing, MPUSEL = 1 . . . . .	7-8
Figure 6-6.	MPU Write Timing, MPUSEL = 0 . . . . .	7-9
Figure 6-7.	MPU Read Timing, MPUSEL = 0 . . . . .	7-9
Figure 6-8.	68-Pin PLCC Package Drawing . . . . .	7-10
Figure 6-9.	80-Pin PQFP Mechanical Specification . . . . .	7-11

## List of Tables

Table 2-1.	Pin Assignments . . . . .	2-5
Table 2-2.	Signal Definitions . . . . .	2-7
Table 3-1.	PCM And HDSL Loopbacks . . . . .	3-18
Table 3-2.	HDSL Frame Structure and Overhead Bit Allocation . . . . .	3-20
Table 3-3.	HDSL Frame Mapping Examples . . . . .	3-23
Table 3-4.	2B1Q Encoder Alignment . . . . .	3-26
Table 3-5.	Z-Bit Definitions . . . . .	3-26
Table 3-6.	2B1Q Decoder Alignment . . . . .	3-28
Table 4-1.	Register Summary Address Map . . . . .	5-3
Table 4-2.	HDSL Transmit Write Registers . . . . .	5-9
Table 4-3.	HDSL Receive Write Registers . . . . .	5-18
Table 4-4.	PCM Formatter Write Registers . . . . .	5-25
Table 4-5.	HDSL Channel Configuration Write Registers . . . . .	5-29
Table 4-6.	DPLL Configuration Write Registers . . . . .	5-35
Table 4-7.	Data Path Options Write Registers . . . . .	5-39
Table 4-8.	Common Command Write Registers . . . . .	5-44
Table 4-9.	Interrupt and Reset Write Registers . . . . .	5-51
Table 4-10.	Receive and Transmit Status Read Registers . . . . .	5-53
Table 4-11.	Common Status Read Registers . . . . .	5-60
Table 4-12.	PRA Transmit Read Registers . . . . .	5-65
Table 4-13.	PRA Transmit Write Registers . . . . .	5-69
Table 4-14.	PRA Receive Read Registers . . . . .	5-73
Table 4-15.	PRA Receive Write Registers . . . . .	5-77
Table 6-1.	Absolute Maximum Ratings . . . . .	7-1
Table 6-2.	Recommended Operating Conditions . . . . .	7-1
Table 6-3.	Electrical Characteristics . . . . .	7-2
Table 6-4.	Clock Timing Requirements . . . . .	7-3
Table 6-5.	Data Timing Requirements . . . . .	7-4
Table 6-6.	Input Clock Edge Selection . . . . .	7-4
Table 6-7.	Clock and Data Switching Characteristics . . . . .	7-5
Table 6-8.	Output Clock Edge Selection . . . . .	7-6
Table 6-9.	MPU Interface Timing Requirements . . . . .	7-7
Table 6-10.	MPU Interface Switching Characteristics . . . . .	7-7
Table A-1.	Pin Definitions . . . . .	A-1
Table A-2.	Power Consumption . . . . .	A-2



# 1.0 HDSL Systems

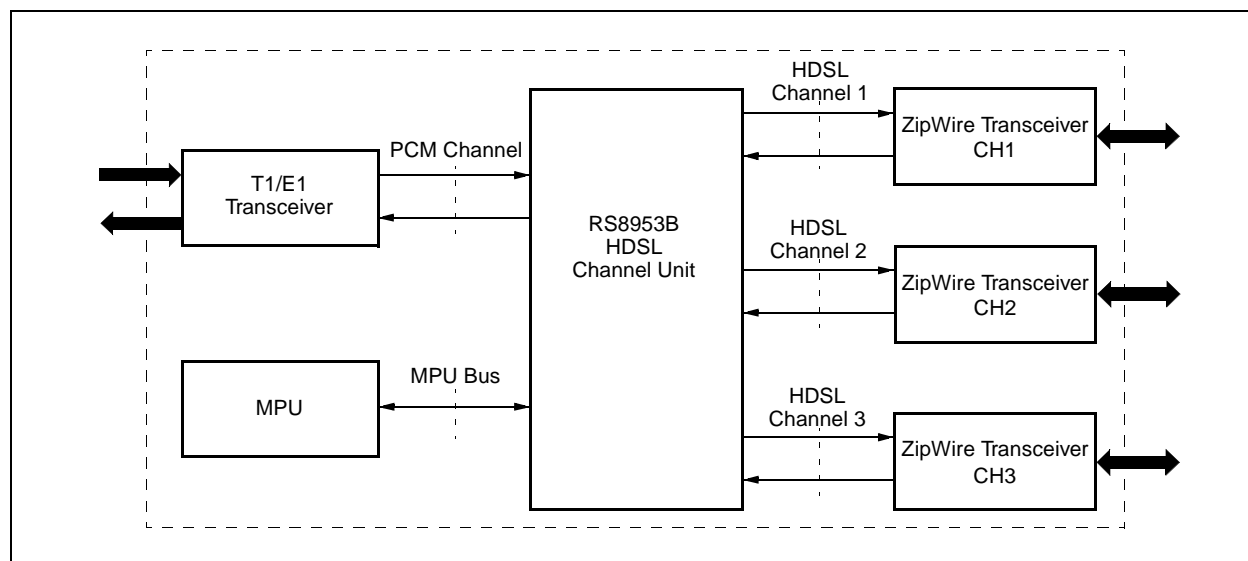
---

## 1.1 HTU Applications

The High-Bit-Rate Digital Subscriber Line (HDSL) is a simultaneous full-duplex transmission scheme which uses twisted-pair wire cables as the physical medium to transport signals between standard types of network or subscriber communication interfaces. A complete HDSL system consists of two pieces of terminal equipment connected by 1, 2, or 3 wire pairs. Each HDSL Terminal Unit (HTU) translates standard interface signals into HDSL payload for transmission, and reconstructs the standard interface from received payload. Bellcore standards define a 1.544 Mbps T1 transport application that uses two HDSL wire pairs (2T1), each operating at 784 kpbs. ETSI standards define a 2.048 Mbps E1 transport application using either two wire pairs (2E1), each operating at 1168 kpbs, or three wire pairs (3E1), each operating at 784 kpbs.

Figure 1-1 illustrates how an HDSL Terminal Unit (HTU) transports standard T1/E1 signals. T1/E1 transceivers convert T1/E1 interface signals into a Pulse Code Multiplexed (PCM) channel of clock, serial data, and optional frame sync. ZipWire transceivers convert 2B1Q line signals to HDSL channels of clock, serial data, and quat sync. The RS8953B translates between PCM and HDSL by performing PCM timeslot and HDSL payload routing, data scrambling and descrambling, overhead insertion and extraction, clock synchronization and clock synthesis. The Microprocessor Unit (MPU) configures devices for the intended application, manages overhead protocol, and monitors real-time performance.

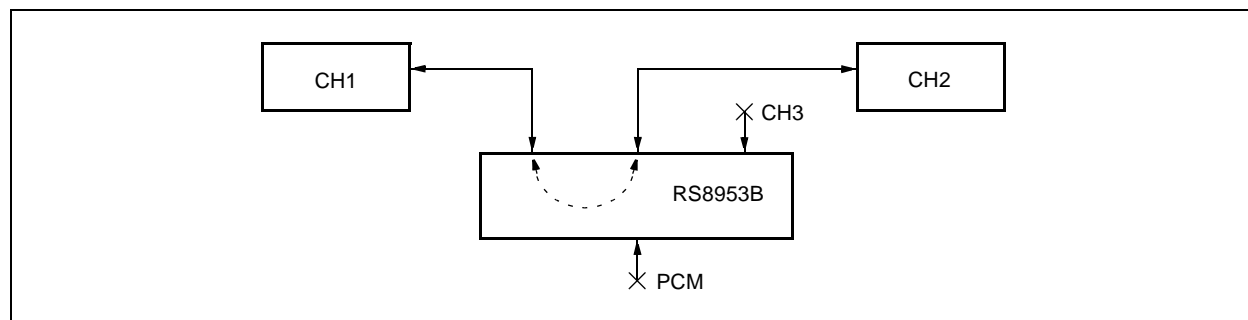
**Figure 1-1. HTU Block Diagram**



### 1.1.1 Repeaters

Figure 1-2 shows single pair repeaters placed in line between HDSL terminals to extend transmission distance. RS8953B provides an internal cross-connect path between HDSL channels 1 and 2 to support single pair repeaters.

**Figure 1-2. Repeater System Block Diagram**

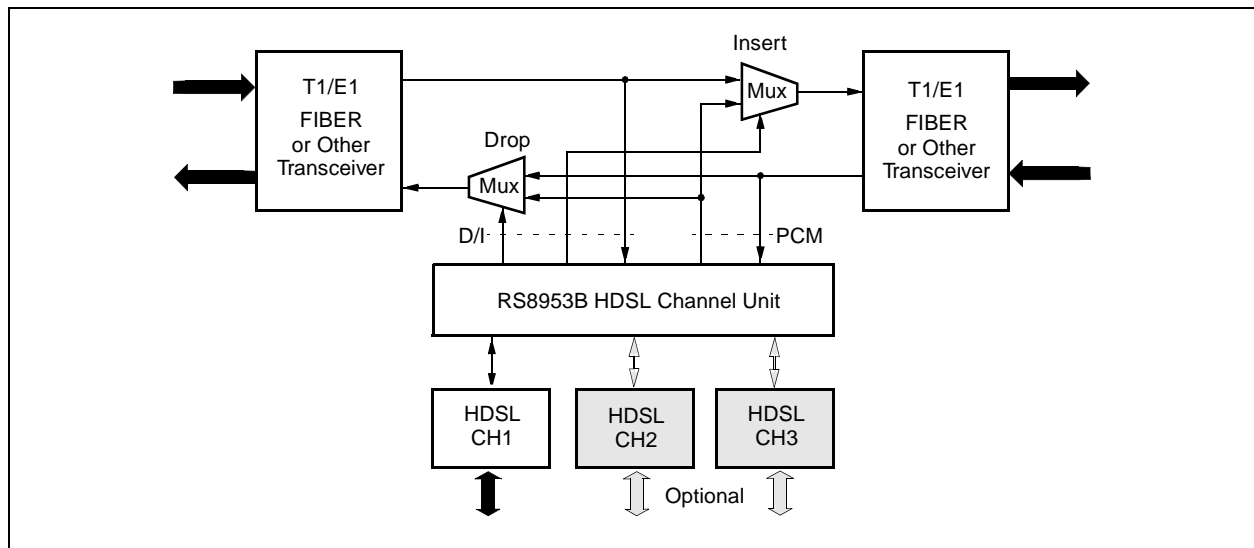




## 1.1.2 Fractional Transport

Figure 1-3 illustrates a drop/insert application where only a portion of the PCM channel bandwidth is transported over one or more HDSL wire pairs. The RS8953B provides drop/insert indicator signals to control external data MUXes and internal routing tables to map timeslots from either one of two synchronized PCM data sources. For remote terminals using partial payloads, the PCM channel may be configured to operate either at the standard interface rate or at the Nx64 effective payload rate.

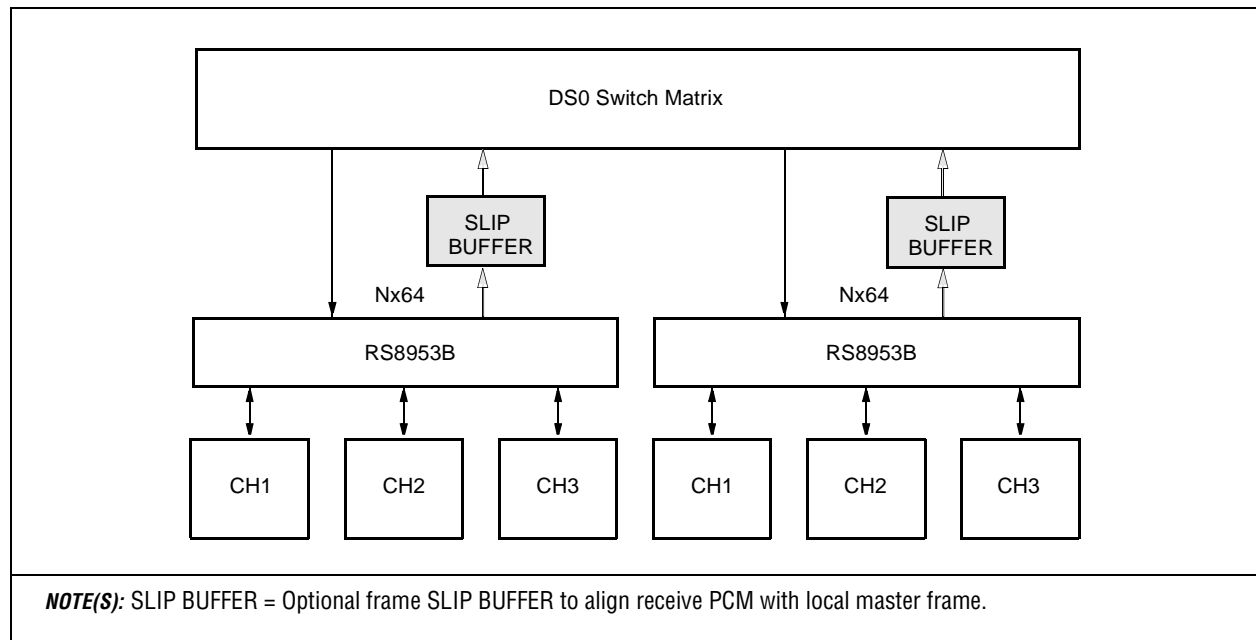
**Figure 1-3. Drop/Insert System Block Diagram**



### 1.1.3 Switching Systems

Figure 1-4 illustrates how the RS8953B is incorporated into a digital switch or multiplexer system that uses multiple HDSL lines to transport Nx64 or standard T1/E1 applications. The RS8953B's PCM timeslot router contains 64 table entries that extends the maximum PCM channel rate to 64x64 or 4.096 Mbps. RS8953B allows PCM channels at the central office (CO) and remote ends to operate at different rates. For example, the PCM channel in a digital switch may connect to a 4.096 Mbps shelf bus, while the remote terminal connects to a T1/E1 standard PCM channel.

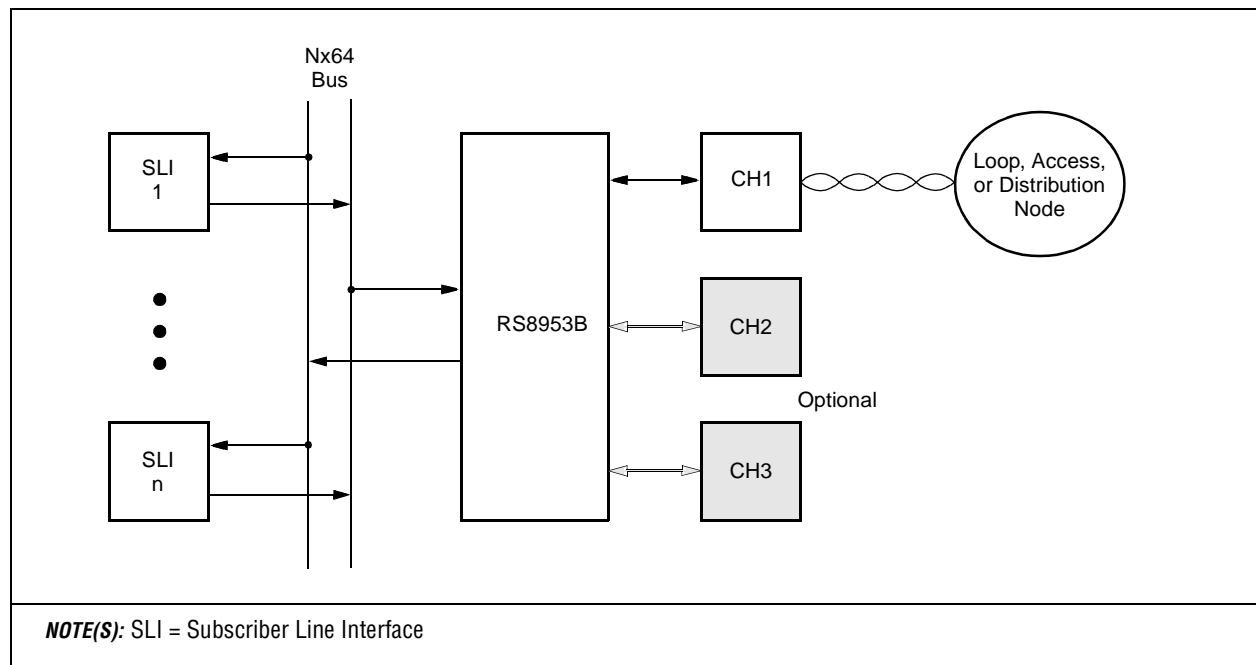
**Figure 1-4. Switch/Mux System Block Diagram**



### 1.1.4 Loop Carrier/Pair Gain

Figure 1-5 shows a channel bank application where the PCM channel connects a bank of voice and/or data subscriber line interfaces using an Nx64 bus. The total number of subscriber lines determines the PCM channel rate and determines how many HDSL wire pairs are needed to transport the application up to the digital loop carrier, cellular base station, network distribution element, or to the private branch exchange. The RS8953B supplies the PCM frame sync reference and acts as the PCM bus master for the remote channel bank. The RS8953B's Digital Phase Locked Loop (DPLL) clock recovery allows PCM channel rates down to 2x64 or 128 kpbs. Unpopulated PCM timeslots or HDSL payload bytes can be replaced by an 8-bit programmable fixed pattern, or one of four Pseudo-Random Bit Sequence (PRBS) patterns.

**Figure 1-5. Voice (Pairgain/Cellular/PCS) System Block Diagram**

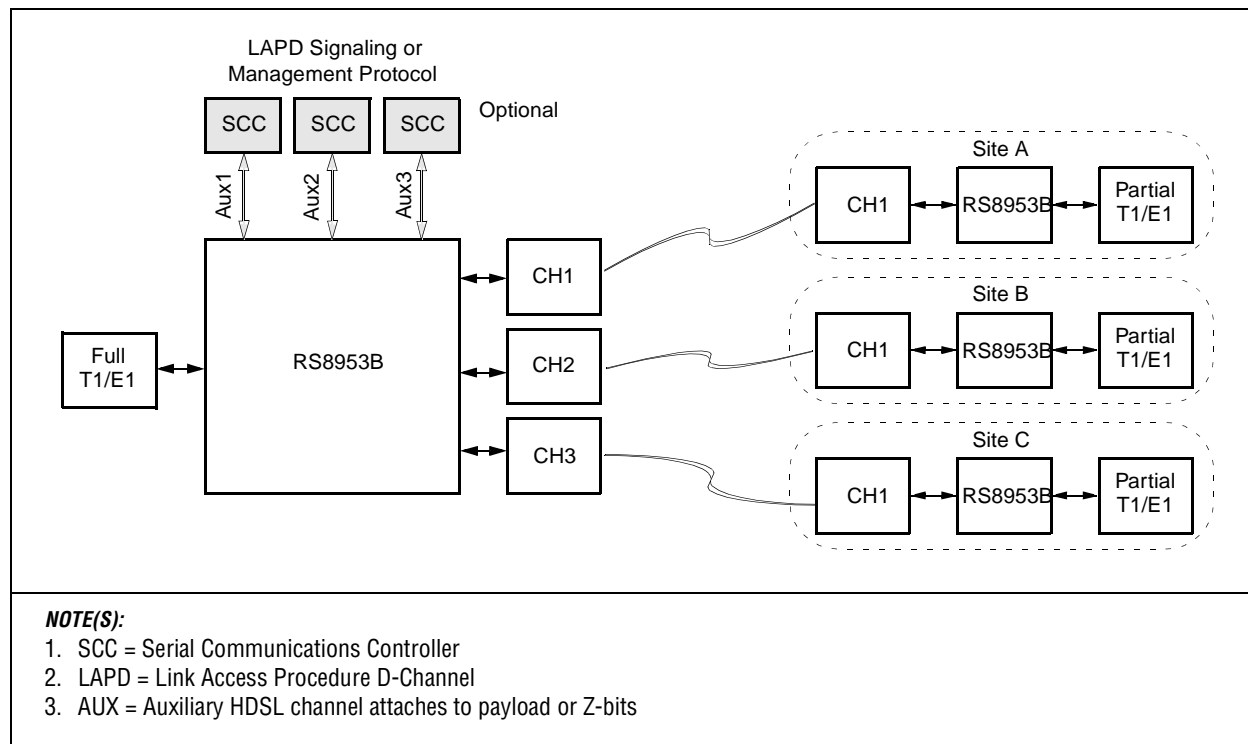


### 1.1.5 Point-to-Multipoint

Figure 1-6 shows fractional T1/E1 services delivered from the CO to multiple remote sites in a Point-to-Multipoint (P2MP) application. The number of HDSL wire pairs and PCM channel rates at each site is variable. The RS8953B provides the ability to measure and compensate for misalignment between separate PCM frame syncs coming from each remote site. By programming transmit delays from PCM to HDSL frame syncs, each remote site can send its HDSL frames back to the central office. The HDSL frames are then sufficiently aligned with the others to be reconstructed into a single PCM frame at the central site. The RS8953B accommodates large differential delays associated with the P2MP application. It receives HDSL frame offsets to groom Channel Associated Signaling (CAS) from different sites.

P2MP applications of primary rate ISDN transport are also supported, where different LAPD channels are received from each remote site. The RS8953B provides auxiliary HDSL channel inputs and outputs for the system to externally insert and monitor transmitted or received HDSL payload bytes. Auxiliary HDSL channels may alternately be configured to terminate the last 40 Z-bits through an external data link controller.

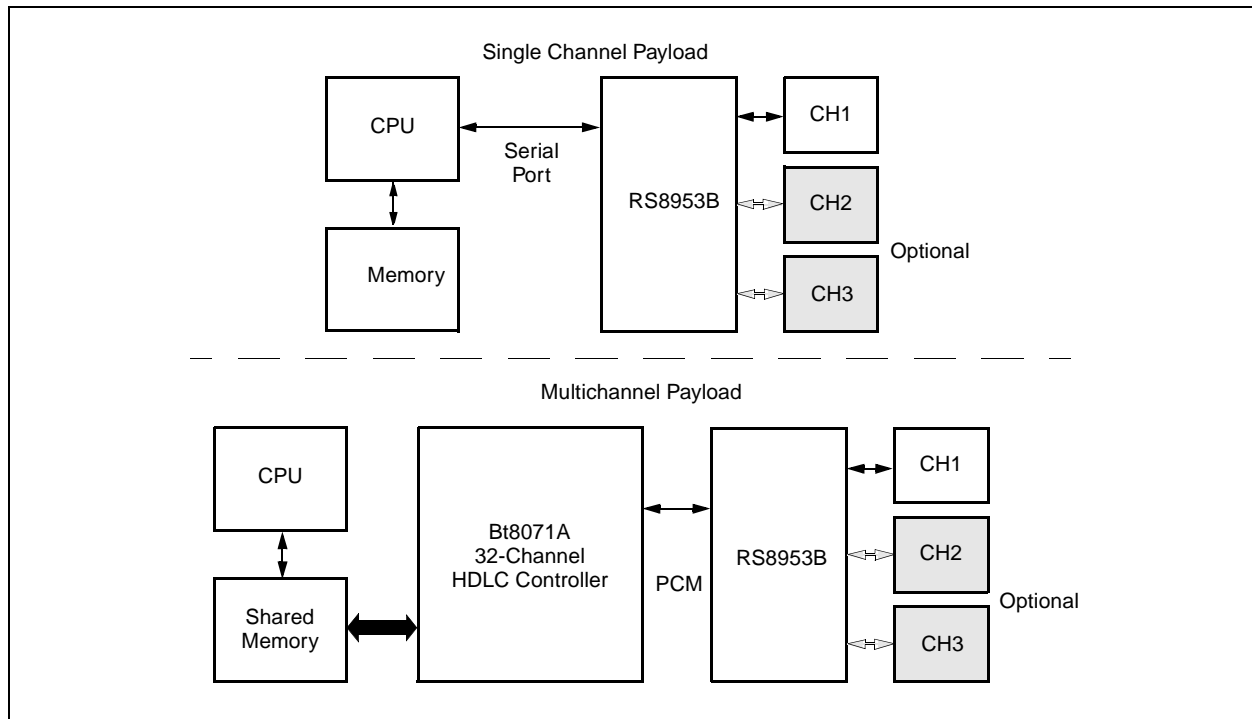
**Figure 1-6. Point-to-Multipoint (Fractional) System Block Diagram**



### 1.1.6 Subscriber Modem

Figure 1-7 shows an HDSL data modem application where a CPU processor delivers PCM data directly to the RS8953B. Alternately, a multichannel communications controller such as the Bt8071A can be used to manage the transfer of data between the CPU and PCM channel through a local shared memory.

**Figure 1-7. Subscriber Modem (Terminal) System Block Diagram**

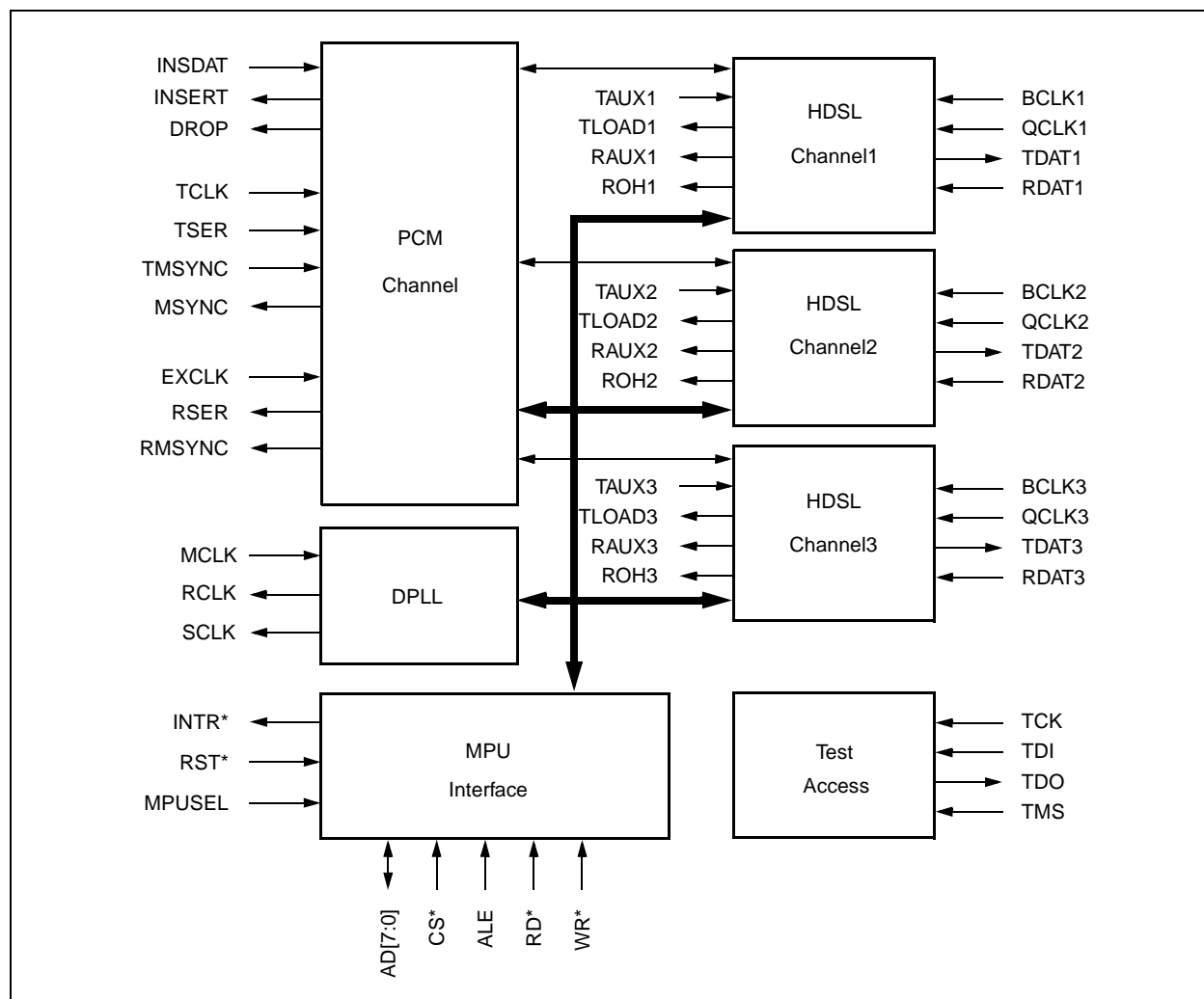


## 1.2 System Interfaces

System interfaces and associated signals for the RS8953B functional circuit blocks are shown in Figure 1-8. Circuit blocks are described in sections 3 and 4, and signals are defined in Table 2-2.

The single-pair version (RS8953SPBEPF and RS8953SPBEPJ) only supports HDSL Channel 1. HDSL Channels 2 and 3 are not usable. Although only 1 HDSL channel is usable, the internal registers are not changed from the 3 HDSL channel versions. The single-pair versions (RS8953SPBEPF and RS8953SPBEPJ) only supports HDSL Channel 1. HDSL Channels 2 and 3 are not usable. Although only 1 HDSL channel is usable, the internal registers are not changed from the 3 HDSL channel versions. This means that the registers should be programmed with the same value as if only HDSL channel 1 was used in a 3 channel version. This allows the 3 channel version to be used for development, and without a software change, a single-pair version used for production.

Figure 1-8. RS8953B System Interfaces



## 2.0 Pin Descriptions

---

### 2.1 Pin Assignments

The RS8953B pin assignments for the 68-pin Plastic Leaded Chip Carrier (PLCC) package are shown in [Figure 2-1](#) and [Figure 2-2](#). The RS8953B pin assignments for the 80-pin Plastic Quad Flat Pack (PQFP) are shown in [Figure 2-3](#) and [Figure 2-4](#). The pinouts for RS8953B packages are listed in [Table 2-1](#) and defined in [Table 2-2](#). The input/output (I/O) column in [Table 2-1](#) is coded as follows:

I = Input, O = Output, I/O = Bidirectional, VCC = Power, GND = Ground, and NC = No Connection.

Figure 2-1. Three-Pair PLCC Pin Assignments

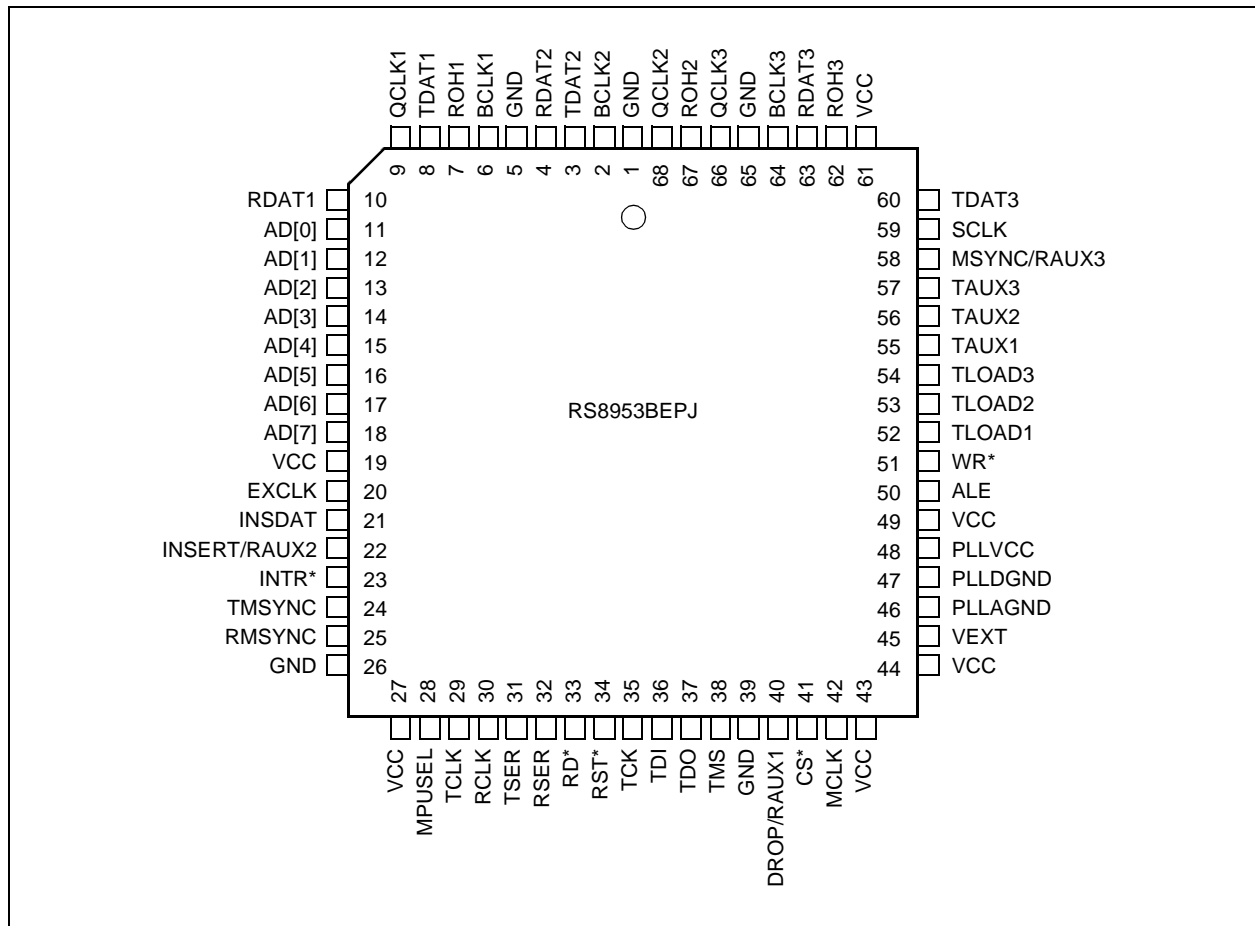




Figure 2-2. Single-Pair PLCC Pin Assignments

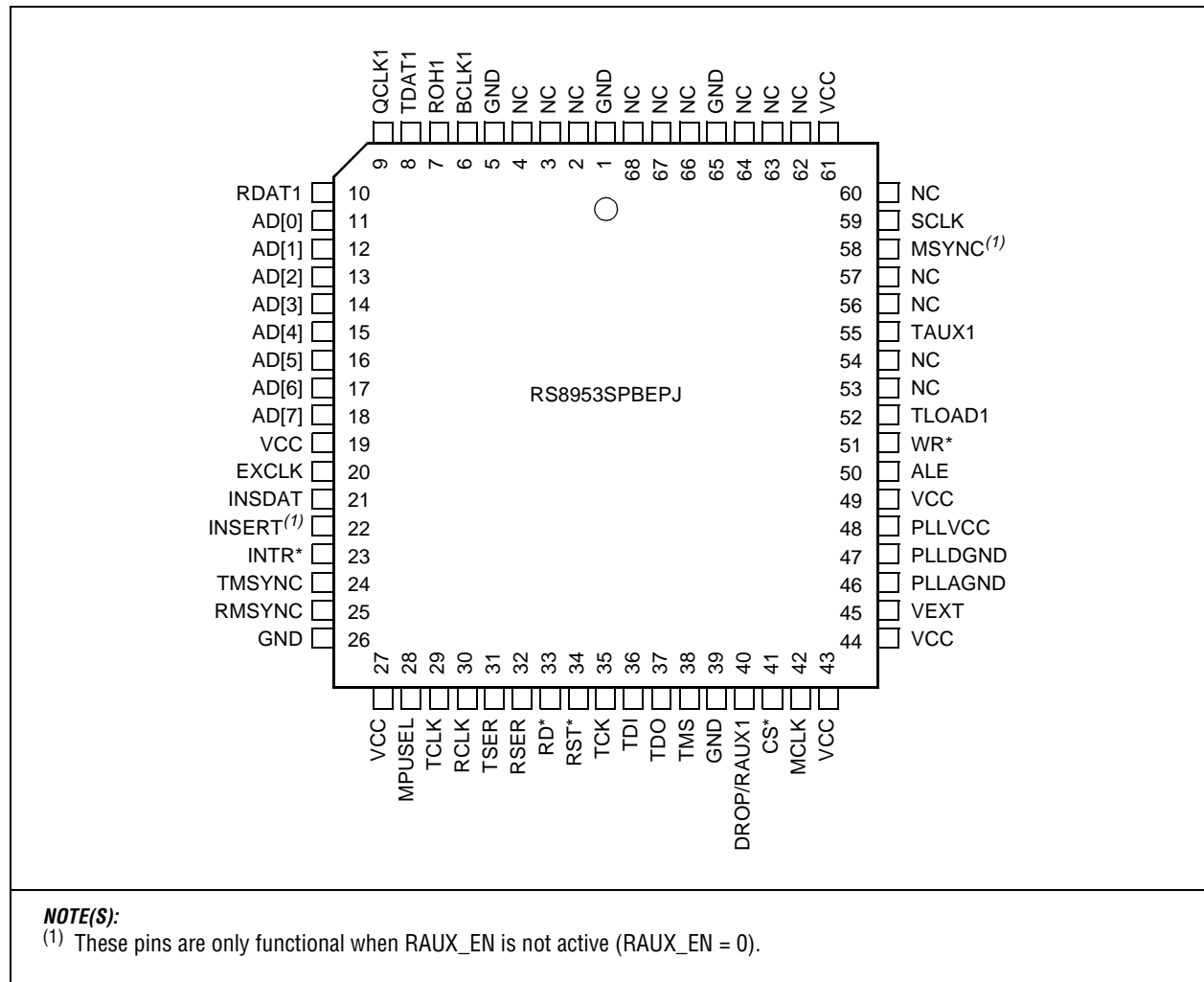


Figure 2-3. Three-Pair PQFP Pin Assignments

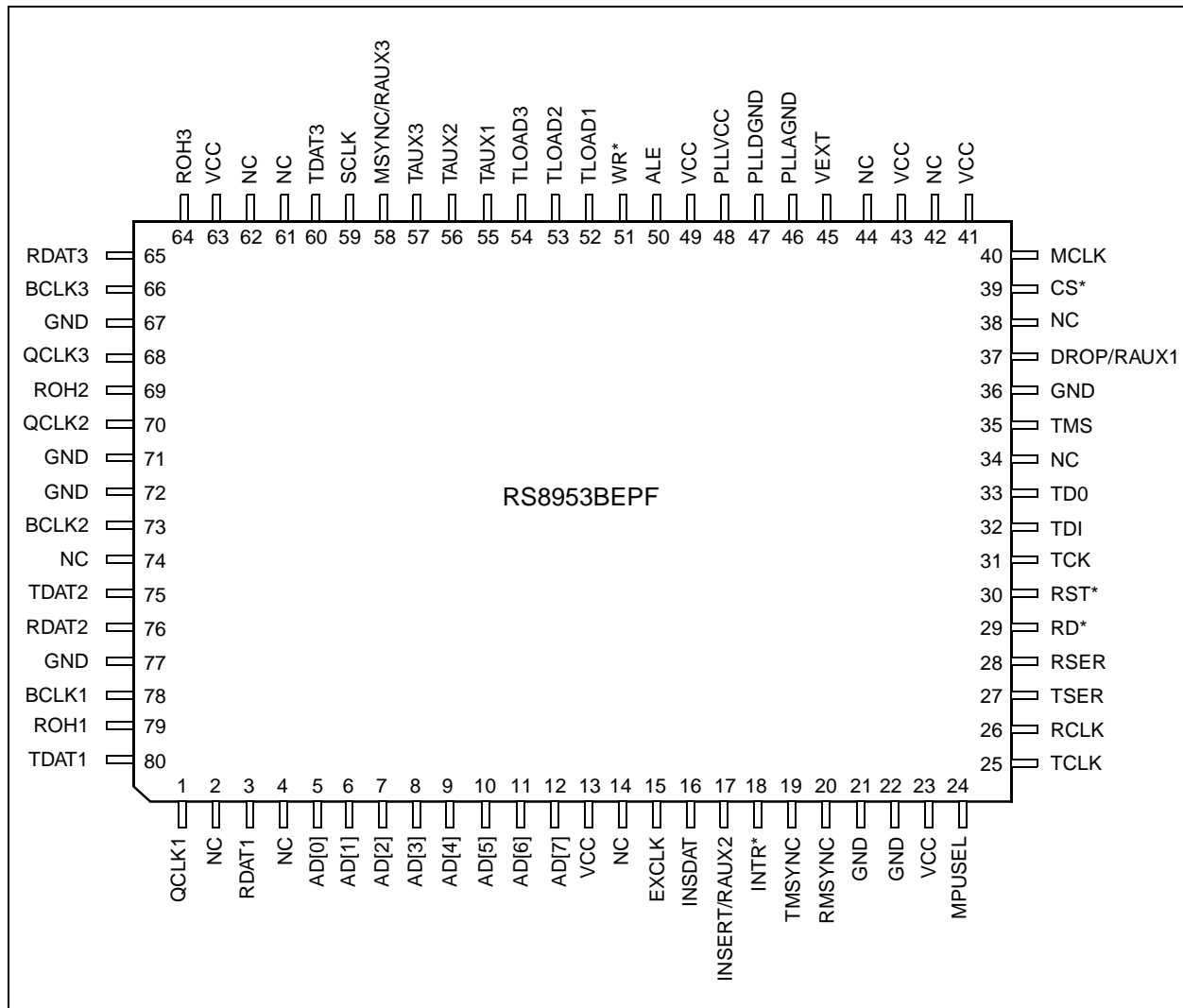


Figure 2-4. Single-Pair PQFP Pin Assignments

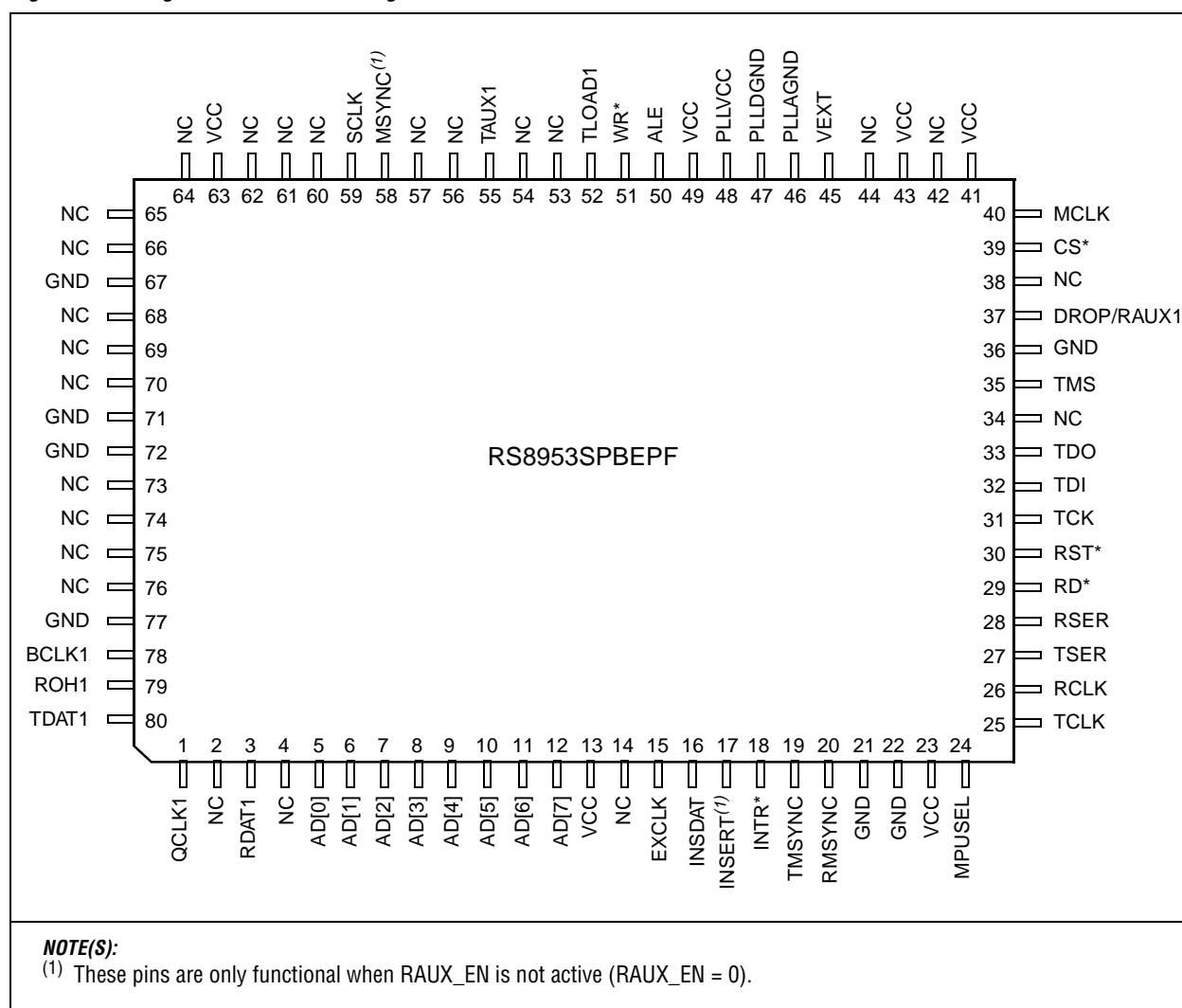


Table 2-1. Pin Assignments (1 of 2)

80-Pin PQFP	68-Pin PLCC	Signal	I/O
71, 72	1	GND	GND
73	2	BCLK2 <sup>(1)</sup>	I
75	3	TDAT2 <sup>(1)</sup>	O
76	4	RDAT2 <sup>(1)</sup>	I
77	5	GND	GND
78	6	BCLK1	I
79	7	ROH1	O
80	8	TDAT1	O
1	9	QCLK1	I

80-Pin PQFP	68-Pin PLCC	Signal	I/O
33	37	TDO	O
35	38	TMS	I
36	39	GND	GND
37	40	DROP/RAUX1	O
39	41	CS*	I
40	42	MCLK	I
41	43	VCC	VCC
43	44	VCC	VCC
45	45	VEXT	I

Table 2-1. Pin Assignments (2 of 2)

80-Pin PQFP	68-Pin PLCC	Signal	I/O	80-Pin PQFP	68-Pin PLCC	Signal	I/O
3	10	RDAT1	I	32	36	TDI	I
5	11	AD[0]	I/O	46	46	PLLAGND	GND
6	12	AD[1]	I/O	47	47	PLLDGND	GND
7	13	AD[2]	I/O	48	48	PLLVC	VCC
8	14	AD[3]	I/O	49	49	VCC(SCAN_MD)	VCC
9	15	AD[4]	I/O	50	50	ALE	I
10	16	AD[5]	I/O	51	51	WR*	I
11	17	AD[6]	I/O	52	52	TLOAD1	O
12	18	AD[7]	I/O	53	53	TLOAD2 <sup>(1)</sup>	O
13	19	VCC	VCC	54	54	TLOAD3 <sup>(1)</sup>	O
15	20	EXCLK	I	55	55	TAUX1	I
16	21	INSDAT	I	56	56	TAUX2 <sup>(1)</sup>	I
17	22	INSERT/RAUX2 <sup>(2)</sup>	O	57	57	TAUX3 <sup>(1)</sup>	I
18	23	INTR*	O	58	58	MSYNC/RAUX3 <sup>(2)</sup>	O
19	24	TMSYNC	I	59	59	SCLK	O
20	25	RMSYNC	O	60	60	TDAT3 <sup>(1)</sup>	O
21, 22	26	GND	GND	63	61	VCC	VCC
23	27	VCC	VCC	64	62	ROH3 <sup>(1)</sup>	O
24	28	MPUSEL	I	65	63	RDAT3 <sup>(1)</sup>	I
25	29	TCLK	I	66	64	BCLK3 <sup>(1)</sup>	I
26	30	RCLK	O	67	65	GND(SCAN_EN)	GND
27	31	TSER	I	68	66	QCLK3 <sup>(1)</sup>	I
28	32	RSER	O	69	67	ROH2 <sup>(1)</sup>	O
29	33	RD*	I	70	68	QCLK2 <sup>(1)</sup>	I
30	34	RST*	I	2, 4, 14, 34, 38, 42, 44, 61, 62, 74	—	NC	—
31	35	TCK	I	—	—	—	—

**NOTE(S):**  
<sup>(1)</sup> These pins do not perform the functions in RS8953SPBEPF and RS8953SPBEPJ.  
<sup>(2)</sup> These pins are only functional in RS8953SPBEPF and RS8953SPBEPJ when RAUX\_EN is not active (RAUX\_EN = 0).

