CX28560
HDLC Controller

Data Sheet

28560-DSH-001-B
April 2004

MINDSPEED™

Ordering Information

Model Number	Package	Operating Temperature
—	TBGA 40 mm x 40 mm	–40 °C to +85 °C

Revision History

Revision	Level	Date	Description
A	Advance	December 2000	Initial release (document No. 101302A).
A	Advance	April 2001	◆ 500031A Formerly document No. 101302A. ◆ Correction of technical inaccuracies for first full release.
B	Advance	October 2001	◆ Corrected fuzzy drawings (Chapter 8.0). ◆ In Table 9-3, replaced signal names to match Pin Description. ◆ Created Table 1-13 for ONESEC signal.
C	Advance	July 2002	◆ Based on preliminary characterization, updated EBUS timing specification (Section 8.2.4) and few other electrical specifications (Chapter 8.0). ◆ Corrected technical inaccuracies.
A	Advance	March 2003	◆ Released with new document number: 28560-DSH-001-A.
B	Advance	April 2004	◆ Corrected pin assignment on AD[6] - AD[10] in Table 9-3. ◆ TDATA[0] mislabelled TDAT[0] in Table 9-3. ◆ RDAT[20] mislabelled RDAT[16] in Table 9-3.

CX28560

HDLC Controller

The CX28560 is an advanced Multichannel Synchronous Communications Controller (MUSYCC™) that formats and deformats up to 2047 HDLC channels in a CMOS integrated circuit. MUSYCC operates at Layer 2 of the Open Systems Interconnection (OSI) protocol reference model and provides a comprehensive, high-density solution for processing HDLC channels for inter-networking applications.

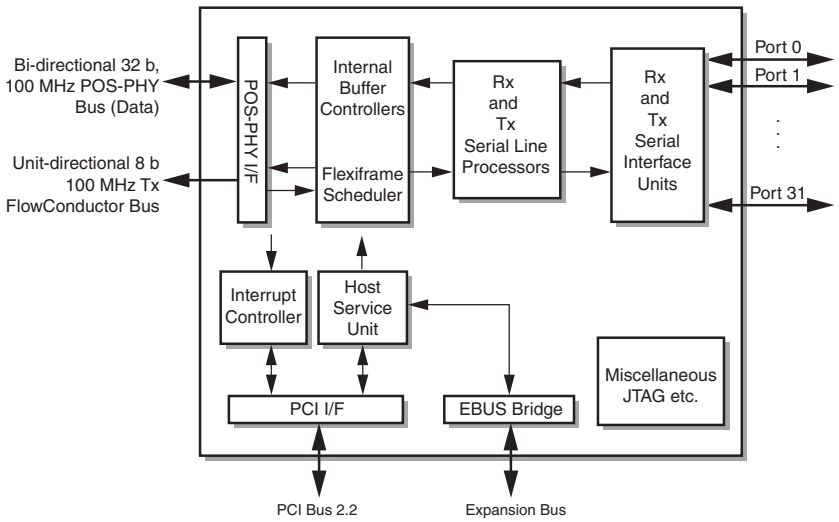
All packet data passed between the system and the CX28560 is passed across the POS-PHY interface (POS-PHY). The POS-PHY operates in packet mode as a 32-bit wide point-to-point interface at 100 MHz. Data is transferred in fragments of user-configurable length (minimum 32 bytes per fragment).

The CX28560 supports a PCI interface for initial configuration as well as to perform dynamic activation and deactivation of channels. In addition, the CX28560's configuration and performance monitoring counters can be read over the PCI interface.

The scheduling system for the receive and transmit data flow is based on the unique Flexiframe™ algorithm. Flexiframe enables efficient memory utilization and provides support for various channels operating at extremely different rates. Flexiframe allows dynamic resizing of every channel's rate without affecting the other channels. The order in which message fragments are transferred across the POS-PHY is fixed by the Flexiframe structure, each fragment having been tagged with a 4-byte fragment header. The fragment header contains the channel number and relevant status information.

A dedicated 8-bit bus provides the system the necessary feedback to determine the amount of data contained in each channel's transmit buffers. This is achieved by the CX28560 sending requests to the system for more transmit data. In the receive direction, the CX28560 operates autonomously without any need for system intervention or guidance.

Functional Block Diagram



Distinguishing Features

- 2047-channel HDLC controller
- OSI Layer 2 protocol support
- 32-bit full duplex standard POS-PHY Level 3 bus
- Aggregate bandwidth of 700 Mbps full duplex
- 32 bits, 33 MHz PCI 2.2 bus interface for configuration and monitoring
- Dedicated feedback bus for Tx buffers fill level
- 32 independent serial interfaces support:
  - T1 data stream
  - N * 64 Kb/s data stream
  - TSBUS interfaces
  - Unchannelized data stream
- Configurable logical channels
  - Standard DS0 (56, 64 Kbps)
  - Hyperchannel (N x 64)
- Channels' bit rate can be dynamically changed.
- Per channel protocol mode selection
- Per-channel message length check
  - Select no length checking
  - Select from three 14-bit registers to compare message length
- HDLC maximum packet length 16,384 bytes
- 3 separate HDLC modes, configurable per channel:
  - no FCS
  - 16-bit FCS
  - 32-bit FCS
- Transparent (not HDLC) mode
- Autonomous Rx operation and arbitration between the channels
- Selectable endian configuration for control information (PCI)
- Per-channel buffer management
- Full set of 10 performance monitoring counters per channel
- Transfer of partial HDLC messages over the POS-PHY interface
- Low power, 1.8 volt core, 3.3 volt I/O, CMOS operation.
- Local expansion bus interface (EBUS) for accessing non-PCI components (framers, LIUs)
- JTAG boundary scan access port
- 40 mm TBGA package

The CX28560 supports four serial port modes: Conventional Channelized, Conventional Unchannelized, Conventional T1, and TSBUS. In TSBUS mode, the CX28560 supports a special Mindspeed proprietary interface: the TSBUS (Time Slot BUS). The TSBUS allows for mapping of all tributary signals to time slots for a transmission to external devices, such as Mindspeed's BAM (Broadband Access Multiplexer) device family. The TSBUS interface consists of two serial interfaces: a 51.84 MHz payload interface and a 12.96 MHz overhead interface. Payload of tributary signals is mapped to time slots on the payload bus allowing for a transmission of the following signals' payload:

- 28 x DS1, VT1.5, or VC-11 signals
- 21 x E1, VT2.0, or VC-12 signals
- 1 x DS3, E3, or STS-1 signal

Overhead information including SONET/SDH/PDH overhead and control information is transmitted in time slots on the overhead TS-Bus interface. The first 12 ports of the CX28560, when configured in TSBUS mode, also support DS0 extraction.

Contents

Figures . . . . . **xiii**

Tables . . . . . **xv**

1.0 Introduction **1-1**

1.1 External Interfaces 1-2
1.1.1 CX28560 Serial Interface 1-2
1.1.1.1 CX28560 Serial Port Modes Description 1-2
1.1.1.2 CX28560 Serial Port Throughput Limits 1-3
1.1.1.3 TSBUS—Time Slot Bus 1-5
1.2 System-Side Interfaces 1-6
1.2.1 POS-PHY Interfaces 1-6
1.2.1.1 POS-PHY Data Interface—CX28560 1-6
1.2.1.2 Transmit FlowConductor Interface 1-7
1.2.2 Expansion Bus (EBUS) 1-7
1.2.3 PCI Bus Interface 1-7
1.3 Feature Summary 1-8
1.4 Applications Examples 1-11
1.5 System Overview 1-12
1.6 Block Diagram 1-14
1.7 Data Flow 1-16
1.7.1 Receive Data Path 1-16
1.7.2 Transmit Data Path 1-16
1.8 CX28560 Pin List 1-17
1.8.1 Pin Descriptions 1-17

2.0 Host Interfaces 2-1

2.1 Host Interface 2-1
2.1.1 PCI Interface 2-3
2.1.1.1 PCI Initialization 2-3
2.1.1.2 PCI Bus Operations 2-3
2.1.1.3 Fast Back-to-Back Transactions 2-4
2.1.1.4 PCI Configuration Space 2-5
2.2 PCI Configuration Registers 2-7
2.2.1 PCI Master and Slave 2-7
2.2.1.1 Register 0, Address 00h 2-7
2.2.1.2 Register 1, Address 04h 2-8
2.2.1.3 Register 2, Address 08h 2-10
2.2.1.4 Register 3, Address 0Ch 2-10
2.2.1.5 Register 4, Address 10h 2-11
2.2.1.6 Register 5–10, Address 14h–28h 2-11
2.2.1.7 Register 11, Address 2Ch 2-11
2.2.1.8 Register 12–14, Address 30h–38h 2-11
2.2.1.9 Register 15, Address 3Ch 2-12
2.2.2 PCI Reset 2-12
2.2.3 PCI Throughput and Latency Considerations 2-12
2.2.4 Host Interface 2-13
2.3 POS-PHY 2-14
2.3.1 POS-PHY Interfaces 2-14
2.3.1.1 POS-PHY Registered Mode 2-14
2.3.1.2 POS-PHY Data Interface 2-14
2.3.1.3 POS-PHY Flow Conductor Interface 2-15
2.3.1.4 Receive POS-PHY Initialization 2-15
2.3.1.5 Transmit POS-PHY Initialization 2-15

3.0 Expansion Bus (EBUS) 3-1

3.1 EBUS—Operational Mode 3-3
3.1.1 Initialization 3-3
3.1.2 Clock 3-5
3.1.3 Interrupt 3-5
3.1.4 Address Duration 3-5
3.1.5 Data Duration 3-6
3.1.6 Bus Access Interval 3-6
3.1.7 PCI to EBUS Interaction 3-6
3.1.8 Microprocessor Interface 3-7
3.1.9 Arbitration 3-8
3.1.10 Connection 3-9
3.1.10.1 Multiplexing Address 3-11

4.0 CX28560 Serial Interface 4-1

4.1 Functional Description 4-1
4.2 Serial Interface Unit (SIU) 4-3
4.3 Serial Line Processor (SLP) 4-4
4.4 Buffer Controller 4-5
4.5 Interrupt Controller 4-7
4.6 Serial Port Interface Definition in Conventional Mode 4-7
4.6.1 Frame Synchronization Flywheel 4-8
4.6.2 Change Of Frame Alignment (COFA) 4-8
4.6.3 Out Of Frame (OOF)/Frame Recovery (FREC) 4-9
4.6.4 General Serial Port Interrupt 4-9
4.6.5 Channel Clear To Send (CTS) 4-10
4.6.6 Frame Alignment 4-10
4.7 Serial Port Interface Definition TSBUS Mode 4-11
4.7.1 TSBUS Frame Synchronization Flywheel 4-11
4.7.2 TSBUS Group Synchronization Flywheel 4-11
4.7.3 TSBUS Change Of Frame Alignment (COFA) 4-12
4.7.4 TSBUS Out Of Frame (OOF)/Frame Recovery (FREC) 4-12
4.7.5 TSBUS Frame Alignment 4-12
4.7.6 TSBUS Channel Clear To Send 4-12
4.7.7 TSBUS Interface 4-13
4.7.7.1 Payload TSBUS 4-14
4.7.7.2 Overhead TSBUS 4-14

5.0 The CX28560 Memory Organization 5-1

5.1 Memory Architecture 5-1
5.1.1 Register Map and Shared Memory Access 5-1
5.2 Global Registers 5-5
5.2.1 Service Request Mechanism 5-5
5.2.1.1 Service Request Descriptors 5-6
5.2.1.2 Service Request Descriptors 5-8
5.2.2 Port Alive Registers 5-10
5.2.3 Soft Chip Reset Register 5-10
5.3 Interrupt Level Descriptors 5-11
5.3.1 Interrupt Queue Register 5-11
5.3.1.1 Interrupt Descriptors 5-12
5.3.1.2 Interrupt Status Register 5-19
5.3.2 Interrupt Handling 5-20
5.3.2.1 Initialization 5-20
5.3.2.2 Interrupt Descriptor Generation 5-20
5.3.2.3 INTA# Signal Line 5-21
5.4 Global Configuration Register 5-22
5.5 EBUS Configuration Register 5-23
5.6 POS-PHY Control Registers 5-24
5.6.1 Transmit POS-PHY Thresholds Register 5-24
5.6.2 Transmit POS-PHY Control Register 5-24
5.6.3 Receive POS-PHY Control Register 5-25

5.7 Receive Path Registers 5-26
5.7.1 RSLP Channel Status Register 5-26
5.7.2 RSLP Channel Configuration Register 5-26
5.7.3 RSLP Maximum Message Length Register 5-28
5.7.4 RBUFFC Channel Configuration Register 5-29
5.7.5 RBUFFC Flexiframe Memory 5-30
5.7.6 RBUFFC Flexiframe Control Register 5-30
5.7.7 RBUFFC DATA FIFO Size Register 5-31
5.7.8 RBUFFC Fragment Size Register 5-31
5.7.9 RBUFFC Flexiframe Slot Time Register 5-31
5.7.10 RBUFFC Counter Memory 5-32
5.7.11 RSIU Time Slot Configuration Register 5-33
5.7.11.1 Receive Time Slot Map 5-33
5.7.11.2 RSIU Time Slot Configuration Descriptor 5-34
5.7.12 RSIU Time Slot Pointer Allocation Register 5-36
5.7.12.1 Time Slot Allocation Rules 5-36
5.7.13 RSIU Group Map Pointer Allocation Register 5-36
5.7.14 RSIU Group State Register 5-37
5.7.15 RSIU Port Configuration Register 5-37
5.8 Transmit Path Registers 5-39
5.8.1 TSLP Channel Status Register 5-39
5.8.2 TSLP Channel Configuration Register 5-40
5.8.3 TBUFFC Channel Configuration Register 5-41
5.8.4 TBUFFC Flexiframe Memory 5-42
5.8.5 TBUFFC Flexiframe Control Register 5-43
5.8.6 TBUFFC DATA FIFO Size Register 5-44
5.8.7 TBUFFC Flexiframe Slot Time Register 5-44
5.8.8 TBUFFC Counter Memory 5-45
5.8.9 TSIU Time Slot Configuration Register 5-45
5.8.9.1 Transmit Time Slot Map 5-45
5.8.9.2 TSIU Time Slot Configuration Descriptor 5-47
5.8.10 TSIU Time Slot Pointer Allocation Register 5-49
5.8.10.1 Time Slot Allocation Rules 5-49
5.8.11 TSIU Group Time Slot Map Pointers Register 5-50
5.8.12 TSIU Group State Register 5-50
5.8.13 TSIU Port Configuration Register 5-51
5.9 POS-PHY Transaction Headers and Packets 5-53
5.9.1 Receive POS-PHY Data Bus 5-53
5.9.2 Transmit POS-PHY Data Bus 5-54
5.9.3 Transmit Flow Conductor Bus 5-55