## 2.2 Signal Definitions

**Table 2-2. Signal Definitions (1 of 4)**

Signal	Name	I/O	Description
<b>Microprocessor (MPU) Interface</b>			
MPUSEL	MPU Select	I <sup>(1)</sup>	Determines the type of MPU bus control signals expected during data transfers. Intel (MPUSEL = 0) or Motorola (MPUSEL = 1) bus types are supported. RD* and WR* signal functions are affected.
AD[0:7]	Address/Data Bus	I/O <sup>(1)</sup>	Eight multiplexed address and data signals. The address is latched on the falling edge of ALE and selects one of 256 internal register locations (0x00-0xFF). The data bus transfers the contents of the latched address location during the read or write cycle.
CS*	Chip Select	I <sup>(1)</sup>	Active-low input enables MPU read and write cycles. The rising edge of CS* completes the read or write data transfer cycle and places the address/data bus (AD[0]–AD[7]) in a high impedance state.
ALE	Address Latch Enable	I <sup>(1)</sup>	Active-high input enables the address bus. The falling edge of ALE latches the address internally.
RD*	Read Strobe	I <sup>(1)</sup>	Signal function determined by MPUSEL: MPUSEL = 0; RD* is an active low data strobe for read cycles. MPUSEL = 1; RD* is an active low data strobe for read/write cycles.
WR*	Write Strobe	I <sup>(1)</sup>	Signal function determined by MPUSEL: MPUSEL = 0; WR* is an active low data strobe for write cycles. MPUSEL = 1; WR* controls the data bus transfer direction: high during read cycles and low during write cycles.
INTR*	Interrupt Request	O	Active low, open-drain output indicates when any one or more Interrupt Request Register (IRR) bit is high and its respective Interrupt Mask Register (IMR) bit is low. INTR* remains active until all pending interrupts are cleared by writing 0s to their corresponding Interrupt Clear Register (ICR) bits.
RST*	Reset	I <sup>(1)</sup>	Active low input required to initialize internal circuits after power and master clock have been applied. All MPU registers remain accessible while reset is active. Unless stated otherwise, reset activation does not affect the MPU register contents. RS8953B reset activation disables interrupts on the INTR* output by forcing all 1s in the Interrupt Mask Register (IMR), and zeros in the TX_ERR_EN, DPLL_ERR_EN, and RX_ERR_EN bits. RS8953B reset activation disables auxiliary channels by forcing zeros in all TAUX_EN and RAUX_EN bits. To facilitate system upgrades from prototype Bt8953EPF, RS8953B reset activation also forces zeros in those command register bits which do not exist on Bt8953EPF, but were added on RS8953B.

Table 2-2. Signal Definitions (2 of 4)

Signal	Name	I/O	Description
<b>HDSL Channels</b>			
BCLK1	Bit Clock	I <sup>(1)</sup>	Corresponds to three HDSL and three Auxiliary channels. BCLKn operates at twice the 2B1Q symbol rate. The rising edge of BCLKn outputs TDATn, TLOADn, RAUXn and ROHn; the falling edge samples QCLKn, RDATn, and TAUXn inputs.
BCLK2 <sup>(3)</sup>			
BCLK3 <sup>(3)</sup>			
QCLK1	Quaternary Clock	I <sup>(1)</sup>	Operates at the 2B1Q symbol rate (half-bit rate) and identifies sign and magnitude alignment of both RDATn and TDATn serially encoded bit streams. The falling edge of BCLKn samples QCLKn: 0 = sign bit; 1 = magnitude bit.
QCLK2 <sup>(3)</sup>			
QCLK3 <sup>(3)</sup>			
TDAT1	Transmit Data	O	HDSL transmit data output at the bit rate on the rising edge of BCLKn. Serially encoded with the 2B1Q sign bit aligned to the QCLKn low level and the 2B1Q magnitude bit aligned to the QCLKn high level.
TDAT2 <sup>(3)</sup>			
TDAT3 <sup>(3)</sup>			
TAUX1	Transmit Auxiliary Data	I <sup>(1)</sup>	HDSL transmit auxiliary data input sampled on the falling edge of BCLKn when TLOADn is active. TAUXn replaces data normally supplied by PCM or HDSL transmitters to the HDSL scrambler input. Payload bytes or Z-bits can be mapped from TAUXn.
TAUX2 <sup>(3)</sup>			
TAUX3 <sup>(3)</sup>			
RDAT1	Receive Data	I <sup>(1)</sup>	HDSL receive data input sampled on the falling edge of BCLKn. The serially encoded 2B1Q sign bit is sampled when QCLKn is low, and the 2B1Q magnitude bit is sampled when QCLKn is high.
RDAT2 <sup>(3)</sup>			
RDAT3 <sup>(3)</sup>			
RAUX1	Receive Auxiliary Data	O	Receives data from the HDSL descrambler output on the rising edge of BCLKn. Includes all SYNC, STUFF, HOH, payload, and Z-bits. RAUXn shares pin locations with DROP, INSERT, and MSYNC, as controlled by RAUX_EN (CMD_6; addr 0xF3).
RAUX2 <sup>(3)</sup>			
RAUX3 <sup>(3)</sup>			
TLOAD1	Transmit Load Indicator	O	Active-high output that indicates when specific payload or Z-bits are sampled at TAUXn. TLOADn is active for 8 bits coincident with each marked payload byte or 1 bit for Z-bits. The last 40 Z-bits or any combination of payload bytes may be marked.
TLOAD2 <sup>(3)</sup>			
TLOAD3 <sup>(3)</sup>			
ROH1	Receive Overhead Indicator	O	Indicate when overhead is received. Has two modes of operation: <ul style="list-style-type: none"> <li>• RAS = 0. ROHn is high to mark only data passed into the RFIFO.</li> <li>• RAS=1. ROHn is high to mark only the last 40 Z-bits.</li> </ul>
ROH2 <sup>(3)</sup>			
ROH3 <sup>(3)</sup>			

Table 2-2. Signal Definitions (3 of 4)

Signal	Name	I/O	Description
<b>PCM Channel</b>			
TCLK	Transmit Clock	I <sup>(2)</sup>	Operates at the PCM bit rate and samples the PCM transmit inputs: TSER, TMSYNC, and INSDAT; and clocks the PCM transmit output, INSERT. Falling edge samples and rising edge outputs are normal, inverted TCLK edges are selectable. Optionally, RCLK or EXCLK can be programmed as the PCM transmit clock for loopback or externally timed applications.
RCLK	Receive Clock	O	Operates at the PCM bit rate and clocks the PCM receive outputs: RSER, RMSYNC, and DROP. Normally, RCLK is supplied by the internal clock recovery DPLL. Optionally, EXCLK or TCLK can be programmed as the receive source during loopback or externally timed applications. Rising-edge (normal) or falling-edge (inverted) output transitions are selectable.
EXCLK	External Clock	I <sup>(2)</sup>	Optionally sources the PCM Receive Clock (RCLK), or both RCLK and PCM Transmit Clock (TCLK) for systems that supply a local master clock. Normal or inverted edges are also selectable.
TSER	Transmit Serial Data	I <sup>(1)</sup>	Accepts up to 64 timeslots (1 timeslot = 8 bits) of data and an optional framing bit per PCM frame. TSER data and F-bits are then routed and mapped into the HDSL transmit channel payload.
RSER	Receive Serial Data	O	Outputs up to 64 timeslots of data and an optional F-bit per PCM frame. Receive serial data and F-bits are constructed by mapping and combining payload from the HDSL receive channels.
TMSYNC	Transmit Multiframe Sync	I <sup>(1)</sup>	Active-high input resets the PCM transmit time base during framed applications. TMSYNC is ignored in unframed or asynchronously mapped applications. The low to high input state transition is detected and internally delayed by a programmable bit and frame offset to coincide with the TSER and INSDAT sample location of bit 0, frame 0. The programmable sample point accommodates any system's rising edge frame or multiframe sync signal.
RMSYNC	Receive Multiframe Sync	O	Active-high output from the receive timebase, typically programmed to mark PCM multiframe boundaries during framed applications, and remains unused during unframed or asynchronously mapped applications. RMSYNC pulses high for one RCLK coincident with RSER output of bit 0, frame 0. Bit 0 is the first bit in TSO of an E1 or Nx64 frame, or the F-bit of a T1 frame. Programmable bit and frame delays allow RMSYNC to mark any desired RSER bit.
MSYNC	Transmit Master Sync	O	Active-high output pulses high for one TCLK to mark two clock cycles before the TSER and INSDAT sample point of bit 0, frame 0, of a transmit multiframe. MSYNC references the TMSYNC applied by the system or supplies the system with a master PCM frame/multiframe sync signal.

Table 2-2. Signal Definitions (4 of 4)

Signal	Name	I/O	Description
<b>Drop/Insert</b>			
DROP	Drop Indicator	0	Active-high output indicates when specific PCM timeslots are present on RSER. DROP is high for 8 bits coincident with each marked timeslot, or 1 bit when marking F-bits. Any combination of timeslots and F-bits within the PCM frame can be marked.
INSDAT	Insert Data	I <sup>(1)</sup>	Alternate source of PCM transmit serial data. INSDAT is sampled by TCLK and replaces TSER when INSERT is active. INSDAT and TSER use the same frame format. INSDAT can be programmed to replace TSER data on a per-timeslot-basis.
INSERT	Insert Indicator	0	Active-high output indicates when specific INSDAT timeslots are sampled. INSERT is high for 8 bits coincident with each marked timeslot or for 1 bit when marking F-bits. Any combination of timeslots and F-bits within the PCM frame can be marked.
<b>DPLL and Power</b>			
MCLK	Master Clock	I	Runs through a multiplier PLL to create an internal 60–80 MHz reference clock for the DPLL. The 16 times symbol rate clock from a Conexant HDSL transceiver typically connects to MCLK. However, MCLK is not required to be synchronized to any HDSL or PCM channel. The DPLL reference clock is used to synthesize the PCM Recovered Clock (RCLK) based on DPLL programmed values. Optionally, a 60–80 MHz clock can be input directly on MCLK.
SCLK	System Clock	0	The internal 60–80 MHz DPLL reference clock is divided by 4 to create a 15–20 MHz system clock output on SCLK. SCLK can be applied to other devices requiring a system clock (i.e., Bt8360 or Bt8510).
VEXT	External Voltage	I	Used to bias input protection diodes. If interfacing to 5 V powered devices, connect this pin to 5 V. Otherwise, connect 3.3 V to this pin.
PLLVCC	PLL Power	I	3.3 Vdc +/- 0.3 V power input for the PLL.
PLLDGND	PLL Ground	I	0 Vdc ground reference for the PLL.
PLLAGND	PLL Analog Ground	I	0 Vdc analog ground reference for the PLL. Tied to GND unless PLL operation is desired above 80 MHz.
VCC	Power	I	3.3 Vdc +/- 0.3 V power input.
GND	Ground	I	0 Vdc ground reference.
TCK	Test Clock	I	Boundary scan clock samples and outputs test access signals.
TMS	Test Mode Select	I <sup>(1)</sup>	Active-high enables test access port. Sampled by TCK rising edge.
TDI	Test Data Input	I <sup>(1)</sup>	Serial data for boundary scan chain. Sampled by TCK rising edge.
TDO	Test Data Output	0	Outputs serial data from boundary scan chain on TCK falling edge.
<b>NOTE(S):</b> <sup>(1)</sup> Internal pull-ups (80-100 kΩ) are present on inputs to allow unused inputs to remain disconnected. <sup>(2)</sup> Internal pull-downs (80-100 kΩ) are present on inputs to allow unused inputs to remain disconnected. <sup>(3)</sup> The pins do not perform these functions in RS8953SPBEPF and RS8953SPBEPJ.			



## 3.0 Circuit Descriptions

### 3.1 MPU Interface

The Microprocessor Unit (MPU) interface consists of an 8-bit parallel multiplexed address-data bus, an associated bus control signal, and a maskable interrupt request output, as illustrated in [Figure 3-1](#) and [Figure 3-2](#). The MPU interface is compatible with 8-bit processors running at bus cycle speeds up to 16 MHz. Systems that use 16 or 32-bit processors can add an external address buffer and data transceiver to connect the RS8953B. Faster bus speeds require external wait-state insertion logic.

**Figure 3-1. MPU Bus Control Logic**

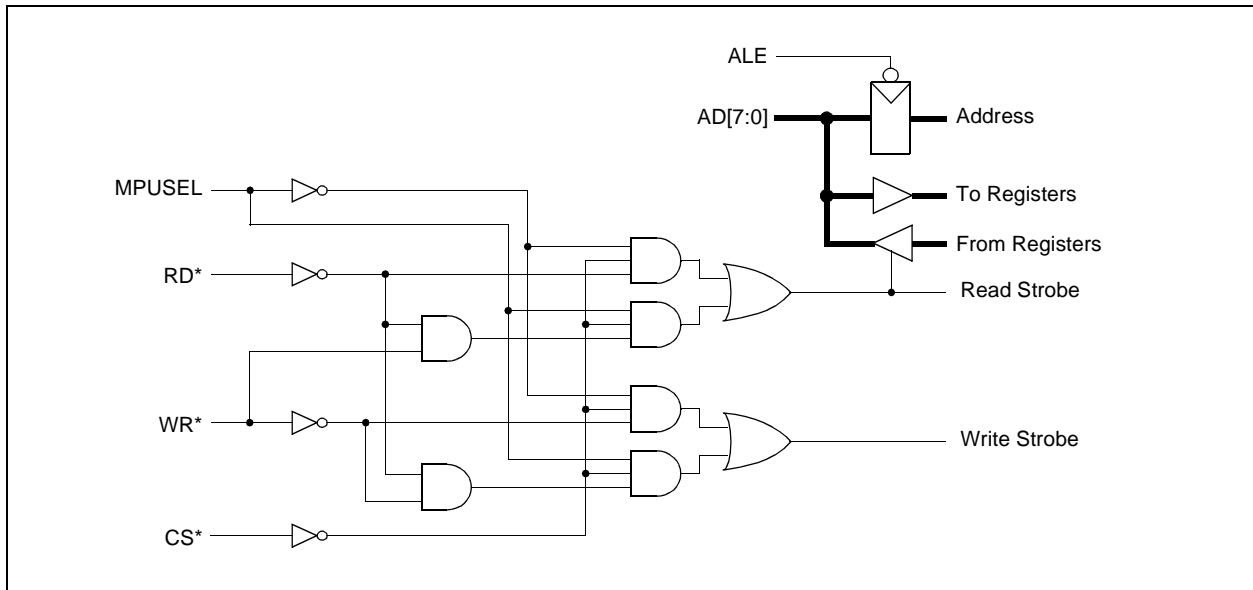
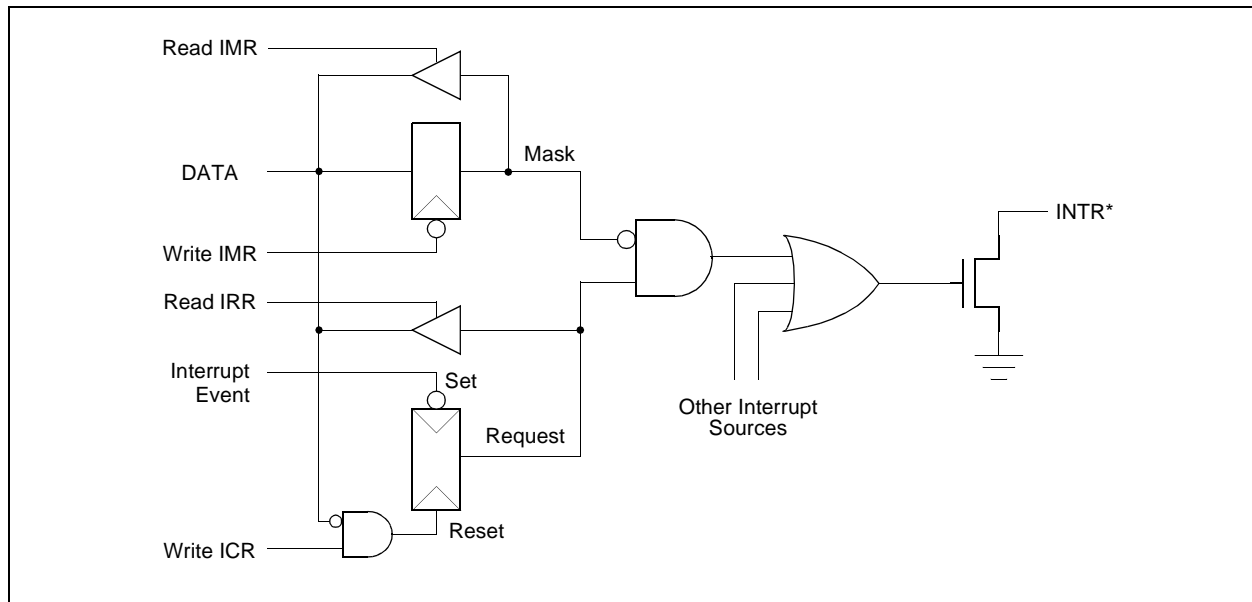


Figure 3-2. MPU Interrupt Logic



### 3.1.1 Address/Data Bus

Address/data bus pins AD[7:0] allow MPU access to RS8953B internal registers. Read and write access is allowed at any of the 256 address locations, but only defined register address locations are applicable (see [Table 4-1](#)).

### 3.1.2 Bus Controls

Five signals control register access: ALE, CS\*, RD\*, WR\*, and MPUSEL. The address on AD[7:0] is latched on the falling edge of ALE, and CS\* is an active-low port enable for all read and write operations. If CS\* is high, the MPU port is inactive.

Different styles of bus control are supported using separate read and write strobes for Intel-style buses, or common data strobe with a combined read/write signal for Motorola-style buses. When MPUSEL = 0 (Intel bus), RD\* is an active-low read enable and WR\* is an active-low write strobe. While RD\* and CS\* are low, the addressed register's data is driven onto AD[7:0]. If WR\* and CS\* are low, the rising edge of WR\* or CS\* latches data from AD[7:0] into the register. When MPUSEL = 1 (Motorola bus), RD\* is an active-low data strobe for both read and write cycles, and WR\* is a read/write select. While RD\* and CS\* are low and WR\* is high, the addressed register's data is driven onto AD[7:0]. If RD\*, CS\*, and WR\* are low, the rising edge of RD\*, CS\*, or WR\* latches data from AD[7:0].

### 3.1.3 Interrupt Request

The open-drain interrupt request output (INTR\*) indicates when a particular set of transmit, receive, or common status registers have been updated. Eight maskable interrupt sources are requested on the common INTR\* pin:

1. TX1 = Channel 1 Transmit 6 ms Frame
2. TX2 = Channel 2 Transmit 6 ms Frame
3. TX3 = Channel 3 Transmit 6 ms Frame
4. RX1 = Channel 1 Receive 6 ms Frame
5. RX2 = Channel 2 Receive 6 ms Frame
6. RX3 = Channel 3 Receive 6 ms Frame
7. TX\_ERR = Logical OR of 3 Transmit Channel Errors
8. RX\_ERR = Logical OR of 3 Receive Channel Errors and DPLL Errors

All interrupt events are edge-sensitive and synchronized to their respective HDSL channel's 6 ms frame. The basic structure of each interrupt source is shown in [Figure 3-2](#), with three associated registers: Interrupt Mask Register [IMR; addr 0xEB], where writing a 1 to an IMR bit prevents the associated interrupt source from activating INTR\*; Interrupt Request Register [IRR; addr 0x1F], where active interrupt events are indicated by IRR bits that are read high; and Interrupt Clear Register [ICR; addr 0xEC], where writing a 0 to an ICR bit clears the associated IRR bit, and if no other interrupts are pending, deactivates INTR\*. Error interrupts (TX\_ERR and RX\_ERR) are combined from multiple sources, each source having its own interrupt enable. Individual errors are reported in the common Error Status Register [ERR\_STATUS; addr 0x3C] which is cleared by an MPU read.

### 3.1.4 Hardware Reset

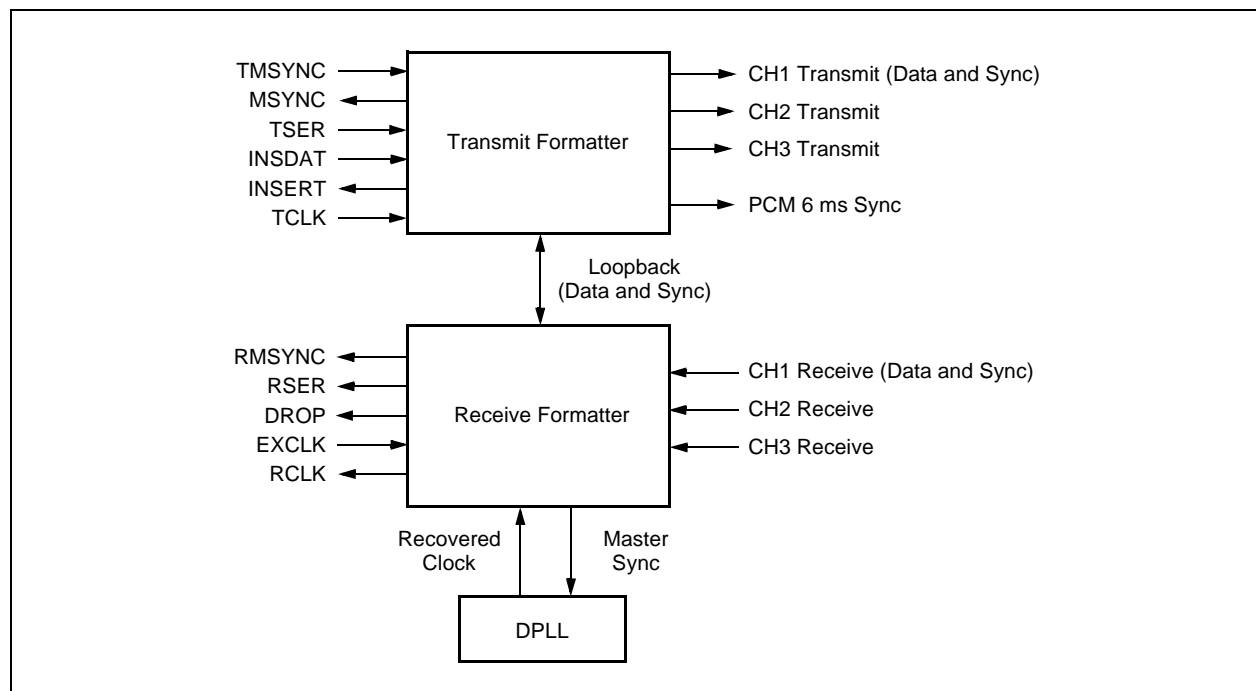
Assertion of hardware reset (RST\*) is required to preset all IMR bits, clear all error interrupt enables, and thus disable INTR\* output. For backward compatibility with Bt8953 software, RST\* also clears the command register bits added to RS8953B which aren't present on prototype Bt8953. All other registers are MPU accessible while RST\* is asserted.

## 3.2 PCM Channel

The Pulse Code Multiplexed (PCM) channel displayed in [Figure 3-3](#) consists of independent transmit and receive formatter circuits to control the flow of serial data between PCM and HDSL channels, to establish alignment between PCM and HDSL frames, and maintain synchronization between PCM and HDSL clocks. Framed serial data consists of a variable number of multiplexed 8-bit timeslots, plus an optional framing bit (F-bit), a variable number of PCM frames repeated to form a PCM multiframe, and a variable number of multiframe concatenated to form a PCM 6 ms frame. T1, E1, or custom Nx64 frame formats are selected by programming the PCM Formatter Registers (see [Table 4-4](#)) to define the number of bits per frame [FRAME\_LEN; addr 0xC8], frames per multiframe [MF\_LEN; addr 0xC6], and multiframe per 6 ms frame [MF\_CNT; addr 0xC7]. Unframed serial data is selected in the same manner; however, the number of bits per frame act as a single channel rather than individual timeslots and can support PCM frame lengths that are not integer multiples of 8-bits.

In framed or unframed applications, PCM timebases create a 6 ms frame period based on the Transmit Clock (TCLK) and Receive Clock (RCLK). PCM timebases are programmed to equal approximately the HDSL 6 ms frame period defined by the HDSL Frame Length [HFRAME\_LEN; addr 0xCA] in relation to the master HDSL channel's Bit Clock (BCLKn). The resultant PCM and HDSL 6 ms frame intervals are used to establish alignment between PCM and HDSL frames, to maintain synchronization between transmit clocks by performing bit stuffing, and to recover PCM receive clock by comparing phase offset between frames.

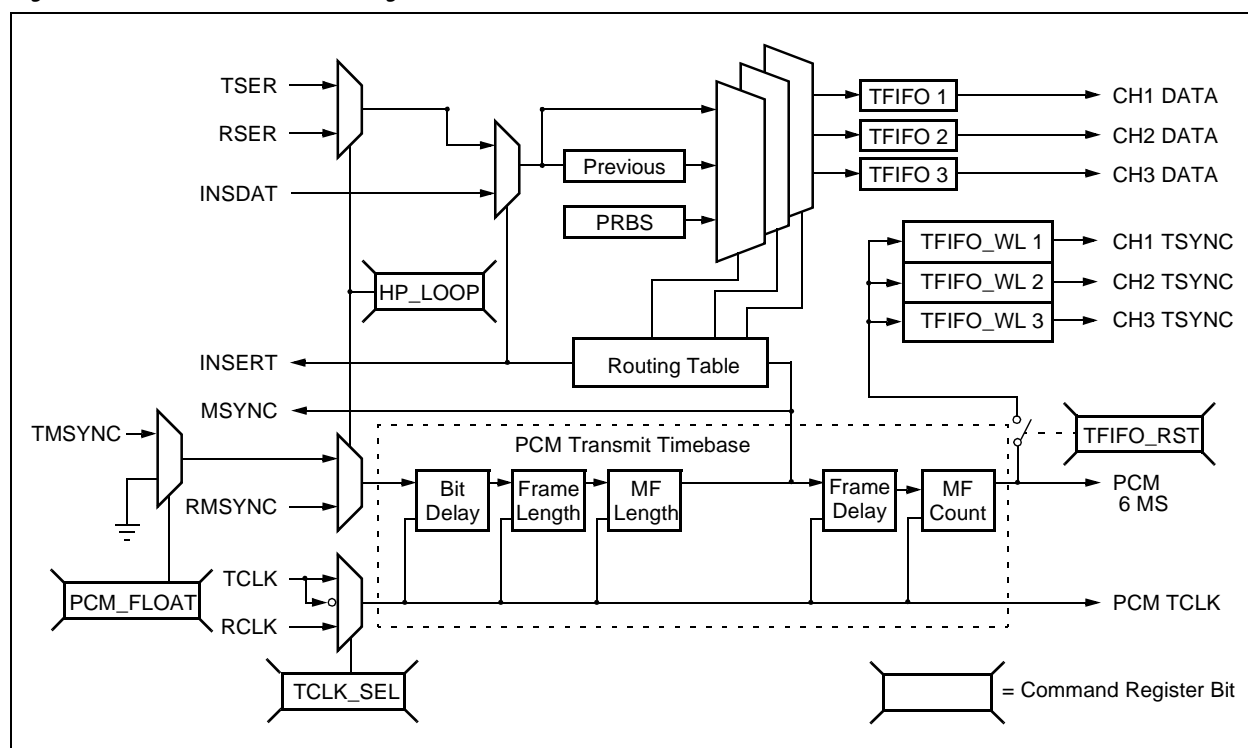
**Figure 3-3. PCM Channel Block Diagram**



### 3.2.1 PCM Transmit

The PCM transmit formatter shown in [Figure 3-4](#) accepts framed or unframed serial data on the TSER and INSDAT inputs. Both inputs are sampled on the clock edge selected by TCLK\_SEL [CMD\_2; addr 0xE6] according to the format of the PCM Multiframe Sync (MSYNC) output. The PCM transmit timebase outputs MSYNC to mark two clock cycles before the PCM input sample point of bit 0, frame 0. The timebase either references the system's Transmit Multiframe Sync (TMSYNC) input or supplies MSYNC without regard to TMSYNC, as controlled by the PCM\_FLOAT setting [CMD\_2; addr 0xE6].

**Figure 3-4. PCM Transmit Block Diagram**



The MSYNC leads the sampling of bit 0, frame 0, on TSER and INSDAT by two TCLK bit positions.

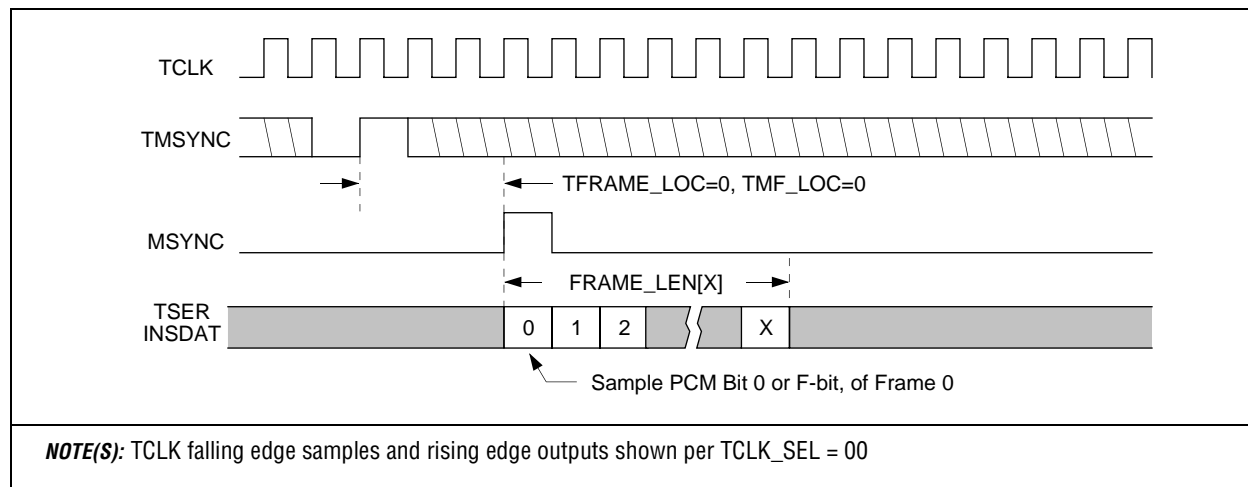
If PCM\_FLOAT is active, the transmit timebase ignores TMSYNC and outputs MSYNC according to the PCM formatter register values: FRAME\_LEN, MF\_LEN, and MF\_CNT. In this case, MSYNC acts as PCM bus master and supplies a multiframe sync reference to the system as illustrated in [Figure 3-5](#), but without a specific TMSYNC relationship.

If PCM\_FLOAT is inactive, MSYNC is aligned to TMSYNC (as shown in [Figures 3-5](#) and [3-6](#)). The system locates the sampling point of bit 0, frame 0, with respect to TMSYNC by programming the number of bit delays [TFRAME\_LOC; addr 0xC0] from TMSYNC's rising edge to bit 0 of the PCM frame. In addition, it locates the frame 0 input sample point by programming the additional number of frame delays [TMF\_LOC; addr 0xC2] needed to mark the first frame of a PCM multiframe.

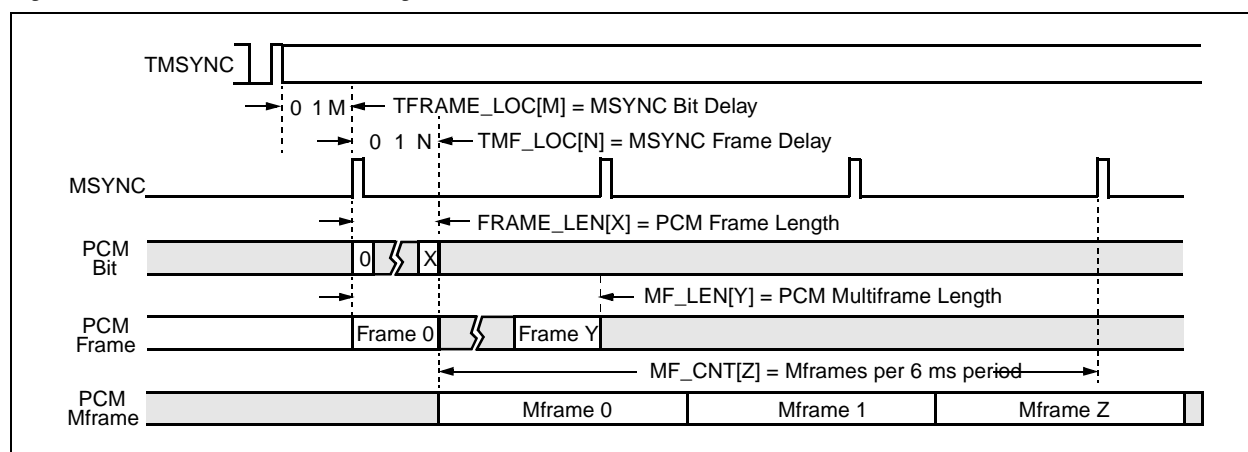
Figure 3-5 shows the phase relationship between TMSYNC and MSYNC when TFRAME\_LOC is equal to 0. Figure 3-6 illustrates the progression of MSYNC with increasing bit and frame delays.

**NOTE:** MSYNC can optionally mark the start of every PCM frame (bit 0, all frames) by setting MF\_LEN equal to 1 frame per multiframe.

**Figure 3-5. PCM Transmit Sync Timing**



**Figure 3-6. PCM Transmit Data Timing**



### 3.2.1.1 Transmit Synchronization

Alignment of transmit PCM data in relation to MSYNC determines whether PCM and HDSL frames are synchronously mapped. The RS8953B does not examine transmit data for T1, E1, or application framing patterns. Therefore, the system must apply PCM data aligned to MSYNC when synchronous mapping is desired.

If the system applies PCM bit 0, frame 0 coincident with MSYNC, then the transmit router guarantees that each PCM timeslot placed in the TFIFO will be aligned and mapped into a specific HDSL payload byte. In addition, timeslots from the first PCM frame are mapped to payload bytes in the first HDSL payload block, and the start of a PCM multiframe is aligned with the start of an HDSL frame.

If the system does not apply PCM data aligned to MSYNC, then the application is asynchronously mapped, and the placement of timeslots, frames and multiframes is not aligned to HDSL payload bytes, blocks, or frames. Asynchronously mapped applications require the entire PCM serial data stream be transported; the transmitter cannot discern timeslot or frame boundaries.

Synchronous mapping allows selective timeslot routing to HDSL channels, thus enabling transport to multiple remote sites and allowing PCM to operate at rates which exceed available HDSL payload. However, synchronously mapped channels are subject to changes in transmit frame alignment, resulting from changes of the TMSYNC reference. ETSI defines synchronous and asynchronous mapping depending on the type of E1 transport. Bellcore requires synchronous T1 frame mapping for F-bits to align with Z-bit positions. (Refer to frame formats and mapping arrangements illustrated in [Figures 3-16 through 3-18](#), and [Tables 3-2 and 3-3](#)).

### 3.2.1.2 Transmit Routing Table

Timeslot and F-bit data are shifted from PCM inputs into the TFIFO according to the programmed transmit Routing Table [ROUTE\_TBL; addr 0xED] assignments. The routing table contains an entry for each PCM timeslot and the system selects 1, 2, 3, or none of the HDSL transmit channels as the timeslot's destination. The system also selects which source (TSER, INSDAT, PRBS generator or previous timeslot) supplies data for the destination. In this manner, the routing table allows a single timeslot to be routed to more than one HDSL channel, and a single timeslot to supply a repeated value to destination channels. If INSDAT supplies source data, then the INSERT output marks PCM sampling times corresponding to that timeslot (refer to [Figure 3-7](#) for INSERT signal timing). Note that INSDAT is sampled through the previous buffer and is routed in the subsequent timeslot table entry.

### 3.2.1.3 PRBS Generator

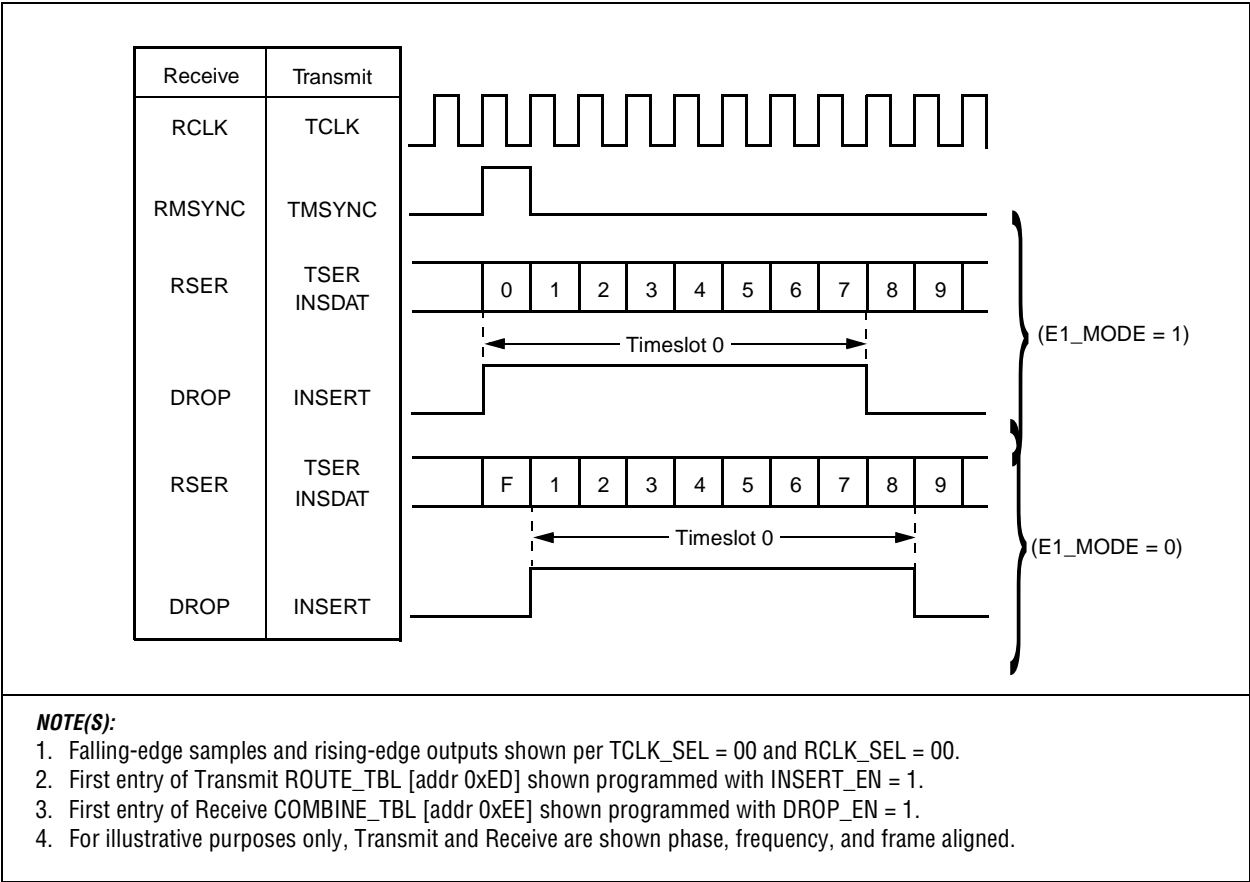
Incoming PCM transmit timeslots can be replaced by a test pattern on a per-timeslot basis, or the entire framed or unframed PCM transmit channel can be replaced by a test pattern (see PRBS\_MODE in CMD\_3; addr 0xE7 and BER\_SEL in CMD\_6; addr 0xF3). When test pattern is enabled on a per-timeslot basis according to the programmed transmit routing table assignments, the PRBS generator is only clocked during enabled timeslots and may output a single test pattern sequence over multiple discontinuous timeslots. The test pattern is selected from one of four Pseudo-Random Bit Sequence (PRBS) patterns or a programmable 8-bit fixed pattern [FILL\_PATT; addr 0xEA]. PRBS pattern selections are:  $2^4-1$ ,  $2^{15}-1$ ,  $2^{23}-1$  and Quasi-Random Signal Sequence (QRSS), where QRSS equals  $2^{20}-1$  PRBS with 14-zero limit. The  $2^{15}-1$  test pattern has an inverter in the data path. RS8953B does not provide a mechanism to automatically insert logic errors in the test pattern, although the capability to synchronize and measure test pattern errors is provided by the BER meter.

3.2.1.4 Drop/Insert Channel

PCM channels can carry timeslot data along a backplane that serves multiple interfaces or subscriber line cards (see [Figure 1-3](#)) which requires that each interface or line card be able to drop or insert individual PCM timeslots. The RS8953B provides DROP and INSERT signals to facilitate external multiplexing of individual timeslots from a shared PCM backplane, but does not provide the capability to three-state its data outputs during specific PCM timeslots. DROP and INSERT signals are programmed to mark RSER data output and INSDAT data input timeslots via the receive Combination Table [COMBINE\_TBL; addr 0xEE] and the transmit Routing Table [ROUTE\_TBL; addr 0xED] assignments.

**NOTE:** Only INSDAT provides an alternate source for each PCM transmit timeslot and does not expand the total number of available timeslots. [Figure 3-7](#) shows DROP and INSERT timing as it relates to PCM bus timing during T1/E1 applications.

Figure 3-7. Drop/Insert Channel Timing





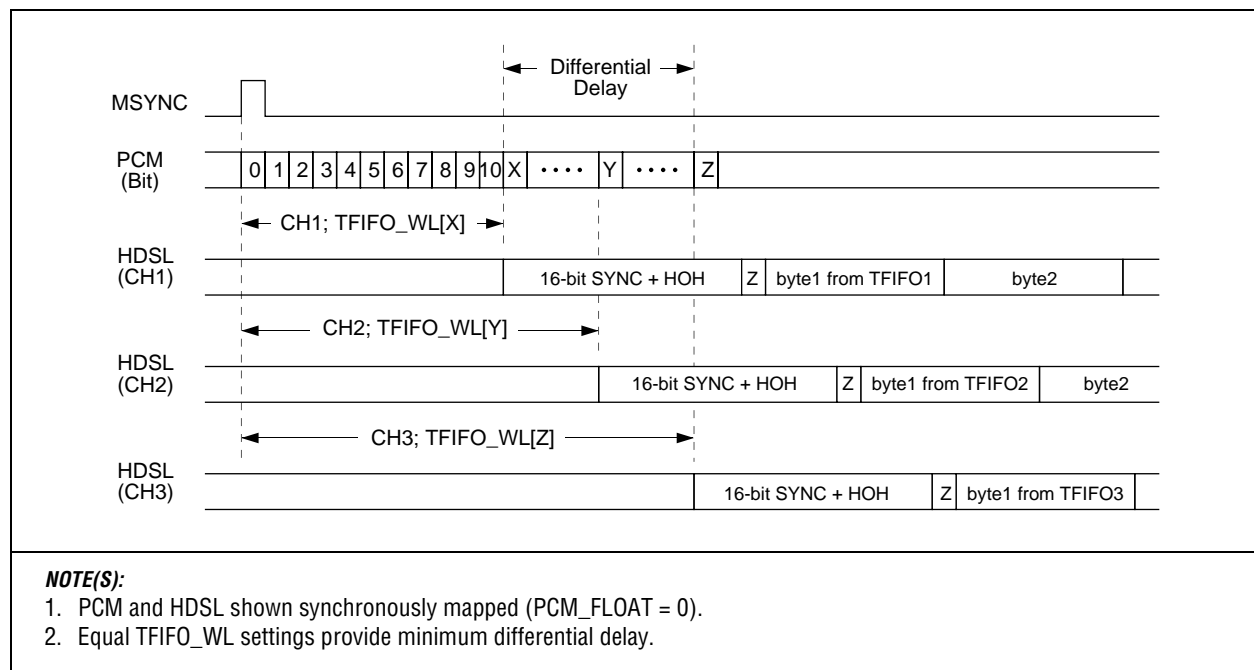
### 3.2.1.5 TFIFO Water Levels

Each HDSL transmit channel aligns the start of its output frame with respect to the PCM 6 ms sync according to the programmed TFIFO water level values [TFIFO\_WL; addr 0x05]. PCM 6 ms sync is created from MSYNC by the divisor programmed in MF\_CNT [addr 0xC7]. The HDSL 6 ms frame is created from PCM 6 ms by adding the TFIFO\_WL phase offset programmed for each channel, as shown in Figure 3-8. In this manner, HDSL output frames are slaved to PCM frame timing regardless of whether the system chooses to synchronize PCM data to MSYNC.

The phase offset between PCM and HDSL 6 ms frames is programmed by TFIFO\_WL as the number of TCLK cycle delays from the start of PCM 6 ms sync to the start of HDSL 6 ms frame. Thus, this phase offset determines the amount of PCM data written to the TFIFO before the HDSL transmitter begins extracting data from the TFIFO, which also defines each transmitter's data throughput delay and subsequently the differential delay with respect to other HDSL channels. The actual phase offset varies over time as a result of stuff bit insertion as well as PCM and HDSL clock jitter and wander. Therefore, TFIFO\_WL is only used to establish the initial phase offset between PCM and HDSL frames when the MPU issues the TFIFO\_RST [addr 0x0D] command, or after a stuffing error.

Because all or part of the PCM frame can be routed to each HDSL channel, the system must consider transmit routing table assignments and other data path delays when programming TFIFO\_WL values. Sufficient phase offset must be established to allow time for the first programmed timeslot to be routed from the PCM frame into the TFIFO, to absorb the phase offset created by HDSL overhead, to stuff bit insertion and clock frequency variation, and to unload the first timeslot from the TFIFO and map data into the HDSL payload byte. Conversely, to avoid TFIFO overflow, phase offset must be limited so the amount of data residing in the TFIFO does not exceed the number of PCM bits routed during one PCM frame, the maximum TFIFO depth (186 bits), or the total HDSL payload block length [HFRAME\_LEN; addr 0xCA].

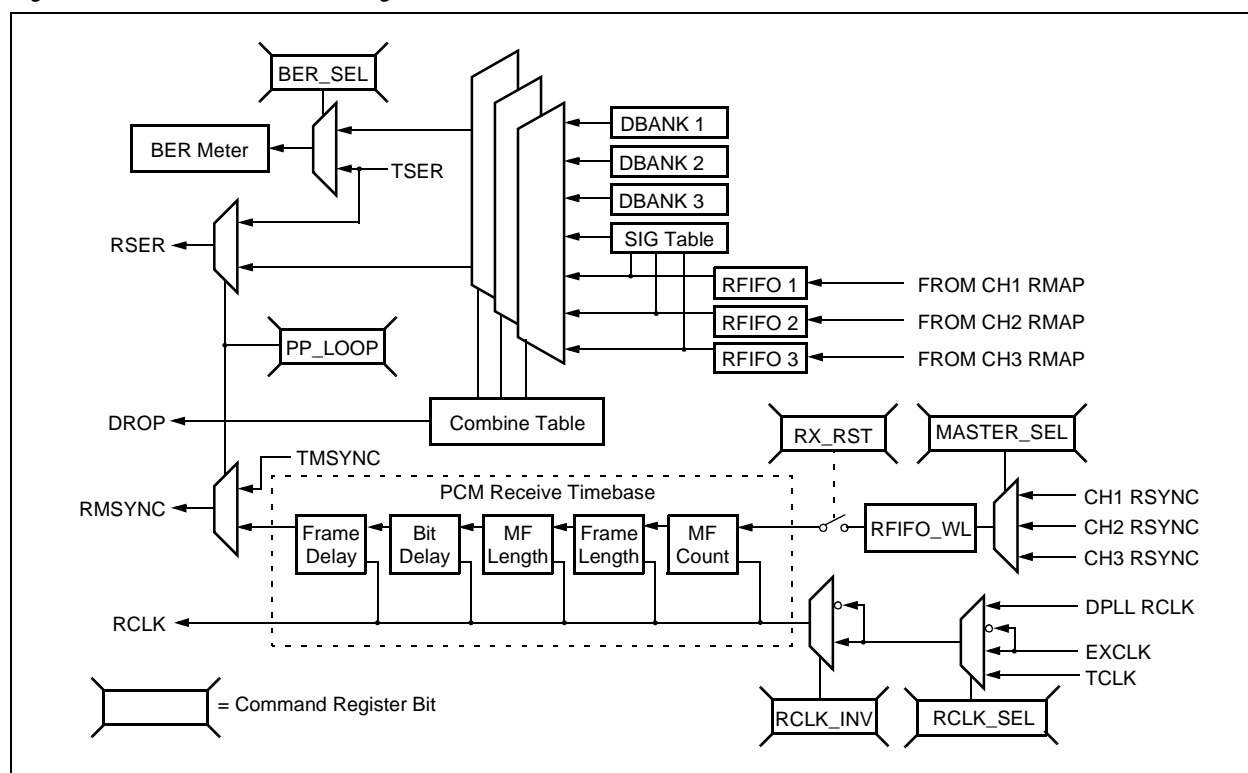
**Figure 3-8. TFIFO Water Level Timing**

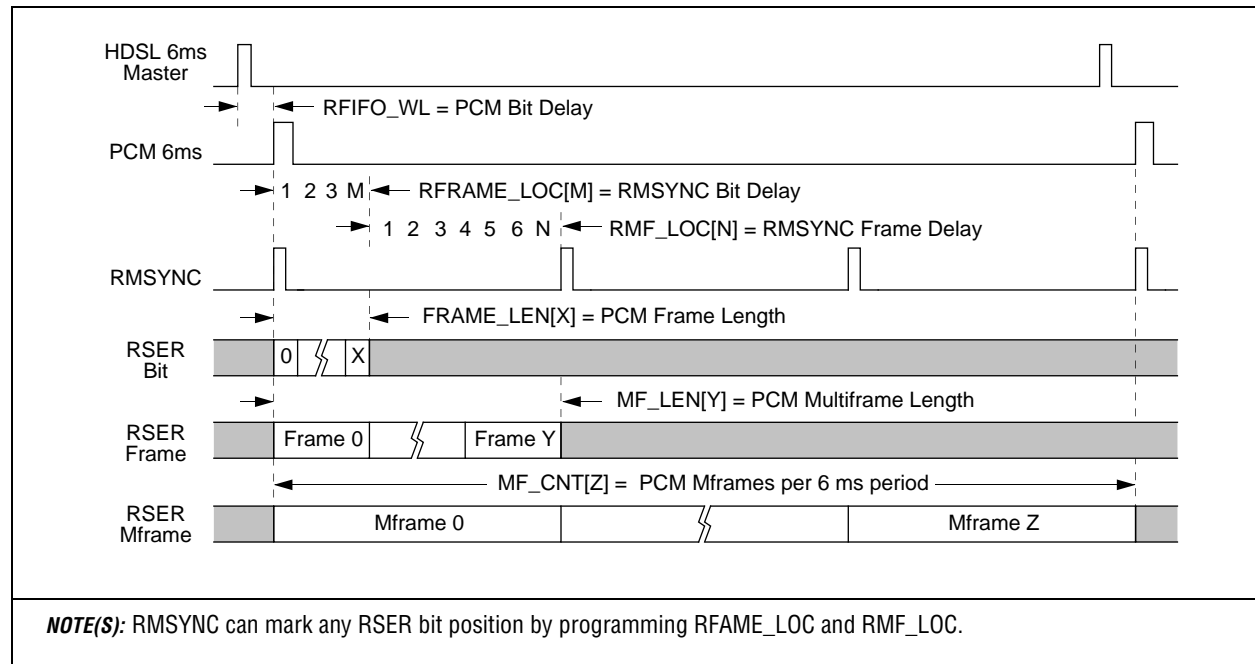


### 3.2.2 PCM Receive

The PCM receive formatter shown in [Figure 3-9](#) constructs the serial data (RSER) output according to receive combination table settings and the frame format defined by the PCM Formatter Registers (see [Table 4-4](#)). The PCM receiver operates on the clock edge selected by RCLK\_SEL [CMD\_2; addr 0xE6] and references the PCM receive timebase and RSER frame location to the alignment provided by the master HDSL channel's receive 6 ms frame. Therefore, the position of bit 0, frame 0 output on RSER, is slaved to the HDSL receiver selected as master by MASTER\_SEL [CMD\_5; addr 0xE9]. The RSER timing relationship with respect to PCM 6 ms sync is shown in [Figure 3-10](#). PCM 6 ms sync is created from the HDSL 6 ms frame delayed by the programmed RFIFO\_WL [addr 0xCD] value, as shown in [Figure 3-13](#).