6.0 Functional Description 6-1

6.1 Initialization 6-1
6.1.1 Reset 6-1
6.1.1.1 Hard PCI Reset 6-2
6.1.1.2 Soft Chip Reset 6-2
6.1.2 Configuration 6-3
6.1.2.1 PCI Configuration 6-3
6.1.2.2 Service Request Mechanism 6-3
6.1.2.3 Global Configuration 6-4
6.1.2.4 Interrupt Queue Configuration 6-4
6.1.2.5 POS-PHY Configuration 6-4
6.1.2.6 Chip-Level Configuration 6-4
6.1.2.7 Channel and Port Configuration 6-5
6.1.2.8 Typical Initialization Procedure 6-5
6.2 Channel Operations 6-8
6.2.1 Channel Activation 6-8
6.2.1.1 Transmit Channel Activation 6-9
6.2.1.2 Receive Channel Activation 6-9
6.2.2 Channel Deactivation 6-10
6.2.2.1 Transmit Channel Deactivation 6-10
6.2.2.2 Receive Channel Deactivation 6-10
6.2.3 Channel Reactivation 6-10
6.3 Port Operations 6-11
6.3.1 Unmapped Time Slots 6-11
6.3.2 Enabling a Port 6-11
6.3.3 Disabling a Port 6-11

7.0 Basic Operations 7-1

7.1 Protocol-Independent Operations 7-1
7.1.1 Transmit 7-1
7.1.2 Receive 7-2
7.2 HDLC Mode 7-2
7.2.1 Frame Check Sequence 7-3
7.2.2 Opening/Closing Flags 7-3
7.2.3 Abort Codes 7-3
7.2.4 Zero-Bit Insertion/Deletion 7-4
7.2.5 Message Configuration Bits—HDLC Mode 7-4
7.2.5.1 Idle Code 7-4
7.2.5.2 Intermessage Pad Fill 7-4
7.2.5.3 Ending a Message with an Abort or Sending an Abort Sequence 7-5
7.2.6 Transmit Events 7-5
7.2.6.1 End Of Message (EOM) 7-5
7.2.6.2 Transmit COFA Recovery (TCREC) 7-5
7.2.7 Receive Events 7-5
7.2.7.1 End Of Message (EOM) 7-5
7.2.7.2 Change to Abort Code (CHABT) 7-6

7.2.7.3 CHABT Interrupt (if IDLEIEN = 1 in Chapter 5.0, RSLP Channel Configuration register). . . . 7-6
7.2.7.4 Change to Idle Code (CHIC) . . . 7-6
7.2.7.5 Frame Recovery (FREC) or Generic Serial PORT (SPORT) Interrupt . . . 7-6
7.2.7.6 Receive COFA Recovery (RCREC) . . . 7-7
7.2.8 Transmit Errors . . . 7-7
7.2.8.1 Transmit Underrun (BUFF) . . . 7-7
7.2.8.2 Transmit Change Of Frame Alignment (COFA) . . . 7-7
7.2.8.3 Buffer Controller Channel FIFO Overflow (BOVFLW) . . . 7-8
7.2.9 Receive Errors . . . 7-8
7.2.9.1 Receive Overflow (BUFF) . . . 7-8
7.2.9.2 Receive Change Of Frame Alignment (COFA) . . . 7-9
7.2.9.3 Out-Of-Frame (OOF) . . . 7-10
7.2.9.4 Frame Check Sequence (FCS) Error . . . 7-10
7.2.9.5 Octet Alignment Error (ALIGN) . . . 7-11
7.2.9.6 Abort Termination (ABT) . . . 7-11
7.2.9.7 Long Message (LNG) . . . 7-12
7.2.9.8 Short Message (SHT) . . . 7-12
7.3 Transparent Mode . . . 7-13
7.3.1 Message Configuration Bits—Transparent Mode . . . 7-13
7.3.1.1 Idle Code . . . 7-13
7.3.1.2 Intermessage Pad Fill . . . 7-13
7.3.1.3 Ending a Message with an Abort or Sending an Abort Sequence . . . 7-13
7.3.2 Transmit Events . . . 7-14
7.3.2.1 End Of Message (EOM) . . . 7-14
7.3.3 Receive Events . . . 7-14
7.3.3.1 End Of Message (EOM) . . . 7-14
7.3.3.2 Frame Recovery (FREC) . . . 7-15
7.3.3.3 Receive COFA Recovery (RCREC) . . . 7-15
7.3.4 Transmit Errors . . . 7-15
7.3.4.1 Transmit Underrun (BUFF) . . . 7-15
7.3.4.2 Transmit Change Of Frame Alignment (COFA) . . . 7-16
7.3.4.3 Buffer Controller Channel FIFO Overflow (BOVFLW) . . . 7-16
7.3.5 Receive Errors . . . 7-16
7.3.5.1 Receive Overflow (BUFF) . . . 7-16
7.3.5.2 Receive Change Of Frame Alignment (COFA) . . . 7-17
7.3.5.3 Out Of Frame (OOF) . . . 7-17
7.3.5.4 Short COFA (SHT COFA) . . . 7-18

8.0 Electrical and Mechanical Specification . . . 8-1

8.1 Electrical and Environmental Specifications . . . 8-1
8.1.1 Absolute Maximum Ratings . . . 8-1
8.1.2 Recommended Operating Conditions . . . 8-2
8.1.3 Electrical Characteristics . . . 8-2
8.2 Timing and Switching Specifications . . . 8-3
8.2.1 Overview . . . 8-3
8.2.2 Host Interface (PCI) Timing and Switching Characteristics . . . 8-3

8.2.3 Data Interface (POS-PHY) Timing and Switching Characteristics 8-7
8.2.4 Expansion Bus (EBUS) Timing and Switching Characteristics 8-11
8.2.5 EBUS Arbitration Timing Specification 8-13
8.2.6 Serial Interface Timing and Switching Characteristics 8-15
8.2.7 Test and Diagnostic Interface Timing 8-20
8.3 Package Thermal Specification 8-21
8.4 Mechanical Specification 8-22

9.0 Package Description 9-1

A. Counters A-1
A.1 One-Second Pin A-1
A.2 Counter Latching A-2
A.3 Counter Descriptions A-3
A.3.1 Receive Counters A-3
A.3.1.1 Multiple Errors on A Single Message A-3
A.3.2 Transmit Counters A-4
A.4 Reading Counters A-5
A.4.1 Receive Direction A-5
A.4.2 Transmit Direction A-6

B. Flexiframe Algorithm B-1
B.1 Overview B-1
B.2 New Flexiframe Required B-2
B.3 Algorithm B-2
B.3.1 Splitting Channel Bit Rates into Groups B-2
B.3.2 Harmonic Bit Rates B-3
B.3.3 Calculating Step Size Per Group B-4
B.3.4 Assigning Channels to Slots/Tracks B-5
B.4 Pseudo-Code B-5
B.4.1 Assigning Input Channels to Groups B-5
B.4.1.1 Computing the Number of Tracks to be Used B-7
B.4.2 Building the Output B-8
B.5 Analysis B-9
B.5.1 Equations for Analysis B-10
B.5.2 Solution for Equations B-10
B.5.3 Building the Flexiframe B-11

C. Flow Conductor Interface C-1
C.1 Overview C-1
C.2 Example C-3

D. TSBUS D-1
D.1 Connection Between CX28560 and Other TSBUS Device D-1
D.1.1 VSP Mapping of Intermixed Digital Level 2 Signals D-5
D.2 Timing Details D-7
D.2.1 Payload Bus, AC Characteristics D-7
D.2.2 Transmit Timing D-8
D.2.3 Receive Timing D-10

D.3 Overhead Bus, AC Characteristics D-15
D.3.1 Transmit Timing D-15
D.3.2 Receive Timing D-15
E. Buffer Controller FIFO Size Calculation E-1
E.1 Introduction E-1
E.1.1 Terminology E-1
E.1.2 Assumptions E-2
E.1.3 Overkill E-3
E.2 Expanding Data In E-4
E.2.1 Ending a 57-Byte Message E-4
E.2.2 Byte Message E-5
E.2.3 Ending a Fragment with No End of Message E-5
E.2.4 Not Ending a Fragment E-5
E.3 General Buffer Wastage E-6
E.4 Overview of Analysis E-6
E.4.1 Preliminary Calculations E-6
E.5 Receive Analysis E-7
E.5.1 Preliminaries E-7
E.5.1.1 Missed Services E-7
E.5.2 Calculation of Step Size Between Services E-9
E.5.2.1 Starting Position E-9
E.5.2.2 Servicing E-10
E.5.2.3 57-Byte Messages E-10
E.5.2.4 Last Bit (Byte) E-10
E.5.3 Example E-11
E.5.3.1 Channels of Same Bit Rate, Large Minimum Packet Size E-11
E.6 Transmit FIFO Calculation E-12
E.6.1 Service Request Scheme E-12
F. Example of Little-Big Endian Byte Ordering F-1
G. Example of an Arbitration for Fast and Non-Fast Back-to-Back Transactions G-1
H. PCI Utilization H-1
H.1 Overview H-1
H.2 Analysis H-1
H.2.1 Internal Considerations H-1
H.2.2 Conditions H-2
I. Maximum Number of Channels Calculation I-1

Figures

Figure 1-1. OC-12 Application . . . 1-11
Figure 1-2. System Overview. . . . 1-12
Figure 1-3. CX28560 Block Diagram . . . 1-14
Figure 2-1. The CX28560 Host Interface Functional Block Diagram . . . 2-2
Figure 3-1. EBUS Functional Block Diagram with Local MPU . . . 3-1
Figure 3-2. EBUS Functional Block Diagram without Local MPU . . . 3-2
Figure 3-3. EBUS Connection, Non-Multiplexed Address/Data, 8 Framers, No Local MPU . . . 3-9
Figure 3-4. EBUS Connection, Non-Multiplexed Address/Data, 16 Framers, No Local MPU . . . 3-10
Figure 3-5. EBUS Connection, Multiplexed Address/Data, 8 Framers, No Local MPU . . . 3-11
Figure 3-6. EBUS Connection, Multiplexed Address/Data, 4 Framers, No Local MPU . . . 3-11
Figure 4-1. Serial Interface Functional Block Diagram . . . 4-2
Figure 5-1. Interrupt Notification to Host. . . . 5-21
Figure 5-2. Receive Time Slot Map Pointers . . . 5-33
Figure 5-3. Transmit Time Slot Map Pointers . . . 5-46
Figure 8-1. PCI Clock (PCLK) Waveform, 3.3 V Clock . . . 8-4
Figure 8-2. PCI Output Timing Waveform . . . 8-6
Figure 8-3. PCI Input Timing Waveform . . . 8-7
Figure 8-4. Transmit Physical Timing . . . 8-8
Figure 8-5. Receive Physical Timing . . . 8-10
Figure 8-6. EBUS Reset Active to Inactive Delay . . . 8-11
Figure 8-7. EBUS Output Timing Waveform . . . 8-12
Figure 8-8. EBUS Input Timing Waveform . . . 8-12
Figure 8-9. EBUS Write/Read Cycle, Intel-Style . . . 8-13
Figure 8-10. EBUS Write/Read Cycle, Motorola-Style . . . 8-14
Figure 8-11. Serial Interface Clock (RCLK,TCLK) Waveform . . . 8-15
Figure 8-12. Serial Interface Data Input Waveform . . . 8-16
Figure 8-13. Serial Interface Data Delay Output Waveform . . . 8-17
Figure 8-14. Transmit and Receive T1 Mode . . . 8-18
Figure 8-15. Transmit and Receive Channelized Non-T1 (i.e., N x 64) Mode . . . 8-19
Figure 8-16. JTAG Interface Timing . . . 8-20
Figure 8-17. Package Diagram . . . 8-22
Figure 9-1. Pin Diagram . . . 9-12
Figure C-1. Data and Command Storage in Internal Buffer . . . C-2
Figure D-1. CX28560 Time Slot Interface Pins . . . D-2
Figure D-2. Source/Destination of TSBUS Block Line-Side Signals . . . D-4
Figure D-3. Payload Time Slot Bus Transmit Data (TSB_TDAT) . . . D-8
Figure D-4. Payload Time Slot Bus Transmit Stuff Indicator (TSB_TSTUFF) . . . D-9
Figure D-5. Payload Time Slot Bus Receive Data (TSB_RDAT) . . . D-10
Figure D-6. Payload Time Slot Bus Receive Stuff Indicator (TSB_RSTUFF) . . . D-11

Figure D-7. TSBUS Interface to CX28560 Transmit SYNC Timing (TSB_TSYNCO) D-12
Figure D-8. TSBUS Interface to CX28560 Transmit SYNC Timing (TSB_TSYNCI) D-13
Figure D-9. TSBUS Interface to CX28560 Receive SYNC Timing (TSB_RSYNC) D-14
Figure E-1. BUFFC Internal FIFO E-2
Figure E-2. Worst Case on a Frame E-7
Figure E-3. Servicing a Normal Channel E-8
Figure E-4. Worst Case Servicing of a Mid-range Channel E-8
Figure E-5. Worst Case Servicing of a Mid-range Channel E-10
Figure G-1. PCI Burst Write: Two 32-bit Fast Back-to-Back Transactions to Same Target G-1
Figure G-2. PCI Burst: Two 32-bit Transactions G-2

Tables

Table 1-1. Supported CX28560 Serial Port Modes . . . 1-2
Table 1-2. Allowed CX28560 Port Configurations . . . 1-3
Table 1-3. Data Path Configurations . . . 1-4
Table 1-4. Examples of Serial Port Configurations . . . 1-4
Table 1-5. Pin Summary . . . 1-17
Table 1-6. Serial Interface (General) . . . 1-18
Table 1-7. With DS0 extraction Mode Additional Pins (12 Ports Only) . . . 1-21
Table 1-8. CX28560 POS-PHY Interface (Transmit) . . . 1-21
Table 1-9. CX28560 POS-PHY Interface (Receive) . . . 1-23
Table 1-10. PCI Interface . . . 1-25
Table 1-11. EBUS Interface (Communication with Peripheral Components) . . . 1-27
Table 1-12. Boundary Scan and Test Access . . . 1-28
Table 1-13. Performance Monitoring . . . 1-28
Table 2-1. PCI Configuration Space . . . 2-6
Table 2-2. Register 0, Address 00h . . . 2-7
Table 2-3. Register 1, Address 04h . . . 2-8
Table 2-4. Register 2, Address 08h . . . 2-10
Table 2-5. Register 3, Address 0Ch . . . 2-10
Table 2-6. Register 4, Address 10h . . . 2-11
Table 2-7. Register 5–10, Address 14h–28h . . . 2-11
Table 2-8. Register 11, Address 2Ch . . . 2-11
Table 2-9. Register 12–14, Address 30h–38h . . . 2-11
Table 2-10. Register 15, Address 3Ch . . . 2-12
Table 3-1. EBUS Service Request Descriptor . . . 3-3
Table 3-2. EBUS Service Request Field Descriptions . . . 3-4
Table 5-1. PCI Register Map (Direct Access) . . . 5-3
Table 5-2. Indirect Register Map Address Accessible via Service Request Mechanism . . . 5-3
Table 5-3. Service Request Length Register . . . 5-5
Table 5-4. Service Request Pointer Register . . . 5-6
Table 5-5. Service Request Descriptor—OPCODE Description . . . 5-7
Table 5-6. Device Configuration Descriptor . . . 5-8
Table 5-7. DCD Field Descriptions . . . 5-8
Table 5-8. EBUS Configuration Service Request Descriptor . . . 5-8
Table 5-9. ECD Field Descriptions . . . 5-9
Table 5-10. Channel Configuration Service Request Descriptor . . . 5-9
Table 5-11. CCD Field Descriptions . . . 5-9
Table 5-12. Receive Port Alive Register . . . 5-10
Table 5-13. Transmit Port Alive Register . . . 5-10
Table 5-14. Interrupt Queue Pointer . . . 5-12

Table 5-15. Interrupt Queue Length 5-12
Table 5-16. BUFFC Interrupt Descriptors Format 5-14
Table 5-17. Non-BUFFC Interrupt Descriptors Format 5-17
Table 5-18. Interrupt Status Register 5-19
Table 5-19. Global Configuration Register 5-22
Table 5-20. EBUS Configuration Register 5-23
Table 5-21. Transmit POS-PHY Thresholds Register 5-24
Table 5-22. Transmit POS-PHY Control Register 5-24
Table 5-23. Receive POS-PHY Control Register 5-25
Table 5-24. RSLP Channel Status Register 5-26
Table 5-25. RSLP Channel Configuration Register 5-26
Table 5-26. Maximum Message Length Register 5-28
Table 5-27. RBUFFC Channel Configuration Register 5-29
Table 5-28. RBUFFC Flexiframe Memory 5-30
Table 5-29. RBUFFC Flexiframe Control Register 5-30
Table 5-30. RBUFFC Data FIFO Size Register 5-31
Table 5-31. RBUFFC Fragment Size Register 5-31
Table 5-32. RBUFFC Flexiframe Slot Time Register 5-31
Table 5-33. RBUFFC Counter Memory 5-32
Table 5-34. RSIU TS/Group Map 5-34
Table 5-35. RSIU Group Map 5-35
Table 5-36. RSIU Time Slot/Group Map Pointer Allocation Register 5-36
Table 5-37. RSIU Group Map Pointer Allocation Register 5-36
Table 5-38. RSIU Group State Register 5-37
Table 5-39. RSIU Port Configuration Register 5-37
Table 5-40. TSLP Channel Status Register 5-39
Table 5-41. TSLP Channel Configuration Register 5-40
Table 5-42. TBUFFC Channel Configuration Register 5-41
Table 5-43. TBUFFC Flexiframe Memory 5-42
Table 5-44. TBUFFC Flexiframe Control Register 5-43
Table 5-45. TBUFFC Data FIFO Size Register 5-44
Table 5-46. TBUFFC Flexiframe Slot Time Register 5-44
Table 5-47. TBUFFC Counter Memory 5-45
Table 5-48. TSIU TS/Group Map 5-47
Table 5-49. TSIU Group Map 5-48
Table 5-50. TSIU Time Slot/Group Map Pointer Allocation Register 5-49
Table 5-51. TSIU Group Map Pointers Register 5-50
Table 5-52. TSIU Group State Register 5-50
Table 5-53. TSIU Port Configuration Register 5-51
Table 5-54. CX28560 Receive Header Format 5-53
Table 5-55. CX28560 Transmit Data Header Format 5-54
Table 5-56. CX28560 Flow Conductor Packet Format 5-55
Table 8-1. Absolute Maximum Ratings 8-1
Table 8-2. Recommended 3.3 V Operating Conditions 8-2
Table 8-3. DC Characteristics for 3.3 V Operation 8-2
Table 8-4. PCI Interface DC Specifications 8-3

Table 8-5. PCI Clock (PCLK) Waveform Parameters, 3.3 V Clock 8-4
Table 8-6. PCI Reset Parameters 8-4
Table 8-7. PCI Input/Output Timing Parameters 8-5
Table 8-8. PCI I/O Measure Conditions 8-6
Table 8-9. Transmit Interface Timing 8-7
Table 8-10. Receive Interface Timing 8-9
Table 8-11. EBUS Reset Parameters 8-11
Table 8-12. EBUS Input/Output Timing Parameters 8-11
Table 8-13. EBUS Input/Output Measure Conditions 8-12
Table 8-14. Serial Interface Clock (RCLK, TCLK) Parameters 8-15
Table 8-15. Serial Interface Input/Output Timing Parameters 8-15
Table 8-16. Serial Interface Input/Output Measure Conditions 8-16
Table 8-17. Test and Diagnostic Interface Timing Requirements 8-20
Table 8-18. Test and Diagnostic Interface Switching Characteristics 8-20
Table 9-1. Pin List for 28560 HDLC Controller—Alphabetic Order (1 of 2) 9-2
Table 9-2. BGA Assignments for Power (Vddc, Vddo and Vgg) 9-4
Table 9-3. Signals (1 of 7) 9-5
Table A-1. Service Request Routine Field for Counter Read (Receive) A-5
Table A-2. Receive Counters in Shared Memory A-6
Table A-3. Service Request Routine Field for Counter Read (Transmit) A-6
Table A-4. Transmit Counters in Shared Memory A-7
Table B-1. The Flexiframe Structure B-4
Table B-2. Flexiframe Analysis Parameters B-9
Table D-1. System Side Interface: Payload Time Slot Bus D-3
Table D-2. System Side Interface: Overhead Time Slot Bus D-3
Table D-3. System Side Interface: Overhead Time Slot Bus Frame D-5
Table D-4. VSP Mapping of Intermixed Digital Level 2 Signals Containing Either DS1 or E1 Signals . . . D-6
Table E-1. Data/Status Combinations E-3
Table E-2. Servicing Sequence E-11
Table F-1. Little Endian F-1
Table F-2. Big Endian F-1

1.0 Introduction

The CX28560 is a 2047-channel communications controller targeted at synchronous link layer applications that provides a comprehensive, high density solution for HDLC internetworking applications.

- HDLC/SDLC
- LAPB, LAPD
- Digital Access Cross-Connect (DAC)
- Frame Relay Switches and Access Devices (FRAD)
- ISDN-D channel signaling
- X.25
- SMDS/ATM DXI
- LAN/WAN access data
- SONET/SDH add/drop multiplexers (ADMs)
- Terminal multiplexers (TMs)
- High Range/Gigabit Routers

The CX28560 HDLC controller interfaces to 32 independent serial data streams such as DS0, T1/E1, T3/E3, STS-1/STM-1. Data is transferred between the system and the CX28560 across a standard, high performance, 32-bit POS-PHY bus as fragments of complete packets. In the Transmit direction, the CX28560 provides the system with buffer state information across an 8-bit/100 MHz standard POS-PHY bus level 3 (FlowConductor™ bus).

1.1 External Interfaces

1.1.1 CX28560 Serial Interface

Each serial port can be configured to support different types of interfaces. Each of the CX28560's 32 full-duplex serial ports are individually programmable to operate as conventional or TSBUS serial ports.

The CX28560 supports five different operating modes for each of its serial ports (some limitations apply, see below): Conventional Unchannelized, Conventional Channelized, Conventional T1, TSBUS (no DS0 extraction), and TSBUS (with DS0 extraction). A brief description of each of these modes is listed in Table 1-1, or, for a fuller description, see Chapter 4.0.

1.1.1.1 CX28560 Serial Port Modes Description

In all conventional modes the Group Sync signals are ignored.

Table 1-1. Supported CX28560 Serial Port Modes (1 of 2)

CX28560 Serial Port Mode	Description
Conventional Unchannelized *(1)*	The serial input/output data stream is a bit stream without any framing or alignment. The bit stream belongs to a single logical channel. The CX28560 conventional unchannelized mode can be configured for all 32 serial ports. The first twelve serial ports can operate unchannelized T3/E3, HSSI, or STS-1/STM-1 bit stream up to 52 Mbps per serial interface (for reference, see Section 1.1.1.2).
Conventional Channelized *(1)(2)*	The serial bit stream is treated as a frame of N time slots (where N is ≤ 8192, given that other restrictions are met). The maximum bandwidth embedded into the PCM highway for the first thirteen ports is STS-1 rate (51.84 Mbps). The byte and frame synchronization performed is based on receive and transmit sync pulse (RSYNC and TSYNC). (For a detailed description of these signals, see Chapter 4.0.)
Conventional T1 mode *(1)*	The serial bit stream is treated as a frame of 24 time slots and the first bit of each T1 frame is discarded by the CX28560 hence, if the serial port is configured in T1 mode, the port operates according to the T1 framing definition.
TSBUS *(2)(3)* (No DS0 Extraction)	The TSBUS serial interface bit stream is treated as a frame of N time slots or variable bandwidth time slots called Virtual Serial Ports (VSPs) where N is defined as ≥ 5, and the aggregate number of time slots across all ports in any direction does not exceed the 8192 available time slots in each direction, (receive or transmit). Byte synchronization and frame synchronization is performed based on the TSBUS sync pulse TSTB (i.e., bus strobe). Mixed T1/E1 paths in one T3, mixed VT1.5/VT2 paths mapped to VTGs in one STS1, and mixed VC11/VC12 paths mapped to TUG2 in STM-1 are allowed using this serial port configuration. No DS0 extraction is performed, but separate logical channels can be configured within the frame.
TSBUS*(2)(4)* (With DS0 Extraction)	This mode is identical to TSBUS no DS0 extraction, except that further multiplexing can be performed by synchronizing T1/E1 frames within the STS-1 bit stream with the Group Sync pulse (RGSYNC and TGSYNC). According to this pulse, DS0 extraction is performed. Normally this mode will be used to extract DS0 signals from T1/E1 frames within a higher multiplexed hierarchy (see Appendix D for full explanation). Hyper channeling of channels within groups is possible. A minimum of 5 slots must be programmed per group.

Table 1-1. Supported CX28560 Serial Port Modes (2 of 2)

CX28560 Serial Port Mode	Description

Note(s):
1. A conventional serial port is defined as 7 input/output signals as follows: Transmit Clock (TCLK), Transmit Synchronization (TSYNC), Transmit Data (TDAT), Receive Clock (RCLK), Receive Synchronization (RSYNC), Receive Data (RDAT), Receive Out-Of-Frame, or Clear To Send (ROOF/ CTS). In conventional mode, the TGSYNC and RGSYNC signals are ignored. (For a detailed description of these signals, see Section 4.2.)
2. Channelized mode refers to a data bit stream segmented into frames. Each frame consists of a series of 8-bit time slots. The frame synchronization is maintained in both the transmit and receive direction by using the Transmit Synchronization (TSYNC) and Receive Synchronization (RSYNC) input signals.
3. An Time Slot Bus (TSBUS) (no DS0 extraction) is defined as 7 input/output signals as follows: Transmit Clock (TCLK), Transmit Stuff (TSTUFF), Transmit Data (TDAT), Transmit Strobe (TSTB), Receive Clock (RCLK), Receive Stuff (RSTUFF), Receive Data (RDAT). (For a detailed description of these signals, see Chapter 4.0).
4. A Time Slot Bus (TSBUS) (with DS0 extraction) is defined as 9 input/output signals: Transmit Clock (TCLK), Transmit Stuff (TSTUFF), Transmit Data (TDAT), Transmit Strobe (TSTB), Receive Clock (RCLK), Receive Stuff (RSTUFF), Receive Data (RDAT), Receive Group Sync (RGSYNC), Transmit Group Sync (TGSYNC). For a detailed description of these signals, see Chapter 4.0.

1.1.1.2 CX28560 Serial Port Throughput Limits

Each of the CX28560 serial ports can be configured to operate in any of the preceding operational modes. The following restrictions apply:

- The overall number of time slots cannot exceed 8192.
- The overall number of channels cannot exceed 2047.
- The total accumulated speed of all serial port clocks in either receive or transmit direction must be less than 800 MHz.
- The aggregated serial port data bandwidth cannot exceed 700 Mbps in each direction.
- The maximum speed per port is less than 52 Mbps (HSSI) where the maximum number of high speed ports is 12.
- The maximum number of ports that can run 44.7 Mbps (T3) is 15.
- The maximum number of ports that can run 32.768 Mbps (E3) 21.

Allowed CX28560 port configurations are listed in Table 1-2.

Table 1-2. Allowed CX28560 Port Configurations

Speed of Port	4 Mbps	8 Mbps	13 Mbps	32.8 Mbps	44.7 Mbps	52 Mbps
Number of Ports	32	32	32	21*(1)*	15*(1)*	12*(1)*

Note(s):
(1) For high speed ports such as DS3 (44.736 Mbps), E3 (34.368 Mbps), and HSSI (52 Mbps), the data path of the remaining ports may operate according to the CX28560's bandwidth restrictions.

Table 1-3. Data Path Configurations

Speed of Port (Mbps)	Low Speed Ports				High Speed Ports			Aggregated Port Speed
	4	8	10	13	32	44	52	
Allowed Number of Ports	—	—	—	12	—	—	12	780[1]
	—	—	7	—	—	—	12	694
	—	9	—	—	—	—	12	696
	19	—	—	—	—	—	12	700
	—	—	17	—	—	12	—	698
	—	20	—	—	—	12	—	688
	20	—	—	—	—	12	—	608
	—	—	20	—	12	—	—	584
	—	20	—	—	12	—	—	544
	20	—	—	—	12	—	—	464

The CX28560's configuration options are extremely flexible. Each logical channel can be assigned to a physical stream ranging from a DS-0 (64 Kbps) to 52 Mbps. The CX28560's serial ports can interface to a standard PCM highway or TSBUS. Examples of configurations are listed in Table 1-4.

Table 1-4. Examples of Serial Port Configurations

Configuration	Total Bit Rate
15 x T3	671.040 Mbps
21 x E3	688.128 Mbps
12 x 52 + 12 x 12.96 (SONET)	622.080 Mbps[1]
32 x 12*T1	592.896 Mbps
Maximum	700 Mbps

Note(s):
[1] The guaranteed total bit rate of the channel (622.080 Mbps) is lower than the aggregate port rate (780 MHz). This is an example of a TSBUS application where stuffing would guarantee the bit rate to be within the maximum tolerated by the CX28560.

The send and receive data can be formatted in the HDLC messages or left unformatted (transparent mode) over any combination of bits within a selected time slot. The CX28560's protocol message type is specified on a per-channel basis.

1.1.1.3 TSBUS—Time Slot Bus

The CX28560 provides a TSBUS interface for variable bandwidth time slots, Virtual Serial Port (VSP). A VSP is defined as an entity—quantified by clock bus rate divided by number of time slots—which provides multiple asynchronous paths over a single serial port. A programmable number of VSPs per TSBUS are allowed by using the existing start and end address time slot pointer mechanism. This mechanism allows the CX28560 to allocate any number of VSPs on a given serial port.

The total number of time slots allocated across all ports must not exceed 8192, the total number of logical channels must not exceed 2047, and the serial port clock speed must not exceed 52 MHz.

While operating in TSBUS mode, the minimum number of time slots required is 5. The programmable number of time slots, implemented by the pointer mechanism (i.e., configurable start and end addresses) allows any number of time slots to be concatenated into a single logical channel. This concatenation allows mixed VTG path options without changing the number of time slots assigned to the TSBUS port. The stuff signal provides the CX28560 with the information necessary to pad time slots in the transmit direction, or to ignore them in the receive.

When working in TSBUS mode, DS0 signals can be extracted from a higher level (SONET/SDH/DS3 or E3) payload bit stream. This is performed in a similar manner to the TSBUS frame mechanism, by using a group synchronizing pulse and a pointer mechanism. Each group occupies fixed places in the time slot map. When this position is reached, a group time slot map is consulted in order to retrieve the relevant channel number. (See Appendix D for a detailed description).

1.2 System-Side Interfaces

1.2.1 POS-PHY Interfaces

1.2.1.1 POS-PHY Data Interface—CX28560

Data is transferred between the System and the CX28560 over a bidirectional standard POS-PHY level 3 100 MHz, 32-bit interface, working in packet mode. The data is transferred as fragments accompanied by a 4-byte fragment header.

Data transferred on this POS-PHY interface is in fragments of a user- configurable length (minimum 32 bytes, maximum 256 bytes) together with a fragment header (4 bytes).