**Figure 3-9. PCM Receive Block Diagram**

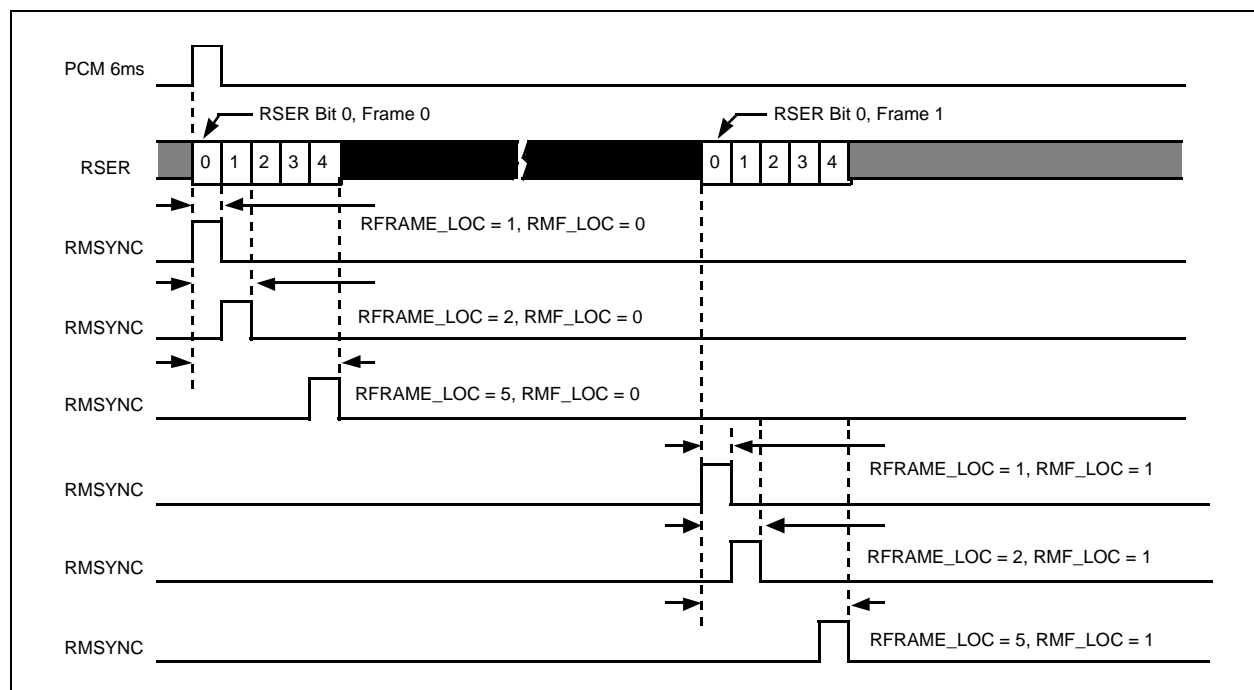


**Figure 3-10. PCM Receive Data Timing**

### 3.2.2.1 Receive Synchronization

The Receive Multiframe Sync (RMSYNC) output can be programmed to mark any bit position within the receive multiframe and does not affect RSER alignment with respect to the PCM 6 ms frame. Figure 3-11 shows the phase offset between PCM 6 ms sync and RMSYNC for various bit and frame delay values [RFRAME\_LOC and RMF\_LOC; addr 0xC3-C5]. The RS8953B does not search receive data for T1, E1, or other specific framing patterns and must always infer PCM receive frame timing from the master HDSL channel's RSYNC reference. When transmit PCM frames are synchronously mapped, the system can program fixed receive delay values for RFRAME\_LOC and RMF\_LOC so that RMSYNC marks the desired RSER bit position. For unframed or asynchronously mapped applications, the RMSYNC output can be ignored, or the remote system can measure transmit phase offset and communicate the necessary phase displacement to the central site.

Figure 3-11. PCM Receive Sync Timing

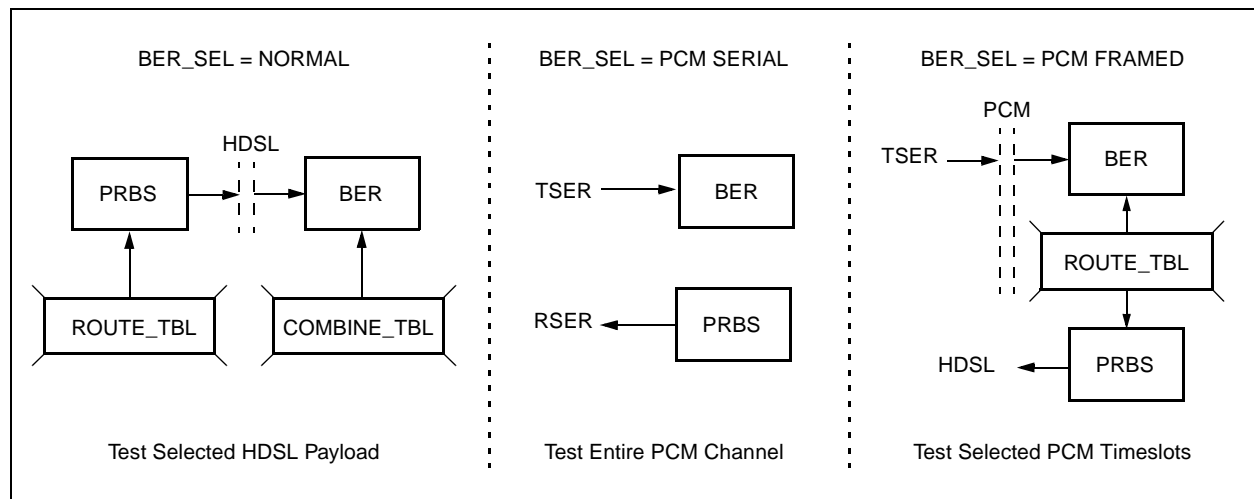


### 3.2.2.2 Receive Combination Table

RSER data output for each PCM timeslot is supplied from one of seven data sources via programmed assignments in the receive Combination Table [COMBINE\_TBL; addr 0xEE]. RSER can be supplied by payload bytes from one of three HDSL receive channels, from fixed 8-bit patterns from one of three Data Bank Registers [DBANK1-3; addr DC-DE] or from groomed Channel Associated Signaling (CAS) from the Receive Signaling Table [RSIG\_TBL; addr 0xF2]. The receive combination table contains up to 64 table entries corresponding to RSER timeslot destinations and each table entry selects one of seven data sources. The first PCM timeslot destination (counting from timeslot 0) that selects a particular HDSL channel's payload byte receives the first payload byte mapped into the RFIFO from that particular HDSL channel's payload block, regardless of whether PCM is synchronously mapped. Asynchronously mapped data is reconstructed into a serial PCM bit stream which maintains bit sequence integrity, provided that the entire PCM channel is formed from combined payload bytes. Each receive combination table entry also selects whether the associated data is copied to the BER meter for test pattern examination.

**3.2.2.3 BER Meter**

PCM timeslots from TSER or RSER can be examined for test patterns on a per timeslot-basis, or the entire framed or unframed PCM channel from TSER can be examined (see PRBS\_MODE in CMD\_3; addr 0xE7 and BER\_SEL in CMD\_6; addr 0xF3). When a test pattern is examined on a per timeslot-basis from receive combination or transmit routing table assignments, the BER meter is only clocked during enabled timeslots and expects a single test pattern to arrive in one sequence from all enabled timeslots. The expected test pattern is selected from one of four Pseudo-Random Bit Sequence (PRBS) patterns or a programmable 8-bit fixed pattern [FILL\_PATT; addr 0xEA]. PRBS pattern selections are:  $2^4-1$ ,  $2^{15}-1$ ,  $2^{23}-1$  and Quasi-Random Signal Sequence (QRSS), where QRSS equals  $2^{20}-1$  PRBS with 14-zero limit. The MPU configures BER\_SCALE [CMD\_3; addr 0xE7] to determine the test measurement interval from a range of  $2^{21}-2^{31}$  bit lengths, starts BER measurement by issuing BER\_RST [addr 0xEF], then monitors test results [BER\_METER; addr 0x1D] and test status [BER\_STATUS; addr 0x1E].

**Figure 3-12. PRBS/BER Measurements**

### 3.2.2.4 RFIFO Water Level

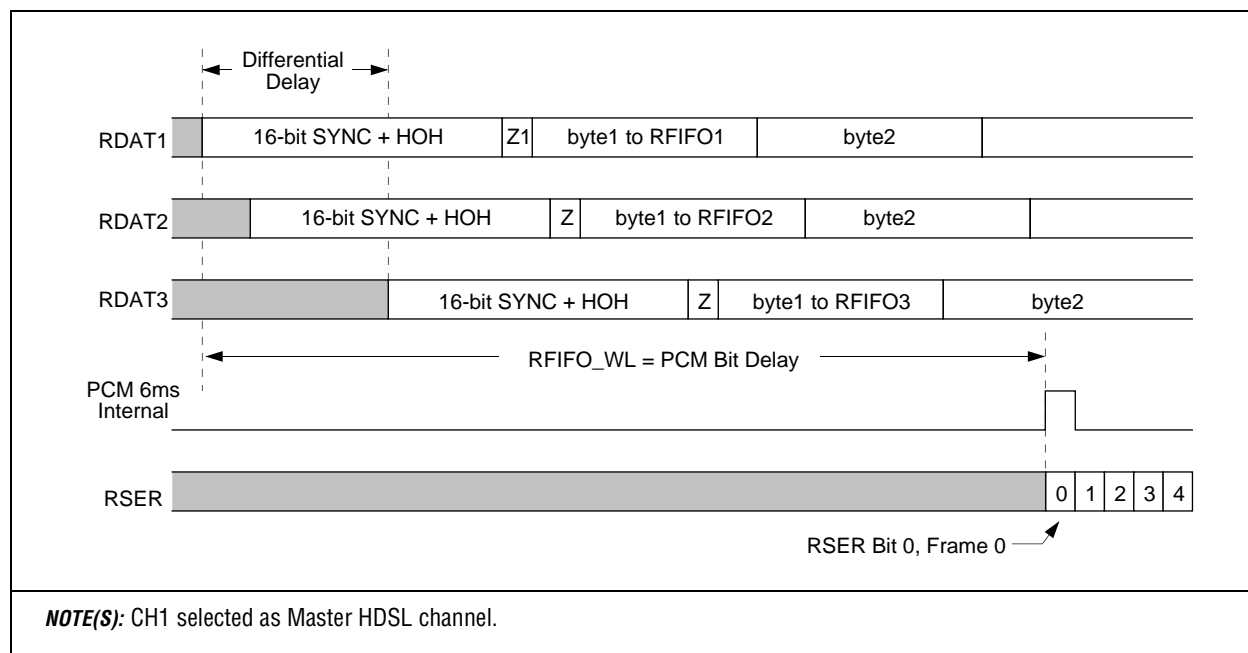
The RFIFO Water Level [RFIFO\_WL; addr 0xCD] determines the PCM and HDSL receiver's phase error tolerance and receive throughput data delay by establishing a fixed phase offset between the master HDSL channel's receive 6 ms frame and the PCM 6 ms sync, as shown in Figure 3-13. RFIFO\_WL selects the number of RCLK bit delays from HDSL to PCM 6 ms frames and controls the amount of time available for the HDSL receiver to map data into the RFIFO before the PCM receiver begins extracting data from the RFIFO. Because all or part of an HDSL payload block can be mapped into a PCM frame, the system must consider Receive Payload Map [RMAP; addr 0x64], Combination Table [COMBINE\_TBL; addr 0xEE] and other data path delays when programming RFIFO\_WL values.

Sufficient phase offset must be established to allow time for HOH, SYNC, and STUFF bit extraction (20 HDSL bits), to load one payload byte (8 HDSL bits), to unload one PCM timeslot (8 PCM bits), to account for differential transmission delay (up to 65  $\mu$ s) and PCM reconstruction (up to 96 PCM bits in T1 mode), and time to tolerate clock variance (1 to 8 PCM bits).

Conversely, to avoid RFIFO overflow, phase offset must be limited so the amount of data residing in the RFIFO never exceeds the number of PCM bits mapped during one PCM frame, the maximum RFIFO depth (185 bits), or the total HDSL payload block length [HFRAME\_LEN; addr 0xCA].

The actual phase offset between HDSL and PCM 6 ms frames varies over time as a result of STUFF bit extraction, clock variance, and differential phase delays. Therefore, RFIFO\_WL is only used to establish the initial phase offset between HDSL and PCM receive frames when the MPU issues the Reset Receiver command [RX\_RST; addr 0xF1].

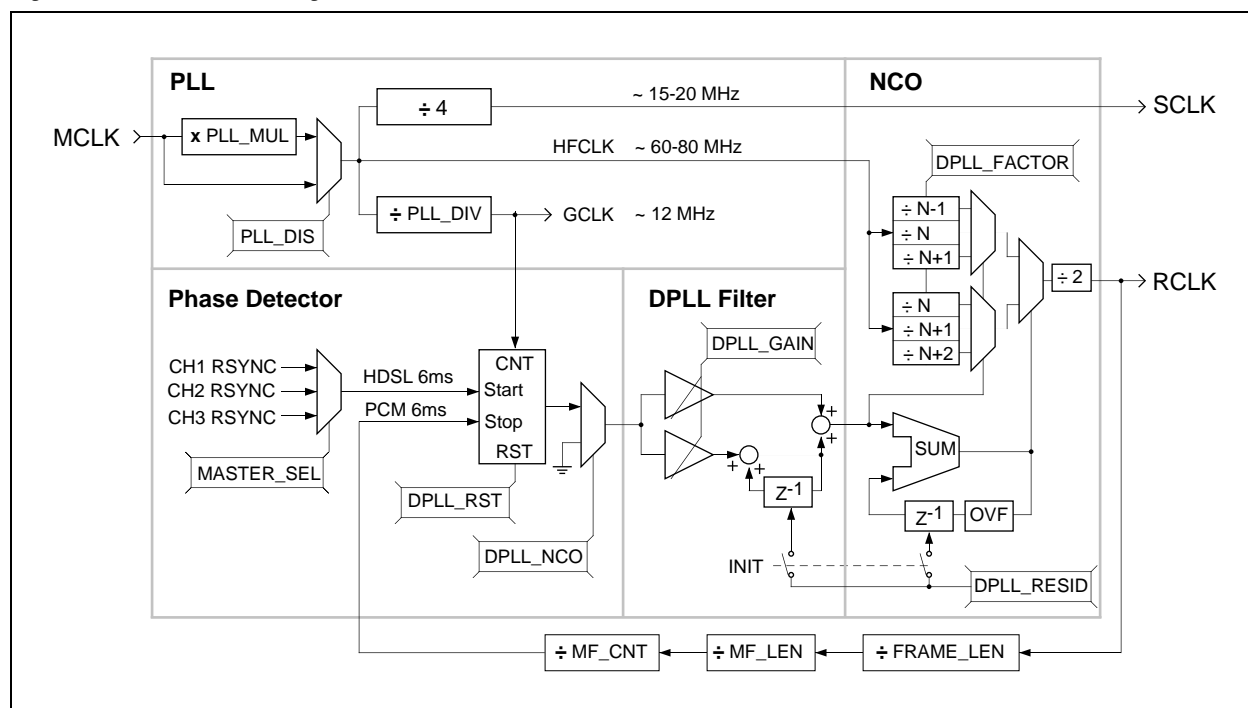
**Figure 3-13. RFIFO Water Level Timing**



### 3.3 Clock Recovery DPLL

The Digital Phase Locked Loop (DPLL) shown in [Figure 3-14](#) synthesizes the PCM Receive Clock (RCLK) from a 60–80 MHz High Frequency Clock (HFCLK). HFCLK is developed by analog PLL multiplication of the MCLK input frequency, or HFCLK is applied directly to the MCLK input (see PLL\_MUL and PLL\_DIS in CMD\_1; addr 0xE5). The analog PLL requires external loop filter components and connections as shown in Figure 6-1. HFCLK must be in the range of 60–80 MHz, but requires no specific frequency or phase relationship to PCM or HDSL clocks. Open or closed loop operation is selected by DPLL\_NCO [CMD\_5; addr E9].

**Figure 3-14. DPLL Block Diagram**



In closed loop operation, the Numerical Controlled Oscillator (NCO) synthesizes the nominal RCLK frequency according to the programmed HFCLK integer scale factor [DPLL\_FACTOR; addr 0xD7] and the fractional scale factor [DPLL\_RESID; addr 0xD5]. The NCO locks the RCLK frequency to the HDSL reference by varying the RCLK phase based on the filtered phase error from the DPLL filter and the DPLL phase detector. Phase error is the phase difference measured from the receive PCM 6 ms sync to the master HDSL channel's 6 ms frame. Phase error is quantized in units of GCLK, where GCLK is set to approximately 12 MHz the from division of HFCLK by the programmed value of PLL\_DIV [CMD\_1; addr 0xE5]. The phase detector measures and reports the Phase Error [PHS\_ERR; addr 0x38] coincident with the master HDSL channel's receive 6 ms frame interrupt. The phase detector automatically reinitializes, if phase error exceeds  $\pm 511$  GCLK cycles according to the initialization mode selected by PHD\_MODE [CMD\_7; addr 0xF4]. The DPLL filter is a Type II digital filter in which the gain [DPLL\_GAIN; addr 0xD8] determines the closed loop DPLL filter bandwidth.

During open loop operation, the NCO synthesizes the RCLK frequency according to the programmed HFCLK integer and fractional scale factors, but ignores phase detector error outputs. In this case, RCLK frequency accuracy is dependent on HFCLK accuracy ( $\pm 20$  ppm) and programmed scale factor accuracy ( $\sim 2$  Hz). Open loop operation is useful during remote HTU applications to provide a stable RCLK output frequency while HDSL channels are performing startup activities.



## 3.4 Loopbacks

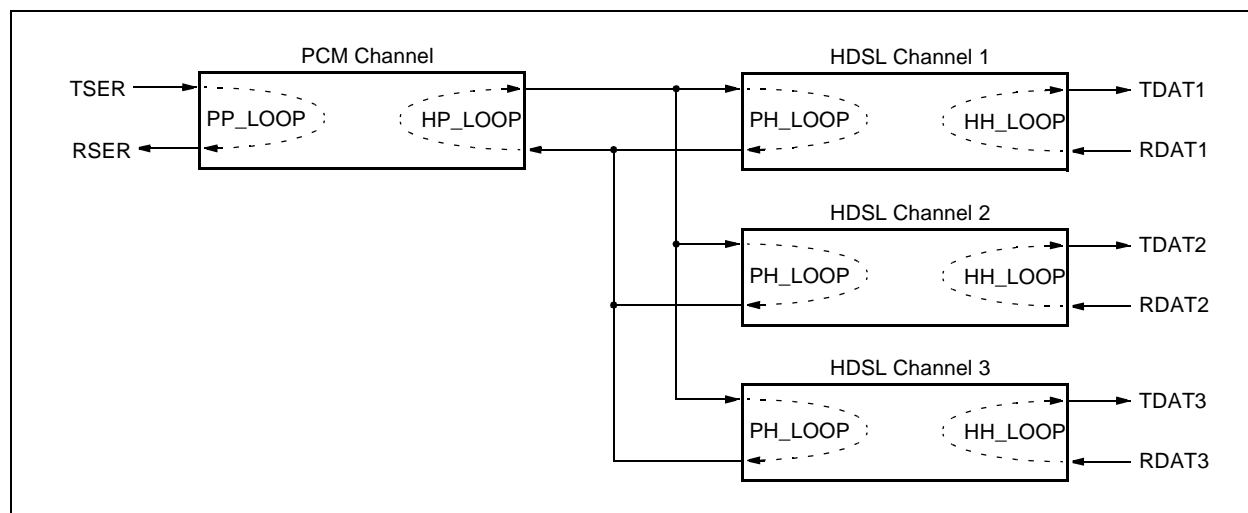
The RS8953B provides multiple PCM and HDSL loopbacks, as shown in [Figure 3-15](#). The output towards which data is looped is called the test direction. Loopback activation in the test direction does not disrupt the through data path in the non-test direction. Data path options (refer to [Table 4-7](#)) are provided to replace data in the non-test direction with fixed or PRBS test patterns. [Table 3-1](#) shows the loopback controls which are designated by initials corresponding to test direction and the channel from which data is looped.

PP\_LOOP and HP\_LOOP automatically switch both PCM data and PCM multiframe sync signals to the test direction. In these loopback modes, the PCM transmit and receive clocks are not automatically switched to the test direction. The PCM transmit and receive clocks must be properly configured for these loopback modes to operate.

When performing PH\_LOOP, all HDSL channels that have payload data mapped require that PH\_LOOP mode be enabled to complete PCM channel loopback on the HDSL side. Also, the same tap must be used for the RS8953B scrambler and descrambler, or both the RS8953B scrambler and descrambler must be disabled.

When performing HH\_LOOP, the scrambler and descrambler in the HDSL transceivers must be enabled. Different tap must be used in each direction. This is required to prevent the exact same data to be sent in both directions. If the exact same data is sent in both directions, the echo canceler in the HDSL transceivers will consider the data an echo and cancel the data. Also, the same tap must be used for the RS8953B scrambler and descrambler, or both the RS8953B scrambler and descrambler must be disabled.

**Figure 3-15. PCM and HDSL Loopbacks**



**Table 3-1. PCM And HDSL Loopbacks**

Loopback	Command Register	Test Direction	Loopback Description
PP_LOOP	CMD_2; addr 0xE6	Receive	PCM Loopback on PCM Side
HP_LOOP	CMD_2; addr 0xE6	Transmit	HDSL Loopback on PCM Side
PH_LOOP	RCMD_2; addr 0x61	Receive	PCM Loopback on HDSL Channel 1
PH_LOOP	RCMD_2; addr 0x81	Receive	PCM Loopback on HDSL Channel 2
PH_LOOP	RCMD_2; addr 0xA1	Receive	PCM Loopback on HDSL Channel 3
HH_LOOP	TCMD_2; addr 0x07	Transmit	HDSL Loopback on HDSL Channel 1
HH_LOOP	TCMD_2; addr 0x27	Transmit	HDSL Loopback on HDSL Channel 2
HH_LOOP	TCMD_2; addr 0x47	Transmit	HDSL Loopback on HDSL Channel 3

### 3.5 HDSL Channel

The three identical HDSL channels (CH1, CH2, and CH3) consist of separate transmit and receive circuits that are responsible for the assembly of HDSL output frames and the disassembly of HDSL receive frames. The basic structure of an HDSL frame is shown in [Table 3-2](#). Each frame is nominally 6 ms in length and consists of 48 payload blocks with each block containing a single Z-bit, plus an application-specific number of payload bytes. The MPU selects the desired payload block length in HFRAME\_LEN [addr 0xCA], in which the length is programmed to equal the number of payload and Z-bits. Groups of 12 payload blocks are concatenated and separated by an ordered set of HDSL overhead bits, in which a 14-bit SYNC word pattern identifies the starting location of the HDSL frame. 50 overhead bits are defined in one HDSL frame, but the last 4 STUFF (sq1–sq4) bits are nominally present in alternate frames. Therefore, one frame contains an average of 48 overhead bits.

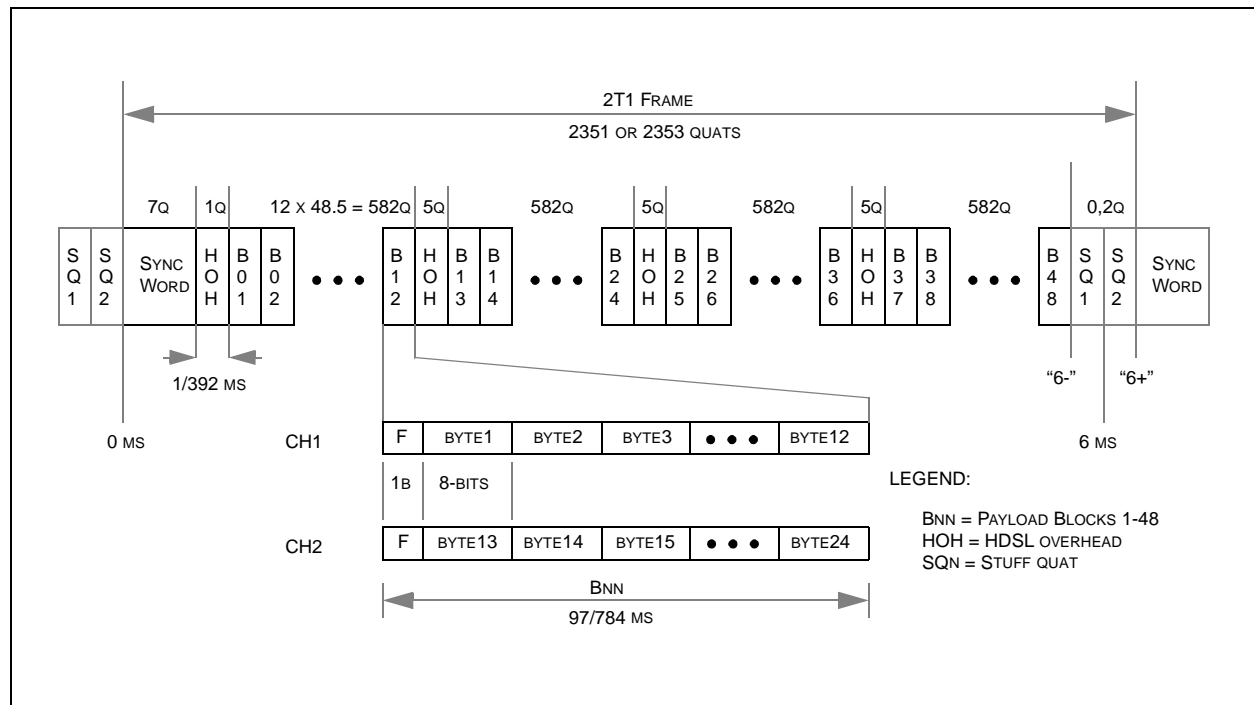
**Table 3-2. HDSL Frame Structure and Overhead Bit Allocation**

HOH Bit #	Symbol	Bit Name	HOH Register Bit
1–14	sw1–sw14	SYNC word	—
15	losd	Loss of Signal	IND[0]
16	febe	Far End Block Error	IND[1]
Payload Blocks 1–12			
17–20	eoc1–eoc4	Embedded Operations Channel	EOC[0]–EOC[3]
21–22	crc1–crc2	Cyclic Redundancy Check	—
23	ps1	HTU-R Power Status	IND[2]
24	ps2	Power Status Bit 2	IND[3]
25	bpv	Bipolar Violation	IND[4]
26	eoc5	Embedded Operations Channel	EOC[4]
Payload Blocks 13–24			
27–30	eoc6–eoc9	Embedded Operations Channel	EOC[5]–EOC[8]
31–32	crc3–crc4	Cyclic Redundancy Check	—
33	hrp	HDSL Repeater Present	IND[5]
34	rrbe	Repeater Remote Block Error	IND[6]
35	rcbe	Repeater Central Block Error	IND[7]
36	rega	Repeater Alarm	IND[8]
Payload Blocks 25–36			
37–40	eoc10–eoc13	Embedded Operations Channel	EOC[9]–EOC[12]
41–42	crc5–crc6	Cyclic Redundancy Check	—
43	rta	Remote Terminal Alarm	IND[9]
44	rtr	Ready to Receive	IND[10]
45	uib	Unspecified Indicator Bit	IND[11]
46	uib	Unspecified Indicator Bit	IND[12]
Payload Blocks 37–48			
47	sq1	Stuff Quat Sign	STUFF[0]
48	sq2	Stuff Quat Magnitude	STUFF[1]
49	sq3	Stuff Quat Sign	STUFF[2]
50	sq4	Stuff Quat Magnitude	STUFF[3]

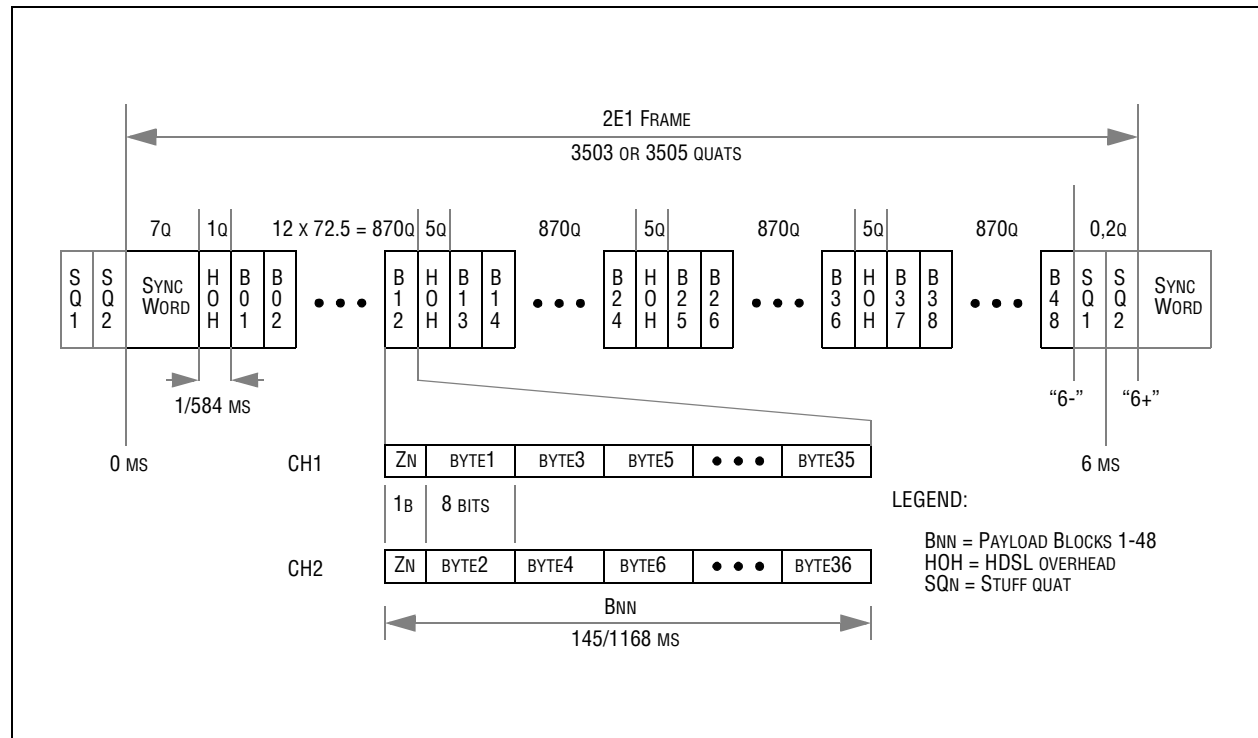
In T1 framing mode [E1\_MODE = 0 in CMD\_1; addr 0xE5], Z-bit positions are replaced by F-bits and are treated as payload with respect to the PCM channel. [Figure 3-16](#) shows a standard application 2T1 frame format in which each payload block contains 1 F-bit, plus 12 payload bytes. The figure also illustrates F-bits routed as payload to both HDSL channels and demonstrates the order in which PCM timeslots are routed to payload bytes: bytes 1 through 12 correspond to PCM timeslots 1–12 routed on CH1, bytes 13 through 24 correspond to PCM timeslots 13–24 routed on CH2. CH3 is unused in 2T1 application.

Standard application 2E1 and 3E1 frame formats are shown in [Figure 3-17](#) and [Figure 3-18](#), respectively. Standard mapping of PCM data places alternating bytes in each HDSL channel, as shown by byte numbering. The 2E1 payload block contains 18 payload bytes, and the 3E1 payload block contains 12 bytes. In E1 framing mode [E1\_MODE = 1 in CMD\_1; addr 0xE5], 48 Z-bits are treated as overhead and are under MPU control. (Refer to [Table 3-5](#) for Z-bit definitions). [Table 3-3](#) shows additional examples of frame mapping options.

**Figure 3-16. 2T1 Frame Format**



**Figure 3-17. 2E1 Frame Format**



**Figure 3-18. 3E1 Frame Format**

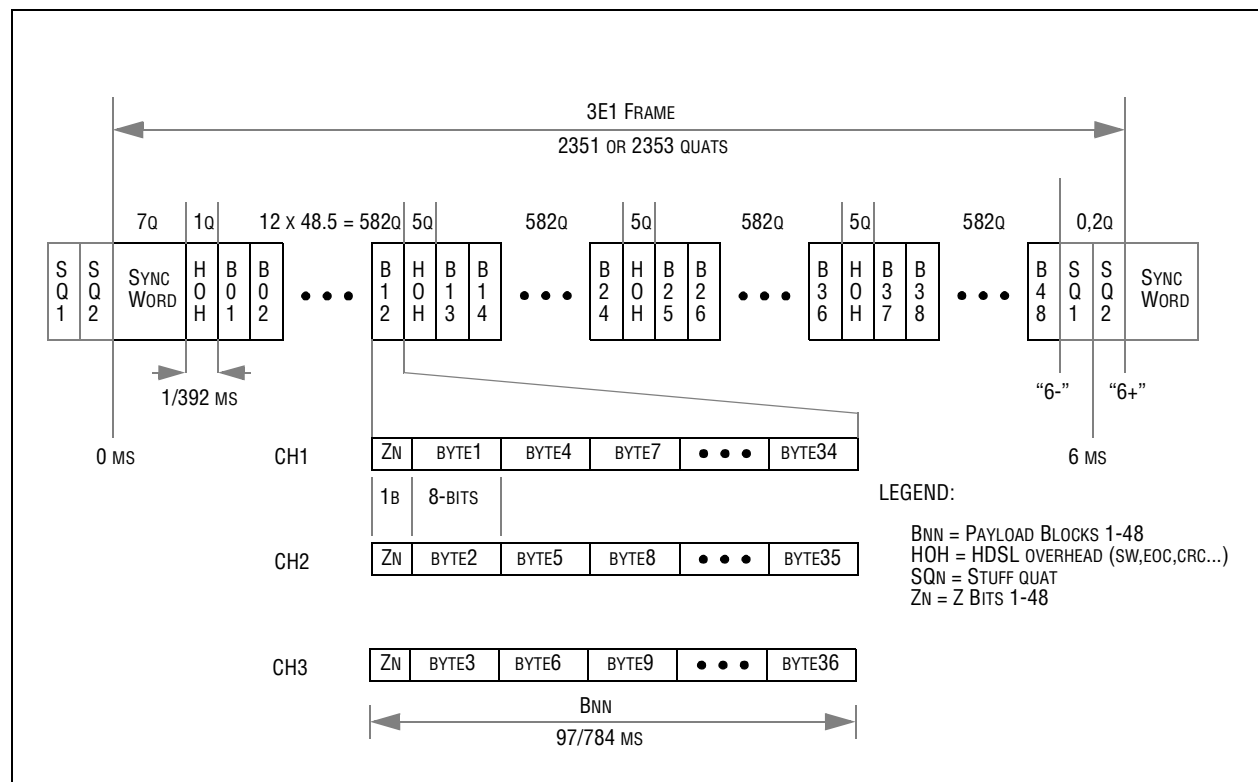


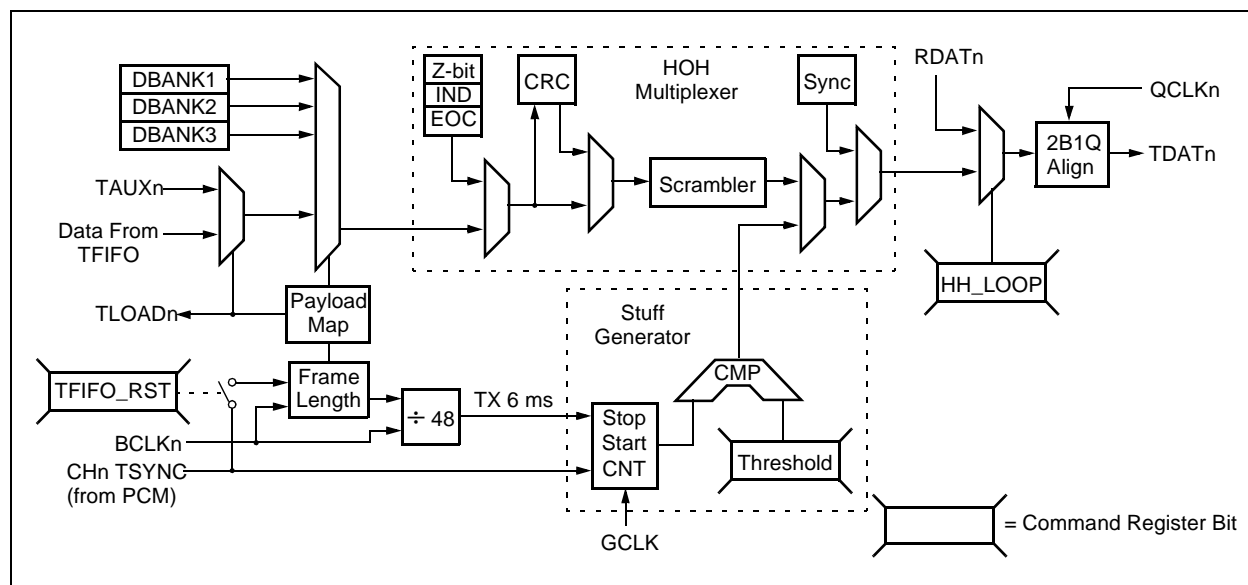
Table 3-3. HDSL Frame Mapping Examples

	Payload	2E1	VC-12	3E1-P2MP
Payload Block (B)	BYTE1	R	V5	Channel 0
	BYTE2	R	R	Channel 0
	BYTE 3-35	32 BYTES	32 BYTES	Channel 0
				Channels 1-15
				Channel 16
Payload Block (B+1)	BYTE36	R	R	Channel 16
	BYTE37	Y	Y	Channel 16
	BYTE 38-71	32 BYTES	C1 C2 0000 RR	Channels 17-31
				Channel 0
				Channel 0
Payload Block (B+2)	BYTE 72	32 BYTES	32 BYTES	Channel 0
				Channels 1-15
				Channel 16
				Channel 16
				Channel 16
Payload Block (B+3)	BYTE 73-107	R	R	Channels 17-31
		Y	Y	Channel 0
		32 BYTES	C1 C2 0000 RR	Channel 0
				Channel 0
				Channels 1-15
Payload Block (B+3)	BYTE108	R	R	Channel 16
	BYTE109-143	Y	Y	Channel 16
		32 BYTES	32 BYTES	Channel 16
				Channel 16
				Channel 16
Payload Block (B+3)	BYTE 144	R	R	Channels 17-31
		Y	Y	Channel 0
				Channel 0
Payload Block (B+3)			C1 C2 0000	Channel 0
			S2 IIIIII	Channels 1-15
			31 BYTES	Channel 16
Payload Block (B+3)				Channel 16
				Channel 16
				Channel 16
Payload Block (B+3)			R	Channels 17-31
			Y	Channel 0
				Channel 0

### 3.5.1 HDSL Transmit

Three identical HDSL transmitters accept data and sync from the PCM channel, insert HDSL overhead, and output serially encoded 2B1Q data on TDATn. One HDSL transmitter, shown in [Figure 3-19](#), consists of a transmit payload mapper, HOH multiplexer, STUFF generator and 2B1Q encoder. All transmitter circuits are clocked by BCLKn, where n corresponds to HDSL channels numbered 1, 2, or 3. The HDSL transmit timebase develops 6 ms frame timing based upon the programmed HFRAME\_LEN [addr 0xCA] and initial phase alignment established from PCM transmit 6 ms sync plus the TFIFO\_WL delay. Each HDSL transmitter automatically manages SYNC, STUFF, and CRC overhead protocols and provides the MPU with write register access for insertion of IND, EOC, and Z-bit overhead bits, but does not automatically manage IND, EOC, or Z-bit protocols.

**Figure 3-19. HDSL Transmitter Block Diagram**



### 3.5.1.1 Transmit Payload Mapper

The transmit payload mapper controls the contents of HDSL transmit payload blocks by selecting data for each payload byte from one of five data sources according to selections made in the TMAP Registers [TMAP\_1; addr 0x08]. TMAP selects one of five sources for each byte within the payload block: PCM timeslot or F-bit data from the TFIFO, one of three fixed pattern Data Bank Registers (DBANK1–DBANK3), or data sampled from the HDSL auxiliary input (TAUXn).

### 3.5.1.2 HOH Multiplexer

Placement of HDSL Overhead (HOH) bits in the output frame is performed by the HOH multiplexer. HOH bits are grouped into the following categories: SYNC, IND, EOC, CRC, STUFF, and Z-bits. (Refer to [Table 3-2](#) for HOH bit positions within the output frame.) The MPU controls the contents of the HOH bits by writing SYNC\_WORD [addr 0xCB], TIND, TEOC, TZBIT (see [Table 4-2](#)) and TSTUFF [addr 0xE4] register values. CRC bits are calculated autonomously and inserted into the appropriate HOH bit positions.



**3.5.1.3 CRC Calculation**

The Cyclic Redundancy Check (CRC) calculation is performed on all transmit data, and the HOH multiplexer inserts the resulting 6-bit CRC into the subsequent output frame. CRC is calculated over all bits in the (N)th frame except the SYNC, STUFF, and CRC bits, and then is inserted into the (N+1)th frame. The MPU can choose to inject CRC errors on a per frame-basis by setting ICRC\_ERR [TCMD\_1; addr 0x07]. The six CRC bits are calculated as follows:

1. All bits of the (N)th frame, except the 14 SYNC, 6 CRC, and any STUFF bits are used to calculate CRC. A total of 4,682 bits are used, in order of occurrence, to construct a polynomial in X, such that bit 0 of the (N)th frame is the coefficient of the term  $X^{4681}$  and bit 4681 of the (N)th frame is the coefficient of the term  $X^0$ .
2. The polynomial is multiplied by the factor  $X^6$  and the result is divided, modulo 2, by the generator polynomial  $X^6+X+1$ . Coefficients of the remainder polynomial are used, in order of occurrence, as an ordered set of check bits, CRC1–CRC6, for the (N+1)th frame. Ordering is such that the coefficient of term  $X^5$  in the remainder polynomial is check bit CRC1, and the coefficient of term  $X^0$  is check bit CRC6.
3. Check bits CRC1–CRC6 contained in a frame are associated with the contents of the preceding frame. When there is no immediately preceding frame, check bits may be assigned any value.

**3.5.1.4 Scrambler**

The scrambler operates at the BCLKn bit rate on all HDSL transmit data, except for the 14-bit SYNC words and the four STUFF bits. The MPU enables the scrambler by setting SCR\_EN [TCMD\_1; addr 0x06] and selects the scrambler algorithm in SCR\_TAP [TCMD\_2; addr 0x07]. Two scrambler algorithms are implemented for HTU-R or HTU-C data transmission:

- In the HTU-R to HTU-C direction, the polynomial shall be  $X^{-23} \oplus X^{-18} \oplus 1$ , where  $\oplus$  is equal to modulo 2 summation.
- In the HTU-C to HTU-R direction, the polynomial shall be  $X^{-23} \oplus X^{-5} \oplus 1$ , where  $\oplus$  is equal to modulo 2 summation.

**3.5.1.5 STUFF Generator**

Transmit bit stuffing synchronizes the HDSL channel's transmit 6 ms frame period to the PCM channel's 6 ms sync by adding 0 or 4 STUFF bits to the HDSL output frame. The STUFF generator decides whether 0 or 4 STUFF bits are inserted and reports the result of each decision in TX\_STUFF [STATUS\_3; addr 0x07]. When 4 STUFF bits are inserted, sign/magnitude values are taken from TSTUFF [addr 0xE4]. Stuffing decisions are based on comparison of the phase difference measured between PCM and HDSL 6 ms frame intervals in relation to the programmed STUFF thresholds [STF\_THRESH\_B; addr 0xD1] and threshold [STF\_THRESH\_C; addr 0xD3]. If the measured phase difference is equal to or less than threshold B, then no STUFF bits are inserted for that output frame. If the measured phase difference exceeds threshold B and is less than or equal to threshold C, then 4 STUFF bits are inserted. When the measured phase difference exceeds threshold C, the STUFF generator reports a transmit Stuffing Error, STUFF\_ERR [STATUS\_3; addr 0x07] and automatically resets the transmit FIFO by performing the TFIFO\_RST [addr 0x0D] command.

The MPU can bypass the STUFF generator and select an alternate source of transmit STUFF bits by setting SLV\_STUF [TCMD\_2; addr 0x07] and selecting the alternate source in STUFF\_SEL [CMD\_5; addr 0xE9]. Alternate STUFF bits can be supplied by other HDSL channels, or the MPU can directly manipulate EXT\_STUFF [CMD\_5; addr 0xE9]. For systems that externally synchronize PCM and HDSL clock phase, the STUFF generator can also be programmed to insert an alternating pattern of 0 and 4 STUFF bits.

### 3.5.1.6 2B1Q Encoder

The 2B1Q (2 Binary, 1 Quaternary) encoder provides the ability to directly interface to the Conexant HDSL transceiver. The 2B1Q encoder converts HDSL data generated internally at the bit rate into sign and magnitude data according to the quaternary alignment provided on the QCLKn input. (Refer to [Table 3-4](#) for sign and magnitude bits used to generate 2B1Q coded outputs on TDATn.)

**Table 3-4. 2B1Q Encoder Alignment**

First Bit (Sign)	Second Bit (Magnitudes)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1
0	1	–1
0	0	–3

**Table 3-5. Z-Bit Definitions**

Z-bit (Zn)	Loop1	Loop2	Loop3	Comments
1	1	0	0	Pair Identification
2	0	1	0	Pair Identification
3	0	0	1	Pair Identification
4	x	x	x	Not defined
5	x	x	x	Not defined
6	x	x	x	Not defined
7	x	x	x	Not defined
8 to 46	x	x	x	Not defined
47	x	x	x	Not defined
48	x	x	x	Not defined

3.5.1.7 HDSL Auxiliary Transmit

The HDSL auxiliary transmit channel provides an alternate source of HDSL payload bytes and optionally, an alternate source for the last 40 Z-bits transmitted in each HDSL frame. Auxiliary transmit data (TAUXn) is sampled by BCLKn whenever TLOADn is active-high, as shown in Figures 3-20 and 3-21. TLOADn is enabled by TAUX\_EN [TCMD\_2; addr 0x07] and programmed in the Transmit Payload Map Registers [TMAP; addr 0x08]. TLOADn marks specific payload bytes selected in the TMAP registers or marks the last 40 Z-bits depending on the setting of EXT\_ZBIT [TCMD\_2; addr 0x07].

Figure 3-20. HDSL Auxiliary Channel Payload Timing

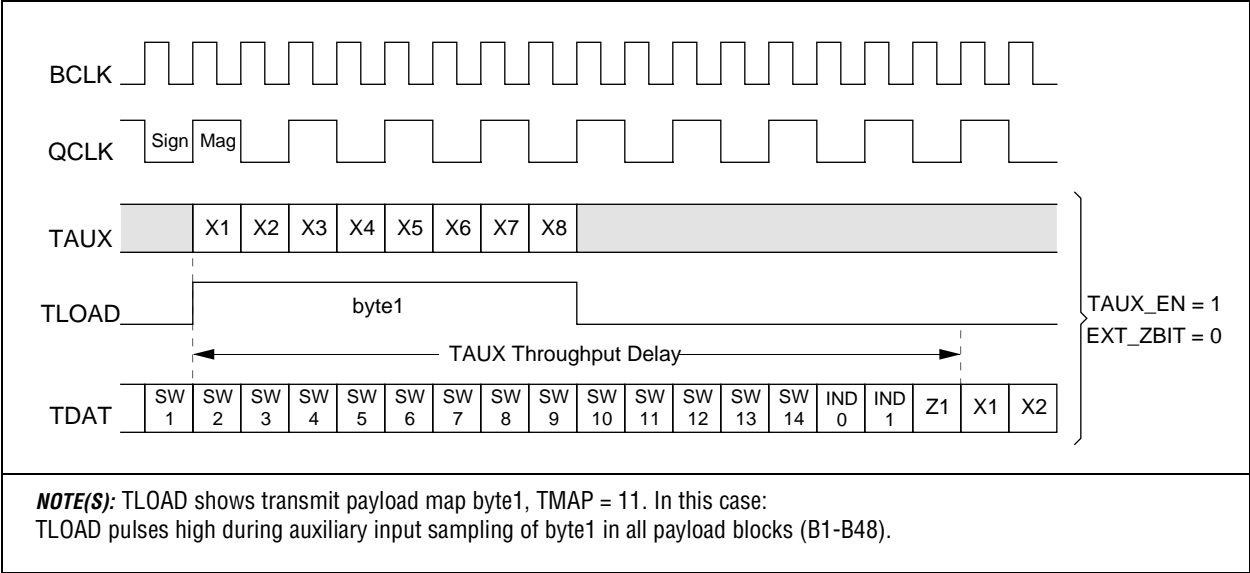
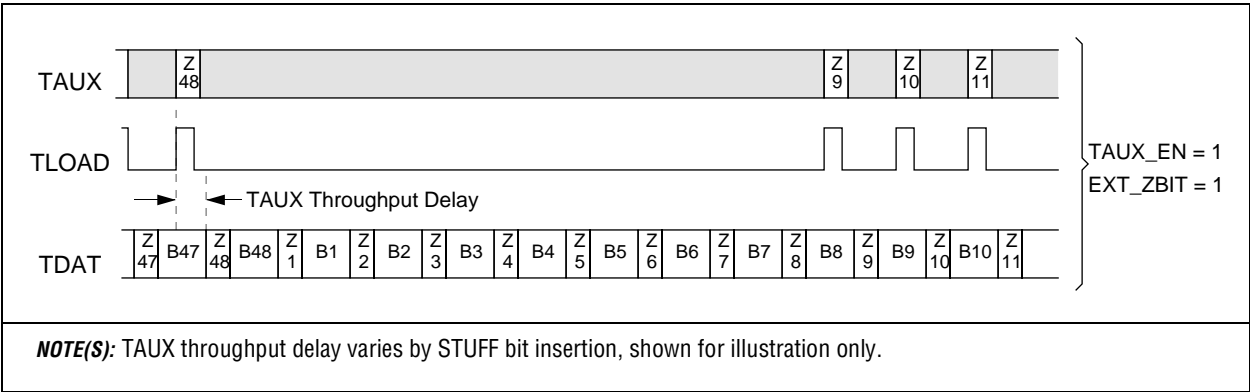


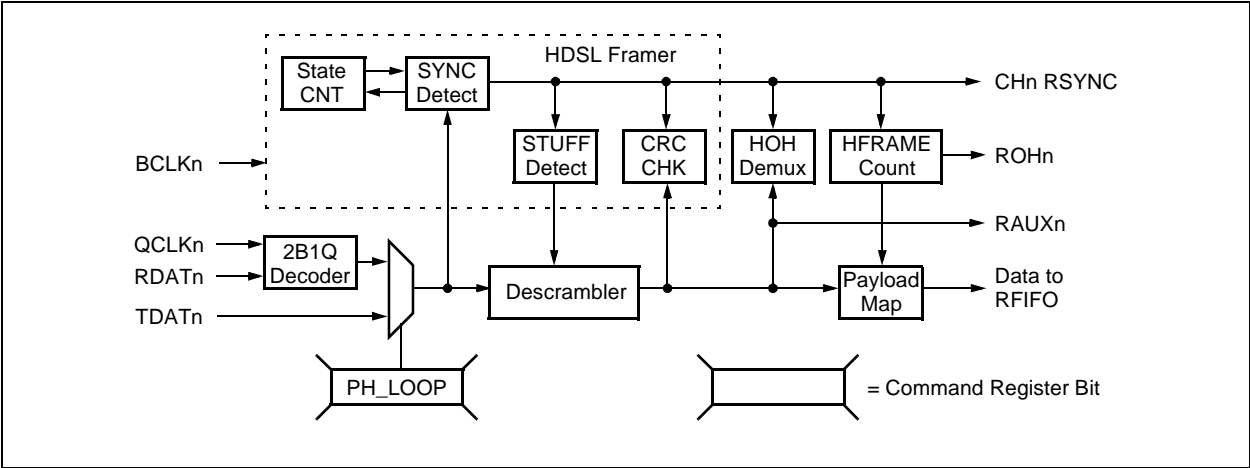
Figure 3-21. HDSL Auxiliary Channel Z-bit Timing



3.5.2 HDSL Receive

The RS8953B contains three identical HDSL receivers, each receiver the same as the one shown in Figure 3-22. The receiver is responsible for frame alignment, destuffing, overhead extraction, descrambling of payload data, error performance monitoring, and payload mapping of HDSL data from received frames into the RFIFO. The receive framer monitors incoming HDSL data to locate SYNC words and to identify frame boundaries for use by other circuits that locate and remove bit stuffing, to check CRC errors, to extract HOH bits and to map payload data to the RFIFO. One of the receivers is configured to act as master reference for the PCM receive channel and from which T1 framing bits are extracted (see MASTER\_SEL, CMD\_5; addr 0xE9). The master channel also supplies its 6 ms frame reference for DPLL clock recovery.

Figure 3-22. HDSL Receiver Block Diagram



3.5.2.1 2B1Q Decoder

The 2B1Q decoder provides the capability to directly connect to the Conexant HDSL transceiver. The 2B1Q decoder samples and aligns the incoming sign and magnitude data. (Refer to Table 3-6 for 2B1Q mapping.) All three HDSL channels operate independent of one another to allow separate, asynchronous clock signals, to be applied from the system at each HDSL interface.

Table 3-6. 2B1Q Decoder Alignment

First Bit (Sign)	Second Bit (Magnitudes)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1
0	1	-1
0	0	-3

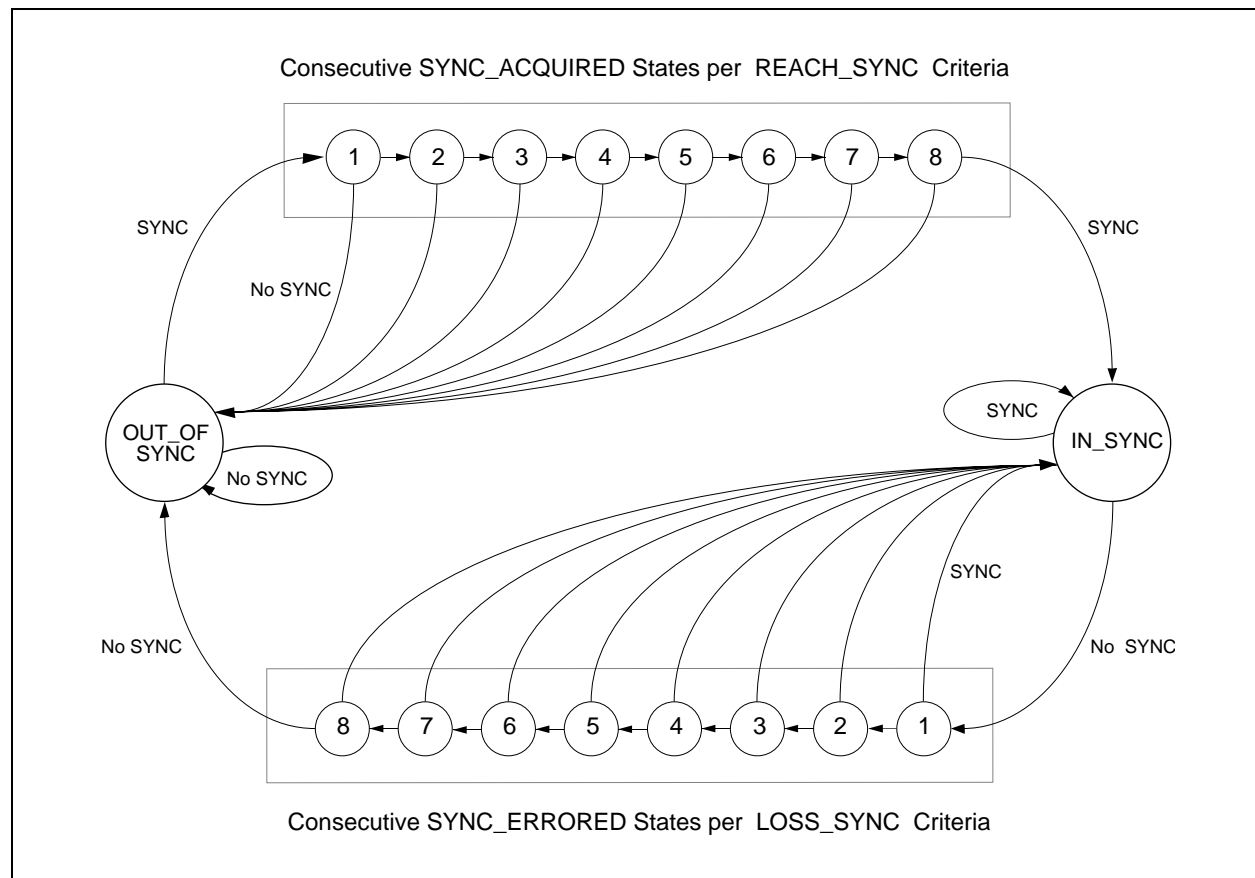
### 3.5.2.2 HDSL Receive Framer

The HDSL receive framer acquires and maintains synchronization of the HDSL channel and generates pointers that control overhead extraction in the STUFF, CRC and HOH demux circuitry. The MPU initializes the framer to the Out Of Sync state by writing any data value to SYNC\_RST [addr 0x63]. From the Out Of Sync state, the framer advances to Sync Acquired when a correct SYNC word is detected. The framer searches all bits received on RDATn to locate a match with one or both of the SYNC word patterns, SYNC\_WORD\_A [addr 0xCB] or SYNC\_WORD\_B [addr 0xCC], according to the selection made by FRAMER\_EN [RCMD\_1; addr 0x60].

For T1 applications, the framer is programmed to search for two different sync word values because separate sync words are transmitted on each HDSL channel to specify the wire pair number. During E1 applications, ETSI requires a common sync word be used for all pairs and Z-bits used to define the wire pair number, although the framer may still be programmed to search for two different sync words in non-standard E1 applications. Due to the possibility of tip/ring connector reversal on each wire pair, all sign bits received on RDATn might be inverted. Therefore, the receive framer searches for both the programmed sync word value and the sign-inverted sync word value. Consequently a maximum of four values of the sync word are used in finding the frame location. If the sync word detected is a sign inverted version of one of the configured sync words, the framer sets the Tip/Ring Inversion (TR\_INVERT) status bit [STATUS\_1; addr 0x05] and automatically inverts the sign of all quats received on RDATn.

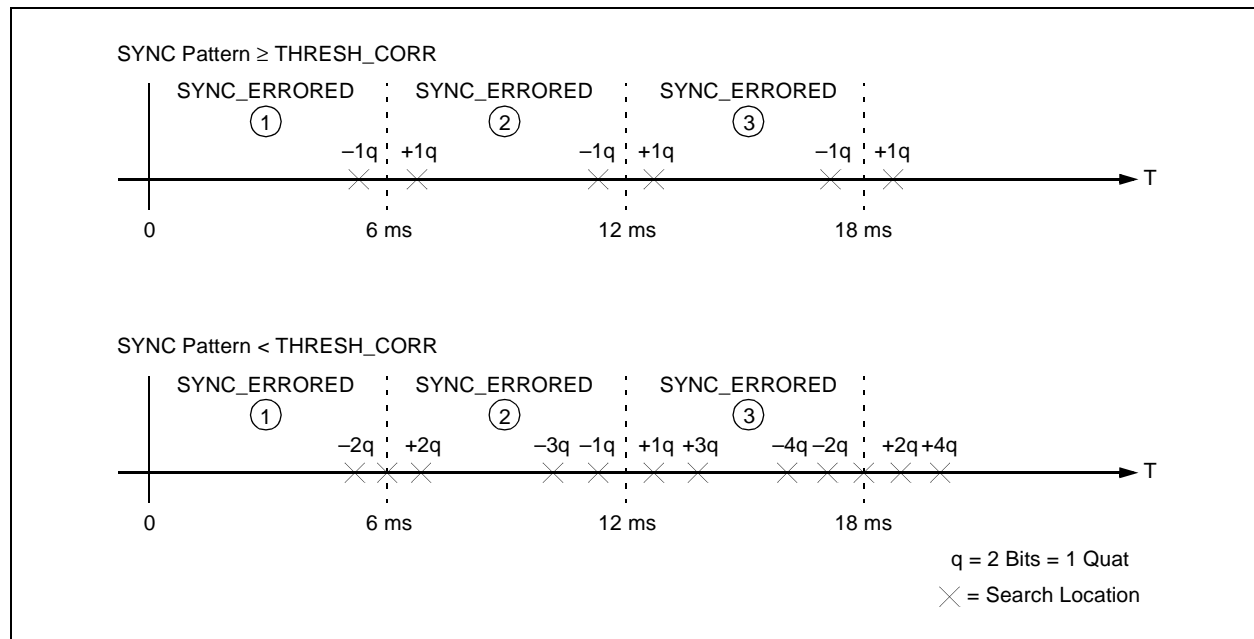
After detecting a sync word and changing to the Sync Acquired state, the framer progresses through a programmable number of intermediate “Sync Acquired” states before entering the In Sync state. In each “Sync Acquired” state, the framer searches for the previously detected sync word value in one of two locations based upon the absence or presence of the 4 STUFF bits. If the sync word is detected in one of the two possible locations, the STATE\_CNT counter is incremented [STATUS\_2; addr 0x06]. When STATE\_CNT increments to the value selected by the REACH\_SYNC criteria [RCMD\_1; addr 0x60], the framer changes to the In Sync state. During the Sync Acquired state, if valid sync is not detected at one of the two possible locations, the framer returns to the Out Of Sync state, as shown in [Figure 3-23](#).

**Figure 3-23. HDSL Receive Framer Synchronization**



After entering In Sync, the framer either remains In Sync as successive sync words are detected, or regresses to the Sync Errored state if sync pattern errors are found. During Sync Errored states, the number of matching bits from each comparison of received sync word and programmed sync word patterns must meet or exceed the programmed pattern match tolerance specified by THRESH\_CORR [RCMD\_2; addr 0x61]. If the number of matching bits falls below tolerance, the framer expands the locations searched to quats on either side of the expected location, as shown in Figure 3-24. After detecting a sync pattern error and changing to the Sync Errored state, the framer passes through a programmable number of intermediate Sync Errored states, before entering the Out Of Sync state. STATE\_CNT increments for each frame in which sync is not detected until the count reaches the LOSS\_SYNC criteria [RCMD\_1; addr 0x60] and the framer enters the Out Of Sync state. If at any time during the Sync Errored state the framer detects a completely correct sync word pattern at one of the valid frame locations, then the framer returns to the In Sync state. The ETSI standard recommends the REACH\_SYNC = 2 and LOSS\_SYNC = 6 framing criteria.

**Figure 3-24. Threshold Correlation Effect on Expected Sync Locations**



### 3.5.2.3 Descrambler

The descrambler operates at the BCLKn bit rate on all HDSL receive data, except for the 14-bit SYNC words and 4 STUFF bits. The MPU enables the descrambler by setting the DSCR\_EN bit and selects the descrambler algorithm via DSCR\_TAP [RCMD\_2; addr 0x61]. Two descrambling algorithms are implemented as follows:

- In the HTU-R to HTU-C direction, the polynomial shall be  $X^{-23} \oplus X^{-18} \oplus 1$ , where  $\oplus$  is equal to modulo 2 summation.
- In the HTU-C to HTU-R direction, the polynomial shall be  $X^{-23} \oplus X^{-5} \oplus 1$ , where  $\oplus$  is equal to modulo 2 summation.

**3.5.2.4 CRC Checking**

The Cyclic Redundancy Check (CRC) error is reported each time the calculated CRC of the (N)th HDSL frame does not match the CRC received in the (N+1)th HDSL frame. Individual block errors are reported in CRC\_ERROR [STATUS\_2; addr 0x06] and are accumulated in CRC\_CNT [addr 0x21]. Each HDSL receiver calculates CRC in the same manner as described for the transmitter.

**3.5.2.5 HOH Demux**

HDSL Overhead (HOH) bits are grouped into the following categories: SYNC, IND, EOC, CRC, and Z-bits. (Refer to [Table 3-2](#) for HOH bit positions within the frame.) HOH demux extracts IND, EOC, and Z-bits from each receive frame and places them into MPU accessible read registers RIND, REOC, and RZBIT (see [Table 4-10](#)). The MPU must read the contents of the HOH registers every 6 ms, or as noted. Otherwise, data is overwritten by new received data.

**3.5.2.6 Receive Payload Mapper**

The receive payload mapper controls placement of receive payload bytes and Z-bits into the RFIFO as programmed by the RMAP Registers [RMAP; addr 0x64]. The payload mapper aligns itself to incoming HDSL 6 ms frames and selectively transfers payload bytes from the received payload block.

**3.5.2.7 HDSL Auxiliary Receive**

The HDSL auxiliary receive channels allow the system to monitor the receive HDSL payload and overhead bits output from the descrambler on RAUXn. The entire received HDSL unscrambled bit stream is output on RAUXn at the BCLKn rate. The MPU selects which category of RAUXn data is marked by ROHn according to programmed values for RAUX\_EN and RAZ [CMD\_6; addr 0xF3]. ROHn either marks all overhead bits (STUFF, SYNC, HOH, and Z-bits) as shown in [Figure 3-25](#), or marks only the last 40 Z-bits, as shown in [Figure 3-26](#). The system can externally decode ROHn to access specific payload bytes or overhead bits, or to qualify receipt of the last 40 Z-bits. RAUXn and ROHn are disabled (output low) when the respective RAUX\_EN is inactive.

**Figure 3-25. HDSL Auxiliary Receive Payload Timing**

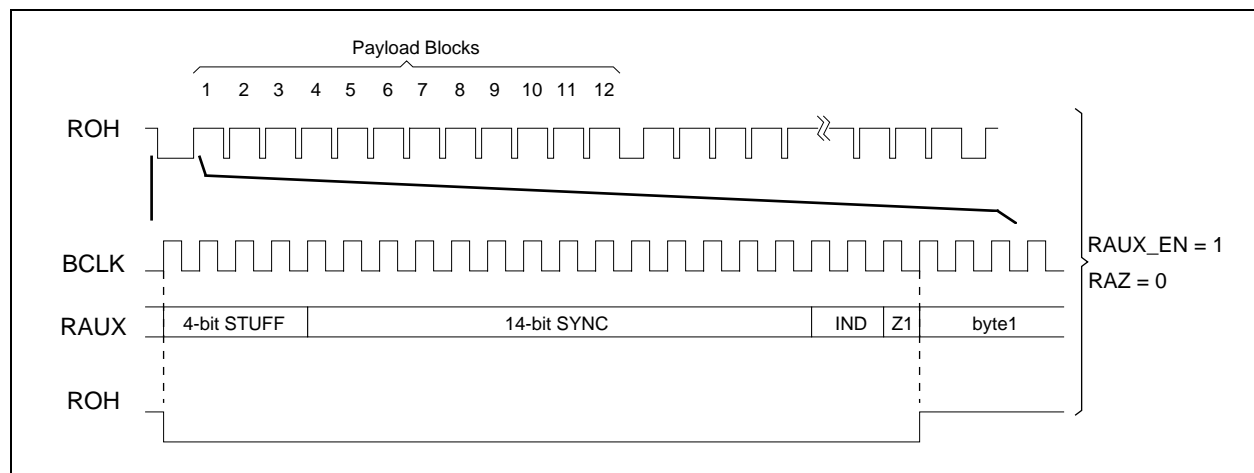
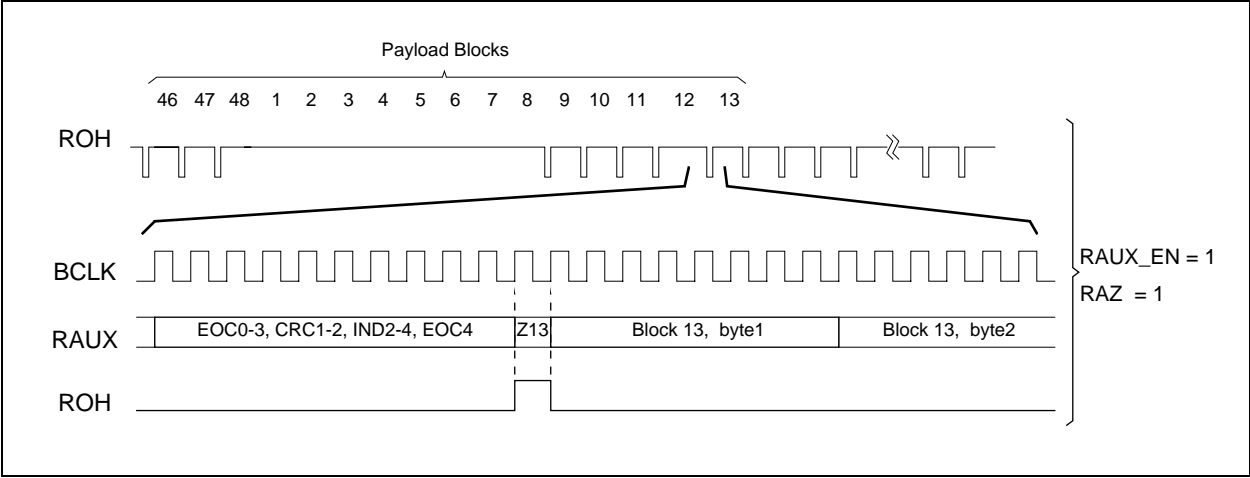




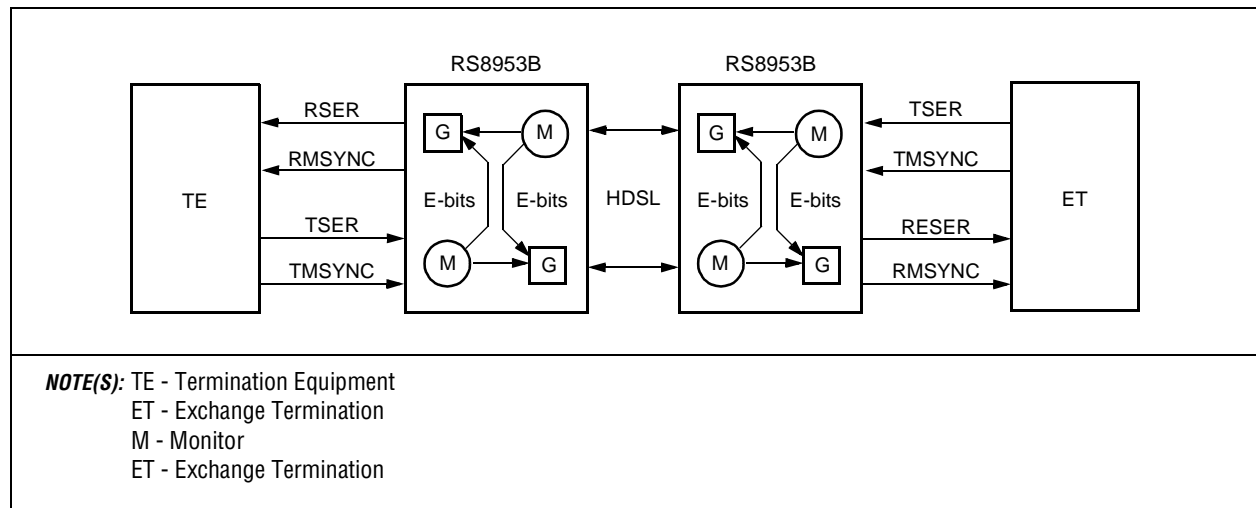
Figure 3-26. HDSL Auxiliary Receive Z-bit Timing



## 3.6 PRA Function

This document specifies requirements for using the integrated service digital network. [Figure 3-27](#) shows an overview of the HDSL link between the Termination Equipment (TE) and the Exchange Termination (ET).

**Figure 3-27. An Overview of the PRA Transfer of Data**



### 3.6.1 Transferring Data from HDSL to RSER

The following functions are available when transferring data from HDSL to RSER: CRC4 monitoring, E-bit insertion, E-bits counter, and CRC4 generator.

When CRC4 monitoring is enabled, E1 data—which is received via HDSL—is checked for error blocks by using the CRC4 procedure, as specified in CCITT recommendation G.704[9], subclause 2.3.3. The check result, represented in E-bits, is inserted into the data stream in the appropriate location.

When CRC4 monitoring is disabled, two options are available for E-bit insertion: New values are inserted for the E-bits, or the E1 data stream remains untouched. If new values are inserted, an external CPU must be used to program the value of these bits. The E-bit insertion mode is repeated each E1 frame until another value is programmed, or until another mode is selected.

The E-bits counter is continuously enabled and causes an 8-bit counter to count the amount of E-bits reflected by the E1 stream. This information is received via HDSL. This counter wraps around on full count. The value of the counter needs to be accessible to an external CPU. The counter is reset upon reading the value.

Enabling the CRC4 generator causes CRC4 regeneration of the E1 data (RSER). The result is inserted into the data stream in the appropriate location in accordance with the CRC4 procedure specified in CCITT recommendation G.704.

If the CRC4 generator is disabled, the following options are available: New values are inserted for the CRC4 bits, or you can leave the CRC4 bits untouched. If new values are inserted, an external CPU must be used to program the value of these bits. To implement this, a simple storage register can be used to insert four bits into the data stream. The CRC4 insertion is repeated each E1 frame until another value is programmed, or until another mode is selected.

### 3.6.2 Definitions of Detection Algorithms

When transferring data from HDSL or RSER, the following definitions of detection algorithms apply:

- Normal operational frames: The algorithm will be in accordance with CCITT Recommendation G.706[7]. This condition is indicated by one bit in a register.
- Loss of frame alignment: The algorithm will be in accordance with CCITT Recommendation G.706[7]. This condition is indicated by one bit in a register.
- Code words: Code words consist of four SA6 bits and the A-bit. A new code is declared only when the value of the SA6 bits and the A-bit remains the same in the last eight frames. The code word is then stored in a 5-bit register.

### 3.6.3 Inserting Data Transferred from HDSL to RSER

When transferring data from HDSL to RSER, bits are transferred in the following manner:

- Each A-bit and SA4, SA5, SA7, and SA8 bit may be selected as transparent or non-transparent. For non-transparent bits, the new value of the certain bit is stored in a register and is inserted into the correct location of the data stream (RSER).
- The SA6 bits may be transferred either transparently or non-transparently. Selecting transparent or non-transparent affects all four SA6 bits. For non-transparent transfers, the new value of the bits is stored in a register and is inserted into the correct location of the data stream (RSER).
- The FAS bits may be transferred either transparently or non-transparently. Selecting transparent or non-transparent affects all the FAS bits. For non-transparent bits, the FAS value is inserted into the correct location of the data stream (RSER).

### 3.6.4 Transferring Data from TSER to HDSL

When CRC4 monitoring is enabled, data received from TSER is checked for error blocks by using the CRC4 procedure, as specified in CCITT recommendation G.704[9], subclause 2.3.3. The check result, reflected by the E-bits, is inserted into the correct location of the data stream.

If CRC4 monitoring is disabled, new values must be inserted into the E-bits, or the E-bits must pass transparently (from the input TSER). If new values are inserted, these bits are obtained by enabling an external CPU to program a 2-bit register.

The E-bits counter is continuously enabled and causes an 8-bit counter to count the amount of E-bits errors. It wraps around on full count. An external CPU must be available to read the value of the counter. The counter is reset upon reading the value.

The CRC4 generator, when enabled, causes the E1 data (TSER) to be fed into a CRC4 generator. The CRC bits are inserted into the correct location of the data stream (TSER) according to the CRC4 procedure which is specified in CCITT recommendation G.704.

If the CRC4 generator is disabled, new values can be inserted for the CRC4 bits, or the CRC4 bits can be passed transparently (from the input TSER). If new values are inserted, the new value is stored in a 4-bits register and is repeatedly inserted into the correct location of the data stream. This insertion process is continuously repeated until a new mode is selected.

### 3.6.5 Inserting Data Transferred from TSER to HDSL

- Each A-bit and SA4, SA5, SA7, and SA8 bit may be selected as transparent or non-transparent. For non-transparent bits, the new value of the certain bit is stored in a register and is inserted into the correct location of the data stream (TSER).
- The SA6 bits may be transferred either transparently or non-transparently. Selecting transparent or non-transparent affects all four SA6 bits. For non-transparent transfers, the new value of the bits is stored in a register and is inserted into the correct location of the data stream (TSER).
- The FAS bits may be transferred either transparently or non-transparently. Selecting transparent or non-transparent affects all the FAS bits. For non-transparent bits, the FAS value is inserted into the correct location of the data stream (TSER). The FAS is a constant pattern.

## 4.0 Registers

---

All RS8953B registers are read-only or write-only. For registers that contain less than 8 bits, assigned bits reside in LSB positions; unassigned bits are ignored during write cycles and are indeterminate during read cycles. The LSB in all registers is bit position 0. All registers are randomly accessible except for the 64 transmit routing table entries, the 64 receive combination table entries, and the 16 receive signaling table entries which are written sequentially to a single register address. After power-up, register initialization is required only for populated HDSL channels. Command and status registers related to disconnected HDSL channels can be ignored (all HDSL inputs are internally pulled high).

The single-pair version (RS8953SPBEPF and RS8953SPBEPJ) only supports HDSL Channel 1. HDSL Channels 2 and 3 are not usable. Although only one HDSL channel is usable, the internal registers are not changed from the three HDSL channel versions. This means that the registers should be programmed with the same value as if only HDSL Channel 1 was used in a three-channel version. This allows the three-channel version to be used for development, and without a software change, a single-pair version used for production.

**Register Types**

The Microprocessor Unit (MPU) must read and write real-time registers, and receive and transmit EOC, IND, Z-bit, and status registers within a prescribed time interval (1–6 ms) after their respective HDSL channel's 6 ms frame interrupt. This must be done to avoid reading or writing transitory data values. Failure to read real-time registers within the prescribed interval results in a loss of data.

The MPU writes to non-real time command registers which are event-driven and which are written when the system initializes, changes modes, or responds to an error condition. Whenever data is written to a RS8953B register, the data is first written to the Shadow Write Register [SHADOW\_WR; addr 0x3B], and then the data is transferred from the SHADOW\_WR register to the addressed register. For diagnostics, software can read-verify the last write cycle by reading the SHADOW\_WR register. This will confirm that the data was written to the SHADOW\_WR register, but does not confirm that the data was transferred to the addressed register. If the Write Pulse Width specification is not met, then the data may not be correctly transferred from the SHADOW\_WR register to the addressed register. To prevent transitory write data in non-real time command registers, the MPU can first write the desired data value to the SHADOW\_WR register, and then write the same data to the desired register.

MPU reads may be interrupt event driven, polled, or a combination of both, thereby allowing the choice to be dictated by system architecture. Polled procedures can avoid reading transitory real-time data by monitoring the Interrupt Request Register [IRR; address 0x1F] bits to determine when a particular group of registers has been updated. Interrupt driven and polled procedures must complete reading within the prescribed 1–6 ms interval following HDSL frame interrupts.

**Register Groups**

RS8953B command, status, and real-time registers are divided into three groups: Common, Transmit, and Receive registers. Common registers effect overall operation, primarily the PCM channel and the DPLL. Three identical groups of Transmit and Receive registers only affect operation or report status of the respective HDSL channel. Transmit registers reference data flow from the PCM channel to the HDSL channel outputs, while Receive registers reference data flow from the HDSL channel to the PCM channel outputs. RS8953B initialization and error handling routines, written in C-language, are available via the terms of the HDSL software license agreement.

The addresses shown for each Transmit and Receive register or bit description reference only HDSL Channel 1. (See the Summary tables at the start of each section to find address locations for HDSL channels 2 and 3.)

## 4.1 Address Map

The channel column (CHn) of [Table 4-1](#) indicates which HDSL channel number (n = 1,2,3) is associated with each register. Common registers are indicated by a 'C' in the CHn column.

**Table 4-1. Register Summary Address Map (1 of 6)**

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0x00	1	TEOC_LO	4-10	1	REOC_LO	4-53
0x01	1	TEOC_HI	4-10	1	REOC_HI	4-54
0x02	1	TIND_LO	4-10	1	RIND_LO	4-54
0x03	1	TIND_HI	4-10	1	RIND_HI	4-54
0x04	1	TZBIT_1	4-10	1	RZBIT_1	4-54
0x05	1	TFIFO_WL	4-12	1	STATUS_1	4-55
0x06	1	TCMD_1	4-12	1	STATUS_2	4-57
0x07	1	TCMD_2	4-13	1	STATUS_3	4-58
0x08	1	TMAP_1	4-15	2	REOC_LO	4-53
0x09	1	TMAP_2	4-15	2	REOC_HI	4-54
0x0A	1	TMAP_3	4-15	2	RIND_LO	4-54
0x0B	1	TMAP_4	4-15	2	RIND_HI	4-54
0x0C	1	TMAP_5	4-15	2	RZBIT_1	4-54
0x0D	1	TFIFO_RST	4-17	2	STATUS_1	4-55
0x0E	1	SCR_RST	4-17	2	STATUS_2	4-57
0x0F	1	TMAP_6	4-16	2	STATUS_3	4-58
0x10	1	TMAP_7	4-16	3	REOC_LO	4-53
0x11	1	TMAP_8	4-16	3	REOC_HI	4-54
0x12	1	TMAP_9	4-16	3	RIND_LO	4-54
0x13	—	—	—	3	RIND_HI	4-54
0x14	—	—	—	3	RZBIT_1	4-54
0x15	—	—	—	3	STATUS_1	4-55
0x16	—	—	—	3	STATUS_2	4-57
0x17	—	—	—	3	STATUS_3	4-58
0x18	—	—	—	C	RZBIT_2	4-55
0x19	—	—	—	C	RZBIT_3	4-55
0x1A	—	—	—	C	RZBIT_4	4-55
0x1B	—	—	—	C	RZBIT_5	4-55

Table 4-1. Register Summary Address Map (2 of 6)

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0x1C	—	—	—	C	RZBIT_6	4-55
0x1D	—	—	—	C	BER_METER	4-60
0x1E	—	—	—	C	BER_STATUS	4-61
0x1F	—	—	—	C	IRR	4-61
0x20	2	TEOC_LO	4-10	C	RESID_OUT_HI	4-62
0x21	2	TEOC_HI	4-10	1	CRC_CNT	4-58
0x22	2	TIND_LO	4-10	1	FEBC_CNT	4-59
0x23	2	TIND_HI	4-10	—	—	—
0x24	2	TZBIT_1	4-10	—	—	—
0x25	2	TFIFO_WL	4-12	—	—	—
0x26	2	TCMD_1	4-12	—	—	—
0x27	2	TCMD_2	4-13	—	—	—
0x28	2	TMAP_1	4-15	C	RESID_OUT_LO	4-62
0x29	2	TMAP_2	4-15	2	CRC_CNT	4-59
0x2A	2	TMAP_3	4-15	2	FEBC_CNT	4-59
0x2B	2	TMAP_4	4-15	—	—	—
0x2C	2	TMAP_5	4-15	—	—	—
0x2D	2	TFIFO_RST	4-17	—	—	—
0x2E	2	SCR_RST	4-17	—	—	—
0x2F	2	TMAP_6	4-16	—	—	—
0x30	2	TMAP_7	4-16	C	IMR	4-62
0x31	2	TMAP_8	4-16	3	CRC_CNT	4-59
0x32	2	TMAP_9	4-16	3	FEBC_CNT	4-59
0x33–0x37	—	—	—	—	—	—
0x38	—	—	—	C	PHS_ERR	4-63
0x39	—	—	—	C	MSYNC_PHS_LO	4-63
0x3A	—	—	—	C	MSYNC_PHS_HI	4-63
0x3B	—	—	—	C	SHADOW_WR	4-64
0x3C	—	—	—	C	ERR_STATUS	4-64
0x3D–0x3F	—	—	—	—	—	—
0x40	3	TEOC_LO	4-10	C	TX_PRA_CTRL0	4-65
0x41	3	TEOC_HI	4-10	C	TX_PRA_CTRL_1	4-66
0x42	3	TIND_LO	4-10	C	TX_PRA_MON1	4-67
0x43	3	TIND_HI	4-10	C	TX_PRA_E_CNT	4-67



Table 4-1. Register Summary Address Map (3 of 6)

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0x44	3	TZBIT_1	4-10	—	—	
0x45	3	TFIFO_WL	4-12	C	TX_PRA_CODE	4-67
0x46	3	TCMD_1	4-12	C	TX_PRA_MON0	4-68
0x47	3	TCMD_2	4-13	C	TX_PRA_MON2	4-68
0x48	3	TMAP_1	4-15	—	—	—
0x49	3	TMAP_2	4-15	—	—	—
0x4A	3	TMAP_3	4-15	—	—	—
0x4B	3	TMAP_4	4-15	—	—	—
0x4C	3	TMAP_5	4-15	—	—	—
0x4D	3	TFIFO_RST	4-17	—	—	—
0x4E	3	SCR_RST	4-17	—	—	—
0x4F	3	TMAP_6	4-16	—	—	—
0x50	3	TMAP_7	4-16	—	—	—
0x51	3	TMAP_8	4-16	—	—	—
0x52	3	TMAP_9	4-16	—	—	—
0x60	1	RCMD_1	4-19	—	—	—
0x61	1	RCMD_2	4-20	—	—	—
0x62	1	RFIFO_RST	4-21	—	—	—
0x63	1	SYNC_RST	4-21	—	—	—
0x64	1	RMAP_1	4-22	—	—	—
0x65	1	RMAP_2	4-22	—	—	—
0x66	1	RMAP_3	4-22	—	—	—
0x67	1	ERR_RST	4-23	—	—	—
0x68	1	RSIG_LOC	4-23	—	—	—
0x69	1	RMAP_4	4-22	—	—	—
0x6A	1	RMAP_5	4-22	—	—	—
0x6B	1	RMAP_6	4-23	—	—	—
0x70	C	TX_PRA_CTRL0	4-69	—	—	—
0x71	C	TX_PRA_CTRL1	4-70	—	—	—
0x72	C	TX_BITS_BUFF1	4-71	—	—	—
0x73	C	TX_PRA_TMSYNC_OFFSET	4-71	—	—	—
0x74	C	TX_BITS_BUFF0	4-72	—	—	—
0x80	2	RCMD_1	4-19	C	RX_PRA_CTRL0	4-73
0x81	2	RCMD_2	4-20	C	RX_PRA_CTRL1	4-74

Table 4-1. Register Summary Address Map (4 of 6)

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0x82	2	RFIFO_RST	4-21	C	RX_BITS_BUFF1	4-75
0x83	2	SYNC_RST	4-21	C	RX_PRA_E_CNT	4-75
0x84	2	RMAP_1	4-22	C	RX_PRA_CRC_CNT	4-75
0x85	2	RMAP_2	4-22	C	RX_PRA_CODE	4-76
0x86	2	RMAP_3	4-22	C	RX_PRA_MON0	4-76
0x87	2	ERR_RST	4-23	C	RX_PRA_MON2	4-76
0x88	2	RSIG_LOC	4-23	—	—	—
0x89	2	RMAP_4	4-22	—	—	—
0x8A	2	RMAP_5	4-22	—	—	—
0x8B	2	RMAP_6	4-23	—	—	—
0xA0	3	RCMD_1	4-19	—	—	—
0xA1	3	RCMD_2	4-20	—	—	—
0xA2	3	RFIFO_RST	4-21	—	—	—
0xA3	3	SYNC_RST	4-21	—	—	—
0xA4	3	RMAP_1	4-22	—	—	—
0xA5	3	RMAP_2	4-22	—	—	—
0xA6	3	RMAP_3	4-22	—	—	—
0xA7	3	ERR_RST	4-23	—	—	—
0xA8	3	RSIG_LOC	4-23	—	—	—
0xA9	3	RMAP_4	4-22	—	—	—
0xAA	3	RMAP_5	4-22	—	—	—
0xAB	3	RMAP_6	4-23	—	—	—
0xB0	C	RX_PRA_CTRL0	4-77	—	—	—
0xB1	C	RX_PRA_CTRL1	4-78	—	—	—
0xB2	C	RX_BITS_BUFF1	4-79	—	—	—
0xB4	C	RX_BITS_BUFF0	4-79	—	—	—
0xC0	C	TFRAME_LOC_LO	4-26	—	—	—
0xC1	C	TFRAME_LOC_HI	4-26	—	—	—
0xC2	C	TMF_LOC	4-26	—	—	—
0xC3	C	RFRAME_LOC_LO	4-26	—	—	—
0xC4	C	RFRAME_LOC_HI	4-27	—	—	—
0xC5	C	RMF_LOC	4-27	—	—	—
0xC6	C	MF_LEN	4-27	—	—	—
0xC7	C	MF_CNT	4-28	—	—	—

**Table 4-1. Register Summary Address Map (5 of 6)**

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0xC8	C	FRAME_LEN_LO	4-28	—	—	—
0xC9	C	FRAME_LEN_HI	4-28	—	—	—
0xCA	C	HFRAME_LEN_LO	4-29	—	—	—
0xCB	C	SYNC_WORD_A	4-31	—	—	—
0xCC	C	SYNC_WORD_B	4-31	—	—	—
0xCD	C	RFIFO_WL_LO	4-31	—	—	—
0xCE	C	RFIFO_WL_HI	4-32	—	—	—
0xCF	C	STF_THRESH_A_LO	4-33	—	—	—
0xD0	C	STF_THRESH_A_HI	4-33	—	—	—
0xD1	C	STF_THRESH_B_LO	4-33	—	—	—
0xD2	C	STF_THRESH_B_HI	4-33	—	—	—
0xD3	C	STF_THRESH_C_LO	4-33	—	—	—
0xD4	C	STF_THRESH_C_HI	4-34	—	—	—
0xD5	C	DPLL_RESID_LO	4-35	—	—	—
0xD6	C	DPLL_RESID_HI	4-36	—	—	—
0xD7	C	DPLL_FACTOR	4-36	—	—	—
0xD8	C	DPLL_GAIN	4-37	—	—	—
0xDB	C	DPLL_PINI	4-38	—	—	—
0xDC	C	DBANK_1	4-39	—	—	—
0xDD	C	DBANK_2	4-39	—	—	—
0xDE	C	DBANK_3	4-40	—	—	—
0xDF	C	TZBIT_2	4-11	—	—	—
0xE0	C	TZBIT_3	4-11	—	—	—
0xE1	C	TZBIT_4	4-11	—	—	—
0xE2	C	TZBIT_5	4-11	—	—	—
0xE3	C	TZBIT_6	4-11	—	—	—
0xE4	C	TSTUFF	4-40	—	—	—
0xE5	C	CMD_1	4-44	—	—	—
0xE6	C	CMD_2	4-45	—	—	—
0xE7	C	CMD_3	4-46	—	—	—
0xE8	C	CMD_4	4-47	—	—	—
0xE9	C	CMD_5	4-47	—	—	—
0xEA	C	FILL_PATT	4-40	—	—	—
0xEB	C	IMR	4-51	—	—	—

Table 4-1. Register Summary Address Map (6 of 6)

Addr	CHn	Write Register	Page Ref.	CHn	Read Register	Page Ref.
0xEC	C	ICR	4-51	—	—	—
0xED	C	ROUTE_TBL	4-41	—	—	—
0xEE	C	COMBINE_TBL	4-42	—	—	—
0xEF	C	BER_RST	4-52	—	—	—
0xF0	C	PRBS_RST	4-52	—	—	—
0xF1	C	RX_RST	4-52	—	—	—
0xF2	C	RSIG_TBL	4-43	—	—	—
0xF3	C	CMD_6	4-48	—	—	—
0xF4	C	CMD_7	4-49	—	—	—
0xF5	C	HFRAME_LEN_HI	4-30	—	—	—
0xF6	C	DPLL_RST	4-38	—	—	—
0xF7–0xFF	—	—	—	—	—	—
F8	2	HFRAME_2LEN_LO	4-30	—	—	—
F9	2	HFRAME2_LEN_HI	4-30	—	—	—
FA	3	HFRAME3_LEN_LO	4-30	—	—	—
FB	3	HFRAME3_LEN_HI	4-31	—	—	—

## 4.2 HDSL Transmit

	HDSL Channel 1 (CH1)	HDSL Channel 2 (CH2)	HDSL Channel 3 (CH3)
Base Address	0x00	0x20	0x40

**Table 4-2. HDSL Transmit Write Registers**

CH1	CH2	CH3	Register Label	Bits	Description
0x00	0x20	0x40	TEOC_LO	8	Transmit EOC Bits
0x01	0x21	0x41	TEOC_HI	5	Transmit EOC Bits
0x02	0x22	0x42	TIND_LO	8	Transmit IND Bits
0x03	0x23	0x43	TIND_HI	5	Transmit IND Bits
0x04	0x24	0x44	TZBIT_1	8	Transmit Z-bits
0xDF			TZBIT_2	8	Common Transmit Z-bits
0xE0			TZBIT_3	8	Common Transmit Z-bits
0xE1			TZBIT_4	8	Common Transmit Z-bits
0xE2			TZBIT_5	8	Common Transmit Z-bits
0xE3			TZBIT_6	8	Common Transmit Z-bits
0x05	0x25	0x45	TFIFO_WL	8	Transmit FIFO Water Level
0x06	0x26	0x46	TCMD_1	7	Configuration
0x07	0x27	0x47	TCMD_2	6	Configuration
0x08	0x28	0x48	TMAP_1	8	Payload Map
0x09	0x29	0x49	TMAP_2	8	Payload Map
0x0A	0x2A	0x4A	TMAP_3	8	Payload Map
0x0B	0x2B	0x4B	TMAP_4	8	Payload Map
0x0C	0x2C	0x4C	TMAP_5	8	Payload Map
0x0F	0x2F	0x4F	TMAP_6	8	Payload Map
0x10	0x30	0x50	TMAP_7	8	Payload Map
0x11	0x31	0x51	TMAP_8	8	Payload Map
0x12	0x32	0x52	TMAP_9	8	Payload Map
0x0D	0x2D	0x4D	TFIFO_RST	—	Transmit FIFO Reset
0x0E	0x2E	0x4E	SCR_RST	—	Scrambler Reset

**0x00—Transmit Embedded Operations Channel (TEOC\_LO)**

7	6	5	4	3	2	1	0
TEOC[7:0]							

**0x01—Transmit Embedded Operations Channel (TEOC\_HI)**

7	6	5	4	3	2	1	0
—	—	—	TEOC[12:8]				

**TEOC[12:0]** The Transmit Embedded Operations Channel (TEOC) holds 13 EOC bits for transmission in the next frame. (Refer to [Table 3-2](#) for EOC bit positions within the frame.) The HOH multiplexer samples TEOC coincident with the respective HDSL channel's transmit 6 ms frame interrupt. Unmodified registers repeatedly output their contents in each frame.

**0x02—Transmit Indicator Bits (TIND\_LO)**

7	6	5	4	3	2	1	0
TIND[7:0]							

**0x03—Transmit Indicator Bits (TIND\_HI)**

7	6	5	4	3	2	1	0
—	—	—	TIND[12:8]				

**TIND[12:0]** The Transmit Indicator holds 13 IND bits for transmission in the next frame and includes the FEBE bit (TIND[1]). (Refer to [Table 3-2](#) for IND bit positions within the frame.) The HOH multiplexer samples TIND coincident with the respective HDSL channel's transmit 6 ms frame interrupt. Unmodified registers repeatedly output their contents in each frame. TIND[0] is transmitted first.

**NOTE:** The RS8953B does not automatically output FEBE. Proper transmit of FEBE requires the MPU to copy the CRC\_ERR bit from STATUS\_2 [addr 0x06] to TIND[1].