As the fragment length increases, the number of configurable channels decreases. When 56-byte fragments are used, up to 2047 channels may be configured; when 112-byte fragments are configured, a maximum of 1024 channels may be configured. See Appendix I for an explanation of the relationship between fragment length, number of channels, and the channels' bandwidths.

The last fragment of each message is marked as an End Of Message (EOM) fragment. The next message starts with the fragment immediately following an EOM fragment.

The receive fragment header contains the following fields:

- Channel Number
- Fragment Length
- End of Message Indicator
- Beginning of Message Indicator
- Message Status

The transmit fragment header contains the following fields:

- Channel Number
- Command Valid
- Idle Code (IC) select (HDLC Flags/Aborts, All Zeros) for padding between messages
- Pad Count (PADCNT) minimum number of idle codes to be inserted after message
- Abort Command

For fragment header formats see Chapter 5.0, POS-PHY transaction headers for full header layout.

1.2.1.2 Transmit FlowConductor Interface

The CX28560 provides the system with information necessary to calculate free buffer space available in the CX28560's transmit buffers. This information is reported in packets of a specified format (see Table 5-56, *CX28560 Flow Conductor Packet Format*, transmitted over an 8-bit, 100 MHz, standard, level 3, unidirectional, POS-PHY dedicated feedback interface. The format of the report is a counter based system whereby the system receives information regarding the amount of data transmitted since the previous report, and keeps track of the amount of space presently available in the channel's transmit buffer. The packets received by the system contain the channel number, and the amount of space freed since the previous report for that channel was sent. (See Appendix C for flow conductor interface).

The system should be able to respond, if necessary, to the reports within 5 µs. This interface may be fully utilized. See Table 5-56, *CX28560 Flow Conductor Packet Format*.

1.2.2 Expansion Bus (EBUS)

CX28560 provides an access to a local 32-bit bus interface called the Expansion Bus (EBUS), which provides a host processor access to any address in the peripheral memory space on the EBUS. Although EBUS use is optional, the most notable applications for EBUS are connections to peripheral devices, such as Bt8370/Bt8398 T1/E1 framers, CX28398 (Octal DS1/E1 framers), and CX29503/M29513 BAM3/BAM3+ and CX29610 OptiPhy that are local to the CX28560's serial port.

The CX28560 provides access to the EBUS through an interface similar to a mailbox interface. This interface provides all the EBUS read and write accesses to be carried over the PCI bus allowing bursts. This mechanism improves the PCI use when multiple EBUS accesses are needed for accessing the configuration of the peripheral devices.

The EBUS may be configured to use the Intel- or Motorola-style using a special bit in the EBUS configuration register within the Host Service Unit (HSU). The EBUS also supports slow devices by allowing the address/data to be transferred over multiple cycles and thus allowing slow devices to read the address, access the data, or write it into their memories.

1.2.3 PCI Bus Interface

A standard PCI 2.2 bus interface is provided to the system as an interface for configuring and reading the CX28560 registers, counters, and interrupts.

The CX28560 acts as a PCI master. Configuration reads and writes and other commands are written by the system to the shared memory. These commands are retrieved via the CX28560's HSU block, and passed internally to the relevant block to be performed. Interrupts collated by the CX28560 are written to a user-configurable address in the shared memory for collection by the system.

1.3 Feature Summary

The following is a summary of the features provided by the CX28560:

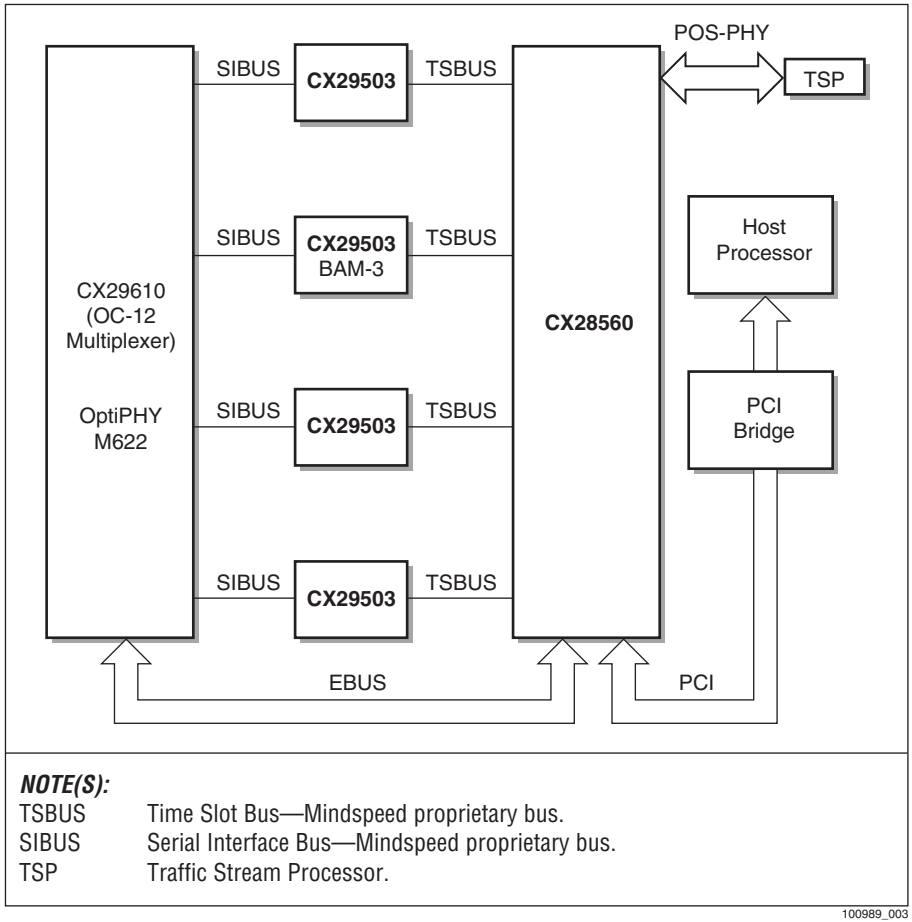
- 2047-channel, full-duplex link layer controller for synchronous applications.
- 32 full-duplex physical interfaces (i.e., ports) with independent clock rates.
  - The CX28560 implements 32 serial ports that are individually programmable to operate either as conventional serial ports or TSBUS serial ports. Both operational modes allow the input/output data stream to be configured as channelized or unchannelized bit streams. Clock rates may be as high as 52 MHz.
- General purpose HDLC (ISO 3309) is supported.
  - HDLC/SDLC
  - HSSI
  - ISDN D-channel (LAPD/Q.921)
  - X.25 (LAPB)
  - Frame Relay (LAPF/ANSI T1.618)
  - Inter-System Link Protocol (ISLP) support
  - LAPDm support
  - ATM/SMDS DXI
  - Transparent unformatted mode
  - Point-to-Point-Protocol (PPP)
  - Multi-Link-Point-to-Point-Protocol (MLPPP)
- Hyper-channels and sub-channels are supported.
  - Applications that require hyper-channeling:
    - ISDN Primary Rate Interface (PRI)
    - ISDN Primary Rate Adapter (PRA)
    - Fractional T1 (FT1)
    - Fractional E1 (FE1)
    - Fractional Nx64K
    - SONET/SDH/PDH paths connected via TSBUS:
      Mixed VT1.5/VT2 paths
      Mixed TU-11/TU-12 paths
      Mixed T1/E1 paths
    - Multiple lines multiplexed to 1 port:
      Digital Subscriber Line Access Multiplexer (DSLAM)
      T1/E1 Frame Relay
  - Sub-channeling mode
    - Each channel can be programmed to either use a complete DS-0 time slot or mask any subset of a time slot. A signal mask is defined per-channel basis that has an enabled bit per time slot. The mask bit dictates whether the whole DS-0 or part of it is enabled.
    - Applications that require sub-channeling:
      ISDN Basic Rate Interface (BRI)
      ISDN Basic Rate Adapter (BRA)
      Frame Relay 56K and Nx56K
      Compressed Voice Transparent Channels (e.g., ADPCM)
      Centralized Signaling Channel Controllers:
      Link Access Procedure D-Channel (LAPD)

- Unchannelized mode
  - Applications that require unchannelized ports:
    - Digital Comm/Termination Equipment (DCE/DTE) Interfaces
    - High Speed Serial Interface (HSSI)
    - Inter-Process Communication (IPC)
    - V-Series DTE/DCE Interfaces (V.35)
    - SDSL Modems and Access Concentrators
    - T3/E3 Frame Relay
- Variable path primitives are supported.
  - Path Payload
    - DS-0 (64 Kbps)
    - Nx64, where N is defined as any number between 1– 810, allows all types of hyper-channeling, channelized, unchannelized, or path payload
  - Higher speed ports
    - Unchannelized DS3 (44.736 Mbps)
    - Unchannelized E3 (34.368 Mbps)
    - HSSI (52 Mbps)
  - Path overhead (Performance Monitoring and Provisioning)
    - T1/E1
    - Facilities Data Link (FDL)
    - Common Channel Signaling (CCS)
    - T3/E3 Terminal Data Link (TDL)
    - V.51 and V.52 signaling channels
- Per-channel protocol selection is supported.
  - Non-FCS mode
  - 16-bit FCS mode
  - 32-bit FCS mode
  - Transparent mode
- Dynamically configurable logical channels are supported.
  - Standard DS0
  - Hyper-channel
  - Sub-channel
- Programmable time slot allocation is supported.
  - Pointer mechanism
- Per-channel BUFFC buffer management is supported.
  - Internal FIFO of size 352 KB in the transmit direction and 320 KB in the receive direction.
  - One size of channel FIFO required regardless of channel bit rate (allows for dynamic reconfiguration of specific channels without full chip reset)
  - Configurable BUFFC threshold set on a per-channel basis in the transmit direction
  - Programmable FIFO size per-chip basis
- Clear to Send (CTS) per-channel control of data transmission in conventional mode ports.
- Direct POS-PHY bus interface is supported.
- PCI Bus Interface (rev. 2.2) for configuration purposes only.
  - 32-bit multiplexed Address/Data bus minimizes pin count
  - 33 MHz operation

- EBUS—Expansion Bus Interface is provided.
  - 32-bit multiplexed Address/Data
  - Allows Host to control other local devices
  - Facilitates Host access to any local memory
- TSBUS interface.
  - Variable bandwidth time slot
  - DS0 level extraction and synchronization
  - Multiple asynchronous paths over single port
  - Allows SONET/SDH/PDH paths connection
    - Mixed VT1.5/VT2 paths
    - Mixed TU-11/TU-12 paths
    - Mixed T1/E1 paths
- Full set of Performance Monitoring counters provided
  - Receive direction:
    - Octets
    - Packets
    - Packets with alignment errors
    - Packets with too short errors
    - Packets with too long errors
    - Packets with FCS errors
    - Packets terminating in an abort
  - Transmit direction:
    - Octets
    - Packets
    - Packets transmitted terminating in an abort signal
- 3.3 V/1.8 V supply; 5 V-tolerant inputs
- JTAG access is provided
- Low power CMOS technology is used
- 1 A at 1.8 V, 0.34 A at 3.3 V (according to preliminary mini-characterization)

1.4 Applications Examples

Figure 1-1. OC-12 Application



NOTE(S):

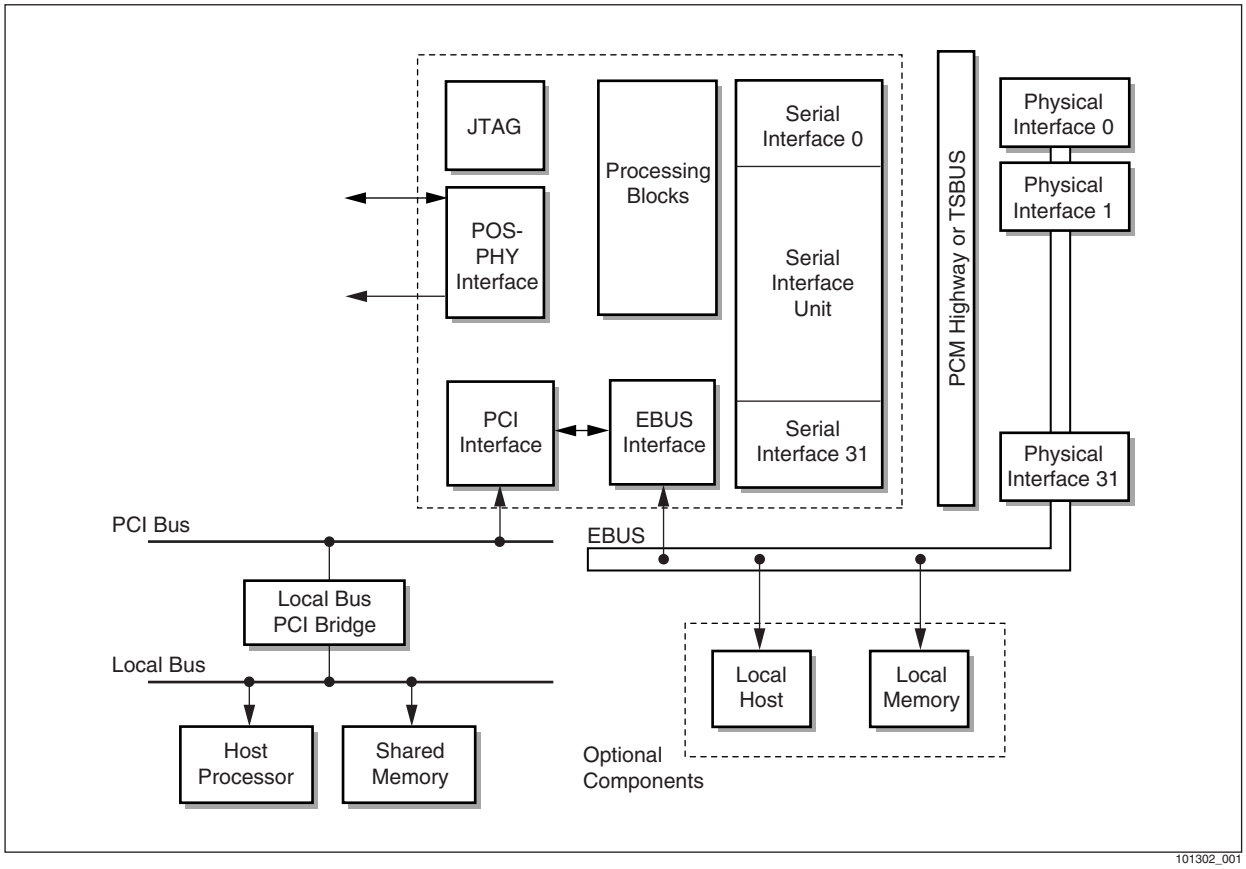
TSBUS	Time Slot Bus—Mindspeed proprietary bus.
SIBUS	Serial Interface Bus—Mindspeed proprietary bus.
TSP	Traffic Stream Processor.

100989_003

1.5 System Overview

CX28560 supports 32 fully independent serial ports that can be configured to run in channelized, unchannelized, T1 or TSBUS (with or without DS0 extraction) mode. For example, in the channelized mode, the first 12 ports can operate up to 51.81 Mbps (STS-1 rate) while the another 20 ports can operate up to 12.96 Mbps (maximum SONET overhead rate). Each STS-1 frame transports 28xT1, 1xT3, 21xE1 or mixed T1/E1 VTG paths, while the overhead bit streams contain the overhead required. The configuration is valid as long as the overall number of time slots per the whole device is 8192 time slots or less. For other restrictions see Section 1.1.1.2. Alternatively, any of the CX28560's ports can interface unchannelized data streams (HDLC or unformatted). In this mode, each of the first twelve ports can be configured to operate up to 52 Mbps. The restriction is that the overall data bandwidth must not exceed 700 Mbps (full duplex). The CX28560 manages uniformly allocated buffer memory according to the Flexiframe algorithm for each of the active data channel, allowing the user to reconfigure any channel without affecting other active channels. The on-device features allow data transmission between buffer memory and the serial interfaces with minimum host processor intervention. This allows the host processor to concentrate on managing the higher layers of the protocol stack.