**0x04—Transmit Z-Bits (TZBIT\_1)**

7	6	5	4	3	2	1	0
TZBIT[7:0]							

**0xDF—Transmit Z-Bits (TZBIT\_2)**

7	6	5	4	3	2	1	0
TZBIT[15:8]							

**0xE0—Transmit Z-Bits (TZBIT\_3)**

7	6	5	4	3	2	1	0
TZBIT[23:16]							

**0xE1—Transmit Z-Bits (TZBIT\_4)**

7	6	5	4	3	2	1	0
TZBIT[31:24]							

**0xE2—Transmit Z-Bits (TZBIT\_5)**

7	6	5	4	3	2	1	0
TZBIT[39:32]							

**0xE3—Transmit Z-Bits (TZBIT\_6)**

7	6	5	4	3	2	1	0
TZBIT[47:40]							

**TZBIT[47:0]** Transmit Z-bits is applicable only in E1\_MODE [CMD\_1; addr 0xE5]; otherwise, Z-bit registers are ignored. TZBIT[47:0] holds 48 Z-bits for transmission in the first bit of each of the 48 payload blocks. (See [Figure 3-16](#) for Z-bit positions within the frame.) The first eight Z-bits are individually output for each channel from TZBIT\_1. The last 40 Z-bits are output to all channels from a single set of TZBIT\_2–TZBIT\_6.

**NOTE:** The system may also supply the last 40 Z-bits individually for each HDSL transmit channel from the TAUXn inputs by setting TAUX\_EN and EXT\_ZBIT [TCMD\_2; addr 0x07].

TZBIT\_1 is sampled on the respective transmit 6 ms frame interrupt, giving the MPU up to 6 ms to modify the TZBIT\_1 contents for output in next frame. TZBIT\_2 through TZBIT\_6 are sampled during their respective output times, giving the MPU up to 1 ms after the transmit frame interrupt to update TZBIT\_2; 2 ms to update TZBIT\_3; and 5 ms to update TZBIT\_6. This assumes all HDSL transmit frames are output aligned. If differential delay exists between the transmit channels (as controlled by TFIFO\_WL; addr 0x05), then less time is available to update TZBIT\_2–TZBIT\_6. Unmodified registers repeatedly output their contents in each frame. TZBIT[0] is transmitted first.

### 0x05—Transmit FIFO Water Level (TFIFO\_WL)

7	6	5	4	3	2	1	0
TFIFO_WL[7:0]							

**TFIFO\_WL[7:0]** Transmit FIFO Water Level contains the number of TCLK cycles to delay from the PCM 6 ms frame to the start of the HDSL transmit SYNC word. A value of zero equals 1 TCLK delay. Minimum water level values compensate for time to unload one timeslot (8 HDSL bits), time to load one timeslot (8 PCM bits), the amount of differential delay created by the PCM router (up to 96 PCM bits in T1 mode), and a phase jitter tolerance (8 to 16 PCM bits). (Refer to TFIFO\_WL description in the PCM Channel Section.)

### 0x06—Transmit Command Register 1 (TCMD\_1)

Real-time commands (Bits 0–5) are sampled by the HOH multiplexer on the respective transmit frame to affect operation in the next outgoing frame. HOH\_EN, TWO\_LEVEL, and FORCE\_ONE command bit combinations provide the transmit data encoding options needed to perform standard HDSL channel startup procedures.

7	6	5	4	3	2	1	0
—	TX_ERR_EN	FORCE_ONE	HOH_EN	SYNC_SEL	ICRC_ERR	TWO_LEVEL	SCR_EN

**SCR\_EN** Scrambler Enable—All transmit HDSL channel bits, except SYNC and STUFF bits, are scrambled per the SCR\_TAP setting in TCMD\_2[0x47]. Otherwise, transmit data passes through the scrambler unchanged.

0 = Scrambler bypassed  
1 = Scrambler enabled

**TWO\_LEVEL** Two Level Transmit Enable—All 2B1Q encoder magnitude bit outputs are forced to 0 to comply with standard requirements for a two-level transmit signal.

0 = Four-level 2B1Q encoder operation  
1 = Two-level 2B1Q encoder operation

**ICRC\_ERR** Inject CRC Error—Logically inverts the six calculated CRC bits in the next frame.

0 = Normal CRC transmission  
1 = Transmit errored CRC

**SYNC\_SEL** SYNC Word Select—Selects one of two SYNC words, SYNC\_WORD\_A or SYNC\_WORD\_B [addresses 0xCB–0xCC], for transmission in the next frame.

0 = SYNC\_WORD\_A is transmitted  
1 = SYNC\_WORD\_B is transmitted



<b>HOH_EN</b>	<p>HDSL Overhead Enable—The HOH multiplexer inserts EOC, IND, and CRC bits. Otherwise, transmit overhead bits, except SYNC and STUFF, are forced to all 1s. HOH_EN = 0 select transmission of two-level or four-level scrambled 1s.</p> <p>0 = HOH transmitted as all 1s 1 = Normal HOH transmission</p>
<b>FORCE_ONE</b>	<p>Force All 1s Payload—Transmit payload data bytes are replaced by all 1s. FORCE_ONE and HOH_EN are both set to enable output of a four-level framed, scrambled-1s signal.</p> <p>0 = Normal payload transmission 1 = Force all 1s payload</p>
<b>TX_ERR_EN</b>	<p>Transmit Error Interrupt Enable—Transmit errors request TX_ERR interrupt and report TXn_ERR status upon detection of TFIFO or TSTUFF errors [STATUS_3; addr 0x07]. Disabled channels are prevented from activating INTR*, or setting TX_ERR [IRR; addr 0x1F]. Transmit errors are always latched in ERR_STATUS [addr 0x3C] regardless of TX_ERR_EN.</p> <p>0 = Disable transmit error interrupts 1 = Enable transmit error interrupts</p>

## 0x07—Transmit Command Register 2 (TCMD\_2)

7	6	5	4	3	2	1	0
—	STUF_CNTR_MODE	EXT_ZBIT	REPEAT_EN	TAUX_EN	SLV_STUF	SCR_TAP	HH_LOOP

<b>HH_LOOP</b>	<p>Loopback to HDSL on the HDSL Side—Receive HDSL data (RDATn) is switched to transmit HDSL data (TDATn) to accomplish a loopback of the HDSL channel on the HDSL side. Loopback data is switched at I/O pins and does not alter HDSL receive operations.</p> <p>0 = Normal transmit 1 = TDATn supplied by RDATn pin</p>
<b>SCR_TAP</b>	<p>Scrambler Tap—Selects which delay stage, 5th or 18th, to tap for feedback in the transmit scrambler. The system's HDSL terminal type dictates which scrambler tap should be selected.</p> <p>0 = HTU-C or LTU terminal type, scrambler taps 5th delay stage 1 = HTU-R or NTU terminal type, scrambler taps 18th delay stage</p>
<b>SLV_STUF</b>	<p>Slave STUFF Bits—Transmit STUFF bits are either generated by a local stuffing mechanism or are slaved to an alternate source of STUFF bits. If enabled, the slave STUFF source is chosen by STUFF_SEL in Common CMD_5 [addr 0xE9].</p> <p>0 = Local STUFF bit generation 1 = Slave STUFF bits to STUFF_SEL source</p>
<b>TAUX_EN</b>	<p>Transmit Auxiliary Enable—Transmit auxiliary data from the TAUX1–TAUX3 inputs are sampled when the respective TLOAD1–TLOAD3 outputs are active. TAUX samples and TLOAD activation are selected for each payload byte via the transmit payload map [TMAP; addr 0x08]. When TAUX_EN is low, TAUX inputs are ignored and TLOAD outputs are forced low.</p> <p>0 = Disable TAUX and TLOAD signals 1 = Enable TAUX and TLOAD signals</p>

<b>REPEAT_EN</b>	<p>Enable Repeater Mode—When set in both CH1 and CH2, REPEAT_EN cross-connects HDSL payload, SYNC, STUFF, and Z-bits from receive to transmit to implement a single pair repeater. REPEAT_EN has no effect in CH3. Transmit 6 ms frames are forced to align to cross-connected receive 6 ms frames. HOH bits (EOC, IND, and CRC) are inserted by each channel's transmit HOH multiplexer to allow for translation of repeater specific IND bits. HDSL bit clocks, BCLK1, and BCLK2 can operate with separate phase, but must be identical in long-term frequency. Receive payload from CH1 and CH2 can still be mapped and PCM combined, but transmit PCM inputs are ignored.</p> <p>0 = Normal transmit 1 = Cross-connect CH1 and CH2</p>
<b>EXT_ZBIT</b>	<p>Enable External Z-bits—Is set in conjunction with TAUX_EN when the system supplies the last 40 Z-bits for transmission from TAUXn input.</p> <p>0 = Last 40 Z-bits supplied by TZBIT2–TZBIT6 registers 1 = Last 40 Z-bits supplied by TAUXn</p>
<b>STUF_CNTR_MODE</b>	<p>Selects the operation of the counter that measures the phase difference between the PCM and HDSL line clocks. If this bit is cleared in any of the registers for the three channels, the stuffing generator is in a mode in which the counter is not allowed to wrap around. In this mode, the stuffing algorithm is based on only current conditions.</p> <p>When this bit is set in each of the registers for the 3 channels, the stuffing generator is in a mode in which the counter is allowed to wrap around. In this mode, the stuffing algorithm is based on both previous decisions and current conditions. This changes the stuffing pattern, which causes reduced low frequency wander and increased high frequency jitter.</p> <p>0 = Increased wander and decreased jitter (in any register) 1 = Decreased wander and increased jitter (in all three registers)</p>

## 4.3 Transmit Payload Mapper

The Transmit Payload Map (TMAP\_1–TMAP\_9) determines whether HDSL payload bytes (byte1 through byte36) are supplied from PCM timeslots, DBANK registers, or the HDSL auxiliary channel data. All routed timeslots to a given channel's TFIFO must also be mapped out of the TFIFO. The RS8953B sequentially maps payload and cannot rearrange byte ordering but allows payload from the DBANK registers to be interleaved with PCM data. If PCM transmit data is input-aligned to MSYNC, then the first TMAP byte to select PCM receives the first routed PCM timeslot from the transmit PCM multiframe (i.e., PCM frame 0 maps to HDSL payload block 1). If PCM data is not aligned to MSYNC, then payload bytes mapped from the TFIFO are not aligned to PCM timeslots and HDSL payload blocks are not aligned to PCM frames. In T1 mode, TMAP must be programmed to supply F-bits, by enabling one extra byte of payload at the end of the payload block.

### 0x08—Transmit Payload Map (TMAP\_1)

7	6	5	4	3	2	1	0
BYTE4 TMAP[1:0]		BYTE3 TMAP[1:0]		BYTE2 TMAP[1:0]		BYTE1 TMAP[1:0]	

### 0x09—Transmit Payload Map (TMAP\_2)

7	6	5	4	3	2	1	0
BYTE8 TMAP[1:0]		BYTE7 TMAP[1:0]		BYTE6 TMAP[1:0]		BYTE5 TMAP[1:0]	

### 0x0A—Transmit Payload Map (TMAP\_3)

7	6	5	4	3	2	1	0
BYTE12 TMAP[1:0]		BYTE11 TMAP[1:0]		BYTE10 TMAP[1:0]		BYTE9 TMAP[1:0]	

### 0x0B—Transmit Payload Map (TMAP\_4)

7	6	5	4	3	2	1	0
BYTE16 TMAP[1:0]		BYTE15 TMAP[1:0]		BYTE14 TMAP[1:0]		BYTE13 TMAP[1:0]	

### 0x0C—Transmit Payload Map (TMAP\_5)

7	6	5	4	3	2	1	0
BYTE20 TMAP[1:0]		BYTE19 TMAP[1:0]		BYTE18 TMAP[1:0]		BYTE17 TMAP[1:0]	

**0x0F—Transmit Payload Map (TMAP\_6)**

7	6	5	4	3	2	1	0
BYTE24 TMAP[1:0]		BYTE23 TMAP[1:0]		BYTE22 TMAP[1:0]		BYTE21 TMAP[1:0]	

**0x10—Transmit Payload Map (TMAP\_7)**

7	6	5	4	3	2	1	0
BYTE28 TMAP[1:0]		BYTE27 TMAP[1:0]		BYTE26 TMAP[1:0]		BYTE25 TMAP[1:0]	

**0x11—Transmit Payload Map (TMAP\_8)**

7	6	5	4	3	2	1	0
BYTE32 TMAP[1:0]		BYTE31 TMAP[1:0]		BYTE30 TMAP[1:0]		BYTE29 TMAP[1:0]	

**0x12—Transmit Payload Map (TMAP\_9)**

7	6	5	4	3	2	1	0
BYTE36 TMAP[1:0]		BYTE35 TMAP[1:0]		BYTE34 TMAP[1:0]		BYTE33 TMAP[1:0]	

Transmit Payload Map code selects one of four data sources for HDSL payload bytes. A maximum of 18 map codes, corresponding to payload byte1 through byte36, are programmed for each HDSL channel. If the payload block length is less than 36 bytes, TMAP codes of the upper bytes are unused.

TMAP[1:0]	Transmit HDSL Payload Source
00	PCM data from TFIFO
01	DBANK_1
10	DBANK_2
11	DBANK_3 <sup>(1, 2)</sup>

- Notes:
1. When DBANK\_3 and TAUX\_EN [TCMD\_2; addr 0x07] are selected, TLOADn output is active and TAUXn supplies data during selected payload byte.
  2. When DBANK\_3, TAUX\_EN and EXT\_ZBIT [TCMD\_2; addr 0x07] are selected, TLOADn output is active and TAUXn supplies data during the last 40 Z-bits.

The following configurations can cause the HDSL frame to become corrupted:

- DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte #33.
- TFIFO or CBank\_1 is the source of data for HDSL payload byte #0.
- The LSB of the DBank\_1 pattern is equal to one (1).
- or
- DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte #33.
- DBank\_2 is the source of data for HDSL payload byte #0.
- The LSB of the DBank\_2 pattern is equal to one (1).
- or
- DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte #33.
- DBank\_3 is the source of data for HDSL payload byte #0.
- The LSB of the DBank\_3 pattern is equal to one (1).

### 0x0D—Transmit FIFO Reset (TFIFO\_RST)

Writing any data value to TFIFO\_RST empties the TFIFO, forces the HDSL transmitter to resample the Transmit FIFO Water Level [TFIFO\_WL; addr 0x05], and realigns the HDSL channel's transmit 6 ms frame to the PCM 6 ms frame. The MPU must write TFIFO\_RST after modifying the TFIFO\_WL value, the Transmit Payload Map [TMAP; addr 0x08], or the PCM Routing Table [ROUTE\_TBL; addr 0xED] each time PCM MultiFrame Sync (TMSYNC) experiences a change of frame alignment and whenever the TFIFO reports an overflow, underflow, or slip error. The RS8953B asserts TFIFO\_RST automatically whenever a transmit STUFF error is detected.

**NOTE:** Each write to TFIFO\_RST may cause TFIFO errors in the next three subsequent HDSL frames. Therefore, the MPU must ignore up to three TFIFO errors reported in the respective channel for the next 3 HDSL frames after writing the TFIFO\_RST command.

### 0x0E—Scrambler Reset (SCR\_RST)

Writing any data value to SCR\_RST sets the 23 stages of the scrambler LFSR to 0x000001. SCR\_RST is used during the Conexant production test to verify scrambler operation, and is not required during normal operation.

## 4.4 HDSL Receive

	HDSL Channel 1 (CH1)	HDSL Channel 2 (CH2)	HDSL Channel 3 (CH3)
Base Address	0x60	0x80	0xA0

**Table 4-3. HDSL Receive Write Registers**

CH1	CH2	CH3	Register Label	Bits	Name/Description
0x60	0x80	0xA0	RCMD_1	8	Configuration
0x61	0x81	0xA1	RCMD_2	8	Configuration
0x62	0x82	0xA2	RFIFO_RST	–	Receive FIFO Reset
0x63	0x83	0xA3	SYNC_RST	–	Receive Framer Reset
0x64	0x84	0xA4	RMAP_1	6	Payload Map
0x65	0x85	0xA5	RMAP_2	6	Payload Map
0x66	0x86	0xA6	RMAP_3	6	Payload Map
0x67	0x87	0xA7	RMAP_4	6	Payload Map
0x68	0x88	0xA8	RMAP_5	6	Payload Map
0x69	0x89	0xA9	RMAP_6	6	Payload Map
0x70	0x70	0xA0	ERR_RST	–	Error Count Reset
0x71	0x71	0xA1	RSIG_LOC	4	Receive Signaling Location

Three identical groups of write-only registers configure the HDSL receivers, and control the mapping of HDSL payload bytes into the receiver elastic stores (RFIFO). Configuration registers define each HDSL receive framer's criteria for loss and recovery of frame alignment by selecting the number of detected SYNC word errors used to declare loss of sync or needed to acquire sync. Refer to the Framer Synchronization State Diagram, [Figure 3-23](#). Frame alignment criteria are programmable to meet different standard application requirements.

**0x60—Receive Command Register 1 (RCMD\_1)**

7	6	5	4	3	2	1	0
FRAMER_EN[1:0]		LOSS_SYNC[2:0]			REACH_SYNC[2:0]		

**REACH\_SYNC[2:0]** Reach Sync Framing Criteria—Contains the number of consecutive HDSL frames in which the SYNC word is detected before the receive framer moves from the OUT\_OF\_SYNC to the IN\_SYNC state. REACH\_SYNC determines the number of SYNC\_ACQUIRED intermediate states the framer must pass through during recovery of frame sync. ETSI standard criteria requires two consecutive frames containing SYNC.

REACH_SYNC	IN_SYNC Criteria
000	1 frame containing SYNC
001	2 consecutive frames
010	3 consecutive frames
011	4 consecutive frames
100	5 consecutive frames
101	6 consecutive frames
110	7 consecutive frames
111	8 consecutive frames

**LOSS\_SYNC[2:0]** Loss of Sync Framing Criteria—Contains the number of consecutive HDSL frames in which the SYNC word is not detected before the receive framer moves from the IN\_SYNC to the OUT\_OF\_SYNC state. LOSS\_SYNC determines the number of SYNC\_ERRORED intermediate states the framer must pass through during loss of frame sync. ETSI standard criteria requires six consecutive frames without SYNC word detected.

LOSS_SYNC	OUT_OF_SYNC Criteria
000	1 frame not containing SYNC
001	2 consecutive frames
010	3 consecutive frames
011	4 consecutive frames
100	5 consecutive frames
101	6 consecutive frames
110	7 consecutive frames
111	8 consecutive frames

**FRAMER\_EN[1:0]** Receive Framer Enable—Instructs the receive framer to search for one or both of the SYNC word patterns programmed in SYNC\_WORD\_A [addr 0xCB] or SYNC\_WORD\_B [addr 0xCC]. If enabled to search for both, then the SYNC acquisition state proceeds with only the first detected pattern. When disabled, the framer does not count errors or generate interrupts.

FRAMER_EN	Receive Framer Search
00	Disabled; framer forced to OUT_OF_SYNC
01	SYNC_WORD_A
10	SYNC_WORD_B
11	Both SYNC_WORD_A and SYNC_WORD_B

**0x61—Receive Command Register 2 (RCMD\_2)**

7	6	5	4	3	2	1	0
RX_ERR_EN	PH_LOOP	DSCR_EN	DSCR_TAP	THRESH_CORR[3:0]			

**THRESH\_CORR[3:0]** SYNC Threshold Correlation—Upon the receive framer's entry to a Sync Errored state, the number of SYNC word locations searched is determined by the result of the previous states' threshold correlation. During an In Sync state, the framer searches the two most probable SYNC word locations at  $6\text{ ms} \pm 1\text{ quat}$ , corresponding to 0 or 4 STUFF bits. One of the two locations searched must correctly match the entire 14-bit SYNC word or the framer enters a Sync Errored state. The highest number of matching bits found among the search locations is compared to the selected THRESH\_CORR value to determine whether the framer should expand the number of search locations. If the highest number of matching bits meets or exceeds the threshold, but was not a complete match, the framer progresses to the next Sync Errored state and continues to each of the two most probable locations. Otherwise, the framer progresses to the next Sync Errored state, increments the number of locations to be searched, and examines quats on either side of the prior search locations. For example, if the location with the highest number of matching bits is below the threshold during In Sync, then the framer enters the first Sync Errored state and searches from the prior location at  $6\text{ ms} \pm 2\text{ quats}$ , and at exactly 6 ms. The effect of threshold correlation on the number of search locations is depicted in [Figure 3-24](#).

THRESH_CORR	SYNC Threshold Correlation
1010	10 or more out of 14 bits
1011	11 or more out of 14 bits
1100	12 or more out of 14 bits
1101	13 or more out of 14 bits
1110	14 out of 14 bits

<b>DSCR_TAP</b>	<p>Descrambler Tap—Selects which delay stage, 5th or 18th, to tap for feedback in the descrambler. The system's terminal type dictates which tap should be selected.</p> <p>0 = HTU-C or LTU terminal type, descrambler selects tap 18</p> <p>1 = HTU-R or NTU terminal type, descrambler selects tap 5</p>
<b>DSCR_EN</b>	<p>Descrambler Enable—When enabled, all receive HDSL channel data, except SYNC and STUFF bits, are descrambled per the DSCR_TAP setting. Otherwise the data passes through the descrambler unchanged. DSCR_EN also determines whether RSER and RAUXn data are descrambled.</p> <p>0 = Descrambler bypassed</p> <p>1 = Descrambler enabled</p>
<b>PH_LOOP</b>	<p>Loopback to PCM on HDSL Side—Transmit HDSL data (TDATn) is connected back towards the PCM interface to accomplish a loopback of the PCM channel on the HDSL side. Receive HDSL data (RDATn) is ignored, but HDSL transmit continues without interruption. PH_LOOP requires the descrambler and scrambler to use the same tap, as opposed to their normal opposing tap selection.</p> <p>0 = Normal receive</p> <p>1 = RDATn supplied by TDATn</p>



**RX\_ERR\_EN** Receive Error Interrupt Enable—Receive errors request RX\_ERR interrupt and report RXn\_ERR status upon detection of RFIFO errors [STATUS\_1; addr 0x05], framer state transitions or error counter overflows [STATUS\_2; addr 0x06]. Disabled channels are prevented from activating INTR\*, or setting RX\_ERR [IRR; addr 0x1F]. Receive errors are always latched in ERR\_STATUS [addr 0x3C] regardless of RX\_ERR\_EN.

0 = Disable RX\_ERR interrupts

1 = Enable RX\_ERR interrupts

## 0x62—Receive Elastic Store FIFO Reset (RFIFO\_RST)

Writing any data value to RFIFO\_RST empties the RFIFO and forces the payload mapper to realign HDSL bytes with respect to the receive HDSL 6 ms frame. The MPU must write RFIFO\_RST after modifying the Receive Payload Map [RMAP; addr 0x64] or the Combination Table [COMBINE\_TBL; addr 0xEE] each time the receive framer changes from the SYNC\_ACQUIRED to the IN\_SYNC state [STATUS\_2; addr 0x06]; whenever a RFIFO error is reported [STATUS\_1; addr 0x05]; and after the DPLL has settled. Writing RFIFO\_RST corrupts up to three receive PCM frames worth of data.

## 0x63—Receive Framer Synchronization Reset (SYNC\_RST)

Writing any data value to SYNC\_RST forces the receive framer to the OUT\_OF\_SYNC state, which restarts the SYNC word search and causes the framer to issue an RX\_ERR interrupt. The MPU must write SYNC\_RST after modifying FRAMER\_EN [RCMD\_2; addr 0x61], SYNC\_WORD\_A, or SYNC\_WORD\_B. Writing SYNC\_RST to the master HDSL channel corrupts up to three receive PCM frames worth of data and may cause a DPLL error interrupt.

## 4.5 Receive Payload Mapper

The Receive Payload Map (RMAP\_1–RMAP\_6) controls placement of HDSL payload bytes (byte1–byte36) into the RFIFO by instructing the mapper to place or discard payload bytes from the received payload block. Payload bytes are mapped sequentially from each payload block and cannot be rearranged. Payload is subsequently combined [COMBINE\_TBL; addr 0xEE] at the RFIFO outputs to reconstruct the PCM channel. RMAP is programmed to discard bytes within the payload block that are not needed for PCM reconstruction. In T1 mode, RMAP must be programmed to choose which HDSL channel supplies F-bits, by enabling one extra byte of payload at the end of the payload block.

### 0x64—Receive Payload Map (RMAP\_1)

7	6	5	4	3	2	1	0
—	—	RMAP[5:0]					

### 0x65—Receive Payload Map (RMAP\_2)

7	6	5	4	3	2	1	0
—	—	RMAP[11:6]					

### 0x66—Receive Payload Map (RMAP\_3)

7	6	5	4	3	2	1	0
—	—	RMAP[17:12]					

### 0x69—Receive Payload Map (RMAP\_4)

7	6	5	4	3	2	1	0
—	—	RMAP[23:18]					

### 0x6A—Receive Payload Map (RMAP\_5)

7	6	5	4	3	2	1	0
—	—	RMAP[29:24]					

**0x6B—Receive Payload Map (RMAP\_6)**

7	6	5	4	3	2	1	0
—	—	RMAP[35:30]					

**RMAP[35:0]** Receive Payload Map—Six registers hold a 36-bit value to define which of the received HDSL payload bytes (byte1 through byte36) are placed into the RFIFO. RMAP[0] corresponds to the first HDSL payload byte (byte1). In T1 mode, F-bits are mapped by enabling one extra byte after the last payload mapped byte. For example, RMAP[12] controls F-bit mapping to the RFIFO in 2T1 applications.

If RMAP[x] = 0, discard payload byte(x+1)

If RMAP[x] = 1, map payload byte(x+1) to RFIFO

**0x67—Error Count Reset (ERR\_RST)**

Writing any data value to ERR\_RST clears the receive CRC Error Counter [CRC\_CNT; addr 0x21], the receive Far End Block Error Counter [FEBE\_CNT; addr 0x22], and consequently clears the Counter Overflow (CRC\_OVR) and FEBE\_OVR bits [STATUS\_2; addr 0x06]. ERR\_RST clears the error counters immediately and must be issued within 6 ms after the respective receive frame interrupt in order to avoid clearing unreported errors. No other receive errors (CRC\_ERR, RFIFO, or RX\_STUFF) are affected by ERR\_RST.

**0x68—Receive Signaling Location (RSIG\_LOC)**

7	6	5	4	3	2	1	0
—	—	—	—	RSIG_LOC[3:0]			

**RSIG\_LOC[3:0]** Receive Signaling Location—Is applicable only if RSIG\_EN [CMD\_6; addr 0xF3] enables LTU grooming in a 2E1 or 3E1 Point-to-Multipoint (P2MP) system. The Receive Signaling Table [RSIG\_TBL; addr 0xF2] compensates for differential frame delays between two or three remote sites by delaying the current PCM receive frame sync according to the RSIG\_LOC frame delay values for each HDSL channel. RSIG\_TBL uses each RSIG\_LOC frame delay to locate frame 0 and to transfer ABCD signaling from the respective channel. RSIG\_LOC sets the number of frame delays, from 1 to 16 frames, therefore RSIG\_TBL needs to delay the current receive PCM frame in order to locate frame 0 of the respective channel. A value of zero signifies a one frame delay. A one frame delay corresponds to frame 0 occurring in the first HDSL payload block. RSIG\_LOC values are calculated for each channel from the remote sites measurement of RMSYNC Phase [MSYNC\_PHS; addr 0x39]:

$$RSIG\_LOC = \text{truncate} \left[ \frac{t(RMP)}{FRAME\_LEN} \right] - 1$$

where: *FRAME\_LEN* = PCM bits per frame

*RSIG\_LOC* = Frame delay

*truncate []* = Integer part only

*t(RMP)* = Remote sites RMSYNC to MSYNC phase  
(measured in PCM bits)

**NOTE:** If RSIG\_LOC is negative, then the programmed value equals 15. EOC messaging capability may be used by the NTU to transfer the results of the RMSYNC phase measurement back to the LTU. Remote sites must align HDSL transmit frames to their respective PCM Transmit Multiframe Sync (TMSYNC) for this equation to remain valid.

## 4.6 PCM Formatter

**Table 4-4. PCM Formatter Write Registers**

Address	Register Label	Bits	Name/Description
0xC0	TFRAME_LOC_LO	8	TSER Frame Bit Location
0xC1	TFRAME_LOC_HI	1	TSER Frame Bit Location
0xC2	TMF_LOC	6	TSER Multiframe Location
0xC3	RFRAME_LOC_LO	8	RSER Frame Bit Location
0xC4	RFRAME_LOC_HI	1	RSER Frame Bit Location
0xC5	RMF_LOC	6	RSER Multiframe Location
0xC6	MF_LEN	6	PCM Multiframe Length
0xC7	MF_CNT	6	PCM Multiframes per HDSL Frame
0xC8	FRAME_LEN_LO	8	PCM Frame Length
0xC9	FRAME_LEN_HI	1	PCM Frame Length

The PCM formatter supports connections to many types of PCM channels by allowing the system to define the PCM bus format and sync timing characteristics. PCM frame length, multiframe length, and PCM multiframes per HDSL frame are programmed in the PCM formatter registers to define receive and transmit timebases. Programmed frame and multiframe lengths for both timebases allows the RS8953B to continue operating at appropriate intervals when PCM transmit sync or HDSL receive sync references are lost, and when RS8953B acts as the PCM bus master. The transmit timebase controls the routing of PCM timeslots into the transmit FIFOs, while the receive timebase controls the extraction of PCM timeslots out of the receive FIFOs. The number of multiframes per HDSL frame is needed to generate PCM 6 ms timebases used for transmit bit stuffing and Digital Phase Lock Loop (DPLL) receive clock recovery.

PCM formatter configuration registers also define the PCM timing relationships between transmit data (TSER, INSDAT) and sync (TMSYNC, MSYNC), and receive data (RSER) and sync (RMSYNC). TMSYNC is delayed by a programmed number of bits and frames to create the MSYNC output signal. MSYNC is then used to locate the first bit (bit 0) of a frame, and the first frame (frame0) of a multiframe at the TSER input. MSYNC is always used to align both PCM and HDSL transmit timebases, regardless of whether TMSYNC is applied. RMSYNC is output from the receive PCM timebase after it is delayed by a programmed number of bits and frames.

**NOTE:** The internal PCM receive timebase is frame and multiframe aligned with respect to the master HDSL channel's receive 6 ms frames [refer to RFIFO\_WL; addr 0xCD]. The internal PCM receive timebase is not affected by programmed bit and frame delays for RMSYNC.

### 0xC0—TSER Frame Bit Location (TFRAME\_LOC\_LO)

TFRAME\_LOC and TMF\_LOC work in conjunction to define the location of bit 0, frame 0, at the TSER data input with respect to TMSYNC.

7	6	5	4	3	2	1	0
TFRAME_LOC[7:0]							

### 0xC1—TSER Frame Bit Location (TFRAME\_LOC\_HI)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	TFRAME_LOC[8]

**TFRAME\_LOC[8:0]** TSER Frame Bit Location—Establishes the number of PCM bit delays, in the range of 1 to 512 bits, from the rising edge of TMSYNC until PCM bit 0 is sampled on TSER. A value of 0 delays TMSYNC by three TCLK periods. If TMSYNC and TSER are input aligned, where TMSYNC's rising edge coincides with TSER input of PCM bit 0, then TFRAME\_LOC is programmed to equal the PCM frame length minus three. The following examples assume TMSYNC and TSER are input aligned:

PCM Frame Length	TFRAME_LOC[8:0] = Decimal (hex)
E1 = 256 bits	253 (0x0FD)
T1 = 193 bits	190 (0x0BE)
64x64 = 512 bits	509 (0x1FD)

### 0xC2—TSER Multiframe Bit Location (TMF\_LOC)

7	6	5	4	3	2	1	0
—	—	TMF_LOC[5]					

**TMF\_LOC[5:0]** TSER Multiframe Bit Location—TMF\_LOC sets the number of frame delays, in the range of 1 to 64 frames, from TMSYNC (delayed by TFRAME\_LOC) until PCM frame 0 is present on TSER. A value of 0 delays TMSYNC by one PCM frame. If TMSYNC and TSER are input aligned, TMF\_LOC is programmed to equal the multiframe length minus two. The following examples assume TMSYNC's rising edge coincides with PCM frame 0 input on TSER:

PCM Multiframe Length	TMF_LOC[5:0] = Decimal (hex)
E1 = 16 frames	14 (0x0E)
SF = 12 frames	10 (0x0A)
ESF = 24 frames	22 (0x16)

### 0xC3—RSER Frame Bit Location (RFRAME\_LOC\_LO)

RFRAME\_LOC and RMF\_LOC work in conjunction to define which RSER bit and frame location is marked by the RMSYNC output. Typically, RMSYNC is used as a PCM multiframe sync signal and is programmed to mark during RSER output of bit0, frame0. However, any RSER bit location within the received multiframe can be marked as desired.

7	6	5	4	3	2	1	0
RFRAME_LOC[7:0]							

### 0xC4—RSER Frame Bit Location (RFRAME\_LOC\_HI)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	RFRAME_LOC[8]

**RFRAME\_LOC[8:0]** RSER Frame Bit Location—Establishes the number of PCM bit delays, in the range of 1 bit to 512 bits, from the internal PCM receive timebase's output of bit 0 to the rising edge of RMSYNC. Due to internal bit delays, a value of two will delay RMSYNC by one RCLK period, in which case the rising edge of RMSYNC coincides with output of RSER bit 1. If the system desires RMSYNC to mark RSER bit0, then RFRAME\_LOC is programmed to equal 1. The following examples assume RMSYNC is desired to mark RSER bit 0:

PCM Frame Length	RFRAME_LOC[8:0] = Decimal (hex)
E1 = 256 bits	1 (0x01)
T1 = 193 bits	1 (0x01)
64x64 = 512 bits	1 (0x01)

### 0xC5—RSER Multiframe Bit Location (RMF\_LOC)

7	6	5	4	3	2	1	0
—	—	RMF_LOC[5:0]					

**RMF\_LOC[5:0]** RSER Multiframe Bit Location—Establishes the number of PCM frame delays, in the range of 1 to 64 frames, from the internal PCM receive timebase's output of frame 0 to the rising edge of RMSYNC. Due to internal frame delay, a value of one delays RMSYNC by one PCM frame. RMF\_LOC enacts the RMSYNC frame delay after the RFRAME\_LOC bit delay. If the system desires RMSYNC to mark RSER frame 0, then RMF\_LOC is programmed to equal 0. The following examples assume RMSYNC is desired to mark RSER frame 0:

PCM Multiframe Length	RMF_LOC[5:0] = Decimal (hex)
E1 = 16 frames	0 (0x00)
SF = 12 frames	0 (0x00)
ESF = 24 frames	0 (0x00)

### 0xC6—PCM Multiframe Length (MF\_LEN)

7	6	5	4	3	2	1	0
—	—	MF_LEN[5:0]					

**MF\_LEN[5:0]** PCM Multiframe Length—Contains the number of PCM frames in one PCM multiframe, in the range of 1 to 64 frames. A value of zero selects one frame per multiframe, which causes TMSYNC and RMSYNC to operate at the PCM frame rate.

### 0xC7—PCM Multiframe per HDSL Frame (MF\_CNT)

7	6	5	4	3	2	1	0
—	—	MF_CNT[5:0]					

**MF\_CNT[5:0]** PCM Multiframe per HDSL Frame—Contains the number of PCM multiframe in one HDSL 6 ms frame, in the range of 1 to 64 multiframe. A value of zero selects one multiframe per HDSL frame. MF\_CNT operates in conjunction with FRAME\_LEN and MF\_LEN to create transmit and receive PCM 6 ms timebases which are needed to perform transmit bit stuffing and DPLL receive clock recovery. The RS8953B requires the product of MF\_LEN and MF\_CNT to always equal 48 to match the number of HDSL payload blocks in an HDSL frame. For example:

PCM Multiframe	MF_LEN[5:0]	MF_CNT[5:0]	Product
E1 = 16 frames	15 (0x0F)	2 (0x02)	16 x 3 = 48
SF = 12 frames	11 (0x0B)	3 (0x03)	12 x 4 = 48
ESF = 24 frames	23 (0x17)	1 (0x01)	24 x 2 = 48
Unframed = 1 frame	0 (0x00)	47 (0x2F)	1 x 48 = 48

### 0xC8—PCM Frame Length (FRAME\_LEN\_LO)

7	6	5	4	3	2	1	0
FRAME_LEN[7:0]							

### 0xC9—PCM Frame Length (FRAME\_LEN\_HI)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	FRAME_LEN[8]

**FRAME\_LEN[8:0]** PCM Frame Length—Contains the number of bits in one PCM frame, in the range of 16 to 512 bits. A value of 255 selects 256 bit PCM frame length. The selected value includes payload (number of timeslots x 8) and framing bits. For example, FRAME\_LEN value equals 192 (0xC0) to select a 193-bit T1 frame.



## 4.7 HDSL Channel Configuration

**Table 4-5. HDSL Channel Configuration Write Registers**

Address	Register Label	Bits	Name/Description
0xCA	HFRAME_LEN_LO	8	HDSL Frame Length
0xF5	HFRAME_LEN_HI	1	HDSL Frame Length
0xF8	HFRAME2_LEN_LO	8	HDSL Frame Length
0xF9	HFRAME2_LEN_HI	1	HDSL Frame Length
0xFA	HFRAME3_LEN_LO	8	HDSL Frame Length
0xFB	HFRAME3_LEN_HI	1	HDSL Frame Length
0xCB	SYNC_WORD_A	7	SYNC Word A (sign only)
0xCC	SYNC_WORD_B	7	SYNC Word B (sign only)
0xCD	RFIFO_WL_LO	8	RX FIFO Water Level
0xCE	RFIFO_WL_HI	2	RX FIFO Water Level
0xCF	STF_THRESH_A_LO	8	Stuffing Threshold A
0xD0	STF_THRESH_A_HI	2	Stuffing Threshold A
0xD1	STF_THRESH_B_LO	8	Stuffing Threshold B
0xD2	STF_THRESH_B_HI	2	Stuffing Threshold B
0xD3	STF_THRESH_C_LO	8	Stuffing Threshold C
0xD4	STF_THRESH_C_HI	2	Stuffing Threshold C

### 0xCA—HDSL Frame Length (HFRAME\_LEN\_LO)

7	6	5	4	3	2	1	0
HFRAME_LEN[7:0]							

**0xF5—HDSL Frame Length (HFRAME\_LEN\_HI)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	HFRAME_LEN[8]

**HFRAME\_LEN[8:0]** HDSL Payload Block Length—Contains the number of BCLKn bits, in the range of 16 to 512, that are transmitted and received in an HDSL payload block. Each payload block is comprised of an integer number of 8-bit bytes plus an additional F-bit or Z-bit. The RS8953B repeats the payload block length 48 times to form one HDSL frame. A value of 15 selects a 16-bit payload block length; therefore, the programmed value of HFRAME\_LEN equals eight times the number of payload bytes. For example, a value of 96 (0x60) selects a 12-byte T1 payload or 144 (0x90) selects an 18-byte E1 payload. Value written to HFRAME\_LEN are copied into HFRAME2\_LEN and HFRAME3\_LEN.

In theory, the HDSL frame length can be different for each loop. But if the HDSL frame lengths are not close to the same value, RFIFO errors will probably occur. For this reason, we do not recommend different values for the HDSL frame length.

**0xF8—HDSL Frame Length (HFRAME2\_LEN\_LO)**

7	6	5	4	3	2	1	0
HFRAME_LEN[7:0]							

**0xF9—HDSL Frame Length (HFRAME2\_LEN\_HI)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	HFRAME_LEN[8]

**HFRAME2\_LEN[8:0]** HDSL Payload Block Length—Contains the number of BCLK2 bits in the range of 16 to 512, that are transmitted and received in an HDSL payload block for Channel 2.

In theory, the HDSL frame length can be different for each loop. But if the HDSL frame lengths are not close to the same value, RFIFO errors will probably occur. For this reason, we do not recommend different values for the HDSL frame length.

**0xFA—HDSL Frame Length (HFRAME3\_LEN\_LO)**

7	6	5	4	3	2	1	0
HFRAME_LEN[7:0]							

**0xFB—HDSL Frame Length (HFRAME3\_LEN\_HI)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	HFRAME_LEN[8]

**HFRAME3\_LEN[8:0]** HDSL Payload Block Length—Contains the number of BCLK3 bits, in the range of 16 to 512, that are transmitted and received in an HDSL payload block for Channel 3.

In theory, the HDSL frame length can be different for each loop. But if the HDSL frame lengths are not close to the same value, RFIFO errors will probably occur. For this reason, we do not recommend different values for the HDSL frame length.

**0xCB—SYNC Word A (SYNC\_WORD\_A)**

7	6	5	4	3	2	1	0
—	SYNC_WORD_A[6:0]						

**SYNC\_WORD\_A[6:0]** SYNC Word A—Holds the 7 sign bits (+/–) of the 7-quat (14-bit) transmit and receive SYNC word. Transmit SYNC word magnitude bits are forced to 0. SYNC\_WORD\_A[0] is the sign bit of the first transmit quat. Sign precedes magnitude on the transmit data (TDATn) output. The receive framer searches HDSL data (RDATn) for patterns matching SYNC\_WORD\_A and/or SYNC\_WORD\_B according to the criteria selected in FRAMER\_EN [RCMD\_1; addr 0x60].

0 = Negative sign bit  
1 = Positive sign bit

**0xCC—SYNC Word B (SYNC\_WORD\_B)**

7	6	5	4	3	2	1	0
—	SYNC_WORD_B[6:0]						

**SYNC\_WORD\_B[6:0]** SYNC Word B—Holds the 7 sign bits (+/–) of the transmit and receive SYNC word. It performs the same function as SYNC\_WORD\_A (see above). SYNC\_WORD\_B is provided for 2T1 applications that use different SYNC patterns on each HDSL channel for loop identification purposes. Transmit selection of SYNC word A or B is programmed by SYNC\_SEL [TCMD\_1; addr 0x06].

0 = Negative sign bit  
1 = Positive sign bit

**0xCD—RX FIFO Water Level (RFIFO\_WL\_LO)**

7	6	5	4	3	2	1	0
RFIFO_WL[7:0]							

**0xCE—RX FIFO Water Level (RFIFO\_WL\_HI)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	RFIFO_WL[9:8]	

**RFIFO\_WL[8:0]** Receive FIFO Water Level—Sets the RCLK bit delay from the master HDSL channel's receive 6 ms frame to the PCM receive 6 ms frame. The delay is programmed in RCLK bit intervals, in the range of 1 to 1,024 bits. A value of zero equals one RCLK bit delay. The minimum RFIFO\_WL value must allow sufficient time to elapse for payload to pass through the RFIFO. The maximum RFIFO\_WL must not allow more than 185 bits to be present in the RFIFO at any given time.

**4.8 Transmit Bit Stuffing Thresholds**

The STUFF generator in each HDSL transmit channel makes bit stuffing decisions based upon phase comparisons of the difference between PCM transmit 6 ms frames and HDSL transmit 6 ms frames, with respect to two programmable Stuffing Thresholds [STF\_THRESH and STF\_THRESH\_C; addr 0xD1–D4]. Results of the phase comparisons determine whether the HDSL channel's STUFF generator inserts 0 STUFF bits or 4 STUFF bits in the outgoing HDSL frame. Inserted STUFF bit values are supplied by TSTUFF [addr 0xE4]. The General Purpose Clock (GCLK) is used to quantize phase differences between PCM and HDSL frame starting locations. GCLK is developed from the MCLK frequency ( $f_{MCLK}$ ), PLL Multiplication (PLL\_MUL) and PLL Division (PLL\_DIV) scale factors [CMD\_1; addr 0xE5]. The STUFF generator makes bit stuffing decisions using the following criteria:

PCM to HDSL Phase Difference	Inserted STUFF Bits
< STF_THRESH_A	0
≥ STF_THRESH_A	4
< STF_THRESH_C	4
≥ STF_THRESH_C	<sup>(1)</sup> 4

<sup>(1)</sup> A phase difference measured to be equal to or in excess of STF\_THRESH\_C is reported as a transmit stuffing error in STUFF\_ERR [STATUS\_3; addr 0x07].

Stuffing threshold values are programmed to set the nominal and maximum tolerable phase difference in units of GCLK phase. STUFF insertion accounts for ± 4 HDSL bits worth of phase error and STUFF thresholds are set to equal 16 or 24 HDSL bits worth of phase at the BCLKn frequency ( $f_{HDSL}$ ), as shown in the following equation:

$$StuffingThreshold = \frac{n \times f_{MCLK}}{f_{HDSL}} \times \frac{PLL\_MUL}{PLL\_DIV}$$

where:  $n = 8$  for STF\_THRESH\_A  
 $n = 12$  for STF\_THRESH\_B  
 $n = 24$  for STF\_THRESH\_C

### 0xCF—Bit Stuffing Threshold A (STF\_THRESH\_A\_LO)

7	6	5	4	3	2	1	0
STF_THRESH_A[7:0]							

### 0xD0—Bit Stuffing Threshold A (STF\_THRESH\_A\_HI)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	STF_THRESH_A[9:8]	

**STF\_THRESH\_A[8:0]** Bit Stuffing Threshold A—Contains the number of GCLK cycles equaling 8 HDSL bit times. If the phase measured from PCM to HDSL 6 ms frames is a positive value greater than or equal to STF\_THRESH\_A, then 4 STUFF bits are inserted in the outgoing HDSL frame. If the phase is a positive value less than STF\_THRESH\_A, then STUFF bits are not inserted in the outgoing HDSL frame. If the phase is a negative value, then the phase tolerance on HDSL, PCM, or GCLK inputs is exceeded and the STUFF generator reports STUFF\_ERR [STATUS\_3; addr 0x07].

### 0xD1—Bit Stuffing Threshold B (STF\_THRESH\_B\_LO)

7	6	5	4	3	2	1	0
STF_THRESH_B[7:0]							

### 0xD2—Bit Stuffing Threshold B (STF\_THRESH\_B\_HI)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	STF_THRESH_B[9:8]	

**STF\_THRESH\_B[8:0]** Bit Stuffing Threshold B—Contains the number of GCLK cycles equaling 12 HDSL bit times.

### 0xD3—Bit Stuffing Threshold C (STF\_THRESH\_C\_LO)

7	6	5	4	3	2	1	0
STF_THRESH_C[7:0]							

**0xD4—Bit Stuffing Threshold C (STF\_THRESH\_C\_HI)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	STF_THRESH_C[9:8]	

**STF\_THRESH\_C[8:0]** Bit Stuffing Threshold C—Contains the number of GCLK cycles equal to 24 HDSL bit times. If the phase measured from PCM to HDSL 6 ms frames is a positive value less than STF\_THRESH\_C, then 4 STUFF bits are inserted in the outgoing frame. If the phase is a positive value greater than or equal to STF\_THRESH\_C, then the phase tolerance on HDSL, PCM, or GCLK inputs is exceeded and the STUFF generator reports STUFF\_ERR [STATUS\_3; addr 0x07].

**NOTE:** STF\_THRESH\_C must be greater than STF\_THRESH\_B by a value of 4 HDSL bit times ( $4 \times \text{HDSL} \div \text{GCLK}$ ) but not greater than a value of 0x3FF.