Figure 1-2. System Overview



The TSBUS interface provides multiple asynchronous paths (all TSBUS frames run at 51.84 Mbps).

The supported TSBUS framer configurations that are mapped into and from VSPs are as follows:

- There are 28 PDH framers and 28 SONET/SDH framers. Either of these two categories of framers can be mapped directly to the VSPs for a given configuration.
- When using the PDH framers, each framer can be independently configured as a DS1 framer or as an E1 framer.
- The SONET/SDH category of framers has two subcategories as the name implies. When using the SONET/SDH framers, they all have to be configured as either SONET framers (one subcategory) or as SDH framers (the other subcategory). The configurations of the SONET/SDH framers When configuring for SONET subcategory of framers, each framer can be independently configured as a VT1.5 framer or as a VT2.0 framer. The SDH subcategory of framers is similarly configured. Each SDH framer can be independently configured as either a C11 framer or an E1 framer.

The supported TSBUS frame structures are DS0s, DS1s, E1s mapped via DS2, VT1.5, VT2.0, C11, and C12.

The following mixed mappings are also supported by selectively configuring each framer:

- DS0s extracted from mixed DS1s via the TSBUS
- DS1s and E1s extracted from mixed DS2s via the PDH framers
- DS1s and E1s extracted from mixed VTGs via the SONET framers
- C11s and C12s extracted from mixed TUG-2s via the SDH framers

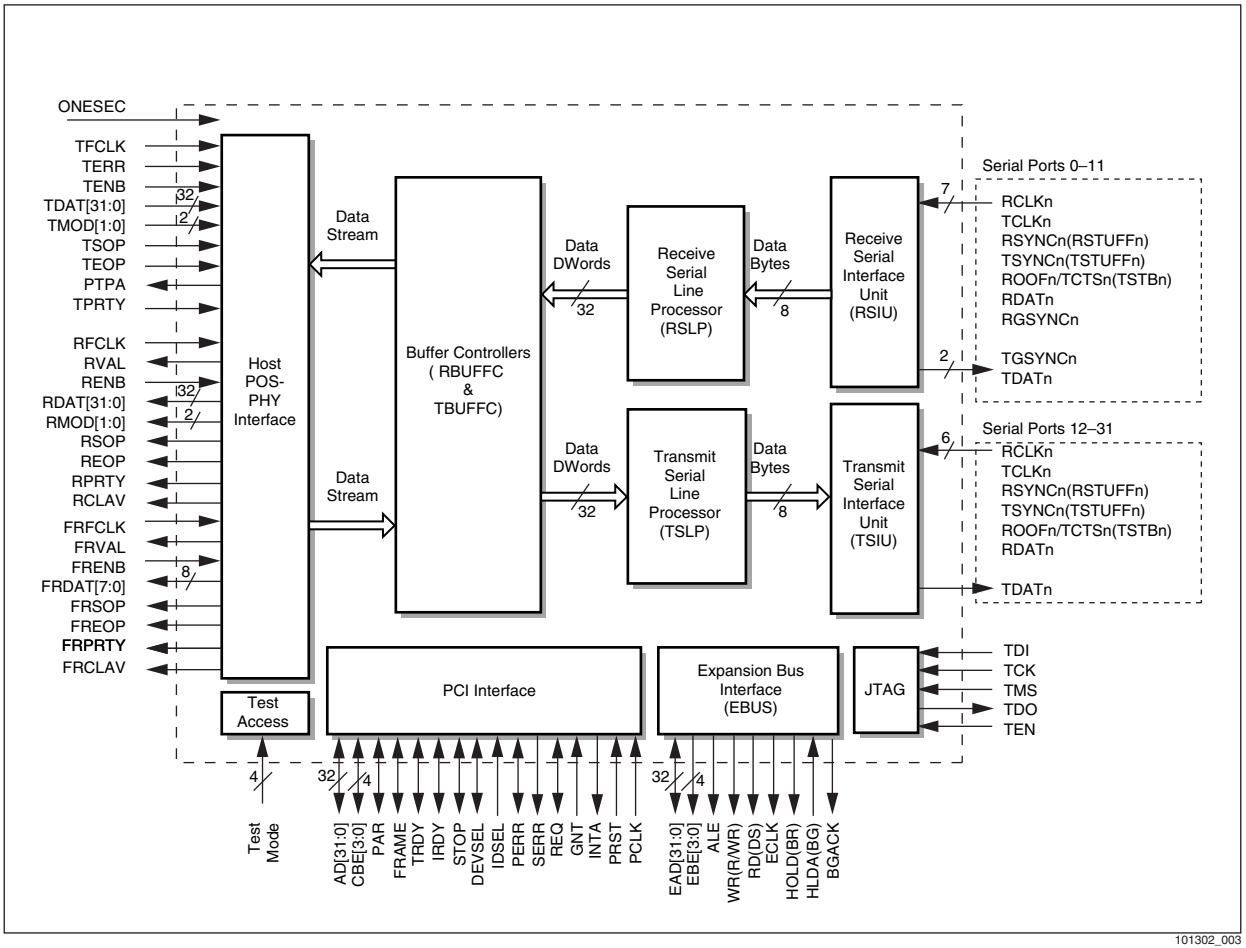
The frame structure is designed to transport the

- unchannelized STS-1 Synchronous Payload Envelope (SPE)
- unchannelized DS3 payload
- 16 E1 signals that are mapped to and from E3

1.6 Block Diagram

Figure 1-3 illustrates the CX28560 conceptual block diagram.

Figure 1-3. CX28560 Block Diagram



The following is a description of the block diagram.

- Host Interface (POS-PHY): This block provides the communication path of the data between the host and the CX28560.
- PCI Host Interface: This block interfaces to the PCI bus over which the host configures and monitors the CX28560 action.
- Expansion Bus (EBUS): The EBUS is an extension of the PCI Host Interface, which provides host with access to control other devices on the local PC board.
- Serial Interface Unit (SIU): This block provides the interface between 32 serial ports and the Receive and Transmit Serial Line Processors block. A temporal buffering space is provided by the SIU that is 56 bits per port, divided as 32 bits (4 bytes) for the transmit direction and 24 bits (3 bytes) for the receive direction. SIU controls the data access to the Rx and Tx Serial Line Processors. Because the CX28560 supports two types of serial ports—one is the conventional interface, the other TSBUS interface—the SIU needs to operate depending on serial port type (for detailed descriptor information, see Chapter 4.0).
- Transmit Serial Line Processor (TSLP): This block provides the interface between the Buffer Controller (BUFFC) and the TSIU. Data provided by the BUFFC is processed by the TSLP according to the channel type and passed to the TSIU for transmission to the line.
- Receive Serial Line Processor (RSLP): This block provides the interface between the SIU and BUFFC. The data provided by RSIU is processed by RSLP according to the channel type before it is transferred to the BUFFC.
- BUFFC: This block provides the interface between the host and the Transmit and Receive Serial Line Processors (TSLP and RSLP). The BUFFC contains the main storage of data—a dual port RAM of 352 KB in the transmit direction and 320 KB in the receive direction. This space acts as a holding buffer for incoming (Rx) and outgoing (Tx) data.
- JTAG: This is a special test port used for serial boundary scan on a PCB, as well as access to internal scan paths and embedded memory for test.
- Onesec: the onesec signal provides the boundaries on which the performance monitoring counters are latched.

1.7 Data Flow

1.7.1 Receive Data Path

Data enters the CX28560 via 32 independently configurable ports. Within the CX28560 data is processed to remove HDLC formatting and to perform message error detection (e.g., FCS error, alignment error, etc.), and concatenated to fixed length chunks (of user-configurable length) that are then transferred to the system. The internal memory required per channel in the CX28560 is constant, regardless of the channel's rate. This is achievable due to the Flexiframe algorithm (see Appendix B).

The Flexiframe method provides a fixed order (frame) for servicing the CX28560 channel internal buffers. The frame structure allocates service time to each channel proportionally to its bit rate. The CX28560 runs through the frame and decides whether or not the next channel in the frame requires servicing (i.e., whether it contains an EOM or enough data to fill a standard fragment). If so, the data is passed to the POS-PHY interface and is transmitted to the system.

1.7.2 Transmit Data Path

The system stores the data until a report is received from the CX28560 via the Flow Conductor bus. On receiving the report, the system calculates whether or not there is room in the channel's CX28560 internal buffer. If so, data is transferred to the CX28560 in complete packets over the 32 bit POS-PHY interface.

The CX28560 processes the data received and outputs HDLC formatted data. According to the Flexiframe algorithm, reports of the amount of buffer freed are sent to the system over the dedicated 8-bit unidirectional Flow Conductor bus. According to the reports received, the system provides fixed size fragments of data over the 32-bit bidirectional POS-PHY bus. See Appendix B and Appendix C for more details.

Formatted, masked, and time slot ordered data is transmitted to the line from the CX28560 via 32 independently configurable ports.

1.8 CX28560 Pin List

1.8.1 Pin Descriptions

Table 1-5 lists the pin summary.

Table 1-5. Pin Summary

Interface	In	Out	I/O	Total	Table
Serial	20 × 6	20 × 1	0	140	Table 1-6, *Serial Interface (General)*
TSBUS (DS0)	12 × 7	12 × 2	0	108	Table 1-7, *With DS0 extraction Mode Additional Pins (12 Ports Only)*
POS-PHY Transmit	40	1	0	41	Table 1-8, *CX28560 POS-PHY Interface (Transmit)*
POS-PHY Receive	2	39	0	41	Table 1-9, *CX28560 POS-PHY Interface (Receive)*
POS-PHY FlowConductor	2	13	0	15	Table 1-9, *CX28560 POS-PHY Interface (Receive)*
PCI	4	0	46	50	Table 1-10, *PCI Interface*
EBUS	1	10	32	43	Table 1-11, *EBUS Interface (Communication with Peripheral Components)*
JTAG	4	1	0	5	Table 1-12, *Boundary Scan and Test Access*
TEST	4	0	0	4	Table 1-12, *Boundary Scan and Test Access*
ONESEC	1	0	0	1	Table 1-13, *Performance Monitoring*
Total	262	108	78	448	—

Note(s):
The SERR and INTA signals are indicated above as Output pins. They are implemented as I/O cells due to design considerations.

Table 1-6 lists pins common to all ports.

Table 1-6. Serial Interface (General) (1 of 3)

Pin Name	I/O	Ref Clk	Description
TCLK[31:0]	I	—	Transmit Clock (TCLK[31:0]. If the serial port is configured in conventional mode, TCLKx controls the rate at which data is transmitted and synchronizes transitions for TDATx and sampling of TSYNCx. If the port is configured as a TSBUS port, TCLKx controls the rate at which data is transmitted and synchronizes transitions for TDATx and sampling of TSTUFFx and TSTBx (TSTBx only for transmit circuitry). If in TSBUS with DS0 extraction mode in addition to the above, TCLKx also synchronizes transitions of TGSYNCx.
TSYNC[31:0]/ TSTUFF[31:0]	I	TCLK[31:0]	Transmit Synchronization/TSBUS Transmit Stuff (TSYNCx[31:0]/TSTUFF[31:0]). If the serial port is configured in conventional mode, this signal is defined as TSYNCx. TSYNCx is sampled on the specified active edge of the corresponding TCLKx clock. When TSYNCx signal goes from low to high, the start of transmit frame is indicated. TSYNCx is ignored if the serial port is configured to operate in conventional unchannelized mode. If the serial port is configured in T1 mode, the corresponding data bit that latched out during the same bit time period (but not necessarily sampled at the same clock edge) is the F-bit of the T1 frame. If the serial port is configured in conventional channelized mode, the corresponding data bit that latched out during the same bit time period (but not necessarily sampled at the same clock Edge) is bit 0 of the first time slot of the N.... 64 frame. Because the CX28560's flywheel mechanism is always used in channelized mode, no other synchronization signal is required to track the start of each subsequent frame. If the port is configured to operate as TSBUS port, this signal is defined as TSTUFF. The TSTUFF values are to either stuff (no TDAT output) or not stuff (TDAT valid). TSTUFF is sampled on the specified active edge of the corresponding TCLKx. If the serial port operates in TSBUS mode, TSTUFF assertion indicates that no data needs to be transmitted in the 8^{th} time slot after the assertion of the TSTUFF. While operating in TSBUS mode, the CX28560 requires the following: The stuff status for each time slot to be presented at its TSTUFF input exactly eight time slots in advance of the actual time slot for which the stuff status is to be applied. The amount of the TSTUFF advance is fixed at eight time slots, even though the number of time slots within a frame may vary. The CX28560 expects assertion of this signal within the first two bits of the time slot. Assertion of this signal elsewhere in the time slot might result in undefined behavior.
TDAT[31:0]	O	TCLK[31:0]	Transmit Data (TDAT[31:0]) Serial data latched out on active edge of transmit clock, TCLKx. If channel is unmapped to time slot, data bit is considered invalid and the CX28560 outputs either three-state signal or logic 1 (user-configurable, see Table 5-53, *TSIU Port Configuration Register*, field TRITX).

Table 1-6. Serial Interface (General) (2 of 3)

Pin Name	I/O	Ref Clk	Description
ROOF[31:0]/ CTS [31:0]/ TSTB[31:0]	I	TCLK[31:0]/ RCLK[31:0]	This pin has three separate definitions. Control of the use of the pin is configured in the *Table 5-39, RSIU Port Configuration Register* and *Table 5-53, TSIU Port Configuration Register using the CTSENB and ROOFABT fields.* *(2)(3)(5)* Receiver Out-Of-Frame ROOF[31:0]. If the pin is configured to be ROOFx, it is sampled on the configured active edge of the corresponding receive clock RCLKx. If ROOFx signal performs a transition from low to high (assertion), an Out-Of-Frame (OOF) condition interrupt is generated if the interrupt is enabled. While ROOFx is asserted, the received serial data stream is considered Out-Of-Frame. If OOFABT bit field is configured to 1, the receive channel processing is disabled for the entire port and it remains disabled until ROOFx is deasserted; otherwise, the receive channel processing is enabled. Upon ROOFx deassertion, if OOFIEN bit field is set to 1, an interrupt Frame Recovery (FREC) is generated. The data processing resumes for all affected channels. General Interrupt Line. This signal can also operate as a general Serial Port Interrupt (SPORT) by clearing the OOFABT bit field and setting the OOFIEN bit field (i.e., OOFABT = 0 and OOFIEN = 1). When the ROOFx signal transitions from high-to-low (deassertion), a SPORT interrupt is generated and data stream is not affected. If this signal is used as a general purpose interrupt, no interrupt is generated until this signal goes from high to low. *(2)(3)(5)* Channel Clear To Send (CTS[31:0]). If CTSx, the signal is sampled on the specified active edge of the corresponding transmit clock, TCLKx. If CTS transitions from high-to-low (is deasserted), the channel assigned to the time slot sends continuous idle characters after the current message has been completely transmitted. The message transmission data restarts when this CTS transitions from low to high again (is asserted). The response time to CTS is a 32 bit-time, meaning that a new message might be transmitted if the message starts within the next 32 bits after CTS was deasserted. *(2)(4)(5)* TSBUS Strobe (TSTB[31:0]) If the port is configured in TSBUS mode the this pin is used as TSTBx. The signal is sampled twice, once by the receive circuitry on the specified edge of the corresponding receive clock, RCLKx, and once by the transmit circuitry on the specified edge of the corresponding transmit clock, TCLKx. If TSTB transitions from low to high, it marks the first bit of time slot 0 within the TSBUS frame. Because there is a single TSTB for both directions, receive and transmit, the number of configured time slots and the RPORT_TYPE or TPORT_TYPE value specifying whether the serial port operates in TSBUS or non-TSBUS mode must be identically configured for both directions per serial port. Unexpected CX28560 behavior may be generated if this restriction is violated.
RCLK[31:0]	I	—	Receive Clock (RCLK[31:0]). This clock controls the rate at which data is received and synchronizes sampling of RDATx, RSYNCx (non-TSBUS mode only), RSTUFFx (TSBUS mode only), and TSTBx (TSBUS mode only, and only for receive path circuitry). If in TSBUS (with DS0 extraction) mode, RCLKx also synchronizes transitions of RGSYNCx.