## 4.9 DPLL Configuration

**Table 4-6. DPLL Configuration Write Registers**

Address	Register Label	Bits	Name/Description
0xD5	DPLL_RESID_LO	8	DPLL Residual
0xD6	DPLL_RESID_HI	8	DPLL Residual
0xD7	DPLL_FACTOR	8	DPLL Factor
0xD8	DPLL_GAIN	7	DPLL Gain
0xDB	DPLL_PINI	8	DPLL Phase Detector Init (optional for RS8953B)
0xF6	DPLL_RST	—	DPLL Phase Detector Reset

The DPLL synthesizes the PCM Receive Clock (RCLK) output from the 60 through 80 MHz Reference Clock (HFCLK) generated internally by PLL multiplication of MCLK, or input directly on MCLK [see PLL\_MUL and PLL\_DIS in CMD\_1; addr 0xE5]. HFCLK must operate in the 60 to 80 MHz frequency range, but requires no specific phase or frequency relationship to the PCM or HDSL channels. The nominal frequency ( $f_{PCM}$ ) of RCLK is synthesized by setting the DPLL\_FACTOR and DPLL\_RESID values according to the integer and fractional results of the following formula:

$$[\text{INTEGER.FRACTION}] = \left( \frac{f_{MCLK} \times PLL\_MUL}{2 \times f_{PCM}} \right)$$

where:  $f_{MCLK}$  = MCLK input frequency  
 $f_{PCM}$  = RCLK output frequency desired  
 INTEGER = Integer part of result [DPLL\_FACTOR; addr 0xD7]  
 FRACTION = Fractional part of result [DPLL\_RESID; addr 0xD5]  
 PLL\_MUL = PLL multiplication factor [CMD\_1; addr 0xE5]  
 PLL\_DIV = PLL scale factor [CMD\_1; addr 0xE5]

The DPLL phase detector operates from the 10–15 MHz General Purpose Clock (GCLK) which equals HFCLK divided by PLL scale factor:

$$\text{GCLK} = \left( \frac{f_{MCLK} \times PLL\_MUL}{PLL\_DIV} \right)$$

### 0xD5—DPLL Residual (DPLL\_RESID\_LO)

7	6	5	4	3	2	1	0
DPLL_RESID[7:0]							

0xD6—DPLL Residual (DPLL\_RESID\_HI)

7	6	5	4	3	2	1	0
DPLL_RESID[15:8]							

**DPLL\_RESID[15:0]** DPLL Residual—Works in conjunction with DPLL\_FACTOR to define the DPLL nominal free-running frequency in Open Loop Mode or the DPLL initial frequency in Closed Loop Mode [DPLL\_NCO in CMD\_5; addr 0xE9]. The DPLL\_RESID value is sampled by the DPLL only after the MPU writes RX\_RST [address 0xF1], or after the master HDSL channel’s receive framer transitions to an IN\_SYNC state.

$$DPLL\_RESID = round(FRACTION \times 65535)$$

$$DPLL\_FACTOR = 257 - INTEGER$$

where: *round ( )* = Round to nearest integer  
FRACTION = Fraction from INTEGER.FRACTION calculation  
INTEGER = Integer from INTEGER.FRACTION calculation

Assuming MCLK operates at eight times the BCLKn frequency (16 times symbol rate) and RCLK is desired to operate at standard T1 or E1 clock rates. The following examples show HTU application values for DPLL\_RESID and DPLL\_FACTOR:

HTU	PLL_MUL	PLL_DIV	DPLL_FACTOR	DPLL_RESID
2T1	11	6	0xEB	0x578B
3E1	11	6	0xF1	0xD7FF
2E1	8	6	0xEF	0x4000

0xD7—DPLL Factor (DPLL\_FACTOR)

7	6	5	4	3	2	1	0
DPLL_FACTOR[7:0]							

**DPLL\_FACTOR[7:0]** DPLL Factor—Works in conjunction with DPLL\_RESID.



**0xD8—DPLL Gain (DPLL\_GAIN)**

7	6	5	4	3	2	1	0
—	DC_GAIN[2:0]			DC_INTEG[3:0]			

**DPLL\_GAIN[7:0]** DPLL Gain—Filtering is controlled by two DC parameters: DC\_GAIN, which represents proportional loop gain, and DC\_INTEG, which represents the filter's integration coefficient. The DPLL closed loop bandwidth is programmed to be in the range of 0.2 Hz to 3 Hz. The following approximations are used to calculate DC parameters for a desired DPLL bandwidth:

$$\text{DC\_GAIN} = \left[ \frac{BW}{N \times 26.5} \right] \times 2^{17}$$

$$\text{DC\_INTEG} = \left[ \frac{(BW)^2}{26.5^2} \right] \times \frac{2^{15}}{N}$$

where:  $N$  = RCLK output frequency ÷ 64000

$BW$  = DPLL closed loop bandwidth (in Hz)

Specific DC parameter values are programmed according to the following tables:

DC_GAIN[2:0]	RS8953B
000	$2^5$
001	$2^6$
010	$2^7$
011	$2^8$
100	$2^9$
101	$2^{10}$
110	$2^{11}$
111	$2^{12}$
DC_INTEG[3:0]	RS8953B
0000	$2^{-4}$
0001	$2^{-3}$
0010	$2^{-2}$
0011	$2^{-1}$
0100	1
0101	$2^1$
0110	$2^2$
0111	$2^3$
1000	$2^4$
1001	$2^5$
1010–1110	$2^6$
1111	0 (Type I)

**0xDB—DPLL Phase Detector Init (DPLL\_PINI)**

7	6	5	4	3	2	1	0
DPLL_PINI[7:0]							

**DPLL\_PINI[7:0]** DPLL Phase Detector Init (optional for RS8953B)—Phase detector init mode [PHD\_MODE in CMD\_7; addr 0xF4] selects whether DPLL\_PINI is supplied by the MPU or is calculated automatically. When MPU supplied, DPLL\_PINI sets the initial point within the phase comparator window that the phase detector returns to after detection of a DPLL error. The RS8953B phase window is 1,024 GCLK cycles. For example, the RS8953B requires a programmed value for DPLL\_PINI which is typically set to init phase window at its center point (i.e., 512 GCLK cycles) from the following formula:

$$\text{DPLL\_PINI} = \text{round}\left[\frac{512 \times \text{BCLK}}{4 \times \text{GCLK}}\right]$$

**NOTE:** The loaded value is internally multiplied by 4 when used to initialize the phase detector.

**0xF6—Reset DPLL Phase Detector (DPLL\_RST)**

Writing any data value to DPLL\_RST clears the phase detector error output, restarts the phase comparator window, and clears pending DPLL error interrupts. The MPU is not required to write DPLL\_RST, unless the MPU has instructed the Phase Detector Init Mode [PHD\_MODE in CMD\_7; addr 0xF4] to disable automatic initialization, or unless FAST\_ACQ in CMD\_7 is enabled and the system needs to reacquire the DPLL frequency.

## 4.10 Data Path Options

**Table 4-7. Data Path Options Write Registers**

Address	Register Label	Bits	Name/Description
0xDC	DBANK_1	8	Data Bank Pattern 1
0xDD	DBANK_2	8	Data Bank Pattern 2
0xDE	DBANK_3	8	Data Bank Pattern 3
0xEA	FILL_PATT	8	Programmable Fill Pattern (Data Bank Pattern 4)
0xE4	TSTUFF	4	Transmit Stuff Bit Value
0xED	ROUTE_TBL	7	Transmit Routing Table
0xEE	COMBINE_TBL	6	Receive Combination Table
0xF2	RSIG_TBL	4	Receive Signaling Table

### 0xDC—Data Bank Pattern 1 (DBANK\_1)

7	6	5	4	3	2	1	0
DBANK_1[7:0]							

**DBANK\_1[7:0]** Data Bank Pattern 1—Holds an 8-bit programmable pattern that can be used to replace transmit HDSL payload bytes and/or receive PCM timeslots according to the Transmit Payload Map [TMAP; addr 0x08] and the Receive Combination Table [COMBINE\_TBL; addr 0xEE] selections. Both transmit and receive can simultaneously use the same DBANK contents. DBANK\_1[0] is the first bit inserted in the selected direction.

The following configuration can cause the HDSL frame to become corrupted:  
 DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte 33.  
 TFIFO or DBank\_1 is the source of data for HDSL payload byte 0.  
 The LSB or the DBank\_1 pattern is equal to one (1).

### 0xDD—Data Bank Pattern 2 (DBANK\_2)

7	6	5	4	3	2	1	0
DBANK_2[7:0]							

**DBANK\_2[7:0]** Data Bank Pattern 2—Provides another 8-bit pattern for insertion in transmit HDSL payload bytes or receive PCM timeslots. See DBANK\_1 above. Multiple DBANK registers may be needed to fill transmit HDSL payload bytes reserved by ETSI standards for future applications. For example, ETSI specifies R and Y bytes within a 2E1 payload block that are currently set to all 1s.

The following configuration can cause the HDSL frame to become corrupted:  
 DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte 33.  
 DBank\_2 is the source of data for HDSL payload byte 0.  
 The LSB of the DBank\_2 pattern is equal to one (1).

### 0xDE—Data Bank Pattern 3 (DBANK\_3)

7	6	5	4	3	2	1	0
DBANK_3[7:0]							

**DBANK\_3[7:0]** Data Bank Pattern 3—Holds a third possible 8-bit pattern for transmit or receive insertion. See DBANK\_1 above. If RSIG\_EN = 1 [CMD\_6; addr 0xF3], DBANK\_3 is a receive signaling buffer and is not available as an alternate source for receive PCM timeslots. If TAUX\_EN = 1 [TCMD\_2; addr 0x07], DBANK\_3 is a transmit auxiliary channel data buffer and is not available for insertion into transmit HDSL payload bytes but remains available for insertion into RSER timeslots.

The following configuration can cause the HDSL frame to become corrupted:  
 DBank\_1, DBank\_2, or DBank\_3 is the source of data for HDSL payload byte 33.  
 DBank\_3 is the source of data for HDSL payload byte 0.  
 The LSB of the DBank\_3 pattern is equal to one (1).

### 0xEA—Fill Pattern (FILL\_PATT)

7	6	5	4	3	2	1	0
FILL_PATT[7:0]							

**FILL\_PATT[7:0]** Fill Pattern—When PRBS\_DIS [CMD\_3; addr 0xE7] is set, FILL\_PATT replaces the PRBS generator output with its 8-bit programmable pattern. The transmit Routing Table [ROUTE\_TBL; addr 0xED] may then select FILL\_PATT as a fourth possible data bank to fill idle or unpopulated PCM timeslots and HDSL payload bytes. In this case, FILL\_PATT also establishes an 8-bit pattern checked by the receiver's BER meter when enabled [BER\_EN in COMBINE\_TBL; addr 0xEE].

When PRBS\_DIS is zero (PRBS enabled), FILL\_PATT is used to initialize the least significant byte of the PRBS generator's LFSR. In this case, FILL\_PATT must be initialized to any non-zero value before the MPU issues the PRBS\_RST command.

### 0xE4—Transmit Stuff Bit Value (TSTUFF)

7	6	5	4	3	2	1	0
—	—	—	—	TSTUFF[3:0]			
				MAG1	SIGN1	MAG0	SIGN0

**TSTUFF[3:0]** Transmit Stuffing Bits—Contains the 4-bit STUFF value used by all HDSL transmitters when any HDSL output frame contains bit stuffing. TSTUFF[0] is the sign bit and first bit of the first quat transmitted during STUFF words.

## 0xED—Transmit Routing Table (ROUTE\_TBL)

MPU access to the transmit routing table's single (ROUTE\_TBL) register is enabled by first setting ROUTE\_EN [CMD\_3; addr 0xE7] to reset the table pointer. The MPU can then write up to 64 table entries sequentially to the ROUTE\_TBL address. The RS8953B increments the internal table pointer after each write to ROUTE\_TBL.

Any writes beyond 64 will wrap around and overwrite the initial table entries. The first table entry written corresponds to the first transmit PCM timeslot, which is the 8-bit period starting at MSYNC's rising edge. Subsequent table writes increment the table pointer towards successive PCM timeslots. Standard E1 requires 32 table writes, corresponding to 32 timeslots. Standard T1 requires 25 table writes, where the F-bit location is treated as the 25th timeslot. An Nx64 transmit PCM channel may require up to 64 table writes, corresponding to the 4.096 Mbit/s data rate. After the MPU writes the required number of table entries, the MPU writes zero to ROUTE\_EN to prevent further table access, and then write TFIFO\_RST [addr 0x0D] on every HDSL channel to realign the transmit elastic stores if the aggregate HDSL data rate is modified. Subsequent table changes can rewrite only the necessary entries up to and including the last desired modification.

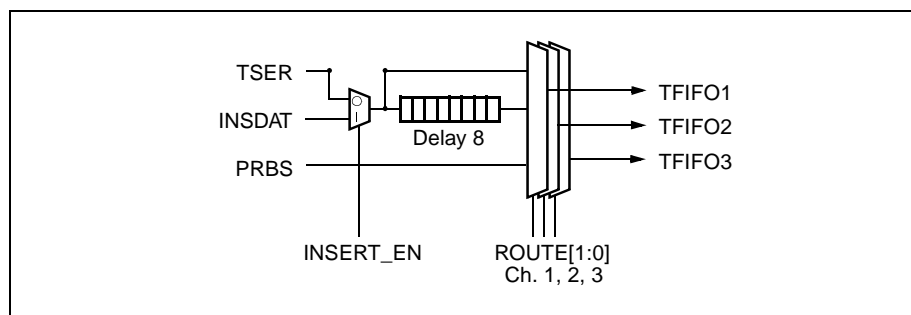
7	6	5	4	3	2	1	0
—	INSERT_EN	ROUTE[1:0] CH3		ROUTE[1:0] CH2		ROUTE[1:0] CH1	

**ROUTE[1:0]** Routing Code—Three identical routing codes are present in each table entry to select which data source is routed to each one of three HDSL channel destinations (CH1–CH3). Route data is available from three sources: PCM Transmit Serial data (TSER), PCM Insert Serial Data (INSDAT), and PRBS generator data. In addition, TSER data is available from an 8-bit delay buffer to allow routing codes to repeatedly (twice) use the same TSER byte as a data source. PCM timeslot data can also be discarded by selecting no destination channels.

Note that INSDAT is available only from the 8-bit delay buffer, and cannot be repeated in the same manner as TSER. INSDAT occupies delay buffer space and prevents routing of previous TSER data during the timeslot following INSERT\_EN. For example, if INSERT\_EN is active in the timeslot 1 table entry, then during timeslot 2, the delay buffer contains INSDAT, and not the previous TSER. The PRBS generator is active only during timeslots that select PRBS data which allows discontinuous timeslots to be tested with a single continuous PRBS test pattern. Sequential timeslot routing is performed from inputs to destination channel(s) without reordering of timeslots. [Figure 4-1](#) illustrates the effect of ROUTE[1:0] and INSERT\_EN on TSER, INSDAT, and PRBS data routing.

ROUTE[1:0]	Source of Transmit HDSL Channel Data
00	Discard, do not route timeslot data
01	TSER
10	PRBS (or FILL_PATT, if PRBS_DIS = 1)
11	Previous TSER (or INSDAT) from delay buffer

**Figure 4-1. Transmit Routing**



**INSERT\_EN** Enable INSERT—Controls the state of the internal MUX and the INSERT output pin during the corresponding PCM timeslot's sample time. The next table entry is programmed to select the previous timeslot (ROUTE = 11) and to place INSDAT data from the previous timeslot into the TFIFO.

0 = INSERT output pin remains inactive (low)

1 = INSERT output pin active (high)

## 0xEE—Receive Combination Table (COMBINE\_TBL)

MPU access to the Receive Combination Table's (COMBINE\_TBL) single register is enabled by writing COMB\_EN [CMD\_3; addr 0xE7], and by writing a maximum of 64 table entries sequentially to COMBINE\_TBL. Each write increments the table pointer, and the first write corresponds to the first receive PCM timeslot. Subsequent writes increment the table pointer to successive timeslots. After writing the required number of table entries (see ROUTE\_TBL), the MPU writes COMB\_EN to disable table access, and then writes RFIFO\_RST [addr 0x62] on every HDSL channel to realign the receive elastic stores. The MPU possibly writes RX\_RST [addr 0xF1] if the aggregate PCM data rate has been modified. Subsequent table changes can rewrite entries only up to and including the last desired modification.

7	6	5	4	3	2	1	0
—	—	DBANK_SEL		DROP_EN	BER_EN	COMBINE[1:0]	

**COMBINE[1:0]** Combine Code—Selects one of four data sources for output on RSER during the respective receive PCM timeslot destination. The data source is selected from one of three HDSL receive channels or the DBANK register. The first combine code that selects data from a HDSL channel will receive the first payload byte mapped from that channel's payload block, as determined by the payload map [RMAP; addr 0x64]. Whenever combine [1:0] is not 00, DBANK\_SEL[1:0] must be 00.

COMBINE[1:0]	Source of RSER Data
00	Determined by DBANK_SEL[1:0]
01	HDSL receive channel 1
10	HDSL receive channel 2
11	HDSL receive channel 3

**BER\_EN** BER Meter Enable—Places a copy of the respective PCM timeslot's data into the BER meter. Any number of timeslots may be copied without affecting throughput.

0 = BER Meter ignores PCM timeslot

1 = BER Meter receives copy of PCM timeslot data from RSER

**DROP\_EN** Enable DROP—Controls the state of the DROP output pin which marks the respective timeslot coincident with data output on RSER.

0 = DROP output pin remains inactive (low)

1 = DROP output pin active (high)

**DBANK\_SEL[1:0]** Data Bank Select (Applicable only if COMBINE = 00)—Selects one of three DBANK registers to output on RSER during the respective timeslot.

DBANK_SEL[1:0]	Source of RSER Output Data
00	Determined by COMBINE[1:0]
01	DBANK_1; addr 0xDC
10	DBANK_2; addr 0xDD
11	Determined by RSIG_EN
RSIG_EN	RSER Source
0	DBANK_3; addr 0xDE
1	RSIG_TBL; addr 0xF2

## 0xF2—Receive Signaling Table (RSIG\_TBL)

Applicable only to the LTU grooming site in a 2E1 or 3E1 Point-to-Multipoint (P2MP) system, the receive signaling table assembles E1 Timeslot 16 (TS16) from the ABCD signaling supplied by the three remote sites. Signaling from each channel is located by RSIG\_TBL selection of a particular timeslot in the receive combination table, and automatically sampled when RSIG\_EN [CMD\_6; addr 0xF3] is active. The groomed signaling table output replaces the DBANK\_3 register selection in the receive Combination Table [COMBINE\_TBL; addr 0xEE].

MPU access to the receive signaling table is provided through the RSIG\_TBL register by first setting RSIG\_WR [CMD\_3; addr 0xE7] to reset the table pointer to zero, and then by writing up to 16 entries sequentially to RSIG\_TBL. The RS8953B increments the table pointer after each write cycle to the RSIG\_TBL address. The first table entry corresponds to the first E1 frame (frame 0) output on RSER and corresponds to subsequent entries to successive frames. Each entry contains two identical RSIG[1:0] grooming codes which select the HDSL channel source for ABCD signaling bits during the respective frame. For example, frame 1 grooming codes select ABCD for E1 channels 1 and 17, frame 2 selects ABCD for E1 channels 2 and 18, and so on. Grooming codes for E1 frame 0 are similar to other E1 frames, and allow the system to select which HDSL channel supplies the CAS Multiframe Alignment Signal (MAS) and which channel supplies the extra and multiframe yellow alarm bits (XYXX). RS8953B does not provide access to the actual received TS16 data and assumes that EOC messages or indicator bits are used to report far-end alarm and status information.

7	6	5	4	3	2	1	0
—	—	—	—	RSIG[1:0]		RSIG[1:0]	

**RSIG[1:0]** Receive Signaling Grooming Code—Selects which HDSL channel supplies ABCD signaling, MAS, or XYXX bits for output on RSER during the PCM timeslot selected by receive combination table. Sixteen table entries correspond to E1 frames 0 through 15, where the most significant grooming code corresponds to the first four bits of the TS16 output.

RSIG[1:0]	TS16 Source
00	None (invalid)
01	HDSL Channel 1
10	HDSL Channel 2
11	HDSL Channel 3

## 4.11 Common Command

**Table 4-8. Common Command Write Registers**

Address	Register Label	Bits	Name/Description
0xE5	CMD_1	8	Configuration
0xE6	CMD_2	8	Configuration
0xE7	CMD_3	8	Configuration
0xE8	CMD_4	8	Configuration
0xE9	CMD_5	8	Configuration
0xF3	CMD_6	8	Configuration
0xF4	CMD_7	7	Configuration

### 0xE5—Command Register 1 (CMD\_1)

7	6	5	4	3	2	1	0
E1_MODE	PLL_DIS	PLL_DIV[1:0]		PLL_MUL[3:0]			

**PLL\_MUL[3:0]** PLL Multiplication Factor—The MCLK input frequency is multiplied from 1 to 16 times by the selected value to create an internal HFCLK approximately equal to 70 MHz and in the range of 60 to 80 MHz for DPLL clock recovery. When PLL\_MUL has a value of 0xF, the PLL output is invalid.

PLL_MUL [hex]	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MCLK Multiplier	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	–

**PLL\_DIV[1:0]** PLL Division Factor—Selects a divisor to scale down the internal HFCLK frequency to create a General Purpose Clock (GCLK) in the frequency range of 10–15 MHz. PLL\_DIV determines the GCLK frequency for the DPLL phase detector and loop filter.

PLL_DIV	HFCLK Divisor
00	2
01	4
10	6
11	8

**PLL\_DIS** PLL Disable—Disables the internal PLL which normally generates HFCLK. When disabled, a 60 to 80 MHz HFCLK must be applied externally on the MCLK input.

- 0 = Normal PLL operation
- 1 = Disable PLL (PLL\_MUL value has no effect)

**E1\_MODE** E1 or Nx64 Mode—Enables insertion of Z-bits from the TZBIT [addr 0x04] registers, and extraction of Z-bits into the RZBIT [addr 0x04] registers. Otherwise, F-bits occupy the first bit of HDSL payload blocks.

- 0 = HDSL payload includes F-bits (T1 mode)
- 1 = HDSL payload includes Z-bits (E1 mode)



**0xE6—Command Register 2 (CMD\_2)**

7	6	5	4	3	2	1	0
GCLK_SEL	PCM_FLOAT	HP_LOOP	PP_LOOP	RCLK_SEL[1:0]		TCLK_SEL[1:0]	

**TCLK\_SEL** PCM Transmit Clock Source—Selects which clock source and clock edge are used for PCM transmit inputs and outputs.

- 00 TCLK (rising edge outputs, falling edge inputs)
- 01 TCLK inverted (falling edge outputs, rising edge inputs)
- 1x PCM receive clock source (see RCLK\_SEL)

**RCLK\_SEL** PCM Receive Clock Source—Selects which clock source and which clock edge is used for PCM receive outputs. See also RCLK\_INV [CMD\_7; addr 0xF4].

- 00 DPLL recovered clock (rising edge outputs)
- 01 EXCLK pin (rising edge outputs)
- 10 EXCLK pin inverted (falling edge outputs)
- 11 PCM transmit clock source (see TCLK\_SEL)

**NOTE:** TCLK\_SEL = 1x and RCLK\_SEL = 11; both must not be set simultaneously.

**PP\_LOOP** Loopback Towards PCM on the PCM Side—The RSER and RMSYNC outputs are connected from TSER and TMSYNC inputs. Signals are switched directly at the I/O pins, without switching the PCM receive clock. The MPU must change RCLK\_SEL to source RCLK from the TCLK input. HDSL transmit and receive channels operate normally, except the receive channel outputs are replaced by loopback signals.

- 0 = Normal PCM receive
- 1 = RSER and RMSYNC supplied by PCM transmit inputs

**HP\_LOOP** Loopback Towards HDSL on the PCM Side—The TSER and TMSYNC inputs are replaced by data and multiframe sync generated from the PCM receive formatter, without switching the PCM transmit clock. The MPU must change TCLK\_SEL to source TCLK from the RCLK output. The PCM receiver operates normally, but the transmit TSER and TMSYNC inputs are ignored.

- 0 = Normal PCM transmit operation
- 1 = Transmit PCM data supplied by PCM receiver channel

**NOTE:** PP\_LOOP and HP\_LOOP cannot be activated simultaneously.

**PCM\_FLOAT** Float PCM Multiframe—Selects whether MSYNC accepts TMSYNC as a frame and/or multiframe sync reference. MSYNC is always used to establish transmit frame and multiframe alignment for PCM and HDSL frames. If PCM\_FLOAT is active, MSYNC ignores TMSYNC and allows unframed or asynchronous payload mapping of PCM frames into HDSL frames. In this case, TFRAME\_LOC and TMF\_LOC [addr 0xC0–0xC2] are also ignored. When PCM\_FLOAT is zero, the TMSYNC input acts as the frame and/or multiframe sync reference for MSYNC.

- 0 = MSYNC accepts TMSYNC as transmit sync reference
- 1 = MSYNC ignores TMSYNC

**GCLK\_SEL** General Purpose Clock Source—Synchronizes MPU bus cycles and quantizes DPLL phase error.

- 0 = GCLK supplied by HFCLK ÷ PLL\_DIV
- 1 = GCLK supplied by TCK pin

**0xE7—Command Register 3 (CMD\_3)**

7	6	5	4	3	2	1	0
RSIG_WR	PRBS_MODE[1:0]		BER_SCALE[1:0]		PRBS_DIS	ROUTE_EN	COMB_EN

**COMB\_EN** Enable Receive Combination Table Access—The write pointer for the combination table [COMBINE\_TBL; addr 0xEE] is reset to 0, and table access is enabled. MPU writes to COMBINE\_TBL are ignored when COMB\_EN is low.

0 = Disable access to COMBINE\_TBL

1 = Enable MPU access to COMBINE\_TBL and reset write pointer

**ROUTE\_EN** Enable Transmit Routing Table Access—The write pointer for the transmit routing table [ROUTE\_TBL; addr 0xED] is reset to 0, and table access is enabled. MPU writes to ROUTE\_TBL are ignored when ROUTE\_EN is low.

0 = Disable access to ROUTE\_TBL

1 = Enable MPU access to ROUTE\_TBL and reset write pointer

**PRBS\_DIS** PRBS Disable—Replaces PRBS generator output with data from the Fill Pattern Register [FILL\_PATT; addr 0xEA]. Fill patterns are routed to the transmit FIFO in the same manner as PRBS patterns.

0 = PRBS generator output enabled

1 = Fill Pattern replaces PRBS data

**BER\_SCALE[1:0]** BER Meter Scale—Selects the test interval over which bit errors are accumulated by the BER Meter [BER\_METER; addr 0x1D]. The test interval is counted only during bits selected and checked by the BER meter. See also BER\_SEL [CMD\_6; addr 0xF3].

BER_SCALE	Test Interval	Approximate Scale
00	$2^{31}$ bits	$2 \times 10^9$
01	$2^{28}$ bits	$2 \times 10^8$
10	$2^{25}$ bits	$3 \times 10^7$
11	$2^{21}$ bits	$2 \times 10^6$

**NOTE:** The time to complete the test interval depends on the number of bytes examined in each frame, where total test time may exceed 9 hours and 19 minutes.

**PRBS\_MODE[1:0]** Pseudo-Random Bit Sequence Length—Establishes the LFSR pattern generated by the transmit PRBS generator and checked by the receive BER meter. There is an inverter in the data path when the  $2^{15}$  test pattern is selected.

PRBS_MODE	Test Pattern	LFSR Tap Selection
00	$2^{23}$	$1 + x^{18} + x^{23}$
01	$2^{20}$ (14-zero limit)	$1 + x^{17} + x^{20}$
10	$2^{15}$	$1 + x^{14} + x^{15}$
11	$2^4$	$1 + x^3 + x^4$

**RSIG\_WR** Enable Receive Signaling Table Access—The write pointer for the Receive Signaling Table [RSIG\_TBL; addr 0xF2] is reset to 0, and table access is enabled. MPU writes to RSIG\_TBL are ignored when RSIG\_WR is low.

0 = Disable access to RSIG\_TBL

1 = Enable MPU access to RSIG\_TBL and reset write pointer

## 0xE8—Command Register 4 (CMD\_4)

Must be set to 0x04 before any other MPU access to device, for normal operation. Other values are reserved for Conexant production test.

## 0xE9—Command Register 5 (CMD\_5)

7	6	5	4	3	2	1	0
DPLL_NCO	MASTER_SEL[1:0]		ZBIT_SEL[1:0]		EXT_STUFF	STUFF_SEL[1:0]	

**STUFF\_SEL[1:0]** Master STUFF source is applicable only if SLV\_STUF [TCMD\_2; addr 0x07] is enabled. The slave's bit stuffing is provided by the master STUFF source.

STUFF_SEL[1:0]	STUFF Source
00	EXT_ STUFF (see below)
01	HDSL Transmit Channel 1
10	HDSL Transmit Channel 2
11	HDSL Transmit Channel 3

**NOTE:** If SLV\_STUF is enabled and is also selected as master, then the master STUFF source automatically inserts 0 and 4 STUFF bits in alternating frames.

**EXT\_STUFF** External STUFF—Controls whether 0 or 4 STUFF bits are inserted for slave channels that select external stuffing. TSTUFF [addr 0xE4] supplies 4 STUFF bit values. The MPU must write EXT\_STUFF at each slave's transmit frame interrupt.

0 = Insert 0 STUFF bits  
1 = Insert 4 STUFF bits

**ZBIT\_SEL[1:0]** Z-bit Monitor Selection—Applicable only in E1 mode. ZBIT\_SEL selects which channel supplies the last 40 Z-bits to fill the RZBIT\_2–RZBIT\_6 registers [addr 0x18–0x1C].

ZBIT_SEL[1:0]	Monitor RZBIT[47:8] from
00, 01	HDSL receive channel 1
10	HDSL receive channel 2
11	HDSL receive channel 3

**MASTER\_SEL[1:0]** Master Channel Selection—Selects which HDSL receive channel provides the 6 ms frame sync signal to the DPLL and PCM formatter. The selected channel's 6 ms frame is used to align the PCM receive timebase and to recover the PCM receive clock.

MASTER_SEL[1:0]	Master HDSL Receive Channel
00, 01	Channel 1
10	Channel 2
11	Channel 3

**DPLL\_NCO** Operates the DPLL as an NCO —The DPLL operates in open loop configuration. Normally, the DPLL operates in closed loop to recover the PCM receive clock from the master HDSL receive channel. However, the DPLL may be operated in open loop as a Numerically Controlled Oscillator (NCO) when the master HDSL reference is unavailable (i.e., during startup procedure or loss of signal conditions). This bit is only monitored when DPLL is not in lock.

0 = Closed loop DPLL operation  
1 = Open loop DPLL operation

**0xF3—Command Register 6 (CMD\_6)**

7	6	5	4	3	2	1	0
RAZ_[1:3]			RAUX_EN	RSIG_EN	MSYNC_MEAS	BER_SEL[1:0]	

**BER\_SEL[1:0]** BER/PRBS Mode—Selects the BER meter source, the PRBS generator output direction, and serial or framed data formats. Refer to [Figure 3-12](#).

BER_SEL	Mode	Mode Description
00	Normal	PRBS outputs data under control of ROUTE_TBL. BER monitors data selected by COMBINE_TBL.
01	—	Reserved
10	PCM Framed	PRBS outputs data under control of ROUTE_TBL. BER monitors TSER during the same timeslots selected by ROUTE_TBL. TCLK and RCLK must be identical. If accompanied by loopback on HDSL side, framed PCM channels are tested.
11	PCM Serial	PRBS output replaces RSER data. BER monitors all data at TSER. TCLK and RCLK must be identical.

**MSYNC\_MEAS** MSYNC Phase Measurement—Selects whether TMSYNC or RMSYNC phase is measured with respect to MSYNC. The result is reported in MSYNC\_PHS [addr 0x39].

0 = TMSYNC to MSYNC phase measurement  
1 = RMSYNC to MSYNC phase measurement

**RSIG\_EN** Receive Signaling Table Enable—Applicable only for an LTU in a P2MP application. When active, the Receive Signaling Table [RSIG\_TBL; addr 0xF2] grooms the ABCD signaling from two or three remote sites and routes the groomed signal via DBANK\_3 in the receive Combination Table [COMBINE\_TBL; addr 0xEE]. When inactive, RSIG\_TBL is unused and the receive combination table regains use of DBANK\_3.

0 = Normal receive  
1 = Enable receive signaling table

**RAUX\_EN** Receive Auxiliary Enable—The RAUX1–RAUX3 outputs share the same pins with DROP, INSERT, and MSYNC, respectively. RAUX\_EN determines which signals are output on these shared pins.

0 = DROP, INSERT, or MSYNC outputs enabled  
1 = RAUXn outputs enabled

**RAZ\_1–RAZ\_3** Receive Auxiliary Z-bit Enable—Applicable only when RAUX\_EN is active. RAZn (n = 1,2,3) selects whether ROHn marks the output of all overhead and Z-bits or the output of only the last 40 Z-bits. If enabled, ROHn is high for one BCLKn, coincident with each of the last 40 Z-bits output on RAUXn. Otherwise, all data passed into the RFIFO is marked high by ROHn.

0 = ROHn marks all non-payload data  
1 = ROHn marks only the last 40 Z-bits

**0xF4—Command Register 7 (CMD\_7)**

7	6	5	4	3	2	1	0
PRA_EN	FEBE_POLARITY	NCO_SCALE	RCLK_INV	PHD_MODE		FAST_ACQ	DPLL_ERR_EN

**DPLL\_ERR\_EN** DPLL Error Interrupt Enable—Enables DPLL errors to request RX\_ERR interrupt when an overflow or underflow condition occurs at the phase detector output. DPLL errors are latched and reported in ERR\_STATUS [addr 0x3C] regardless of DPLL\_ERR\_EN.

0 = DPLL errors do not generate a RX\_ERR interrupt

1 = DPLL errors generate a RX\_ERR interrupt

**FAST\_ACQ** Fast Acquisition—Enables DPLL fast frequency acquisition by instructing the NCO to reuse the residual phase calculated prior to a DPLL error condition. The phase detector initializes according to PHD\_MODE (see below) while the NCO continues tracking the last known phase, thus widening the DPLL bandwidth. FAST\_ACQ is preferable while the master framer remains IN\_SYNC. To avoid RCLK frequency violations, FAST\_ACQ may be disabled when the master framer is OUT\_OF\_SYNC.

0 = Disable fast acquisition

1 = Enable fast acquisition

**NOTE:** If the system determines that the DPLL is not locked, then the MPU must assert DPLL\_RST [addr 0xF6] to force the DPLL to reload DPLL\_RSID [addr 0xD5]. The system may monitor DPLL tracking by reading RESID\_OUT [addr 0x28] or by checking DPLL\_ERR [ERR\_STATUS; addr 0x3C].

**PHD\_MODE** Phase Detector Init Mode—Selects a method to initialize the phase detector window when a DPLL error occurs. The phase detector can either initialize to the center of the phase window or opposing edge, not initialize at all, or use the programmed DPLL\_PINI [addr 0xDB] value.

PHD_MODE	Phase Detector Initialization
00	DPLL_PINI value
01	Opposing edge of phase window
10	Disabled (infinite phase window)
11	Center of phase window

**NOTE:** Disabling the phase detector is not recommended as the error output can remain saturated without reporting the DPLL error status or generating DPLL interrupts.

**RCLK\_INV** Receive Output Clock Inverted—Enables binary inversion of the clock selected by RCLK\_SEL [CMD\_2; addr 0xE6].

0 RCLK = Clock selected by RCLK\_SEL

1 RCLK = Inverted clock selected by RCLK\_SEL

**NCO\_SCALE** NCO Scale Factor—Divides the NCO clock by 4 to allow the NCO to synthesize the RCLK frequency at or below 128 kHz. GCLK and SCLK are not affected.

0 = Normal NCO operation

1 = Divide NCO clock (HFCLK) by 4

**NOTE:** Calculated values for DPLL\_RESID [addr 0xD5] and DPLL\_FACTOR [addr 0xD7] are changed according to the following equation:

$$[\text{INTEGER.FRACTION}] = \left( \frac{f_{MCLK} \times PLL\_MUL}{4 \times 2 \times f_{PCM}} \right)$$

**FEBE\_POLARITY** Determines the value of the FEBE bit that increments the FEBE counter.

0 = FEBE counter increments when FEBE bit is high

1 = FEBE counter increments when FEBE bit is low

**PRA\_EN** Enable or globally disable the PRA circuitry.

0 = Disable ALL PRA functionality

1 = Enable ALL PRA functionality

## 4.12 Interrupt and Reset

**Table 4-9. Interrupt and Reset Write Registers**

Address	Register Label	Bits	Name/Description
0xEB	IMR	8	Interrupt Mask Register
0xEC	ICR	8	Interrupt Clear Register
0xEF	BER_RST	—	Reset BER Meter/Start BER Measurement
0xF0	PRBS_RST	—	Reset PRBS Generator
0xF1	RX_RST	—	Reset Receiver

### 0xEB—Interrupt Mask Register (IMR)

The MPU writes a 1 to an IMR bit to mask the respective interrupt event. Masked interrupt sources are prevented from generating an active low signal on the INTR\* output, but are reported in the Interrupt Request Register (IRR). Writing zero to the IMR bit enables the respective interrupt event to generate an active low signal on the INTR\* output. Upon power-up or RST\* assertion, all IMR bits are automatically set to 1 to disable the INTR\* output.

7	6	5	4	3	2	1	0
RX_ERR	TX_ERR	RX[3:1]			TX[3:1]		

**TX1–TX3** Mask the HDSL 6 ms transmit frame interrupt for the respective channel.

**RX1–RX3** Mask the HDSL 6 ms receive frame interrupt for the respective channel.

**TX\_ERR** Mask the HDSL transmit error interrupt.

**RX\_ERR** Mask the HDSL receive error interrupt.

### 0xEC—Interrupt Clear Register (ICR)

The MPU writes a zero to an ICR bit to reset the respective IRR bit and, if all IRR bits are 0, causes the INTR\* output to enter a high impedance state. Writing a 1 has no effect.

7	6	5	4	3	2	1	0
RX_ERR	TX_ERR	RX[3:1]			TX[3:1]		

**TX1–TX3** Clear the HDSL 6 ms transmit frame interrupt for the respective channel.

**RX1–RX3** Clear the HDSL 6 ms receive frame interrupt for the respective channel.

**TX\_ERR** Clear the HDSL transmit error interrupt.

**RX\_ERR** Clear the HDSL receive error interrupt.

### 0xEF—Reset BER Meter/Start BER Measurement (BER\_RST)

Writing any data value to BER\_RST clears the BER Meter error count [BER\_METER; addr 0x1D]. The BER Meter Status [BER\_STATUS; addr 0x1E] instructs the BER meter to begin searching for pattern sync according to the mode selected by PRBS\_MODE [CMD\_3; addr 0xE7] and BER\_SEL [CMD\_6; addr 0xF3]. It restarts the BER meter test measurement interval defined by BER\_SCALE [CMD\_3; addr 0xE7]. The MPU must configure PRBS\_MODE, BER\_SEL and BER\_SCALE before issuing a BER\_RST command.

After writing BER\_RST, the MPU monitors SYNC\_DONE to determine when the test pattern qualification period has ended and then checks BER\_SYNC [BER\_STATUS; addr 0x1E] to verify that correct test pattern has been received. The BER meter uses a 128-bit qualification period to examine receive data before updating BER\_SYNC, therefore the MPU may wait up to 2 ms before SYNC\_DONE is activated. If BER\_SYNC is not found when the qualification period ends, then the test has failed to detect pattern sync and the MPU should ignore the BER\_METER results. The MPU may optionally repeat BER\_RST in the event of a PRBS test failure since the BER meter may have initialized LFSR with received bit errors. Similarly, the MPU should repeat BER\_RST if BER\_METER reports any bit errors at the end of the qualification period during a PRBS test.

Once BER\_SYNC is detected, the MPU monitors BER\_DONE to determine the end of the test measurement interval. BER\_METER results are updated in real time during the measurement interval and latched at the end of the interval. The MPU can restart the test measurement interval and thereby extend the measurement indefinitely by applying another BER\_RST command before BER\_DONE is activated.

### 0xF0—Reset PRBS Generator (PRBS\_RST)

Writing any data value to PRBS\_RST loads an 8-bit pattern from the FILL\_PATT Register [addr 0xEA] into the least significant byte of the PRBS generator's 23-stage LFSR and clears all other LFSR bits. The MPU writes PRBS\_RST prior to the start of a PRBS or fixed pattern test.

**NOTE:** Before issuing PRBS\_RST to start a PRBS test, the MPU must initialize the FILL\_PATT value to something other than 0x00, or the LFSR output will be stuck at all 0s.

### 0xF1—Reset Receiver (RX\_RST)

For the RS8953B, writing any data value to RX\_RST forces the PCM formatter to align the PCM receive timebase with respect to the master HDSL channel's receive 6 ms frame by reloading the RFIFO\_WL value [addr 0xCD]. The MPU must write RX\_RST after modifying the RFIFO\_WL value in the RS8953B. The RS8953B automatically performs RX\_RST each time the master HDSL channel's receive framer changes alignment and transitions to the IN\_SYNC state.

Issuing RX\_RST while the PCM formatter is aligned causes no change in alignment of the PCM receive timebase.



## 4.13 Receive/Transmit Status

	HDSL Channel 1 (CH1)	HDSL Channel 2 (CH2)	HDSL Channel 3 (CH3)
Base Address	0x00	0x08	0x10

**Table 4-10. Receive and Transmit Status Read Registers**

CH1	CH2	CH3	Register Label	Bits	Register Description
0x00	0x08	0x10	REOC_LO	8	Receive EOC Bits
0x01	0x09	0x11	REOC_HI	8	Receive EOC Bits
0x02	0x0A	0x12	RIND_LO	8	Receive IND Bits
0x03	0x0B	0x13	RIND_HI	8	Receive IND Bits
0x04	0x0C	0x14	RZBIT_1	8	Receive Z-bits
0x18			RZBIT_2	8	Common Receive Z-bits (CHn = ZBIT_SEL)
0x19			RZBIT_3	8	Common Receive Z-bits (CHn = ZBIT_SEL)
0x1A			RZBIT_4	8	Common Receive Z-bits (CHn = ZBIT_SEL)
0x1B			RZBIT_5	8	Common Receive Z-bits (CHn = ZBIT_SEL)
0x1C			RZBIT_6	8	Common Receive Z-bits (CHn = ZBIT_SEL)
0x05	0x0D	0x15	STATUS_1	8	Receive Status
0x06	0x0E	0x16	STATUS_2	8	Receive Status
0x07	0x0F	0x17	STATUS_3	8	Transmit Status
0x21	0x29	0x31	CRC_CNT	8	CRC Error Count
0x22	0x2A	0x32	FEBC_CNT	8	Far-End Block Error Count

The MPU may read all receive and transmit status registers non-destructively at any time. All status registers are updated coincident with their respective HDSL channel's receive or transmit 6 ms frame interrupts indicated in the Interrupt Request Register [IRR; addr 0x1F]. Therefore, the MPU may poll the IRR or enable interrupts to determine whether a status update has occurred. Real-time receive status (REOC, RIND, and RZBIT) register updates are suspended when the respective HDSL channel's receive framer reports an OUT\_OF\_SYNC state [STATUS\_1; addr 0x05].

### 0x00—Receive Embedded Operations Channel (REOC\_LO)

7	6	5	4	3	2	1	0
REOC[7:0]							

**0x01—Receive Embedded Operations Channel (REOC\_HI)**

7	6	5	4	3	2	1	0
MFG[2:0]			REOC[12:8]				

**REOC[12:0]** Receive EOC—Holds 13 EOC bits received during the previous HDSL frame. Refer to [Table 3-2](#) (Overhead Bit Allocation) for EOC bit positions within the frame. The LSB REOC[0] is received first.

**MFG[2:0]** Manufacture Code—Contains the device manufacture ID code.

CH1 (address 0x01)	001
CH2 (address 0x09)	010
CH3 (address 0x11)	100

**0x02—Receive Indicator Bits (RIND\_LO)**

7	6	5	4	3	2	1	0
RIND[7:0]							

**0x03—Receive Indicator Bits (RIND\_HI)**

7	6	5	4	3	2	1	0
MINOR_VER[2:0]			RIND[12:8]				

**RIND[12:0]** Receive IND—Holds 13 IND bits received during the previous HDSL frame. Refer to [Table 3-2](#) (Overhead Bit Allocation) for the IND bit positions within the frame. The receive framer updates the RIND registers on receive frame interrupt boundaries. The LSB RIND[0] is received first.

**MINOR\_VER[2:0]** Minor Version Number—Contains the device minor revision level which the MPU can read to determine the installed device, enabled new software features, and remove unnecessary software corrections from older versions.

	Rev A	Rev B	Rev C
CH1 (address 0x03)	000	000	000
CH2 (address 0x0B)	010	010	010
CH3 (address 0x13)	000	001	010

**0x04—Receive Z-Bits (RZBIT\_1)**

7	6	5	4	3	2	1	0
RZBIT[7:0]							

**0x18—Receive Z-Bits (RZBIT\_2)**

7	6	5	4	3	2	1	0
RZBIT[15:8]							

**0x19—Receive Z-Bits (RZBIT\_3)**

7	6	5	4	3	2	1	0
RZBIT[23:16]							

**0x1A—Receive Z-Bits (RZBIT\_4)**

7	6	5	4	3	2	1	0
RZBIT[31:24]							

**0x1B—Receive Z-Bits (RZBIT\_5)**

7	6	5	4	3	2	1	0
RZBIT[39:32]							

**0x1C—Receive Z-Bits (RZBIT\_6)**

7	6	5	4	3	2	1	0
RZBIT[47:40]							

**RZBIT[47:0]** Receive Z-bits—Applicable only in E1\_MODE [CMD\_1; addr 0xE5]. RZBIT holds 48 Z-bits received during the previous HDSL frame. Refer to [Figures 3-21](#) and [3-26](#) for Z-bit positions within the frame. The LSB RZBIT[0] is received first. The first 8 received Z-bits from each HDSL channel are individually monitored in the RZBIT\_1 registers. The last 40 received Z-bits are monitored in the RZBIT\_2–RZBIT\_6 registers from only the single receive channel selected by ZBIT\_SEL [CMD\_5; addr 0xE9]. Systems which need individual channel monitoring of the last 40 Z-bits can use external circuitry to capture the Z-bits from the receive HDSL auxiliary channel (RAUXn) outputs.

**0x05—Receive Status 1 (STATUS\_1)**

7	6	5	4	3	2	1	0
MAJOR_VER[1:0]		RFIFO_SLIP	RFIFO_MPTY	RFIFO_FULL	RX_STUFF	TR_INVERT	SYNC_AB

## 4.13 Receive/Transmit Status

HDSL Channel Unit

<b>SYNC_AB</b>	<p>SYNC_WORD_A or SYNC_WORD_B Acquired—Reports which one of the two programmed SYNC words is detected by the receive framer. Updated each time the receive framer state transitions from OUT_OF_SYNC to SYNC_ACQUIRED.</p> <p>0 = SYNC_ACQUIRED with SYNC_WORD_A 1 = SYNC_ACQUIRED with SYNC_WORD_B</p>
<b>TR_INVERT</b>	<p>Tip/Ring Inversion—Indicates the receive framer acquired an inverted SYNC word A or B, indicating that the receive tip and ring wire pair connections are reversed. The RS8953B automatically inverts the sign bits of all received data as it is presented on the RDATn input when inversion is detected. TR_INVERT is updated each time the receive framer state transitions from OUT_OF_SYNC to SYNC_ACQUIRED.</p> <p>0 = SYNC_ACQUIRED with expected SYNC word 1 = SYNC_ACQUIRED with inverted SYNC word</p>
<b>RX_STUFF</b>	<p>Receive STUFF—Indicates whether the receive framer detected 4 STUFF bits or no STUFF bits in the previous frame.</p> <p>0 = No STUFF bits detected 1 = 4 STUFF bits detected</p>
<b>RFIFO_FULL</b>	<p>Receive FIFO Full Error—Indicates the RFIFO has overflowed. Also reported in ERR_STATUS and IRR (if RX_ERR_EN in RCMD_2[addr0x61]), and generates an RX_ERR interrupt (if RX_ERR in IMR is enabled). RFIFO_FULL is indicative of clock problems and may be triggered by DPLL acquisition, DPLL switchover, or by changes to the receive combination table, or to the receive payload map.</p> <p>0 = RFIFO normal 1 = RFIFO overflowed</p>
<b>RFIFO_MPTY</b>	<p>Receive FIFO Empty Error—Indicates the RFIFO has Underrun. Also reported in ERR_STATUS and IRR (if RX_ERR_EN in RCMD_2[addr0x61]), and generates an RX_ERR interrupt (if RX_ERR in IMR is enabled). RFIFO_MPTY is indicative of clock problems and may be triggered by events similar to those which cause RFIFO_FULL errors.</p> <p>0 = RFIFO normal 1 = RFIFO Underrun</p>
<b>RFIFO_SLIP</b>	<p>Receive FIFO Slip—Indicates the number of payload bytes mapped into the RFIFO is not equal to the number of PCM timeslots mapped out of the RFIFO over a 6 ms period, also reported in ERR_STATUS and IRR (if RX_ERR_EN in RCMD_2[addr0x61]). It generates an RX_ERR interrupt (if RX_ERR in IMR is enabled). RFIFO_SLIP errors are caused by a receive framer OUT_OF_SYNC condition, or by improper configuration of the receive payload map, or the receive combination table.</p> <p>0 = RFIFO normal 1 = RFIFO unbalanced</p>
<b>MAJOR_VER[1:0]</b>	<p>Major Version Number—Contains the device major revision level which the CPU can read to determine the installed device, enable new software features, and remove unnecessary software corrections from older versions.</p>

	Bt8953	Bt8953A	RS8953B
CH1 (address 0x05)	01	01	01
CH2 (address 0x0D)	01	01	01
CH3 (address 0x15)	01	10	11

**0x06—Receive Status 2 (STATUS\_2)**

7	6	5	4	3	2	1	0
FEBE_OVR	CRC_OVR	CRC_ERR	SYNC_STATE[1:0]		STATE_CNT[2:0]		

**STATE\_CNT[2:0]** Intermediate State Count—Applicable only if SYNC\_STATE reports SYNC\_ACQUIRED or SYNC\_ERRORED states. STATE\_CNT indicates the framer's progress through the intermediate states.

000	1 Frame
001	2 Consecutive Frames
010	3 Consecutive Frames
011	4 Consecutive Frames
100	5 Consecutive Frames
101	6 Consecutive Frames
110	7 Consecutive Frames
111	8 Consecutive Frames

**SYNC\_STATE[1:0]** Receive Framer Synchronization State—Reports the state of the receive framer. Refer to the Framer Synchronization State Diagram ([Figure 3-23](#)).

00	OUT_OF_SYNC
01	SYNC_ACQUIRED
10	IN_SYNC
11	SYNC_ERRORED

When the framer enters OUT\_OF\_SYNC, the RFIFO is automatically reset, FEBE and CRC error counts are suspended, and RX\_ERR is activated.

When the framer reports SYNC\_ACQUIRED, the RFIFO and the payload mapper are enabled, and RX\_ERR is activated.

When the framer enters IN\_SYNC, the RFIFO Water Level [RFIFO\_WL; addr 0xCD] is re-established, FEBE and CRC counting resumes, and RX\_ERR is activated.

When the framer reports SYNC\_ERRORED, STATE\_CNT indicates the number of consecutive frames in which SYNC was not detected.