Table 1-6. Serial Interface (General) (3 of 3)

Pin Name	I/O	Ref Clk	Description
RSYNC[31:0]/ RSTUFF[31:0]	I	RCLK[31:0]	Receive Synchronization/Receive Stuff (RSYNC[31:0]/ RSTUFF[31:0]). If the port operates in a conventional mode, this signal is defined as RSYNC. RSYNC is sampled on the configured active edge of the corresponding receive clock, RCLKx. RSYNCx is ignored if the serial port is configured to operate in unchannelized mode. If RSYNCx signal transitions from low to high, the start of a receive frame is indicated. For T1 mode, the corresponding sampled and stored data bit during the same bit-time period (not necessarily sampled on the same clock edge) is the F-bit. For the conventional channelized mode, the corresponding data bit sampled and stored during the same bit time period (not necessarily sampled on the same clock edge) is bit 0 of the first time slot of the N.... 64 frame. RSYNCx must remain asserted high for a minimum of PCI setup and hold time relative to the active clock edge of this signal. Since the CX28560's flywheel mechanism is always used in channelized mode, no other synchronization signal is required to track the start of each subsequent frame. If the port operates as a TSBUS port, this signal is RSTUFF. The RSTUFF is sampled on the configured active edge of the corresponding RCLKx. In this case, RSTUFF assertion indicates that this time slot contains no data. While operating in channelized mode, the CX28560 expects assertion of this signal within the first two bits of the time slot. Assertion of this signal elsewhere in the time slot might result in undefined behavior.
RDAT[31:0]	I	RCLK[31:0]	Receive Data (RDAT[31:0]) Serial data sampled on active edge of receive clock, RCLKx. If the channel is mapped to a time slot, input bit is sampled and transferred to memory. If the channel is unmapped to time slot, data bit is considered invalid and the CX28560 ignores the received sample.

Note(s):
1. While operating in TSBUS mode, there is no damage expected when sampling TSTBx twice, because the RCLKx and TCLKx are the same signals for a specific port. However, this may require some additional restrictions for the board designers when these clocks are routed.
2. This signal is used either as Receiver Out-Of-Frame or a Transmit Clear to Send or a TSBUS strobe. (OOF/FREC behavior selected by OOFABT = 1, CTS behavior selected by CTSENB = 1, TSTB behavior selected by TPORT_TYPE or RPORT_TYPE).
3. If the serial port operates in conventional mode, this signal is used either as a ROOFx or CTSx signal.
4. If the port operates in DS0 extraction mode, the signal is used as the TSBUS strobe signal, which indicates the beginning of the TSBUS frame.
5. Only one pin in the device defines all these functions.

Table 1-7. With DS0 extraction Mode Additional Pins (12 Ports Only)

Pin Name	I/O	Ref Clk	Description
TGSYNC[11:0]	O	TCLK[11:0]	Payload Time Slot Bus Transmit DS0 Sync (TGSYNC). When high, indicates that data on TDATA is the first bit of the first group configured for the port of DS0 valid data.
RGSYNC[11:0]	I	RCLK[11:0]	Payload Time Slot Bus Receive DS0 Sync (RGSYNC). When high, indicates that data on RDATA is the first bit of the first group configured for the port of DS0 valid data.

Table 1-8 describes data transfer from the system to the CX28560.

Table 1-8. CX28560 POS-PHY Interface (Transmit) (1 of 2)

Pin Name	I/O	Ref Clk	Description
TFCLK	I	—	Transmit FIFO Write Clock. TFCLK synchronizes data transfer transactions between the system and the CX28560. TFCLK cycles at a rate of 100 MHz. Other signals are sampled on the rising edge of this signal.
TERR	I	TFCLK	Transmit Error Indicator signal. TERR indicates that the current packet should be aborted. When TERR is set high, the current packet is aborted. TERR should only be asserted when TEOP is asserted.
TENB	I	TFCLK	Transmit Write Enable (TENB) signal. The TENB signal controls the flow of data to the transmit FIFOs. When TENB is high, the TDAT, TMOD, TSOP, TEOP, and TERR signals are invalid and are ignored by the CX28560. When TENB is low, the TDAT, TMOD, TSOP, TEOP, and TERR signals are valid and are processed by the CX28560.
TDAT[31:0]	I	TFCLK	Transmit Packet Data Bus. This bus carries the packet octets that are written to the CX28560's FIFO. The TDAT bus is considered valid only when TENB is asserted. Data must be transmitted in big endian order on TDAT[31:0]. In accordance with the HDLC protocol, bit 0 of each byte is transmitted first.
TMOD[1:0]	I	TFCLK	Transmit Word Modulo signal. TMOD[1:0] indicates the number of valid bytes of data in TDAT[31:0]. The TMOD[1:0] bus should always be all zero, except during the last double-word transfer of a packet on TDAT[31:0]. When TEOP is asserted, the number of valid packet data bytes on TDAT[31:0] is specified by TMOD[1:0]. TMOD[1:0] = 00 TDAT[31:0] valid TMOD[1:0] = 01 TDAT[31:8] valid TMOD[1:0] = 10 TDAT[31:16] valid TMOD[1:0] = 11 TDAT[31:24] valid
TSOP	I	TFCLK	Transmit Start of Packet (TSOP) signal. TSOP delineates the packet boundaries on the TDAT bus. When TSOP is high, the start of the packet is present on the TDAT bus. TSOP must be present at the beginning of every packet and is considered valid only when TENB is asserted.
TEOP	I	TFCLK	Transmit End of Packet (TEOP) signal. TEOP delineates the packet boundaries on the TDAT bus. When TEOP is high, the end of the packet is present on the TDAT bus. TMOD[1:0] indicates the number of valid bytes the last double word is composed of when TEOP is asserted. TEOP must be present at the end of every packet and is considered valid only when TENB is asserted.

Table 1-8. CX28560 POS-PHY Interface (Transmit) (2 of 2)

Pin Name	I/O	Ref Clk	Description
PTPA	O	TFCLK	Transmit Packet Available (PTPA) signal. PTPA transitions high when a predefined minimum number of bytes are available in the polled transmit FIFO. Once high, PTPA indicates that the transmit FIFO is not full. When PTPA transitions low, it optionally indicates that the transmit FIFO is full or near full. PTPA allows the polling of the CX28560. The port which PTPA reports is updated on the following rising edge of TFCLK. PTPA is updated on the rising edge of TFCLK.
TPRTY	I	TFCLK	Transmit Bus Parity Signal (TPRTY). TPRTY indicates the parity calculated over the TDAT bus. TPRTY is considered valid only when TENB is asserted. The CX28560 supports odd parity checking which can be disabled by configuring the DISBLPAR bit in the Table 5-21, *Transmit POS-PHY Thresholds Register*.The CX28560 reports any parity error to the system, but shall not interfere with the transferred data.
Note(s): The following pins are supported by the standard POS-PHY, but are not required because the CX28560 supports only packet-level transfers on a single PHY basis: Transmit Start of Transfer (TSX) signal; Transmit PHY Address (TADR[]) bus; Direct Transmit Packet Available (DTPA[]); Selected-PHY Transmit Packet Available (STPA) signal.			

Table 1-9 describes data transfer from the CX28560 to the system. This covers two interfaces—the data interface (32 bit, 100 MHz) and the FlowConductor interface (8 bit, 100 MHz).

Table 1-9. CX28560 POS-PHY Interface (Receive) (1 of 2)

Pin Name	I/O	Ref Clk	Description
RFCLK FRFCLK	I	—	System to the CX28560 Receive FIFO Write Clock (RFCLK). RFCLK is used to synchronize data transfer transactions between the system and the CX28560. Both RFCLK and FRFCLK cycle at 100 MHz and signals are sampled on their rising edges.
RVAL	O	RFCLK	Receive Data Valid (RVAL) signal. RVAL indicates the validity of the receive data signals. RVAL will transition low when a receive FIFO is empty or at the end of a packet. When RVAL is high, the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, and REOP signals are valid. When RVAL is low, the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, and REOP signals are invalid and must be disregarded.
FRVAL		FRFCLK	FRVAL indicates the validity of the receive data signals. FRVAL will transition low when a receive FIFO is empty or at the end of a packet. When FRVAL is high, the FRDAT[7:0], FRPRTY, FRSOP, and FREOP signals are valid. When FRVAL is low, the FRDAT[7:0], FRPRTY, FRSOP, and FREOP signals are invalid and must be disregarded.
RENB	I	RFCLK	Receive Read Enable (RENB) signal. The RENB signal controls the flow of data from the receive FIFO's. During data transfer, RVAL must be monitored as it will indicate if the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, and REOP are valid. The system may deassert RENB at anytime if it is unable to accept data from the CX28560. When RENB is sampled low by the CX28560, a read is performed from the receive FIFO and the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, REOP, and RVAL signals are updated on the following rising edge of RFCLK. When RENB is sampled high by the CX28560, a read is not performed, and the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, REOP, and RVAL signals will not updated on the following rising edge of RFCLK.
FRENB		FRFCLK	The FRENB signal is used to control the flow of data from the receive FIFO's. During data transfer, FRVAL must be monitored as it will indicate if the FRDAT[7:0], FRPRTY, FRSOP, and FREOP are valid. The system may deassert FRENB at anytime if it is unable to accept data from the CX28560. When FRENB is sampled low by the CX28560, a read is performed from the receive FIFO and the FRDAT[7:0], FRPRTY, FRSOP, FREOP, and FRVAL signals are updated on the following rising edge of FRFCLK. When FRENB is sampled low by the CX28560, a read is not performed and the FRDAT[7:0], FRPRTY, FRSOP, FREOP, and FRVAL signals will not updated on the following rising edge of FRFCLK.
RDAT[31:0] FRDAT[7:0]	O	RFCLK FRFCLK	Receive Packet Data Bus (RDAT[31:0] for data interface, FRDAT[7:0] for FlowConductor Interface). The RDAT[31:0]/FRDAT[7:0] bus carries the packet octets that are read from the receive FIFO and the in-band port address of the selected receive FIFO. RDAT[31:0]/FRDAT[7:0] is considered valid only when RVAL/FRVAL is asserted on the 32-bit interface; data must be received in big endian order. In accordance with HDLC protocol, bit 0 of each byte is the first received bit at the serial interface.

Table 1-9. CX28560 POS-PHY Interface (Receive) (2 of 2)

Pin Name	I/O	Ref Clk	Description
RMOD[1:0]	O	RFCLK	Receive Word Modulo (RMOD) signal. RMOD[1:0] indicates the number of valid bytes of data in RDAT[31:0]. The RMOD bus should always be all zero, except during the last double-word transfer of a packet on RDAT[31:0]. When REOP is asserted, the number of valid packet data bytes on RDAT[31:0] is specified by RMOD[1:0]: RMOD[1:0] = 00 RDAT[31:0] valid RMOD[1:0] = 01 RDAT[31:8] valid RMOD[1:0] = 10 RDAT[31:16] valid RMOD[1:0] = 11 RDAT[31:24] valid When the FlowConductor 8-bit interface, the RMOD bus is not considered. RMOD[1:0] is considered valid only when RVAL is asserted
RSOP FRSOP	O	RFCLK FRFCLK	Receive Start of Packet (RSOP/FRSOP) signal. RSOP/FRSOP delineates the packet boundaries on the RDAT/FRDAT bus. When RSOP/FRSOP is high, the start of the packet is present on the RDAT/FRDAT bus. RSOP/FRSOP will be present at the end of every packet and is considered valid when RVAL/FRVAL is asserted.
REOP FREOP	O	RFCLK FRFCLK	Receive End Of Packet (REOP/FREOP) signal. REOP/FREOP delineates the packet boundaries on the RDAT/FRDAT bus. When REOP/FREOP is high, the end of the packet is present on the RDAT/FRDAT bus. On the data 32-bit interface, RMOD[1:0] indicates the number of valid bytes the last double word is composed of when REOP is asserted. On the FlowConductor 8-bit interface, the last byte of the packet is on FRDAT[7:0] when FREOP is asserted. REOP/FREOP is required to be present at the end of every packet and is considered valid only when RVAL/FRVAL is asserted.
RPRTY FRPRTY	O	RFCLK FRFCLK	Receive Parity (RPRTY/FRPRTY) signal. The receive parity (RPRTY/FRPRTY) signal indicates the parity calculated over the RDAT/FRDAT bus. On the FlowConductor 8-bit interface, the CX28560 only supports FRPRTY calculated over FRDAT[7:0]. The CX28560 supports parity calculation.
RCLAV FRCLAV	O	RFCLK FRFCLK	Receive Cell Available (RCLAV/FRCLAV). RCLAV/FRCLAV indicates when the System device has data to transfer. This signal is only relevant in Registered mode (see Chapter 2.0).

Note(s):
1. The Receive Start of Transfer (RSX) pin is supported by the standard POS-PHY, but are not required because the CX28560 supports only packet-level transfers on a single PHY basis.
2. The CX28560 architecture guarantees that the RERR pin, if implemented, would never be asserted. Therefore, to comply fully with POS-PHY, the system should tie RERR to zero.

Table 1-10. PCI Interface (1 of 2)

Pin Name	I/O	Ref Clk	Description
PCLK	I	—	PCI Clock (PCLK). PCLK provides timing for all PCI transitions. All PCI signals except PRST*, INTA*, and INTB* are synchronous to PCLK and are sampled on the rising edge of PCLK. The CX28560 supports a PCI clock up to 33 MHz.
AD[31:0]	I/O	PCLK	PCI Address and Data (AD[31:0]). AD[31:0] is a multiplexed address/data bus. A PCI transaction consists of an address phase during the first clock period followed by one or more data phases. AD[7:0] is the LSB. As both a master and a target, the CX28560 supports only 32-bit operations.
CBE[3:0]	I/O	PCLK	PCI Command and Byte Enables (CBE[3:0]). During the address phase, CBE[3:0] contain command information. During the data phases, CBE[3:0] contain information denoting which byte lanes are valid. Supported PCI commands are defined as follows: CBE[3:0] — Command Type 6h 0110b — Memory Read 7h 0111b — Memory Write Ah 1010b — Configuration Read (target only) Bh 1011b — Configuration Write (target only) Ch 1100b — Memory Read Multiple Eh 1110b — Memory Read Line Fh 1111b — Memory Write and Invalidate (target only)
PAR	I/O	PCLK	PCI Parity (PAR). The number of 1s on PAR, AD[31:0], and CBE[3:0] is an even number. PAR always lags AD[31:0] and CBE* by one clock. During address phases, PAR is stable and valid one clock after the address; during the data phases, it is stable and valid one clock after TRDY on reads and one clock after IRDY on writes. It remains valid until one clock after the completion of the data phase.
FRAME	I/O	PCLK	PCI Frame (FRAME). FRAME is driven by the current master to indicate the beginning and duration of a bus cycle. Data cycles continue as FRAME stays asserted. The final data cycle is indicated by the deassertion of FRAME. For a non-burst, one-data-cycle bus cycle, this pin is only asserted for the address phase.
TRDY	I/O	PCLK	PCI Target Ready (TRDY). Asserted indicates the target's readiness to complete the current data phase.
IRDY	I/O	PCLK	PCI Initiator Ready (IRDY). Asserted indicates the current master's readiness to complete the current data phase.
STOP	I/O	PCLK	PCI Stop (STOP). Asserted indicates the selected target is requesting the master to stop the current transaction.
DEVSEL	I/O	PCLK	PCI Device Select (DEVSEL). Asserted indicates that the driving device has decoded its address as the target of the current cycle.
IDSEL	I	PCLK	PCI Initialization Device Select (IDSEL). This input is used to select the CX28560 as the target for configuration read or write cycles.

Table 1-10. PCI Interface (2 of 2)

Pin Name	I/O	Ref Clk	Description
PERR	I/O	PCLK	Parity Error (PERR). PERR* is asserted by the agent receiving data when it detects a parity error on a data phase. It is asserted one clock after PAR is driven, which is two clocks after the AD and CBE parity was checked. If the CX28560 masters a PCI write cycle, and—after supplying the data during the data phase of the cycle—detects this signal being asserted by the agent receiving the data, then the CX28560 generates a PERR interrupt. If CX28560 masters a PCI read cycle, and—after receiving the data during the data phase of the cycle—calculates that a parity error has occurred, the CX28560 asserts this signal and also generates the PERR Interrupt Descriptor towards the host.
SERR	O	PCLK	System Error (SERR). Any PCI device can assert SERR to indicate a parity error on the address cycle or parity error on the data cycle of a special cycle command or any other system error where the result will be catastrophic. The CX28560 asserts SERR if it detects a parity error on the address cycle or encounters an abort condition while operating as a PCI master. Since SERR is not a sustained three-state signal, restoring it to the deasserted state is done with a weak pull-up (same value as used for sustained three state)(1).
REQ	I/O	PCLK	PCI Bus Request (REQ). The CX28560 drives REQ to notify the PCI arbiter that it desires to master the bus. Every master in the system has its own REQ.
GNT	I	PCLK	PCI Bus Grant (GNT). The PCI bus arbiter asserts GNT when the CX28560 is free to take control of the bus, assert FRAME, and execute a bus cycle. Every master in the system has its own GNT.
INTA	O	PCLK	PCI CX28560 Interrupt (INTA). INTA is driven by the CX28560 to indicate a Layer 2 interrupt condition to the host processor.
PRST	I	None	PCI Reset (PRST). This input resets all functions on the CX28560.

Note(s):
(1) The CX28560 does not input SERR. It is assumed that the host will reset CX28560 in the case of a catastrophic system error.

Table 1-11. EBUS Interface (Communication with Peripheral Components)

Pin Name	I/O	Ref Clk	Description
ECLK	O	—	Expansion Bus Clock (ECLK). The ECLK bit field is an inverted version of the PCI clock.
EAD[31:0]	I/O	ECLK	Expansion Bus Address and Data (EAD[31:0]). EAD[31:0] is a multiplexed address/data bus.
EBE[3:0]	O	ECLK	Expansion Bus Byte Enables (EBE[3:0]). EBE contains byte-enabled information for the EBUS transaction.
ALE (AS)	O	ECLK	Address Latch Enable (ALE (AS)). High-to-low transition indicates that EAD[31:0] bus contains valid address. Remains asserted low through the data phase of the EBUS access. (In Motorola mode, high-to-low transition indicates EBUS contains a valid address. Remains asserted for the entire access cycle.)
WR (R/WR)	O	ECLK	Write Strobe (WR (R/WR)). High-to-low transition enables write data from CX28560 into peripheral device. Rising edge defines write. (In Motorola .mode, R/WR is held high throughout read and held low throughout write. Determines meaning of DS strobe.)
RD (DS)	O	ECLK	Read Strobe (RD (DS)). High-to-low transition enables read data from peripheral into CX28560. Held high throughout write operation. (In Motorola mode, DS transitions low for both read and write operations and is held low throughout the operation.)
HOLD (BR)	O	ECLK	Hold Request (Bus Request) (HOLD (BR)). When asserted, CX28560 requests control of the EBUS.
HLDA (BG*)	I	ECLK	Hold Acknowledge (Bus Grant) (HLDA (BG)). When asserted, CX28560 has access to the EBUS. It is held asserted when there are no other masters connected to the bus, or asserted as a handshake mechanism to control EBUS arbitration.
BGACK	O	ECLK	Bus Grant Acknowledge (BGACK). When asserted, CX28560 acknowledges to the bus arbiter that the bus grant signal was detected and a bus cycle is sustained by CX28560 until this signal is de-asserted.

Table 1-12. Boundary Scan and Test Access

Pin Name	I/O	Ref Clk	Description
TCK	I	—	JTAG Clock (TCK). Used to clock in the TDI and TMS signals and as clock out TDO signal.
TRST	I	TCK	JTAG Enable (TRST). An active-low input used to put the chip into a special test mode. This pin should be pulled up in normal operation.
TMS	I	TCK	JTAG Mode Select (TMS). The test signal input decoded by the TAP controller to control test operations.
TDO	O	TCK	JTAG Data Output (TDO) The test signal used to transmit serial test instructions and test data.
TDI	I	TCK	TDI JTAG Data Input (TDI) The test signal used to receive serial test instructions and test data.
TM[3:0]	I	—	Test Mode (TM). Encodes tests modes (must be pulled low in normal operation).

Table 1-13. Performance Monitoring

Pin Name	I/O	Ref Clk	Description
ONESEC	I	—	An asynchronous pulse provided as an input to CX28560 that causes the latching of the performance monitoring counters. Not necessarily on one-second boundaries.

2.0 Host Interfaces

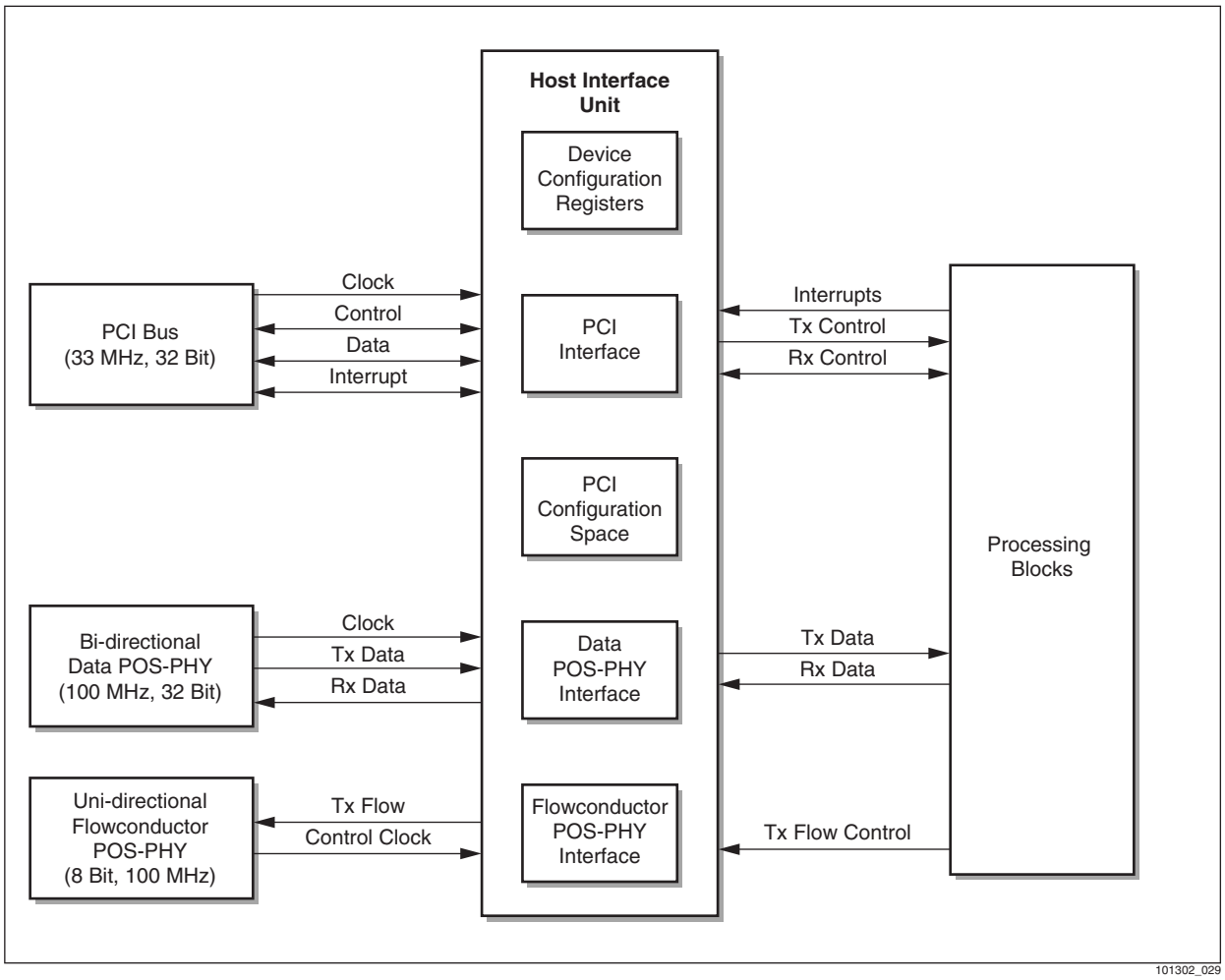
2.1 Host Interface

The CX28560's host interface consists of a PCI interface, a POS-PHY data interface and a POS-PHY Flow Conductor interface. Over these interfaces the following major functions are performed:

- Transfer of data as fragments between the CX28560 and the system;
- Configuration and monitoring of the CX28560 registers and counters;
- Monitoring the fill level of the CX28560's internal per channel buffers.

Figure 2-1 illustrates the CX28560 host Interface block diagram.

Figure 2-1. The CX28560 Host Interface Functional Block Diagram



2.1.1 PCI Interface

The host interface in the CX28560 is compliant with the *PCI Local Bus Specification* (Revision 2.2). The CX28560 provides a PCI interface specific to 3.3 V and 33 MHz operation and supports as master a 32-bit bus with multiplexed address and data lines, and as a slave, a 32-bit PCI bus.

NOTE: The *PCI Local Bus Specification* (Revision 2.2) is an architectural, timing, electrical, and physical interface standard that provides a mechanism for a device to interconnect with processor and memory systems over a standard bus. The host interface can act as a PCI master and a PCI slave, and contains the CX28560's PCI configuration space and internal registers. When the CX28560 needs to access shared memory, it masters the PCI bus and completes the memory cycles without external intervention.

2.1.1.1 PCI Initialization

Generally, when a system initializes a module containing a PCI device, the configuration manager reads the configuration space of each PCI device on a PCI bus. Hardware signals select a specific PCI device based on a bus number, a slot number, and a function number. If a device that is addressed (via signal lines) responds to the configuration cycle by claiming the bus, then that function's configuration space is read out from the device during the cycle. Since any PCI device can be a multi-function device, every supported function's configuration space needs to be read from the device. Based on the information read, the configuration manager will assign system resources to each supported function within the device. Sometimes new information needs to be written into the function's configuration space. This is accomplished with a configuration write cycle.

The CX28560 is a single function device that has device-resident memory to store the required configuration information. The CX28560 supports Function 0 only.

2.1.1.2 PCI Bus Operations

The CX28560 behaves either as a PCI master or a PCI slave device at any time and switches between these modes as required during device operation. The CX28560 supports only dword write transactions.

As a PCI slave, the CX28560 responds to the following PCI bus operations:

- Memory Read
- Memory Write
- Configuration Read
- Configuration Write
- Memory Read Multiple (treated like Memory Read in slave mode)
- Memory Read Line (treated like Memory Read in slave mode)
- Memory Write and Invalidate (treated like Memory Write)

NOTE: As a PCI slave, the CX28560 does not support bursted read or write PCI transactions. The CX28560 ignores all other PCI cycles.

As a PCI master, the CX28560 generates the following PCI bus operations:

- Memory Read
- Memory Read Line
- Memory Read Multiple
- Memory Write

2.1.1.3 Fast Back-to-Back Transactions

Fast back-to-back transactions allow agents to use bus bandwidth more effectively. the CX28560 supports PCI fast back-to-back transactions both as a bus target and bus master. the CX28560 can also execute fast back-to-back transactions regardless of the PCI configuration settings (for details see bit 11 TARGET_FBTB bit field in Chapter 5.0).

NOTE: The CX28560 will only perform Fast Back to Back between transactions from different sources (either the Interrupt Controller or the Host Service Unit) and not between transactions from the same source.

Fast back-to-back transactions are allowed on PCI when contention on TRDY*, DEVSEL*, STOP*, or PERR* is avoided. (for a detailed description of these pins see Chapter 1.0).

The CX28560, as a master supporting fast back-to-back transactions, places the burden of avoiding contention on itself. While acting as a slave, the CX28560 places the burden on all the potential targets. As a master, the CX28560 may remove the Idle state between transactions when it can guarantee that no contention occurs. This can be accomplished when the master's current transaction is to the same target as the previous transaction. While supporting this type of fast back-to-back transaction, the CX28560 understands the address boundaries of the potential target, so that no contention occurs. The target must be able to detect a new assertion of FRAME* without the bus going to idle state.

Operation Mode

During a fast back-to-back transaction, the master starts the next transaction if GNT* is still asserted. If GNT* is deasserted, the master has lost access to the bus and must relinquish the bus to the next master. The last data phase completes when FRAME* is deasserted, and IRDY* and TRDY* (or STOP*) are asserted. The current master starts another transaction on the clock following the completion of the last data phase of the previous transaction. During fast back-to-back transaction, only the master and target involved need to distinguish intermediate transaction boundaries using only FRAME* and IRDY* (there is no bus Idle state). When the transaction is over, all the agents see an Idle state.

Example of an Arbitration for Fast Back-to-Back and Non-Fast Back-to-Back Transactions

Appendix G shows an example of an arbitration for fast back-to-back and non-fast back-to-back transactions. The transactions shown are bursts of 2 or 3 dwords.

2.1.1.4 PCI Configuration Space

This section describes how the CX28560 implements the required PCI configuration register space. The intent of PCI configuration space definition is to provide an appropriate set of configuration registers that satisfy the needs of current and anticipated system configuration mechanisms, without specifying those mechanisms or otherwise placing constraints on their use. These registers allow for the following:

- Full device relocation, including interrupt binding
- Installation, configuration, and booting without user intervention
- System address map construction by device-independent software

The CX28560 responds only to Type 0 configuration cycles. Type 1 cycles, which pass a configuration request on to another PCI bus, are ignored.