**CRC\_ERR** CRC Error—Shows that the CRC comparison in the previous frame resulted in a mismatch of 1 or more CRC bits. CRC\_ERR is invalid in the OUT\_OF\_SYNC state. The MPU may copy CRC\_ERR into the first transmit IND [TIND\_LO; addr 0x02] to report FEBE.

0	CRC pass
1	CRC error detected

**CRC\_OVR** CRC Error Count Overflow—Indicates the CRC Error Count [CRC\_CNT; addr 0x21] has reached its maximum value of 255, and generates an RX\_ERR interrupt.

0	CRC error count below maximum
1	CRC error count equals maximum 255 (0xFF)

**FEBE\_OVR** Far-End Block Error Count Overflow—Indicates the FEBE count [FEBE\_CNT; addr 0x22] has reached its maximum value of 255. Generates an RX\_ERR interrupt.

0	FEBE count below maximum
1	FEBE count equals maximum 255 (0xFF)

**0x07—Transmit Status (STATUS\_3)**

7	6	5	4	3	2	1	0
—	—	—	STUFF_ERR	TFIFO_SLIP	TFIFO_MPTY	TFIFO_FULL	TX_STUFF

<b>TX_STUFF</b>	Transmit STUFF Decision—Indicates whether the last transmitted HDSL frame was output with 4 STUFF bits or none.  0 = No STUFF bits output 1 = 4 STUFF bits output
<b>TFIFO_FULL</b>	Transmit FIFO Full Error—Indicates the TFIFO has overflowed. This is also reported in ERR_STATUS and IRR (if TX_ERR_EN in TCMC_1[addr0x06]) and generates a TX_ERR interrupt (if TX_ERR in IMR is enabled). TFIFO_FULL errors may result from a change of transmit PCM frame alignment when MPU writes to TFIFO_RST, or from any changes in TCLK or BCLKn frequency or to the transmit routing table or the transmit payload map.  0 = TFIFO normal 1 = TFIFO overflowed
<b>TFIFO_MPTY</b>	Transmit FIFO Empty Error—Indicates the TFIFO has Underrun. This is also reported in ERR_STATUS and IRR (if TX_ERR_EN in TCMC_1[addr0x06]), and generates a TX_ERR interrupt (if TX_ERR in IMR is enabled). TFIFO_MPTY errors may be triggered by events similar to those which cause TFIFO_FULL errors.  0 = TFIFO normal 1 = TFIFO Underrun
<b>TFIFO_SLIP</b>	Transmit FIFO Slip—Indicates that the number of PCM timeslots routed into the TFIFO is not equal to the number of payload bytes mapped out of the TFIFO over a 6 ms period. This is also reported in ERR_STATUS and IRR (if TX_ERR_EN in TCMC_1[addr0x06]), and generates a TX_ERR interrupt (if TX_ERR in IMR is enabled). TFIFO_SLIP errors may be triggered by events similar to those which cause TFIFO_FULL errors. Repeated TFIFO_SLIP errors may indicate improper configuration of either the transmit payload map or the transmit routing table.  0 = Transmit FIFO normal 1 = Transmit FIFO unbalanced
<b>STUFF_ERR</b>	Transmit Stuffing Error—Indicates when the phase difference measured from PCM to HDSL 6 ms frames exceeds the maximum bit Stuffing Threshold [STF_THRESH_C; addr 0xD3]. This is also reported in ERR_STATUS and IRR (if TX_ERR_EN) and generates a TX_ERR interrupt (if TX_ERR in IMR is enabled). STUFF_ERR may be triggered by events similar to those which cause TFIFO_FULL errors. The STUFF generator is automatically reset when STUFF_ERR is detected.  0 = STUFF generator normal 1 = STUFF generator error

**0x21—CRC Error Count (CRC\_CNT)**

7	6	5	4	3	2	1	0
CRC_CNT[7:0]							

**CRC\_CNT[7:0]** CRC Error Count—Indicates the total number of received CRC errors detected by the receive framer, and increments by one for each received HDSL 6 ms frame that contains CRC\_ERR [STATUS\_1; addr 0x06]. CRC\_CNT is cleared to 0 by ERR\_RST [addr 0x67]. Error counting is suspended while the receive framer is OUT\_OF\_SYNC or SYNC\_ACQUIRED. CRC\_CNT also sets CRC\_OVR [STATUS\_2; addr 0x06] upon reaching its maximum count value of 255.

0x22—Far End Block Error Count (FEBE\_CNT)

7	6	5	4	3	2	1	0
FEBE_CNT[7:0]							

**FEBE\_CNT[7:0]** Far-End Block Error Count—Indicates the total number of received FEBE errors sent by the far-end transmitter and increments by 1 for each received HDSL 6 ms frame that contains an active FEBE bit. The polarity of active FEBE is determined by FEBE\_POLARITY in CMD\_7 (addr 0xF4). FEBE is the second IND bit received within the indicator bit group and can be monitored separately as the RIND[1] bit in the RIND\_LO [addr 0x02] receive status register.

Refer to [Table 3-2](#) for the FEBE bit position within the frame. FEBE\_CNT is reset to 0 by ERR\_RST [addr 0x67]. Error counting is suspended while the receive framer is OUT\_OF\_SYNC or SYNC\_ACQUIRED. FEBE\_CNT also sets FEBE\_OVR [STATUS\_2; addr 0x06] upon reaching its maximum count value of 255.

## 4.14 Common Status

**Table 4-11. Common Status Read Registers**

Address	Register Label	Bits	Name/Description
0x1D	BER_METER	8	Bit Error Rate Meter
0x1E	BER_STATUS	3	BER Meter Status
0x1F	IRR	8	Interrupt Request Register
0x20	RESID_OUT_HI	8	DPLL Residual
0x28	RESID_OUT_LO	8	DPLL Residual
0x30	IMR	8	Interrupt Mask Register
0x38	PHS_ERR	8	DPLL Phase Error
0x39	MSYNC_PHS_LO	8	Multiframe Sync Phase
0x3A	MSYNC_PHS_HI	5	Multiframe Sync Phase
0x3B	SHADOW_WR	8	Shadow Write
0x3C	ERR_STATUS	7	Error Status

### 0x1D—Bit Error Rate Meter (BER\_METER)

The receive BER meter and the transmit PRBS generator work in conjunction to perform characterization, installation, maintenance, and diagnostic testing on PCM and HDSL channels. PRBS\_MODE and PRBS\_DIS [CMD\_3; addr 0xE7] determine which of the four PRBS patterns or constant pattern is checked by the BER meter.

7	6	5	4	3	2	1	0
BER[7:0]							

**BER[7:0]** Bit Error Ratio—Contains the total number of logical bit errors counted in real time during the test measurement interval defined by BER\_SCALE [CMD\_3; addr 0xE7]. BER stops counting when the test measurement interval is completed or the counter reaches its maximum value of 255, after which the BER\_METER result is latched until the meter is reset [BER\_RST; addr 0xEF].

**NOTE:** BER does not suspend error counting when the HDSL receive framer loses frame alignment. Any time after test completion [see BER\_DONE in BER\_STATUS; addr 0x1E], the MPU can calculate an exact Bit Error Ratio, as follows:

BER_SCALE	Bit Error Ratio
00	$\text{BER}[7:0] \div 2^{31}$
01	$\text{BER}[7:0] \div 2^{28}$
10	$\text{BER}[7:0] \div 2^{25}$
11	$\text{BER}[7:0] \div 2^{21}$



**0x1E—BER Status (BER\_STATUS)**

7	6	5	4	3	2	1	0
—	—	—	—	—	SYNC_DONE	BER_DONE	BER_SYNC

**BER\_SYNC** BER Pattern SYNC—Applicable only if SYNC\_DONE is active. BER\_SYNC reports whether the BER meter acquired test pattern sync during the 128-bit test pattern qualification period. The BER meter must detect fewer than 8 bit errors during examination of the first 128 bits to assert BER\_SYNC. BER\_SYNC can report a false pattern sync if all 1s or all 0s are applied.

0 = No pattern sync  
1 = Pattern sync detected

**BER\_DONE** BER Measurement Complete—Signifies the BER meter has completed examination of the total number of test pattern bits programmed by BER\_SCALE. When BER\_DONE is set, the BER meter stops counting bit errors.

0 = BER measurement in progress  
1 = BER measurement complete

**SYNC\_DONE** Sync Qualification Period Complete—Indicates the BER meter has examined 128 bits and has updated BER\_SYNC. SYNC\_DONE reports the end of the test pattern qualification period.

0 = Qualification period in progress  
1 = Qualification period complete

**0x1F—Interrupt Request Register (IRR)**

The INTR\* output pin is activated and the corresponding IRR bit latched whenever an interrupt event transition is detected from one of eight sources. Interrupt sources that are masked [see IMR; addr 0xEB] do not activate the INTR\* output but are latched and reported in the IRR. Latched IRR bits are reset and the INTR\* output is deactivated by writing a 0 to the corresponding Interrupt Clear Register bits [ICR; addr 0xEC]. However, if IRR is reporting a persistent error condition such as framer OUT\_OF\_SYNC, then writing ICR deactivates the INTR\* pin, but does not clear the IRR bit unless the error condition has ended. INTR\* output activation is triggered by an event edge; therefore, persistent or multiple error conditions only generate one INTR\* request.

7	6	5	4	3	2	1	0
RX_ERR	TX_ERR	RX[3:1]			TX[3:1]		

**TX1-TX3** Transmit HDSL 6 ms Frame Interrupt—Reported coincident with the start of the transmit 6 ms frame for the respective HDSL channel. This allows the MPU to synchronize read access of the transmit status [STATUS\_3; addr 0x07] and to synchronize write access to the real time transmit HDSL registers (see [Table 4-2](#)).

0 = No interrupt  
1 = Transmit frame interrupt

**RX1-RX3** Receive HDSL 6 ms Frame Interrupt—Reported coincident with the start of the receive 6 ms frame for the respective HDSL channel. This allows the MPU to synchronize read access of the real time receive status (see [Table 4-10](#)) and the DPLL status of the master HDSL receive channel (see [Table 4-11](#)).

0 = No interrupt  
1 = Receive frame interrupt

**TX\_ERR** Transmit Error Interrupt—The transmit stuffing and TFIFO errors from all enabled error sources are logically ORed to form TX\_ERR. When active, the MPU reads the Error Status Register [ERR\_STATUS; addr 0x3C] to determine which source caused the interrupt.

0 = No interrupt  
1 = Transmit error interrupt

**RX\_ERR** Receive Error Interrupt—Framer state transitions, RFIFO errors, CRC and FEBE counter overflows, and DPLL errors from all enabled error sources are logically ORed to form RX\_ERR. When active, the MPU reads the Error Status Register [ERR\_STATUS; addr 0x3C] to determine which source caused the interrupt.

0 = No interrupt  
1 = Receive error interrupt

### 0x28—DPLL Residual Output (RESID\_OUT\_LO)

7	6	5	4	3	2	1	0
RESID_OUT[7:0]							

### 0x20—DPLL Residual Output (RESID\_OUT\_HI)

7	6	5	4	3	2	1	0
RESID_OUT[15:8]							

**RESID\_OUT[15:0]** DPLL Residual Output—The NCO's residual phase output equals the synthesized phase needed to construct half-cycle of the recovered clock, given as a fractional result, in units of HFCLK. During DPLL closed loop operation, the RESID\_OUT value should converge to approximately equal the programmed DPLL\_RESID [addr 0xD6] value. The MPU can calculate the recovered clock frequency by substituting the measured value of RESID\_OUT in the synthesis equation, and solving for RCLK. RESID\_OUT is updated coincident with the RX<sub>n</sub> interrupt (where  $n$  = master HDSL channel number) and is provided for diagnostics only.

### 0x30—Interrupt Mask Register (IMR)

This register contains data written to IMR [addr 0xEB] and is provided as an MPU read back register. The MPU interrupt service routine can use the IMR read value to mask read data from the IRR and to avoid processing of masked interrupts.

**0x38—DPLL Phase Error (PHS\_ERR)**

7	6	5	4	3	2	1	0
PHS_ERR[7:0]							

**PHS\_ERR[7:0]** DPLL Phase Error—The DPLL phase detector error output is given in 2's complement format in units of GCLK cycles, where minimum (negative) phase is reported as 0x80 and maximum (positive) phase as 0x7F. The result of the PCM to HDSL 6 ms phase comparison is updated coincident with the RXn interrupt (where n = master HDSL channel number). During DPLL closed loop operation, the phase error's long term average equals zero. PHS\_ERR is provided for diagnostic testing only.

**0x39—Multiframe Sync Phase Low (MSYNC\_PHS\_LO)**

7	6	5	4	3	2	1	0
MSYNC_PHS[7:0]							

**0x3A—Multiframe Sync Phase High (MSYNC\_PHS\_HI)**

7	6	5	4	3	2	1	0
—	—	—	MSYNC_PHS[12:8]				

**MSYNC\_PHS[12:0]** Multiframe Sync Phase—Contains the number of elapsed TCLK cycles measured from the rising edge of the TMSYNC or the RMSYNC signal selected by MSYNC\_MEAS [CMD\_6; addr 0xF3] to the rising edge of MSYNC. A value of zero indicates the phase equals 1 TCLK cycle. Maximum phase equals 1 PCM multiframe. For example, Nx64 multiframe equals 16 frames times [N = 64 the timeslots per frame, times 8 bits per timeslot, for a total length equal to 8,192 PCM bits (0x1FFF)].

For unframed or asynchronously mapped applications, knowing the TMSYNC to MSYNC phase simplifies far-end reconstruction of RMSYNC. Therefore, each terminal measures TMSYNC phase, and sends it to the far-end for calculation of the RFRAME\_LOC [addr 0xC3] and the RMF\_LOC [addr 0xC5] delays needed to recreate RMSYNC. TMSYNC phase measurement is unnecessary when PCM and HDSL frames are synchronized or the far-end does not need to create RMSYNC.

$$\text{RMF\_LOC.RFRAME\_LOC} = \frac{t(\text{TMP})}{\text{FRAME\_LEN}}$$

where: *FRAME\_LEN* = Bits per frame [FRAME\_LEN; address 0xC8]

*RMF\_LOC* = Frame delay (integer part of result)

*RFRAME\_LOC* = Bit delay (fractional part of result)

*t(TMP)* = TMSYNC to MSYNC Phase (in PCM bits)

The NTU in a P2MP application uses both measurements to monitor the phase difference between incoming and outgoing HDSL frames, to adjust its output frame location accordingly to align with other remote sites, and to communicate the resulting transmit frame offset to the LTU for grooming purposes. Refer to the Receive Signaling Location Register [RSIG\_LOC; addr 0x68].

### 0x3B—Shadow Write (SHADOW\_WR)

7	6	5	4	3	2	1	0
WR[7:0]							

**WR[7:0]** Most Recent Write Data—Contains the data latched during the last MPU write cycle to any location within the RS8953B address space. System diagnostics can read-verify the data written to validate MPU access over the address/data bus.

### 0x3C—Error Status (ERR\_STATUS)

ERR\_STATUS is a read-clear register in RS8953B. Reading ERR\_STATUS forces its contents to 0. Transmit and receive HDSL channel errors and DPLL errors are reported individually in ERR\_STATUS, where they are indefinitely latched until cleared. The MPU reads ERR\_STATUS to determine the cause of a TX\_ERR or RX\_ERR interrupt. Each source has independent Interrupt Error Enables (TX\_ERR\_EN, RX\_ERR\_EN and DPLL\_ERR\_EN), which prevent it from setting the corresponding IRR interrupt. See error interrupt enables in TCMD\_1 [addr 0x06], RCMD\_2 [addr 0x61], and CMD\_7 [addr 0xF4].

7	6	5	4	3	2	1	0
—	DPLL_ERR	RX3_ERR	RX2_ERR	RX1_ERR	TX3_ERR	TX2_ERR	TX1_ERR

**TX1\_ERR-TX3\_ERR** Transmit Channel Error—Reported coincident with the TX\_ERR interrupt to indicate which of the three HDSL transmit channels caused the TX\_ERR. The MPU reads the respective channel's transmit status [STATUS\_3; addr 0x07] to determine the specific error.

0 = No error

1 = Transmit error

**RX1\_ERR-RX3\_ERR** Receive Channel Error—Reported coincident with the RX\_ERR interrupt to indicate which of the three HDSL receive channels caused the RX\_ERR. The MPU reads the respective channel's receive status [STATUS\_1-STATUS\_2; addr 0x05-0x06] to determine the specific error.

0 = No error

1 = Receive error

**DPLL\_ERR** DPLL Phase Detector Error—Reported coincident with the RX\_ERR interrupt to indicate when the DPLL phase detector output reached the maximum or minimum phase error limit.

0 = No error

1 = DPLL error

## 4.15 PRA Transmit Read

**Table 4-12. PRA Transmit Read Registers**

Address	Register Label	Bits	Name/Description
0x40	TX_PRA_CTRL0	8	PRA Transmit Control Register 0
0x41	TX_PRA_CTRL1	7	PRA Transmit Control Register 1
0x42	TX_PRA_MON1	8	PRA Transmit Monitor Register 1
0x43	TX_PRA_E_CNT	8	PRA Transmit E-Bits Register 0
0x45	TX_PRA_CODE	6	PRA Transmit In-Band Code
0x46	TX_PRA_MON0	6	PRA Transmit Monitor Register 0
0x47	TX_PRA_MON2	4	PRA Transmit Monitor Register 2

### 0x40—PRA Transmit Control Register 0 (TX\_PRA\_CTRL0)

7	6	5	4	3	2	1	0
E_MODE[1:0]		SA8_MODE	SA7_MODE	SA6_MODE[1:0]		SA5_MODE	SA4_MODE

**SA4\_MODE** Controls the behavior of Sa4 bits transmitted towards the HDSL link, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA5\_MODE** Controls the behavior of Sa5 bits transmitted towards the HDSL link, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**SA6\_MODE** Controls the behavior of Sa6 bits transmitted towards the HDSL link, as follows:

Code	Sa6 Bits
00	Transparent
01	From bits buffer 0
10	Automatic
11	Illegal

The Automatic mode operates based on the result of the receiver (PCM to HDSL) CRC check and E-bits, as follows:

Received E-bits	Receive CRC Check	Sa6
0 (Error)	Error	0011
0 (Error)	No Error	0001
1 (No Error)	Error	0010
1 (No Error)	No Error	From bits buffer 0 (sec0)

**NOTE:** MSB of Sa6 is transmitted first (i.e., in frames 1 and 9).

**SA7\_MODE** Controls the behavior of Sa7 bits, transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 1

**SA8\_MODE** Controls the behavior of Sa8 bits transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 1

**E\_MODE** Controls the behavior of the E-bits transmitted towards the HDSL link, as follows:

Code	E-bits
00	Transparent
01	From bits buffer 0
10	Automatic
11	Illegal

The Automatic mode operates in conjunction with the receiver CRC4 check result (reported also in RX\_PRA\_MON0), as follows:

Receiver CRC Check	E-bits Forced to
Error	0
OK	1

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x41—PRA Transmit Control Register 1 (TX\_PRA\_CTRL1)

7	6	5	4	3	2	1	0
—	RESET_E_CNT	AIS	A_MODE	CRC4_MODE[1:0]		SYNCHR_EN	PRA_EN

**PRA\_EN** Used to enable or globally disable the receive PRA circuitry, as follows:

0 = Disable ALL RX PRA functionality  
1 = Enable ALL RX PRA functionality

**SYNCHR\_EN** Used to enable or disable the PCM multiframe synchronization state machine, as follows:

0 = Enable—Use TMSYNC input pin as a qualifier of the frame, and force the synchronization state machine to HUNT mode  
1 = Disable—Use TMSYNC input as a qualifier of multiframe

**CRC4\_MODE** CRC4\_MODE controls the behavior of the CRC bits transmitted towards the HDSL link, as follows:

Code	CRC4 Bits
00	Transparent
01	All 1
10	Re-calculated
11	Illegal

**A\_MODE** Controls the behavior of A-bits transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 0

**AIS** Enables to override all 32 slots of an PCM frame except Slot 0, transmitted towards the HDSL link, with a constant pattern:

0 = Disable (Normal)

1 = 0xFF

AIS must be activated with reset\_e\_cnt = 1

**NOTE:** AIS enables to achieve framed AIS. To achieve unframed arbitrary AUX pattern generation, use the existing feature of the channel unit.

**RST\_E\_CNT** Clears the TX\_E counter, as follows:

0 = Counter enabled

1 = Clear the E-transmit counter

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x42—PRA Transmit Monitor Register 1 (TX\_PRA\_MON1)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	Sa <sub>5</sub>	A	E2	E1

This register is updated once every PCM multiframe. The bits in this register correspond to the bits in the transmitted PCM multiframe stream, in the PCM to HDSL direction.

**Sa6\_1, \_2, \_3, \_4** Sa6\_1, \_2, \_3, \_4 is updated only if the same Sa6 pattern is detected in the second submultiframe and synchr\_en = 0.

**Sa5** Sa5 is only updated if all 8 corresponding bits of the multiframe were detected as identical.

**A** A-bit is only updated if all 8 corresponding bits of the multiframe were detected as identical.

**E1** E1 is the E-bit detected in Frame 13.

**E2** E2 is the E-bit detected in Frame 15.

## 0x43—PRA Transmit E-Bits Counter (TX\_PRA\_E\_CNT)

7	6	5	4	3	2	1	0
TX_PRA_E_CNT[7:0]							

The register is update twice in an PCM multiframe. It increments each time one of the E-bits is detected active 0.

The counter wraps around at 255. It is cleared or enabled by RESET\_E\_CNT of TX\_PRA\_CTRL1 register.

## 0x45—PRA Transmit In-Band Code (TX\_PRA\_CODE)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	—	—	Sa <sub>5</sub>	A

This register is updated once every PCM multiframe. The bits in this register correspond to the bits in the transmitted PCM multiframe stream, in the PCM to HDSL direction.

**Sa6 \_1, \_2, \_3, \_4** Sa6 \_1, \_2, \_3, \_4 is updated only if detected identical in the last 8 submultiframes, given that the respective field was not masked in TX\_BITS\_BUF1.

**Sa5** Sa5 is only updated only if detected identical in the last 8 submultiframes, given that the respective field was not masked in TX\_BITS\_BUF1.

**A** A-bit is only updated only if detected identical in the last 8 submultiframes, given that the respective field was not masked in TX\_BITS\_BUF1.

### 0x46—PRA Transmit Monitor Register 0 (TX\_PRA\_MON0)

7	6	5	4	3	2	1	0
SYNCH_STATE	Sa <sub>8</sub>	Sa <sub>7</sub>	Sa <sub>4</sub>	—	—	CRC error2	CRC error1

**CRC error 1** Represents the CRC check result in submultiframe 1.

**CRC error 2** Represents the CRC check result in submultiframe 2.

**Sa<sub>4</sub>, Sa<sub>7</sub>, and Sa<sub>8</sub>** Updated with a value that represents the majority over the eight off-frames.

**SYNCH\_STATE** Represents the status of the multiframe synchronization machine.

0 = Not synchronized

1 = Synchronized

If SYNCH\_STATE is 1, the relative frame with which synchronization was achieved in TX\_PRA\_MON2 is readable.

### 0x47—PRA Transmit Monitor Register 2 (TX\_PRA\_MON2)

7	6	5	4	3	2	1	0
—	—	—	—	TX_PRA_MON2[3:0]			

The 4 bits of this register represent the original number of the relative frame with which synchronization was achieved. This is relevant only if bit SYNCH\_STATE of TX\_PRA\_MON0 reads 1.



## 4.16 PRA Transmit Write

**Table 4-13. PRA Transmit Write Registers**

Address	Register Label	Bits	Name/Description
0x70	TX_PRA_CTRL0	8	PRA Transmit Control Register 0
0x71	TX_PRA_CTRL1	7	PRA Transmit Control Register 1
0x72	TX_BITS_BUFF1	6	PRA Transmit Bits Buffer 1
0x73	TX_PRA_TMSYNC_OFFSET	8	PRA Transmit TMSYNC Offset Register
0x74	TX_BITS_BUFF0	8	PRA Transmit Bits Buffer 0

### 0x70—PRA Transmit Control Register 0 (TX\_PRA\_CTRL0)

7	6	5	4	3	2	1	0
E_MODE[1:0]		SA8_MODE	SA7_MODE	SA6_MODE[1:0]		SA5_MODE	SA4_MODE

**SA4\_MODE** Controls the behavior of Sa4 bits transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 1

**SA5\_MODE** Controls the behavior of Sa5 bits transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 0

**SA6\_MODE** Controls the behavior of Sa6 bits transmitted towards the HDSL link, as follows:

Code	Sa6 Bits
00	Transparent
01	From bits buffer 0
10	Automatic
11	Illegal

The Automatic mode operates based on the result of the receiver (HDSL to PCM) CRC check and E-bits, as follows:

Received E-bits	Receive CRC Checks	Sa6
0 (Error)	Error	0011
0 (Error)	No Error	0001
1 (No Error)	Error	0010
1 (No Error)	No Error	From bits buffer 0 (sec 0)

**NOTE:** MSB of Sa6 is transmitted first (i.e., in frames 1 and 9).

**SA7\_MODE** Controls the behavior of Sa7 bits transmitted towards the HDSL link, as follows:

0 = Transparent  
1 = From bits buffer 1

**SA8\_MODE** Controls the behavior of Sa8 bits, transmitted towards the HDSL link, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**E\_MODE** Controls the behavior of the E-bits transmitted towards the HDSL link, as follows:

Code	E-bits
00	Transparent
01	From bits buffer 0
10	Automatic
11	Illegal

The Automatic mode operates in conjunction with the receiver CRC4 check result (reported also in RX\_PRA\_MON0), as follows:

Receiver CRC Check	E-bits Forced to
Error	0
OK	1

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x71—PRA Transmit Control Register 1 (TX\_PRA\_CTRL1)

7	6	5	4	3	2	1	0
—	RESET_E_CNT	AIS	A_MODE	CRC4_MODE[1:0]		SYNCHR_EN	PRA_EN

**PRA\_EN** Enable or globally disable the receive PRA circuitry, as follows:

- 0 = Disable ALL RX PRA functionality
- 1 = Enable ALL RX PRA functionality

**SYNCHR\_EN** Enable or disable the PCM multiframe synchronization state machine, as follows:

- 0 = Enable. Use TMSYNC input pin as a qualifier of the frame and force synchronization state machine to HUNT mode.
- 1 = Disable. Use TMSYNC input as a qualifier of multiframe.

**CRC4\_MODE** Controls the behavior of the CRC bits, transmitted towards the HDSL link, as follows:

Code	CRC4 Bits
00	Transparent
01	All '1's
10	Re-calculated
11	Illegal

**A\_MODE** Controls the behavior of A-bits, transmitted towards the HDSL link, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**AIS** Enables to override all 32 slots of an PCM frame except slot 0 transmitted towards the HDSL link, with a constant pattern:

0 = Disable (Normal)

1 = 0xFF

**NOTE:** AIS enables to achieve framed AIS. To achieve unframed arbitrary AUX pattern generation, use the existing feature of the channel unit.

**RST\_E\_CNT** Clears the TX\_E counter

0 = Counter enabled

1 = Clear the E transmit counter

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x72—PRA Transmit Bits Buffer 1 (TX\_BITS\_BUFF1)

7	6	5	4	3	2	1	0
—	SA6_MASK	SA5_MASK	A_MASK	—	Sa <sub>8</sub>	Sa <sub>7</sub>	Sa <sub>4</sub>

The value of this register is only relevant if the corresponding MODE bit of TX\_PRA\_CTRL0 is set. A new written value takes effect starting with the next PCM multiframe, following the register write access cycle completion.

An in-band code is reported as detected when the pattern in the Sa<sub>6</sub>, Sa<sub>5</sub>, AND A fields remain constant for 8 consecutive multiframes.

**Sa<sub>4</sub>** The new value to be inserted into the Sa<sub>4</sub> location of the data stream, in the PCM to HDSL direction.

**Sa<sub>7</sub>** The new value to be inserted into the Sa<sub>7</sub> location of the data stream, in the PCM to HDSL direction.

**Sa<sub>8</sub>** The new value to be inserted into the Sa<sub>8</sub> location of the data stream, in the PCM to HDSL direction.

**A\_MASK** Determines whether the pattern in the A-bit field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.

**SA5\_MASK** Determines whether the pattern in the SA5 field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.

**SA6\_MASK** Determines whether the pattern in the SA6 field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.

## 0x73—PRA Transmit TMSYNC offset Register (TX\_PRA\_TMSYNC\_OFFSET)

7	6	5	4	3	2	1	0
TX_PRA_TMSYNC_OFFSET[7:0]							

The value of this register is used to enable the accommodation of the RS8953B to any TMSYNC signal shape. When programmed to 0x00, the PRA circuitry assumes that the positive edge of the TMSYNC input signal coincides with the first bit of an PCM framer.

When this assumption is not valid, this register may be used to internally reposition the TMSYNC to coincide with Bit 0.

### 0x74—PRA Transmit Bits Buffer 0 (TX\_BITS\_BUFF0)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	Sa <sub>5</sub>	A	E2	E1

The value of this register is only relevant if the corresponding MODE bit of TX\_PRA\_CTRL0 is set. A new written value takes effect starting with the next PCM multiframe following the register write access cycle completion. Each bit of this register is used in the odd frames of the PCM multiframe.

- E1**                    The new value to be inserted into the E1 location of the data stream, in the PCM to HDSL direction. E1 is used in Frame 13.
- E2**                    The new value to be inserted into the E2 location of the data stream, in the PCM to HDSL direction. E2 is used in Frame 15.
- A**                     The new value to be inserted into the A-bit location of the data stream, in the PCM to HDSL direction. A-bit is used in all odd frames.
- Sa<sub>5</sub>**                  The new value to be inserted into the Sa<sub>5</sub> location of the data stream, in the PCM to HDSL direction. Sa<sub>5</sub> is used in all odd frames.
- Sa<sub>6\_1, \_2, \_3, \_4</sub>**   The new value to be inserted into the Sa<sub>6\_1, \_2, \_3, \_4</sub> location of the data stream, in the PCM to HDSL direction. Sa<sub>6\_1</sub> is used in Frames 1 and 9. Sa<sub>6\_2</sub> is used in Frames 3 and 11. Sa<sub>6\_3</sub> is used in Frames 5 and 13. Sa<sub>6\_4</sub> is used in Frames 7 and 15.

## 4.17 PRA Receive Read

**Table 4-14. PRA Receive Read Registers**

Address	Register Label	Bits	Name/Description
0x80	RX_PRA_CTRL0	7	PRA Receive Read Register 0
0x81	RX_PRA_CTRL1	8	PRA Receive Control Register 1
0x82	RX_BITS_BUFF1	8	PRA Receive Bits Buffer 1
0x83	RX_PRA_E_CNT	8	PRA Receive E Bit Counter
0x84	RX_PRA_CRC_CNT	8	PRA Receive CRC4 Error Counter
0x85	RX_PRA_CODE	6	PRA Receive In-Band Code
0x86	RX_PRA_MON0	6	PRA Receive Monitor Register 0
0x87	RX_PRA_MON2	4	PRA Receive Monitor Register 2

### 0x80—PRA Receive Control Register 0 (RX\_PRA\_CTRL0)

7	6	5	4	3	2	1	0
E_MODE[1:0]		SA8_MODE	SA7_MODE	—	SA6_MODE	SA5_MODE	SA4_MODE

**SA4\_MODE** Controls the behavior of Sa4 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA5\_MODE** Controls the behavior of Sa5 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA6\_MODE** Controls the behavior of Sa6 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**SA7\_MODE** Controls the behavior of Sa7 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA8\_MODE** Controls the behavior of Sa8 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**E\_MODE** Controls the behavior of the E-bits transmitted towards the HDSL link, as follows:

Code	E-bits
00	Transparent
01	From Bits Buffer 0
10	Automatic
11	Illegal

The Automatic mode works in conjunction with the transmitter CRC4 check result (reported also in TX\_PRA\_MON0), as follows:

Receiver CRC Check	E-bits Forced to:
Error	0
OK	1

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x81—PRA Receive Control Register 1 (RX\_PRA\_CTRL1)

7	6	5	4	3	2	1	0
RESET_CRC_CNT	RESET_E_CNT	AIS	A_MODE	CRC4_MODE[1:0]		SYNCHR_EN	PRA_EN

**PRA\_EN** Used to enable or globally disable the transmit PRA circuitry, as follows:

- 0 = Disable ALL TX PRA functionality
- 1 = Enable ALL TX PRA functionality

**SYNCHR\_EN** Used to enable or disable the PCM multiframe synchronization state machine, as follows:

- 0 = Enable synchronization and force HUNT mode. Take RMSYNC as indicator of frame.
- 1 = Disable synchronization. Take RMSYNC as multiframe indicator.

**CRC4\_MODE** Controls the behavior of the CRC bits, transmitted towards the PCM link, as follows:

Code	E-bits
00	Transparent
01	All 1
10	Re-calculated
11	Illegal

**A\_MODE** Controls the behavior of A-bits, transmitted towards the PCM link, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**AIS** Enables to override all 32 slots of an PCM frame except slot 0, transmitted towards the PCM link with a constant pattern:

- 0 = Disable (Normal)
- 1 = 0xFF

**NOTE:** AIS enables to achieve framed AIS. To achieve unframed arbitrary AUX pattern generation, use the existing feature of the channel unit.

**RST\_E\_CNT** Clears the RX\_E counter, as follows:

- 0 = Counter enabled
- 1 = Clear the E-receive counter

**RST\_CRC\_CNT** Clears the RX\_CRC counter, as follows:

- 0 = Counter enabled
- 1 = Clear the E-receive counter

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

## 0x82—PRA Receive Monitor Register 1 (RX\_PRA\_MON1)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	Sa <sub>5</sub>	A	E2	E1

The register is updated once every PCM multiframe. Sa<sub>5</sub> and A-bit are updated with a value that represents the majority of identical corresponding bits (5 or more).

**E1** The value monitored from the E1 location of the data stream, in the HDSL to PCM direction. E1 is the bit detected in Frame 13.

**E2** The value monitored from the E2 location of the data stream, in the HDSL to PCM direction. E2 is the bit detected in Frame 15.

**A** The value monitored from the A-bit location of the data stream, in the HDSL to PCM direction.

**Sa<sub>5</sub>** The value monitored from the Sa<sub>5</sub> location of the data stream, in the HDSL to PCM direction.

**Sa<sub>6\_1, \_2, \_3, \_4</sub>** The value monitored from the Sa<sub>6\_1, \_2, \_3, \_4</sub> location of the data stream, in the HDSL to PCM direction. Sa<sub>6\_1, \_2, \_3, \_4</sub> is updated only if the same Sa<sub>6</sub> pattern is detected in the second submultiframe and synchr\_en = 0.

## 0x83—PRA Receive E bits Counter (RX\_PRA\_E\_CNT)

7	6	5	4	3	2	1	0
RX_PRA_E_CNT[7:0]							

The register is updated twice in a PCM multiframe. It increments each time one of the E-bits is detected active 0.

The counter wraps around at 255. It is cleared/enabled by RESET\_E\_CNT of RX\_PRA\_CTRL1 register.

## 0x84—PRA Receive CRC4 Errors Counter (RX\_PRA\_CRC\_CNT)

7	6	5	4	3	2	1	0
RX_PRA_CRC_CNT[7:0]							

The register is updated twice each PCM multiframe. It increments each time a mismatch between the reported and calculated CRC4 is detected.

The counter wraps around at 255. It is cleared/enabled by RESET\_CRC\_CNT of RX\_PRA\_CTRL1 register.

### 0x85—PRA Receive In-Band Code (RX\_PRA\_CODE)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	—	—	Sa <sub>5</sub>	A

This register is updated with a value, only if it was detected identical in the last 8 submultiframes, given the respective field was not masked in RX\_BITS\_BUF1.

**A** The value from the A-bit location of the data stream, in the HDSL to PCM direction.

**Sa<sub>5</sub>** The value from the Sa<sub>5</sub> location of the data stream, in the HDSL to PCM direction.

**Sa<sub>6\_1, \_2, \_3, \_4</sub>** The value from the Sa<sub>6\_1, \_2, \_3, \_4</sub> location of the data stream, in the HDSL to PCM direction.

### 0x86—PRA Receive Monitor Register 0 (RX\_PRA\_MON0)

7	6	5	4	3	2	1	0
SYNCH_STATE	Sa <sub>8</sub>	Sa <sub>7</sub>	Sa <sub>4</sub>	—	—	CRC error2	CRC error1

**CRC error1** Represents the CRC check result in submultiframe 1.

**CRC error2** Represents the CRC check result in submultiframe 2.

**Sa<sub>4</sub>, Sa<sub>7</sub>, and Sa<sub>8</sub>** Updated with the value that represents the majority of identical respective bits (5 or more).

**SYNCH\_STATE** Represents the status of the multiframe synchronization machine, as follows:

0 = Not synchronized

1 = Synchronized

If SYNCH\_STATE reads 1, the offset frame with which synchronization was achieved in RX\_MON2 is readable.

### 0x87—PRA Receive Monitor Register 2 (RX\_PRA\_MON2)

7	6	5	4	3	2	1	0
—	—	—	—	RX_PRA_MON2[3:0]			

The 4 bits of this register represent the original number of the relative frame with which synchronization was achieved. This is relevant only if bit SYNCH\_STATE of RX\_PRA\_MON0 reads 1.



## 4.18 PRA Receive Write

**Table 4-15. PRA Receive Write Registers**

Address	Register Label	Bits	Name/Description
0xB0	RX_PRA_CTRL0	7	PRA Receive Read Register 0
0xB1	RX_PRA_CTRL1	8	PRA Receive Control Register 1
0xB2	RX_BITS_BUFF1	6	PRA Receive Bits Buffer 1
0xB4	RX_PRA_BUFF0	8	PRA Receive Bit Counter

### 0xB0—PRA Receive Control Register 0 (RX\_PRA\_CTRL0)

7	6	5	4	3	2	1	0
E_MODE[1:0]		SA8_MODE	SA7_MODE	—	SA6_MODE	SA5_MODE	SA4_MODE

**SA4\_MODE** Controls the behavior of Sa4 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA5\_MODE** Controls the behavior of Sa5 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**SA6\_MODE** Controls the behavior of Sa6 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**SA7\_MODE** Controls the behavior of Sa7 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**SA8\_MODE** Controls the behavior of Sa8 bits transmitted towards PCM, as follows:

- 0 = Transparent
- 1 = From bits buffer 1

**E\_MODE** Controls the behavior of the E-bits transmitted towards the HDSL link, as follows:

Code	E-bits
00	Transparent
01	From bits buffer 0
10	Automatic
11	Illegal

The Automatic mode operates in conjunction with the transmitter CRC4 check result (reported also in TX\_PRA\_MON0), as follows:

Receiver CRC Check	E-bits Forced to:
Error	0
OK	1

**NOTE:** The value of this register takes effect starting with the next PCM multiframe following the write access cycle completion.

The unused bit (bit 3) should be set to a value of 0.

## 0xB1—PRA Receive Control Register 1 (RX\_PRA\_CTRL1)

7	6	5	4	3	2	1	0
RESET_CRC_CNT	RESET_E_CNT	AIS	A_MODE	CRC4_MODE[1:0]		SYNCHR_EN	PRA_EN

**PRA\_EN** Used to enable or globally disable the transmit PRA circuitry, as follows:

- 0 = Disable ALL TX PRA functionality
- 1 = Enable ALL TX PRA functionality

**SYNCHR\_EN** Used to enable or disable the PCM multiframe synchronization state machine, as follows:

- 0 = Enable synchronization and force HUNT mode. Take RMSYNC as indicator of frame.
- 1 = Disable synchronization. Take RMSYNC as multiframe indicator.

**CRC4\_MODE** Controls the behavior of the CRC bits transmitted towards the PCM link, as follows:

Code	CRC4 Bits
00	Transparent
01	All 1
10	Re-calculated
11	Illegal

**A\_MODE** Controls the behavior of A-bits transmitted towards the PCM link, as follows:

- 0 = Transparent
- 1 = From bits buffer 0

**AIS** Enables to override all 32 slots of an PCM frame except slot 0, transmitted towards the PCM link with a constant pattern:

- 0 = Disable (Normal)
  - 1 = 0xFF
- AIS must be activated with reset\_e\_cnt = 1.

**NOTE:** AIS enables to achieve framed AIS. To achieve unframed arbitrary AUX pattern generation, use the existing feature of the channel unit.

**RST\_E\_CNT** Clears the RX\_E counter, as follows:

- 0 = Counter enabled
- 1 = Clear the E-receive counter

**RST\_CRC\_CNT** Clears the RX\_CRC counter, as follows:

- 0 = Counter enabled
- 1 = Clear the E-receive counter

**NOTE:** The value of this register takes effect starting with the next PCM multiframe, following the write access cycle completion.

## 0xB2—PRA Receive Bits Buffer 1 (RX\_BITS\_BUFF1)

7	6	5	4	3	2	1	0
—	Sa <sub>6</sub> _MASK	Sa <sub>6</sub> _MASK	Sa <sub>6</sub> _MASK	—	Sa <sub>8</sub>	Sa <sub>7</sub>	Sa <sub>4</sub>

The value of this register is only relevant if its corresponding MODE bit of RX\_PRA\_CTRL0 is set. A new written value takes effect starting with the next PCM multiframe following the register write access cycle completion.

An in-band code is reported as detected when the pattern in the Sa<sub>6</sub>, Sa<sub>5</sub>, AND A fields remain constant for 8 consecutive multiframes.

<b>Sa<sub>4</sub></b>	The new value to be inserted into the Sa <sub>4</sub> location of the data stream, in the HDSL to PCM direction.
<b>Sa<sub>7</sub></b>	The new value to be inserted into the Sa <sub>7</sub> location of the data stream, in the HDSL to PCM direction.
<b>Sa<sub>8</sub></b>	The new value to be inserted into the Sa <sub>8</sub> location of the data stream, in the HDSL to PCM direction.
<b>A_MASK</b>	Determines if the pattern in the A-bit field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.
<b>SA<sub>5</sub>_MASK</b>	Determines if the pattern in the SA <sub>5</sub> field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.
<b>SA<sub>6</sub>_MASK</b>	Determines if the pattern in the SA <sub>6</sub> field must remain constant for 8 consecutive multiframes for an in-band code to be reported as detected.

## 0xB4—PRA Receive Bits Buffer 0 (RX\_BITS\_BUFF0)

7	6	5	4	3	2	1	0
Sa <sub>6_4</sub>	Sa <sub>6_3</sub>	Sa <sub>6_2</sub>	Sa <sub>6_1</sub>	Sa <sub>5</sub>	A	E2	E1

The value of this register is only relevant if the corresponding MODE bit of RX\_PRA\_CTRL0 is set. A new written value takes effect starting with the next PCM multiframe, following the register write access cycle completion. Each bit of this register is used in the odd frames of the PCM multiframe.

<b>E1</b>	The new value to be inserted into the E1 location of the data stream, in the HDSL to PCM direction. E1 is used in Frame 13.
<b>E2</b>	The new value to be inserted into the E2 location of the data stream, in the HDSL to PCM direction. E2 is used in Frame 15.
<b>A</b>	The new value to be inserted into the A-bit location of the data stream, in the HDSL to PCM direction. A-bit is used in all odd frames.

---

**4.18 PRA Receive Write***HDSL Channel Unit*

- Sa<sub>5</sub>** The new value to be inserted into the Sa<sub>5</sub> location of the data stream, in the HDSL to PCM direction. Sa<sub>5</sub> is used in all odd frames.
- Sa<sub>6</sub>\_1, \_2, \_3, \_4** The new value to be inserted into the Sa<sub>6</sub>\_1, \_2, \_3, \_4 location of the data stream, in the HDSL to PCM direction. Sa<sub>6</sub>\_1 is used in Frames 1 and 9. Sa<sub>6</sub>\_2 is used in Frames 3 and 11. Sa<sub>6</sub>\_3 is used in Frames 5 and 13. Sa<sub>6</sub>\_4 is used in Frames 7 and 15.

## 5.0 Applications

---

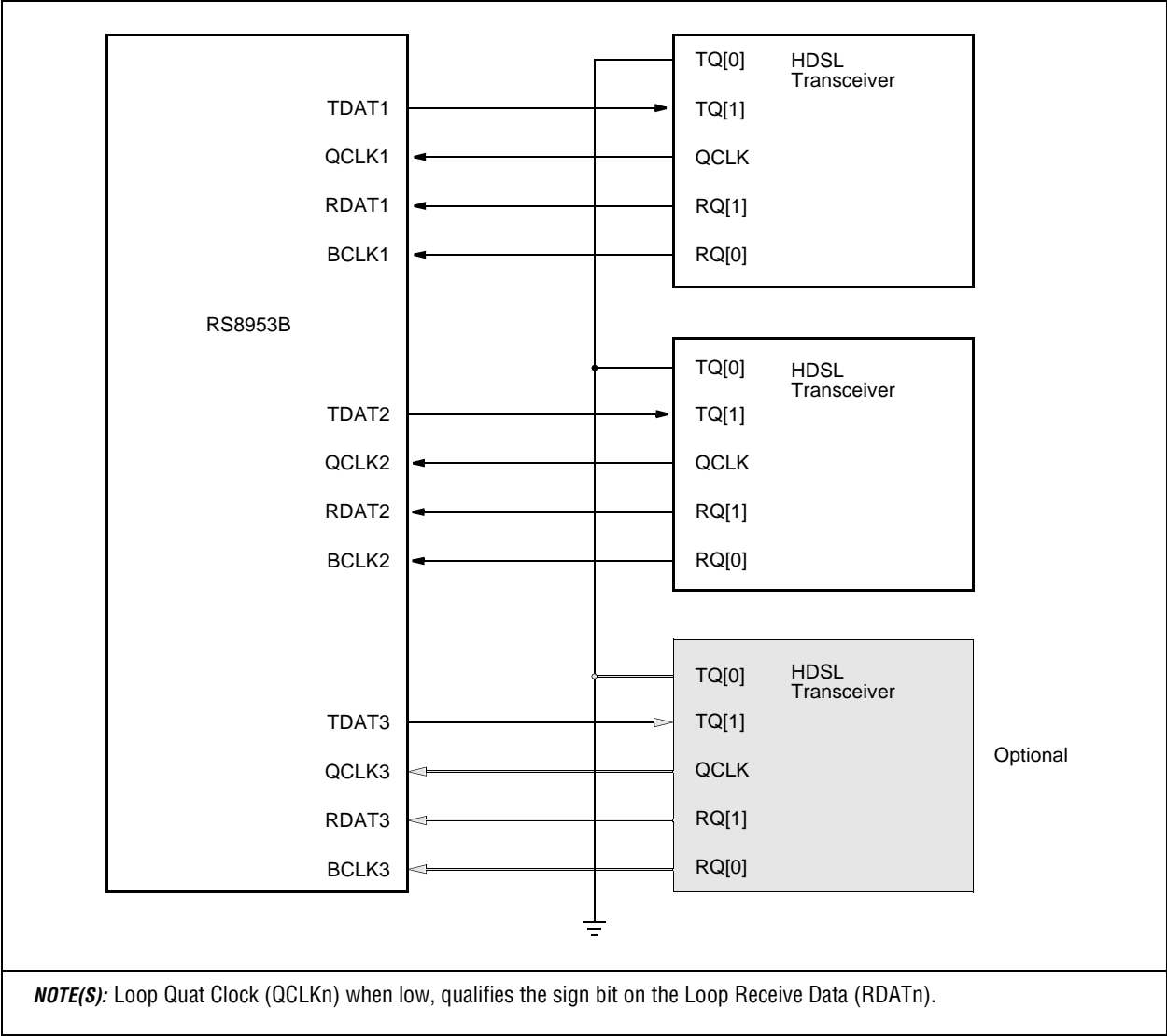
The following chapter shows typical interconnections of the RS8953B HDSL channel unit:

- Conexant HDSL Transceiver
- Bt8370 E1/T1 Primary Rate Framer
- Motorola 68302 16-bit Processor
- Intel 8051 8-bit Processor

# 5.1 Interfacing to a Conexant HDSL Transceiver

Figure 5-1 illustrates a typical interconnection between the RS8953B HDSL channel unit and a Conexant HDSL transceiver.

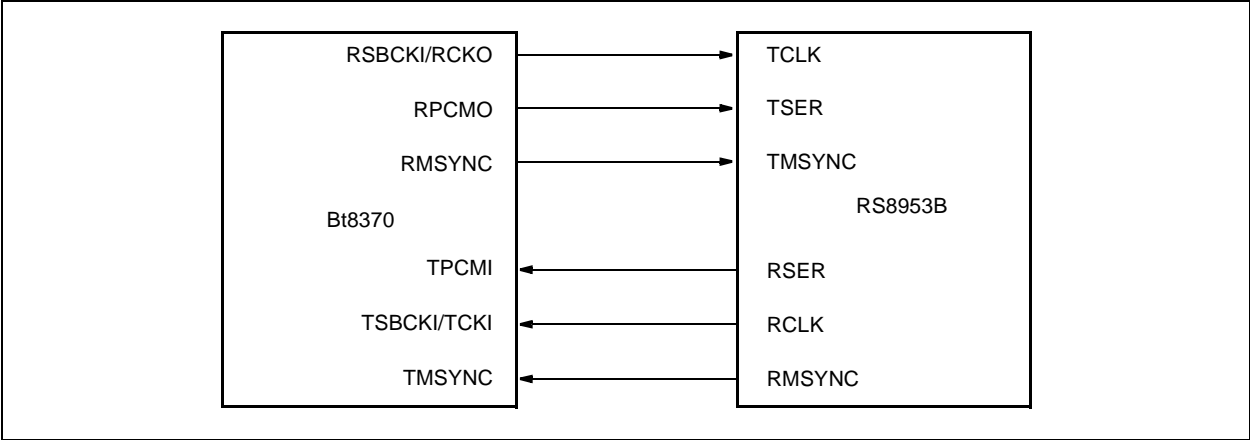
Figure 5-1. RS8953B HDSL Channel Unit to Conexant HDSL Transceiver Interconnection



## 5.2 Interfacing to the Bt8370 E1/T1 Framer

Figure 5-2 illustrates a typical interconnection between the RS8953B HDSL channel unit and the Bt8370 E1/T1 framer.

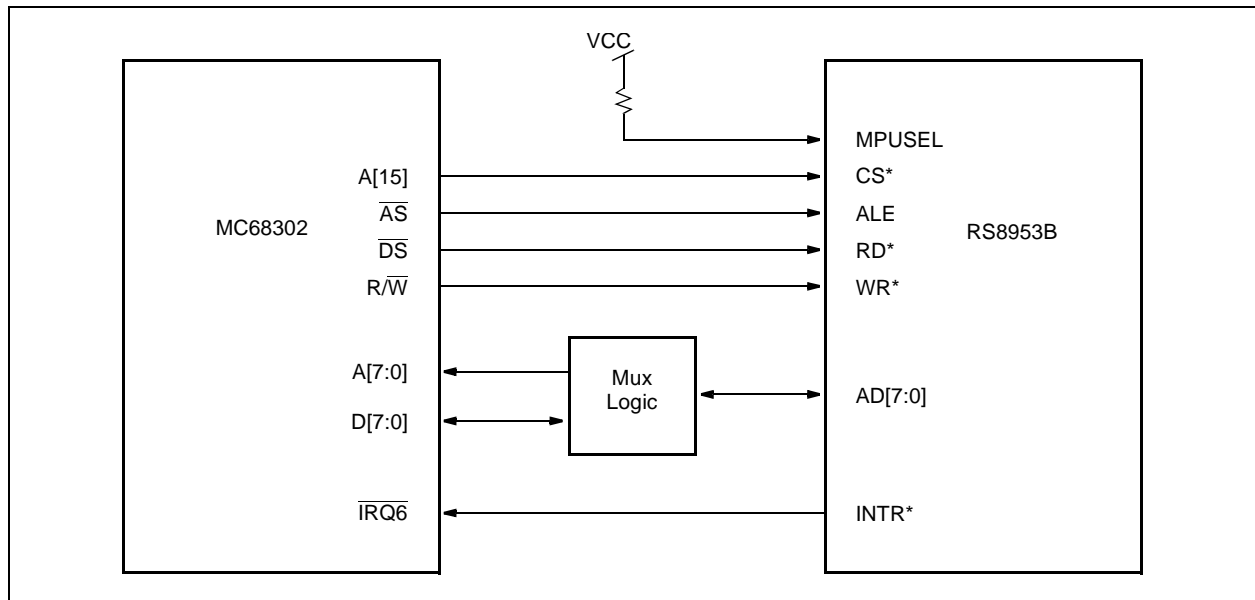
Figure 5-2. RS8953B HDSL Channel Unit to Bt8360 DS1 Framer Interconnection



## 5.3 Interfacing to the 68302 Processor

Figure 5-3 illustrates a typical interconnection between the RS8953B HDSL channel unit and the 68302 processor.

**Figure 5-3. RS8953B to 68302 Processor Interconnection**

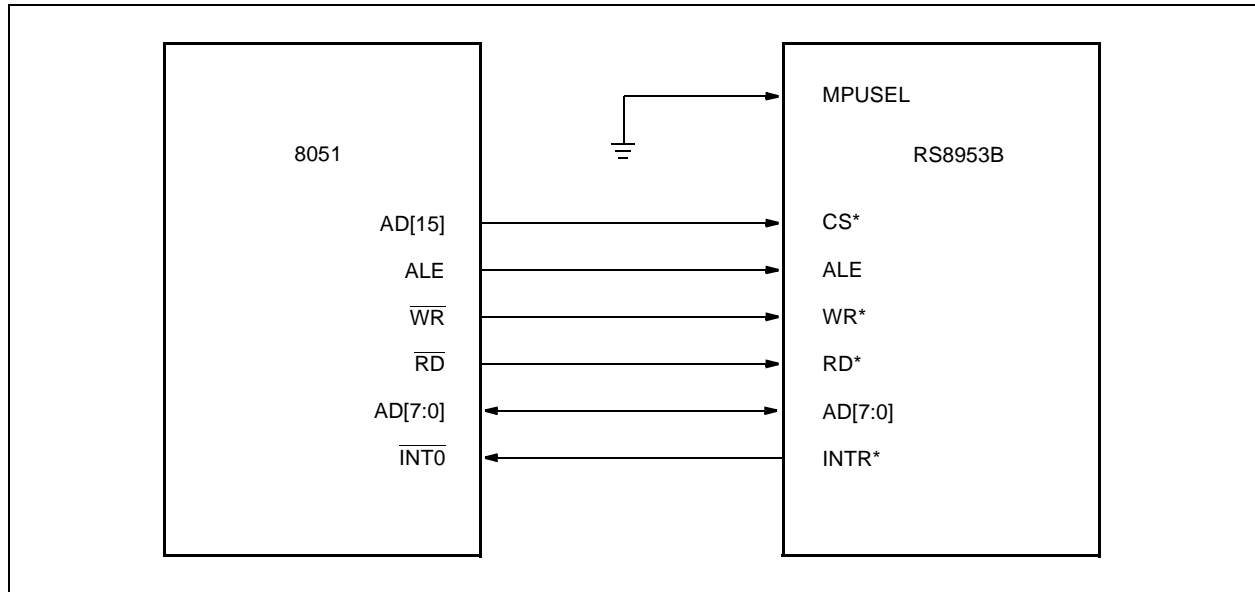




## 5.4 Interfacing to the 8051 Controller

Figure 5-4 illustrates a typical interconnection between the RS8953B HDSL channel unit and the 8051 controller.

**Figure 5-4. RS8953B HDSL Channel Unit to 8051 Controller Interconnection**



## **5.5 References**

Applicable specifications:

- Bellcore TA-NWT-001210
- Bellcore FA-NWT-001211
- ETSI RTR/TM-03036
- CCITT Recommendation G.704
- Bellcore TR-NWT-000499

## 6.0 Electrical and Timing Specifications

---

### 6.1 Absolute Maximum Ratings

**Table 6-1. Absolute Maximum Ratings**

Symbol	Parameter	Minimum	Maximum	Units
VCC	Supply Voltage	−0.3	7	V
V <sub>I</sub>	Voltage on Any Signal Pin	−1.0	5.3	V
T <sub>ST</sub>	Storage Temperature	−40	125	°C
T <sub>VSOL</sub>	Vapor Phase Soldering Temperature (1 minute)		220	°C
θ <sub>JA</sub>	Thermal Resistance (68 PLCC), Still Air		39.8	°C/W

**NOTE(S):** Stresses greater than those listed in this table may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions beyond those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

#### 6.1.1 Recommended Operating Conditions

**Table 6-2. Recommended Operating Conditions**

Symbol	Parameter	Minimum	Maximum	Units
VCC	Supply Voltage	3.0	3.6	V
T <sub>AMB</sub>	Ambient Operating Temperature	−40	85	°C
V <sub>IH</sub>	High-Level Input Voltage	2.0	VEXT +.25	V
V <sub>IL</sub>	Low-Level Input Voltage	−0.3	0.8	V

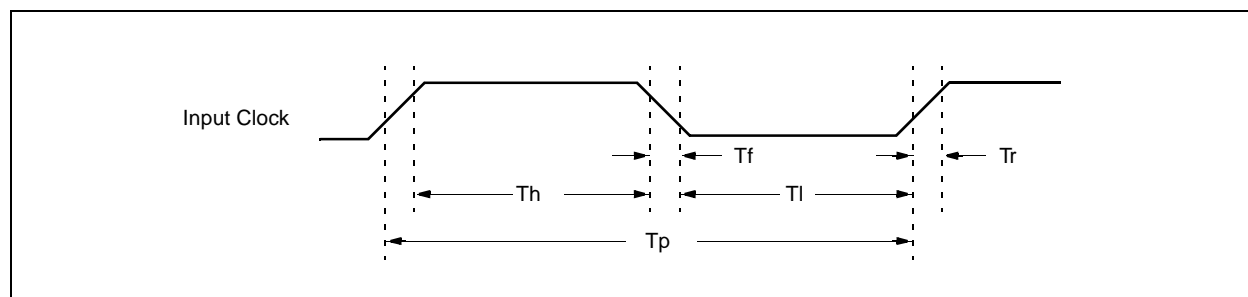
## 6.1.2 Electrical Characteristics

**Table 6-3. Electrical Characteristics**

Symbol	Parameter	Minimum	Maximum	Units
$I_{CC}$	Supply Current PCM = 4.096 Mbps HDSL = 2.320 Mbps		21.1	mA
	Supply Current 1T1		11.6	mA
	Supply Current 2T1		9.8	mA
	Supply Current 1E1		13.8	mA
	Supply Current 2E1		10.9	mA
	Supply Current 3E1		10.7	mA
$V_{OH}$	High-Level Output Voltage	2.4		V
$I_{OH}$	High-Level Output Current Source	200		$\mu$ A
$V_{OL}$	Low-Level Output Voltage		0.4	V
$I_{OL}$	Low-Level Output Current Sink	2	4	mA
$I_{OD}$	Open Drain Output Current Sink		4	mA
$I_{PR}$	Resistive Pullup Current	40	500	$\mu$ A
$I_I$	Input Leakage Current	-10	10	$\mu$ A
$I_{OZ}$	Three-State Leakage Current	-10	10	$\mu$ A
$C_{IN}$	Input Capacitance		2.5	pF
$C_{LD}$	Output Capacitive Loading		70	pF
$C_Z$	High-Impedance Output Capacitance		85	pF
$I_{OSC}$	Short Circuit Output Current	37	160	mA

### 6.1.3 Timing Requirements

**Figure 6-1. Input Clock Timing**



**Table 6-4. Clock Timing Requirements**

Symbol	Parameter	Minimum	Maximum	Units
1/ $T_p$	Mclk Frequency (PII_dis = 0; PII_mul = 16)	3.75	5.0	MHz
	Mclk Frequency (PII_dis = 0; PII_mul = 8)	7.5	10	MHz
	Mclk Frequency (PII_dis = 1)	60	80	MHz
	Tclk, Exclk Frequency	0.128	4.096	MHz
	Bclk Frequency	0.144	2.320	MHz
	Tck Frequency	0	25	MHz
$T_h$	Clock Width High	$0.4 \times T_p$	$0.6 \times T_p$	ns
$T_l$	Clock Width Low	$0.4 \times T_p$	$0.6 \times T_p$	ns
$T_r$	Clock Rise Time		20	ns
$T_f$	Clock Fall Time		20	ns

**Figure 6-2. Input Setup and Hold Timing**

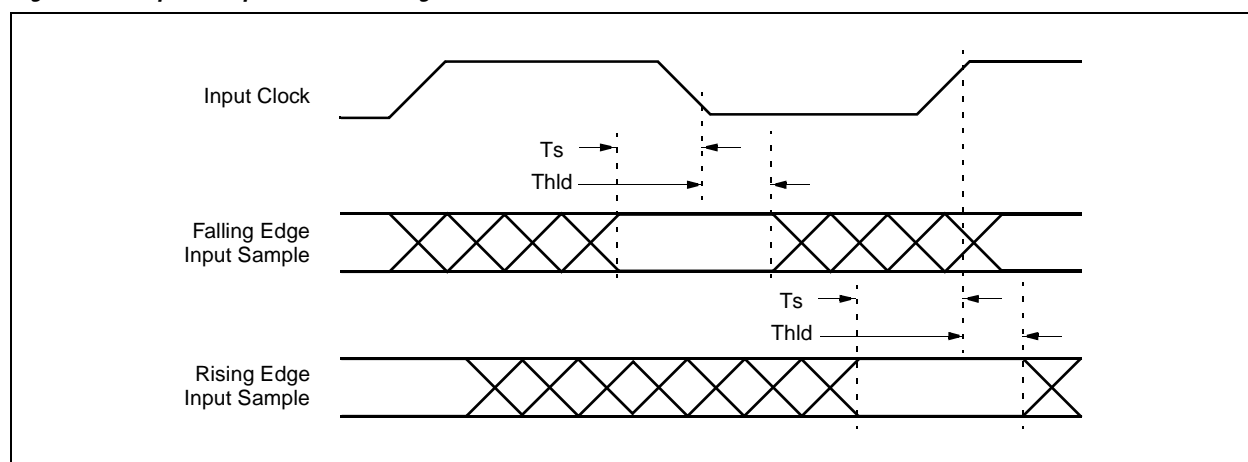


Table 6-5. Data Timing Requirements

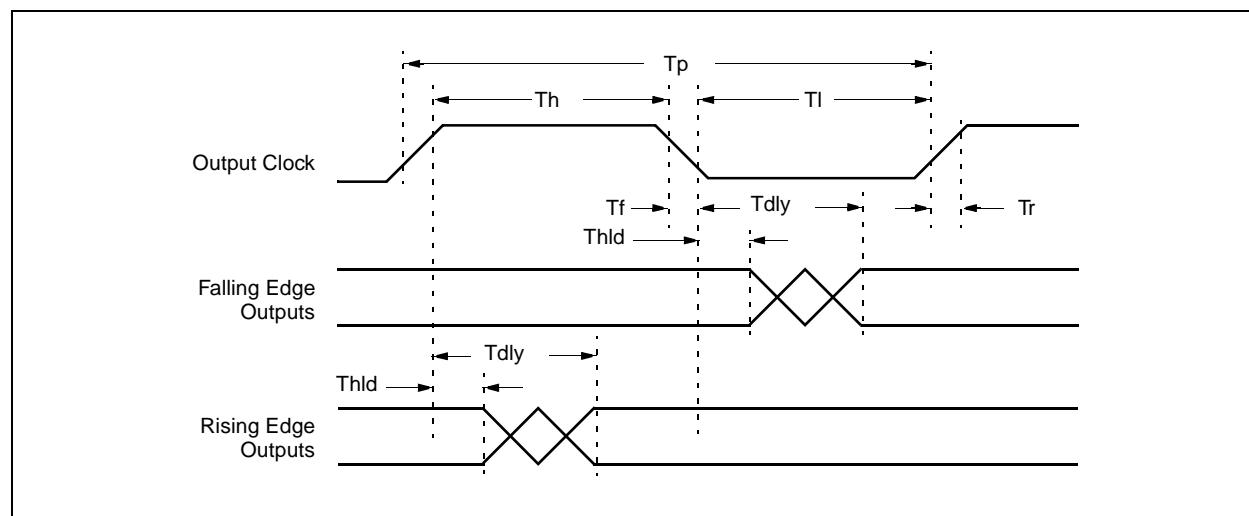
Symbol	Parameter	Minimum	Maximum	Units
Ts	Input Setup Time	35		ns
Thld	Input Hold Time	10		ns

Table 6-6. Input Clock Edge Selection

Clock	Edge	Inputs	TCLK_SEL (CMD_2)	RCLK_SEL (CMD_2)	RCLK_INV (CMD_7)
HDSL Channel Inputs					
BCLK1	Falling	QCLK1, RDAT1, TAUX1	—		
BCLK2	Falling	QCLK2, RDAT2, TAUX2			
BCLK3	Falling	QCLK3, RDAT3, TAUX3			
PCM Channel Inputs					
TCLK	Falling	TSER, INSDAT, TMSYNC	00	—	—
TCLK	Rising	TSER, INSDAT, TMSYNC	01	—	—
RCLK	Falling	TSER, INSDAT, TMSYNC	1x	00	0
RCLK	Rising	TSER, INSDAT, TMSYNC	1x	00	1
EXCLK	Falling	TSER, INSDAT, TMSYNC	1x	01	0
EXCLK	Rising	TSER, INSDAT, TMSYNC	1x	01	1
EXCLK	Falling	TSER, INSDAT, TMSYNC	1x	10	0
EXCLK	Rising	TSER, INSDAT, TMSYNC	1x	10	1
Test Access Inputs					
TCK	Rising	TMS, TDI	—		

### 6.1.4 Switching Characteristics

**Figure 6-3. Output Clock and Data Timing**



**Table 6-7. Clock and Data Switching Characteristics**

Symbol	Parameter	Minimum	Maximum	Units
1/ $T_p$	SCLK Frequency	15	20	MHz
	RCLK Frequency	0.064	4.1	MHz
$T_h$	Clock Width High	$T_p - 20$	$T_p + 20$	ns
$T_l$	Clock Width Low	$T_p - 20$	$T_p + 20$	ns
$T_r$	Clock Rise Time		15	ns
$T_f$	Clock Fall Time		15	ns
$T_{hld}$	Output Data Hold	0		ns
$T_{dly}$	Output Data Delay		25	ns

Table 6-8. Output Clock Edge Selection

Clock	Edge	Outputs	TCLK_SEL (CMD_2)	RCLK_SEL (CMD_2)	RCLK_INV (CMD_7)
HDSL Channel Outputs					
BCLK1	Rising	TDAT1, TLOAD1, RAUX1, ROH1	—	—	—
BCLK2	Rising	TDAT2, TLOAD2, RAUX2, ROH2			
BCLK3	Rising	TDAT3, TLOAD3, RAUX3, ROH3			
PCM Transmit Channel Outputs					
TCLK	Rising	MSYNC, INSERT	00	—	—
TCLK	Falling	MSYNC, INSERT	01	—	—
RCLK	Rising	MSYNC, INSERT	1x	00	0
RCLK	Falling	MSYNC, INSERT	1x	00	1
EXCLK	Rising	MSYNC, INSERT	1x	01	0
EXCLK	Falling	MSYNC, INSERT	1x	10	0
PCM Receive Channel Outputs					
RCLK	Rising	RSER, RMSYNC, DROP	—	00	0
RCLK	Falling	RSER, RMSYNC, DROP	—	00	1
EXCLK	Rising	RSER, RMSYNC, DROP	—	01	0
EXCLK	Falling	RSER, RMSYNC, DROP	—	10	0
TCLK	Rising	RSER, RMSYNC, DROP	00	11	0
TCLK	Falling	RSER, RMSYNC, DROP	01	11	0
Test Access Outputs					
TCK	Falling	TDO	—		



### 6.1.5 MPU Interface Timing

Motorola- (MPUSEL = 1) and Intel- (MPUSEL = 0) style microprocessor bus timing, as follows:

**Table 6-9. MPU Interface Timing Requirements**

Symbol	Parameter	Minimum	Maximum	Units
1	ALE Pulse-Width High	20		ns
2	Address Input Setup to ALE Falling	10		ns
3	Address Input Hold after ALE Low	7		ns
5	Data Input Setup to End of Write Pulse	10		ns
6	Data Input Hold After Write Pulse	8		ns
7	WR* Setup to Start of Read or Write Pulse	10		ns
8	WR* Hold after Read or Write Pulse	10		ns
9	ALE Hold after Read or Write Pulse	8		ns
10	Write Pulse-Width: WR*, RD*, and CS* Low (MPUSEL = 1) RD* = 1, WR*, and CS* Low (MPUSEL = 0)	$2 \times \frac{1}{f_{GCLK}}$		ns
11	Read Pulse Width (WR* = 1, RD* and CS* Low)	26		ns
	Read Pulse Width (WR* = 1, RD* and CS* Low) Address = 0x3C only.	$2 \times \frac{1}{f_{GCLK}}$		ns

**Table 6-10. MPU Interface Switching Characteristics**

Symbol	Parameter	Minimum	Maximum	Units
12	Data Out Enable (Low Z) after Start of Read Pulse	2		ns
13	Data Out Valid After Start of Read Pulse (Access Time)		26	ns
14	Data Out Hold After End of Read Pulse	1		ns
15	Data Out Disable (High Z) after End of Read Pulse		25	ns
16	INTR* Hold After End of Write Pulse (when writing interrupt mask or clear registers)	5		ns
17	INTR* Delay from End of Write Pulse (when writing interrupt mask or enable registers)		20	ns

Figure 6-4. MPU Write Timing, MPUSEL = 1

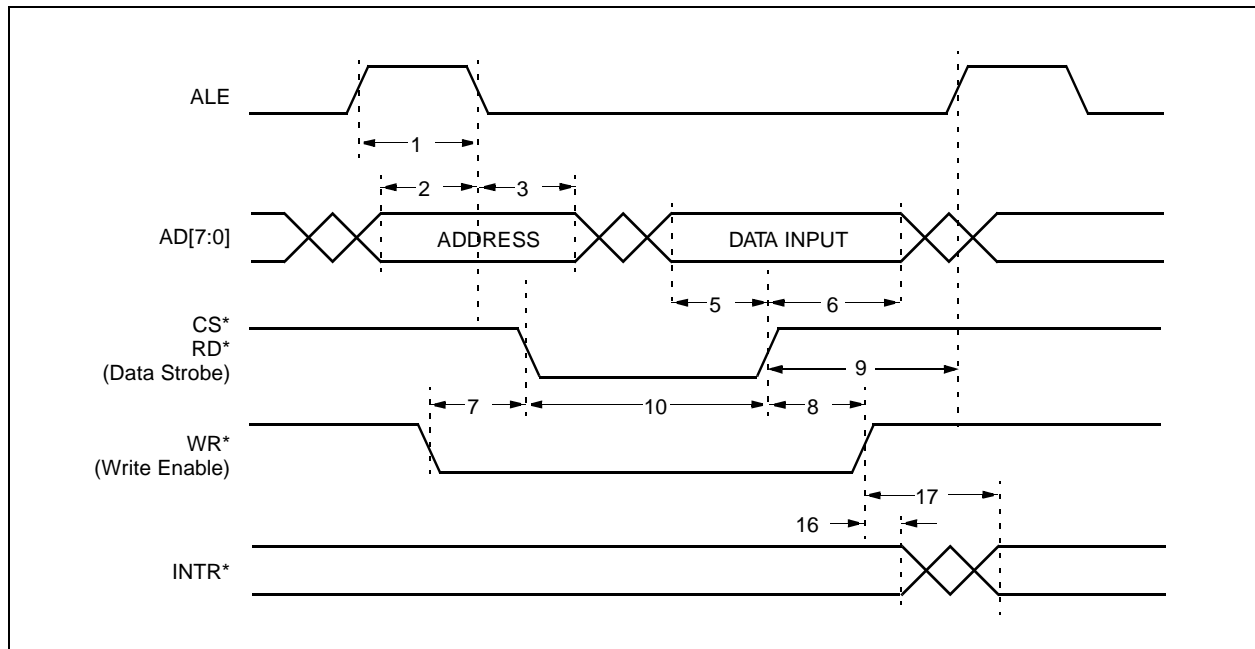
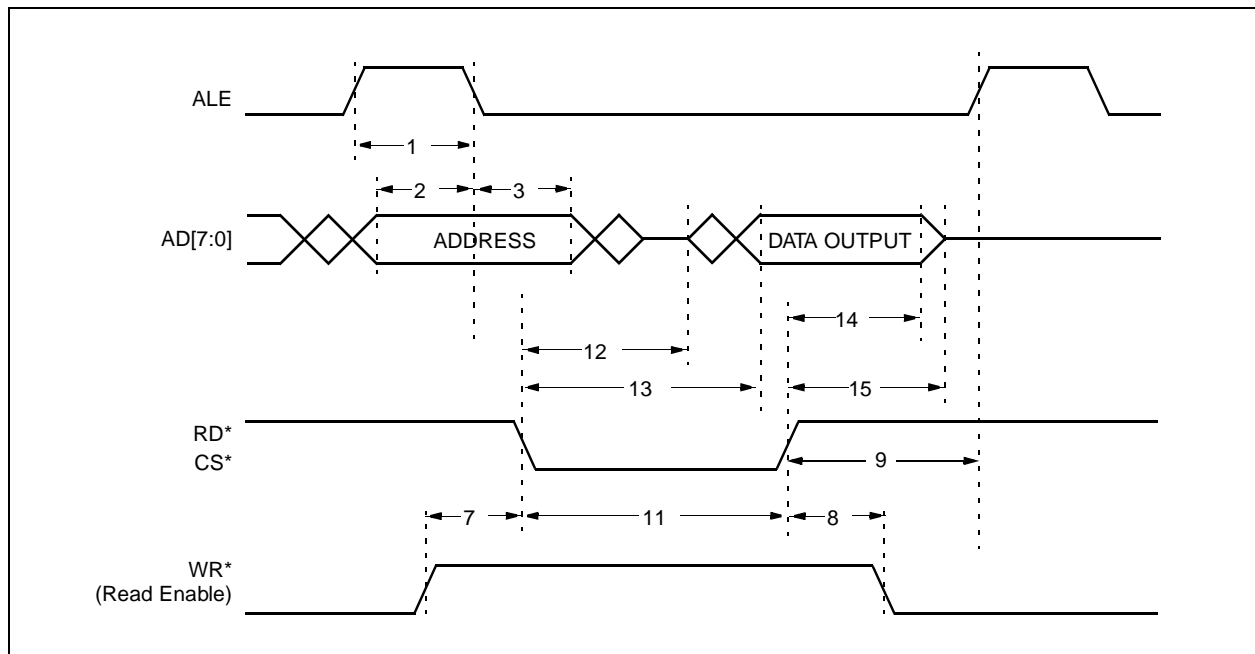
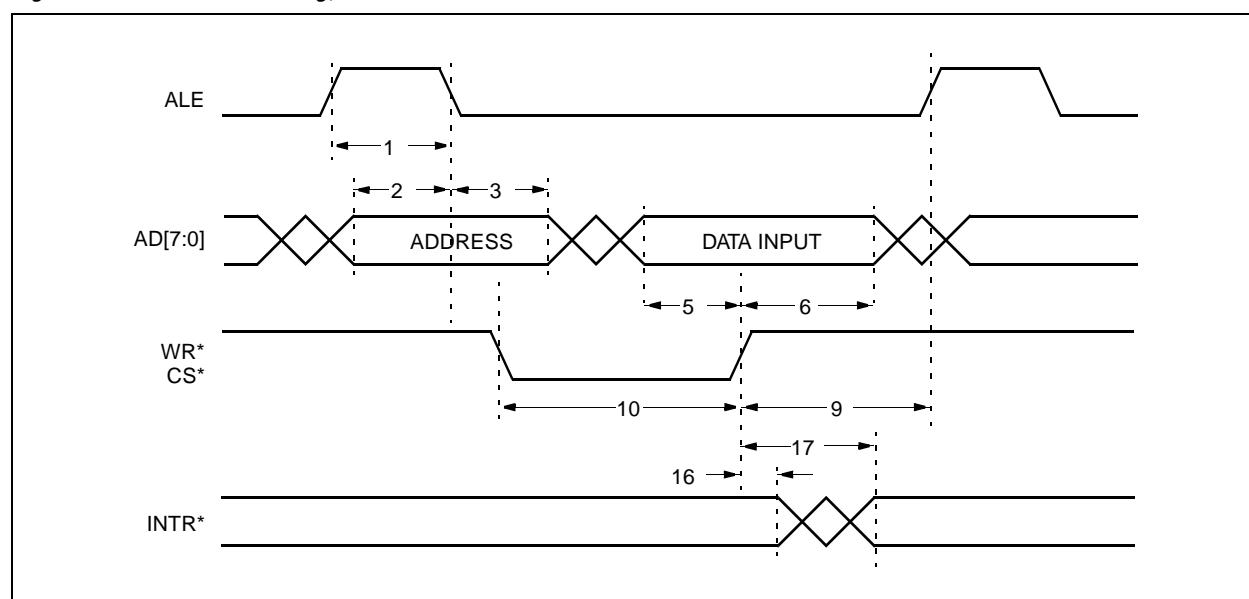
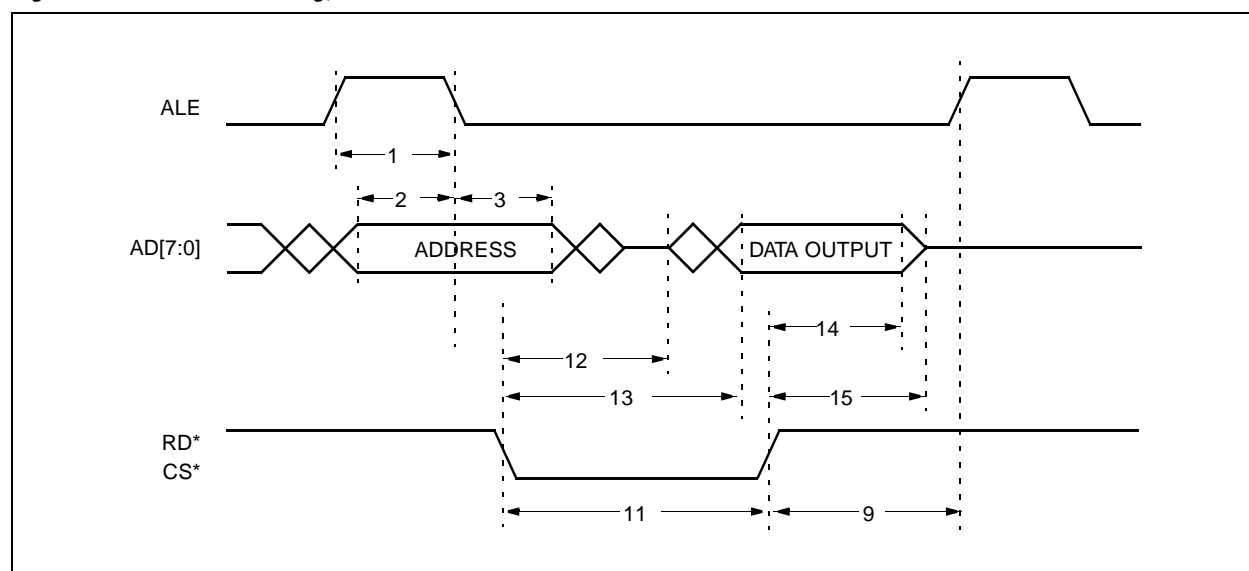


Figure 6-5. MPU Read Timing, MPUSEL = 1



**Figure 6-6. MPU Write Timing, MPUSEL = 0****Figure 6-7. MPU Read Timing, MPUSEL = 0**

6.2 Mechanical Specifications

Figure 6-8. 68-Pin PLCC Package Drawing

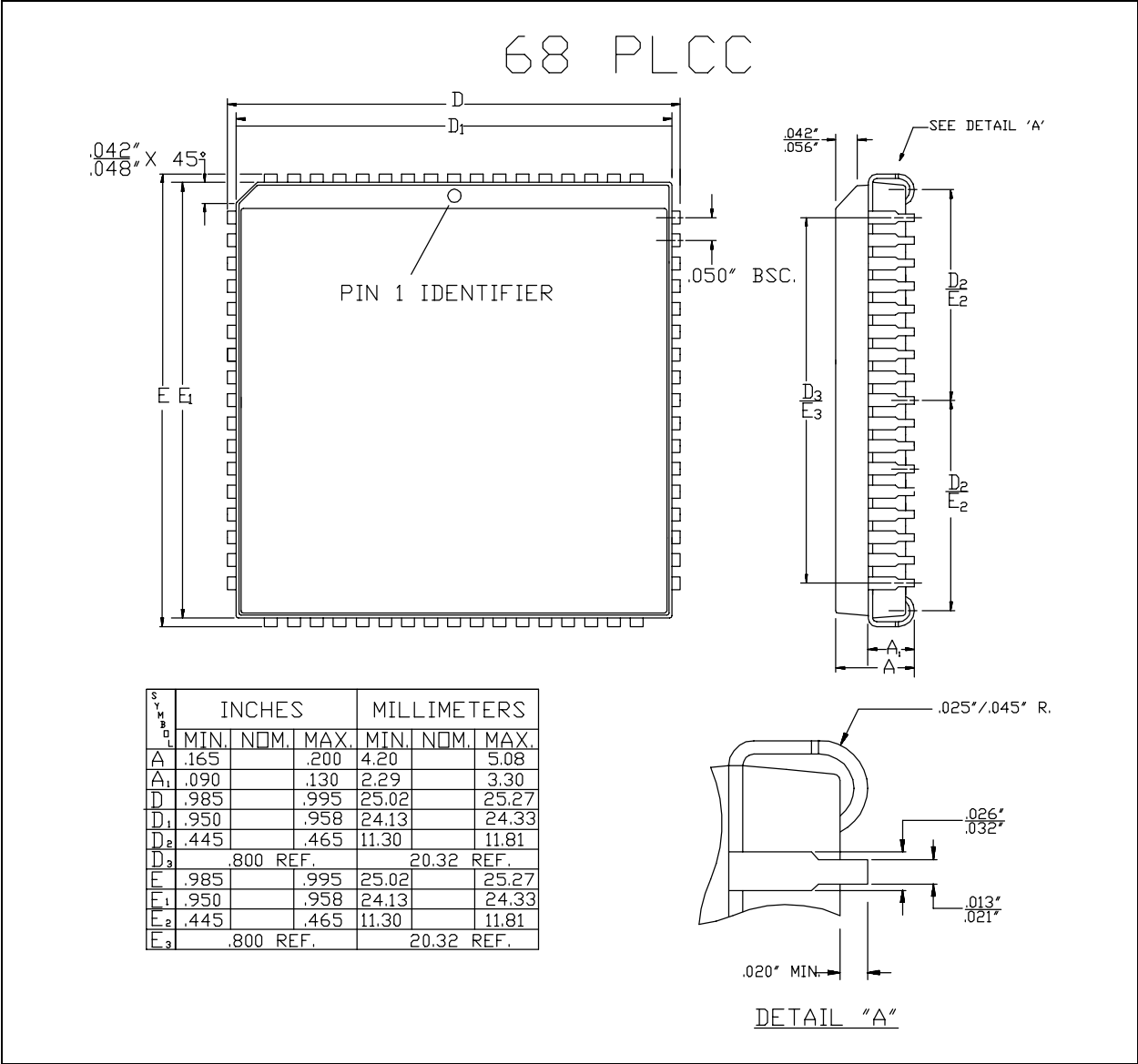
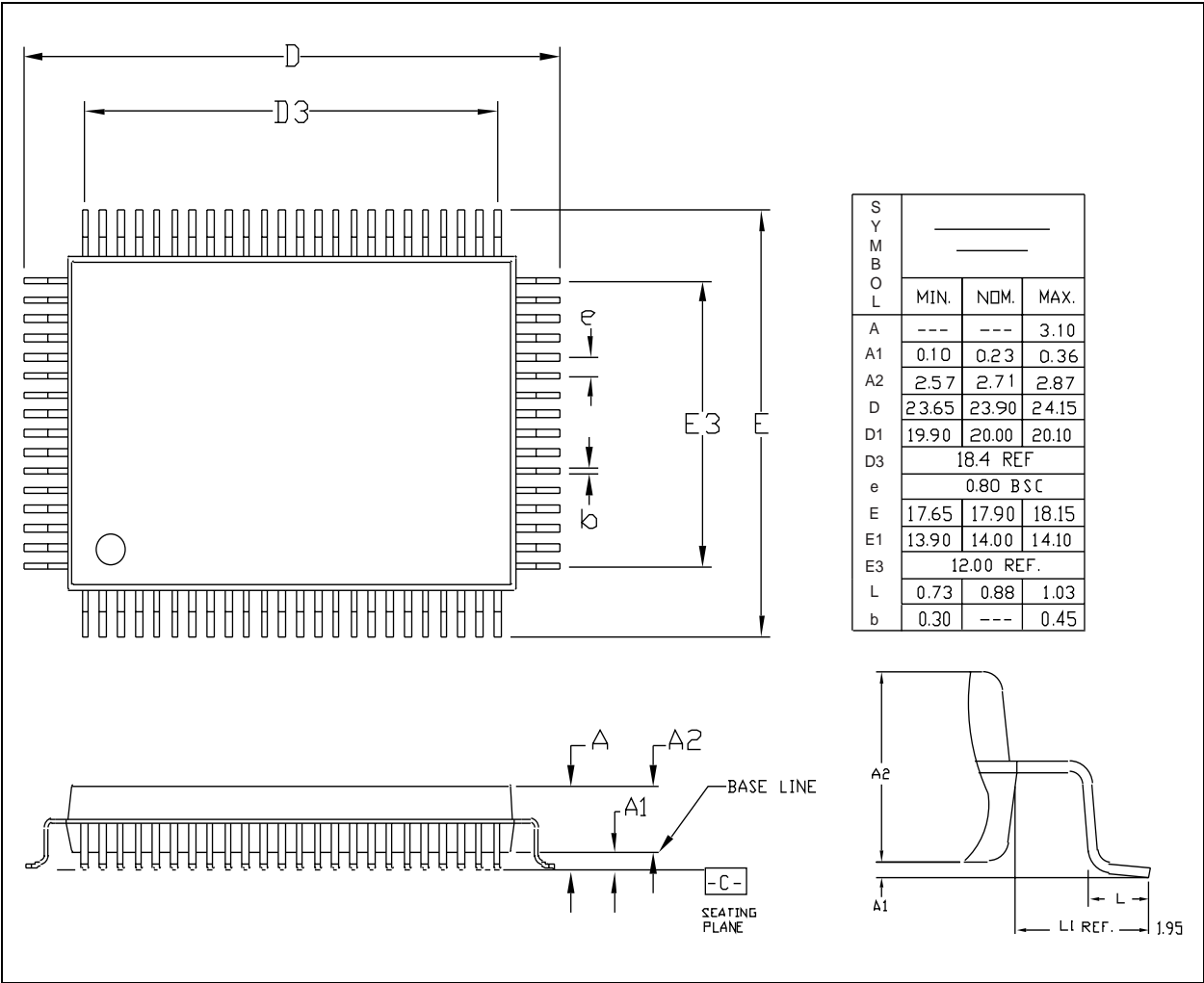


Figure 6-9. 80-Pin PQFP Mechanical Specification





# 7.0 Acronyms, Abbreviations and Notation

---

## 7.1 Arithmetic Notation

### 7.1.1 Bit Numbering

The Least Significant Bit (LSB) having the lowest number represents the lowest number within a bit.

### 7.1.2 Acronyms and Abbreviations

AIS	Alarm Indication Signal
2B1Q	2 Binary, 1 Quaternary
BER	Bit Error Rate
CMOS	Complementary Metal-Oxide Semiconductor
CO	Central Office
CRC	HDSL Cyclic Redundancy Check
DPLL	Digital Phase Lock Loop
EOC	HDSL Embedded Operations Channel
ESF	Extended Superframe
FEBE	HDSL Far-End Block Error
JTAG	Joint Test Action Group
HDSL	High-Bit-Rate Digital Subscriber Line
HOH	HDSL OverHead
HRP	HDSL Repeater Present
HTU-C	HDSL Terminal Unit at the Central Office
HTU-R	HDSL Terminal Unit at the Remote Distribution
LIU	Line Interface Unit
LOSD	Loss of Signal - DS1
LOSW	HDSL Loss-of-Sync Word
LSB	Least Significant Bit
LFSR	Linear Feedback Shift Register
MSB	Most Significant Bit
PQFP	Plastic Quad Flat Pack
PLCC	Plastic Leaded Chip Carrier
PRBS	Pseudo-Random Binary Sequence
QUAT	Quaternary symbol
QRSS	Quasi-Random Sequence Signal
SF	Super Frame
UIB	Unspecified Indicator Bit

VCXO	Voltage-Controlled Crystal Oscillator
------	---------------------------------------



# Appendix A

---

## A.1 Differences Between Bt8953A and RS8953B

**Table A-1. Pin Definitions**

Pin Number		Bt8953A (5.0 V)		RS8953B (3.3 V)	
68 Pin PLCC	80 Pin PQFP	Signal	Description	Signal	Description
19	13	VCC	5.0 V	VCC	3.3 V
27	23	VCC	5.0 V	VCC	3.3 V
48	48	PLLVCC	5.0 V	PLLVCC	3.3 V
49	49	VCC (SCAN_MD)	5.0 V	VCC (SCAN_MD)	3.3 V
61	63	VCC	5.0 V	VCC	3.3 V
44	43	LP1	Multiplier PLL analog phase detector output	VCC	3.3 V
45	45	LP2	Voltage level controls the VCO frequency of the multiplier PLL	VEXT	Input protection diodes bias

Table A-2. Power Consumption

Configuration	Bt8953A (5.0 V)	RS8953B (3.3 V)
Maximum	80 mA	21.1 mA
528 Kbps on single DSL / 2.048 Mbps on PCM	30.5 mA	—
1040 Kbps on single DSL / 2.048 Mbps on PCM	34.8 mA	—
1168 Kbps on single DSL / 2.048 Mbps on PCM	36.2 mA	—
1T1 (1.552 Mbps on single DSL / 1.544 Mbps on PCM)	—	11.6 mA
2T1 (784 Kbps on 2 DSL / 1.544 Mbps on PCM)	—	9.8 mA
1E1 (2.320 Mbps on single DSL / 2.048 Mbps on PCM)	—	13.8 mA
2E1 (1168 Kbps on 2 DSL / 2.048 Mbps on PCM)	38.6 mA	10.9 mA
3E1 (784 Kbps on 3 DSL / 2.048 Mbps on PCM)	41.3 mA	10.7 mA

# Appendix B: Bt8953A/RS8953B

## Product Bulletin

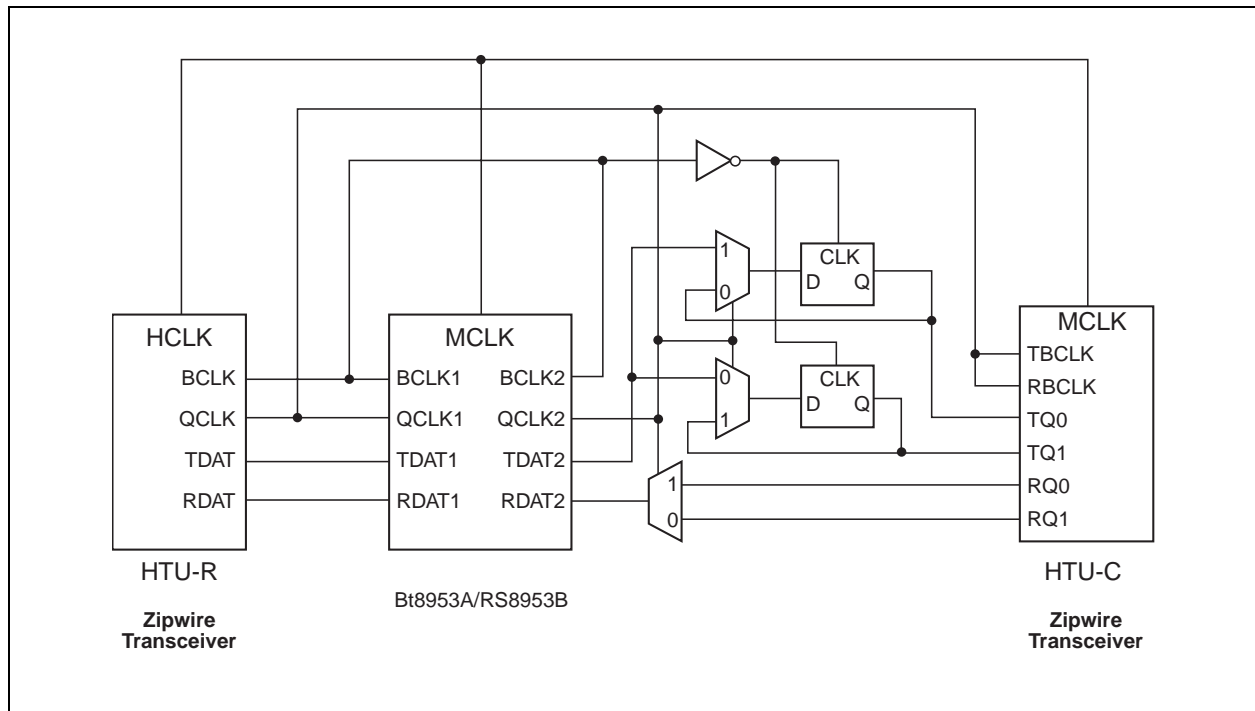
---

### ***B.1 BCLK Phase Constraints In Repeater Mode; Non-Conformance Product Affected: Bt8953A and RS8953B***

While in repeater mode (REPEAT\_EN = 1 for CH1 and CH2), a BCLK1 to BCLK2 phase difference of 180 degrees, +/- 5 nSec, will result in corrupted data transfer. All other phase relationships are acceptable.

In a repeater mode application, to prevent a phase difference of 180 degrees between BCLK1 and BCLK2, the same clock is feed to both BCLK1 and BCLK2. To use the same clock for both HDSL transceivers, one HDSL transceiver needs to be slaved to the other HDSL transceiver. This is accomplished by setting the HDSL transceiver, which is configured as HTU-C, to “parallel slave mode.” Then the parallel signals from the HDSL transceiver need to be externally converted to “serial, sign first mode” to interface to the Bt8953A/RS8953B.

To configure the ZipWire transceiver to “parallel slave mode”, address 0x06 of the ZipWire transceiver needs to be modified. Also to align the sign and magnitude data correctly, the tbcclk\_pol and rbclk\_pol need to set properly. The tbcclk\_pol and rdbclk\_pol are controlled by address 0x06 of the ZipWire transceiver. TQ[1,0] should be sampled on the falling edge of TBCLK. RQ[1,0] should be updated on the falling edge of RBCLK.





**Further Information**  
literature@conexant.com  
1-800-854-8099 (North America)  
33-14-906-3980 (International)

**Web Site**  
www.conexant.com

**World Headquarters**  
Conexant Systems, Inc.  
4311 Jamboree Road  
P. O. Box C  
Newport Beach, CA  
92658-8902  
Phone: (949) 483-4600  
Fax: (949) 483-6375

**U.S. Florida/South America**  
Phone: (813) 799-8406  
Fax: (813) 799-8306

**U.S. Los Angeles**  
Phone: (805) 376-0559  
Fax: (805) 376-8180

**U.S. Mid-Atlantic**  
Phone: (215) 244-6784  
Fax: (215) 244-9292

**U.S. North Central**  
Phone: (630) 773-3454  
Fax: (630) 773-3907

**U.S. Northeast**  
Phone: (978) 692-7660  
Fax: (978) 692-8185

**U.S. Northwest/Pacific West**  
Phone: (408) 249-9696  
Fax: (408) 249-7113

**U.S. South Central**  
Phone: (972) 733-0723  
Fax: (972) 407-0639

**U.S. Southeast**  
Phone: (770) 246-8283  
Fax: (770) 246-0018

**U.S. Southwest**  
Phone: (949) 483-9119  
Fax: (949) 483-9090

**APAC Headquarters**  
Conexant Systems  
Singapore, Pte. Ltd.  
1 Kim Seng Promenade  
Great World City  
#09-01 East Tower  
Singapore 237994  
Phone: (65) 737 7355  
Fax: (65) 737 9077

**Australia**  
Phone: (61 2) 9869 4088  
Fax: (61 2) 9869 4077

**China**  
Phone: (86 2) 6361 2515  
Fax: (86 2) 6361 2516

**Hong Kong**  
Phone: (852) 2827 0181  
Fax: (852) 2827 6488

**India**  
Phone: (91 11) 692 4780  
Fax: (91 11) 692 4712

**Korea**  
Phone: (82 2) 565 2880  
Fax: (82 2) 565 1440

**Europe Headquarters**  
Conexant Systems France  
Les Taissounieres B1  
1681 Route des Dolines  
BP 283  
06905 Sophia Antipolis  
Cedex  
France  
Phone: (33 4) 93 00 33 35  
Fax: (33 4) 93 00 33 03

**Europe Central**  
Phone: (49 89) 829 1320  
Fax: (49 89) 834 2734

**Europe Mediterranean**  
Phone: (39 02) 9317 9911  
Fax: (39 02) 9317 9913

**Europe North**  
Phone: (44 1344) 486 444  
Fax: (44 1344) 486 555

**Europe South**  
Phone: (33 1) 41 44 36 50  
Fax: (33 1) 41 44 36 90

**Middle East Headquarters**  
Conexant Systems  
Commercial (Israel) Ltd.  
P. O. Box 12660  
Herzlia 46733, Israel  
Phone: (972 9) 952 4064  
Fax: (972 9) 951 3924

**Japan Headquarters**  
Conexant Systems Japan  
Co., Ltd.  
Shimomoto Building  
1-46-3 Hatsudai,  
Shibuya-ku, Tokyo  
151-0061 Japan  
Phone: (81 3) 5371-1567  
Fax: (81 3) 5371-1501

**Taiwan Headquarters**  
Conexant Systems, Taiwan  
Co., Ltd.  
Room 2808, 333  
International Trade Building  
Keelung Road, Section 1  
Taipei 110, Taiwan, ROC  
Phone: (886 2) 2720 0282  
Fax: (886 2) 2757 6760