The CX28560 is a single function PCI agent; therefore, it implements configuration space for Function 0 only.

The PCI controller in the CX28560 responds to configuration and memory cycles, but only memory cycles cause bus activity on the EBUS.

The address phase during a the CX28560 configuration cycle indicates the function number and register number being addressed, which can be decoded by observing the status of the address lines AD[31:0]. The figure below illustrates the address lines during configuration cycle.

31 8	7 2	1 0
Don't Care	6 bit register #	2 bit Type #
NOTE(S): The CX28560 supports Function 0 only The CX28560 supports registers 0 through 15 inclusively The CX28560 supports Type 0 configuration cycles.		

The value of the signal lines AD[10:8] selects the function being addressed. Since the CX28560 supports Function 0 only, it ignores these bits. The value of the signal lines AD[7:2] during the address phase of configuration cycles selects the register of the configuration space to access. Valid values are 0 through 15. Accessing registers outside this range results in an all 0s value being returned on reads, and no action being taken on writes.

The value of the signal lines AD[1:0] must be 00b for the CX28560 to respond. If these bits are 0 and the IDSEL* signal line is asserted, then the CX28560 responds to the configuration cycle.

The Base Code register contains the Class Code, Sub Class Code, and Register Level Programming Interface registers. Table 2-1 illustrates the PCI Configuration Space.

Table 2-1. PCI Configuration Space

Register Number	Byte Offset (hex)	31 24	23 16	15 8	7 0
0	00h	Device ID		Vendor ID	
1	04h	Status		Command	
2	08h	Base Code			Revision ID
3	0Ch	Reserved	Header Type	Latency Timer	Reserved
4	10h	The CX28560 Base Address Register			
5–10		Reserved			
11	2Ch	Subsystem ID		Subsystem Vendor ID	
12–14		Reserved			
15	3Ch	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

All writable bits in the configuration space are reset to 0 by the hardware reset, PRST* asserted. After reset, the CX28560 is disabled and only responds to PCI configuration write and PCI configuration read cycles. Write cycles to reserved bits and registers have no effect. Read cycles to reserved bits always result in 0 being read.

2.2 PCI Configuration Registers

2.2.1 PCI Master and Slave

The CX28560 is a single function PCI device that provides the necessary configuration space for a PCI bus controller to query and configure the CX28560's PCI interface. PCI configuration space consists of a device-independent header region (64 bytes) and a device-dependent header region (192 bytes). The CX28560 provides the device-independent header section only. Access to the device-dependent header region results in 0s being read, and no effect on writes.

Three types of registers are available in the CX28560:

1. Read-Only (RO)—Return a fixed bit pattern if the register is used or a 0 if the register is unused or reserved
2. Read-Resettable (RR)—Can be reset to 0 by writing a 1 to the register
3. Read/Write (RW)—Retain the value last written to it.

Sixteen dword registers make up the CX28560's PCI Configuration Space.

The tables below specify the contents of these registers:

2.2.1.1 Register 0, Address 00h

Table 2-2. Register 0, Address 00h

Bit Field	Name	Reset Value	Type
31:16	Device ID	8563 = 2047 Channel HDLC Controller	RO
15:0	Vendor ID	14F1h	RO

2.2.1.2 Register 1, Address 04h

The Status register records status information for PCI bus related events. The Command register provides coarse control to generate and respond to PCI commands.

At reset, the CX28560 sets the bits in this register to 0, meaning the CX28560 is logically disconnected from the PCI bus for all cycle types except configuration read and configuration write cycles.

Table 2-3. Register 1, Address 04h (1 of 2)

Bit Field	Name	Reset Value	Type	Description
31	Status	0	RR	Detected Parity Error. This bit is set by the CX28560 whenever it detects a parity error, even if parity error response is disabled.
30	—	0	RR	Detected System Error. This bit is set by the CX28560 whenever it asserts SERR.
29	—	0	RR	Received Master Abort. This bit is set by the CX28560 whenever a CX28560 initiated cycle is terminated with a master abort.
28	—	0	RR	Received Target Abort. This bit is set by the CX28560 whenever a CX28560 initiated cycle is terminated by a target-abort.
27	—	0	RO	Unused
26:25	—	01b	RO	DEVSEL Timing. Indicates the CX28560 is a medium speed PCI device. This means the longest time it will take the CX28560 to return DEVSEL when it is a target is 3 clocks.
24	—	0	RR	Data Parity Detected. This bit is set by the CX28560 whenever the following 3 conditions are met: The CX28560 asserted PERR or observed PERR The CX28560 was the master for that transactions Parity Error Response bit is set.
23	—	1b	RO	Fast Back-to-Back Capable. Read Only. Indicates that the CX28560 is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.
22	—	0	RO	Unused
21	—	1b	RO	Not 66 MHz Capable.
20:16	—	0	RO	Unused
15:10	Command	0	RO	Unused

NOTE(S):
This value would normally indicate that the device is capable of supporting a 66 MHz clock, but the CX28560 does not.

Table 2-3. Register 1, Address 04h (2 of 2)

Bit Field	Name	Reset Value	Type	Description
9	—	0	RW	Fast back-to-back enable. This bit controls whether or not the CX28560, while acting as master, can perform fast back-to-back transactions to different devices. The configuration software routine sets this bit if all bus agents in the system are fast back-to-back capable. If 1, the CX28560 can generate fast back-to-back transactions to different agents. If 0, the CX28560 can generate fast back-to-back transactions to the same agent. Note: This bit would be presumably set by the system configuration routine after ensuring that all targets on the same bus had the Fast Back-to-Back Capable Bit set. If the target is unable to provide the fast back-to-back capability, the target does not implement this bit and it is automatically returned as zero when Status register is read.
8	—	0	RW	SERR enable. If 1, disables the CX28560's SERR* driver. If 0, enables the CX28560's SERR* driver and allows reporting of address parity errors.
7	—	0	RO	Wait cycle control. The CX28560 does not support address stepping.
6	—	0	RW	Parity error response. This bit controls the CX28560's response to parity errors. If 1, the CX28560 takes normal action when a parity error is detected on a cycle as the target. If 0, the CX28560 ignores parity errors.
5	—	0	RO	VGA palette snoop. Unused.
4	—	0	RO	Memory write and invalidate. The only write cycle type the CX28560 generates is memory write.
3	—	0	RO	Special cycles. Unused. the CX28560 ignores all special cycles.
2	—	0	RW	Bus master. If 1, the CX28560 is permitted to act as bus master. If 0, the CX28560 is disabled from generating PCI accesses.
1	—	0	RW	Memory space. Access control. If 1, enables the CX28560 to respond to memory space access cycles. If 0, disables the CX28560's response.
0	—	0	RO	I/O space accesses. The CX28560 does not contain any I/O space registers.

2.2.1.3 Register 2, Address 08h

This location contains the Class Code and Revision ID registers. The Class Code register contains the Base Code, Sub Class, and Register Level Programming Interface fields. These are used to specify the generic function of the CX28560. The Revision ID register denotes the version of the device.

Table 2-4. Register 2, Address 08h

Bit Field	Name	Reset Value	Type	Description
31:24	Class Code	02h	RO	Function: Network Controller
23:16	Sub Class Code	80h	RO	Type: Other
15:8	Register Level Programming Interface	0	RO	Indicates that there is nothing special about programming the CX28560.
7:0	Revision Id	00h	RO	Denotes the revision number of the CX28560. This revision Id is divided into two 4 bit fields. Upper nibble indicates Die ID which started from 0 for this device. The lower nibble is used for rev number, Rev A = 0, Rev B = 1, etc.

2.2.1.4 Register 3, Address 0Ch

Table 2-5. Register 3, Address 0Ch

Bit Field	Name	Reset Value	Type	Description
31:24	Reserved	0	RO	Unused
23:16	Header Type	0	RO	The CX28560 is a single function device with the standard layout of configuration register space.
15:11	Latency Timer	0	RW	The latency timer is an 8-bit value that specifies the maximum number of PCI clocks that the CX28560 can keep the bus after starting the access cycle by asserting its FRAME*. The latency timer ensures that the CX28560 has a minimum time slot for it to own the bus, but places an upper limit on how long it owns the bus.
10:8	—	0	RO	—
7:0	Reserved	0	RO	Unused

2.2.1.5 Register 4, Address 10h

Table 2-6. Register 4, Address 10h

Bit Field	Name	Reset Value	Type	Description
31:20	CX28560 Base Address Register	0	RW	Allows for 1 MB-bounded PCI bus address space to be blocked off as the CX28560 space. The CX28560 will respond as a PCI slave with DEVSEL* to all memory cycles whose address bits 31:20 match the value of bits 31:20 of this register, and those upper address bits are non-zero, and memory space is enabled in the Register 1, COMMAND bit field. Reads to addresses within this space that are not implemented read back 0; writes have no effect.
19:4	—	0	RO	When appended to bits 31:20, these bits specify a 1 MB bound memory range. 1 MB is the only amount of address space that a CX28560 can be assigned.
3	—	0	RO	The CX28560 memory space is not prefetchable.
2:1	—	0	RO	The CX28560 can be located anywhere in 32-bit address space.
0	—	0	RO	This base register is a memory space base register, as opposed to I/O mapped.

2.2.1.6 Register 5–10, Address 14h–28h

Table 2-7. Register 5–10, Address 14h–28h

Bit Field	Name	Reset Value	Type	Description
31:0	Reserved	0	RO	Unused

2.2.1.7 Register 11, Address 2Ch

Table 2-8. Register 11, Address 2Ch

Bit Field	Description	Reset Value	Type
31:16	Subsystem ID	8563 = 2047 Channel HDLC Controller	RO
15:0	Subsystem Vendor ID	14F1	RO

2.2.1.8 Register 12–14, Address 30h–38h

Table 2-9. Register 12–14, Address 30h–38h

Bit Field	Name	Reset Value	Type	Description
31:0	Reserved	0	RO	Unused

2.2.1.9 Register 15, Address 3Ch

Table 2-10. Register 15, Address 3Ch

Bit Field	Name	Reset Value	Type	Description
31:24	Maximum Latency	0Fh	RO	Specifies how quickly the CX28560 needs to gain access to the PCI bus. The value is specified in 0.25 µs increments and assumes a 33 MHz clock. A value of 0Fh means Tanfo needs to gain access to the PCI bus every 130 PCI clocks, expressed as 3.75 µs in this register.
23:16	Minimum Grant	01h	RO	Specifies, in 0.25 µs increments, the minimum burst period the CX28560 needs. The CX28560 does not have any special MIN_GNT requirements.
15:8	Interrupt Pin	01h	RO	Defines which PCI interrupt pin the CX28560 uses. 01h means the CX28560 uses pin INTA*.
7:0	Interrupt Line	0	RW	Communicates interrupt line routing. System initialization software writes a value to this register indicating which host interrupt controller input is connected to the CX28560's INTA* pin.

2.2.2 PCI Reset

The CX28560 resets all internal functions when it detects the assertion of the PRST* signal line. Upon reset, the following occurs:

- All PCI output signals are three-stated immediately and asynchronously with respect to the PCI clock input, PCLK.
- All EBUS output signals are three-stated immediately and asynchronously with respect to the EBUS clock output, ECLK.
- All writable/resettable internal register bits are set to 0.
- All PCI transfers are terminated immediately.
- All serial data transfers are terminated immediately.
- All POS-PHY transfers are terminated immediately. Output signals are three-stated immediately and asynchronously to the POS-PHY clocks.
- The CX28560 is disabled and responds only to PCI configuration cycles.
- All data is lost.

2.2.3 PCI Throughput and Latency Considerations

For reference to PCI throughput and latency considerations see Appendix H.

2.2.4 Host Interface

After a hardware reset, the PCI configuration space within CX28560 needs to be configured by the host as follows:

- Base address register
- Fast back-to-back enable/disable
- SERR* signal driver enable/disable
- Parity error response enable/disable
- Latency timer register
- Interrupt line register
- Bus mastering enable/disable
- Memory space access enable/disable

2.3 POS-PHY

2.3.1 POS-PHY Interfaces

There are three POS-PHY interfaces—two for the transfer of data, and one that is used as a feedback bus to the system.

In addition to the individual configurations necessary for configuration, the POS-PHY Registered Mode bit in the Global Configuration register (Section 5.4) should be configured.

2.3.1.1 POS-PHY Registered Mode

The POS-PHY standard sets the timing of the RxENB signal. Normal mode (non-registered mode) works exactly according to the standard. Non-registered mode delays the timing of the sampling of the RxENB signal by one clock. In order to use the CX28560 with the TSP (MXT4700), the POS-PHY should be configured in registered mode. When in registered mode, the Cell Available (CLAV) signal is also active. See Table 1-9, *CX28560 POS-PHY Interface (Receive)*.

2.3.1.2 POS-PHY Data Interface

In all places where the POS-PHY Data Interface is referred to, the “Transmit side” is the side on which data is transmitted from the host to the CX28560, and the “Receive side” is the side on which the CX28560 transmits data to the host.

The POS-PHY Data Bus implemented in the CX28560 is compliant to the ATM POS-PHY level 3 standard (AF-PHY-0143.000) and supports other industry standards for Level 3 packet functionality at 100 MHz clock, and 32 bit wide data bus for the transferal of data fragments. The packet functionality is provided by start of packet and end of packet signals that delimit the fragments.

NOTE: The start and end of HDLC packets are indicated in the fragment headers.

Flow control on the Transmit side bus is provided by the PTPA (Polled-PHY Transmit Packet Available) pin. When there is not enough space in the buffer to receive further data for transmission, the PTPA pin is set to low. The decision as to whether there is space in the buffers is decided according to two thresholds: an upper threshold, and a lower threshold. The buffers are initially empty. Crossing thresholds has the following affects:

- Crossing the lower threshold from below has no affect (the PTPA pin remains asserted).
- Crossing the upper threshold from below de-asserts the PTPA pin (there is no space in the buffer).
- Crossing the upper threshold from above has no affect (there is still no space in the buffer).
- Crossing the lower threshold from above causes the PTPA pin to be asserted (there is now space in the buffer).

The thresholds are set in the Transmit POS-PHY Thresholds register (see Chapter 5.0, Transmit POS-PHY Thresholds register).

The upper threshold should be set to the buffer size less the maximum fragment length. The lower threshold should be set in accordance with the latency of the mechanism deciding whether to send data.

2.3.1.3 POS-PHY Flow Conductor Interface

The POS-PHY Flow Conductor bus implemented in the CX28560 is compliant to the ATM POS-PHY level 3 standard (AF-PHY-0143.000) and supports other industry standards for level 3 packet functionality at 100 MHz clock, and 8 bit wide data bus for the transferal of report packets. This interface is identical to the receive side data interface provided.

The Flow Conductor POS-PHY and the Receive data POS-PHY interfaces are not fully compatible with the POS-PHY standard. Not during data transmission, if FRENB/RENB is high (not enabled), the other signals change their value on the following clock. During data transmission if FRENB/RENB is high, the CX28560 POS-PHY interface stores the old values on the next clock. Hence, the system should sample data only if FRENB/RENB is low on the previous clock.

NOTE: The CX28560 will always introduce a minimum of a 2 cycle delay between packets over the POS-PHY interface.

2.3.1.4 Receive POS-PHY Initialization

No initialization of the POS-PHY is necessary.

2.3.1.5 Transmit POS-PHY Initialization

To initialize the Transmit POS-PHY, the flow control thresholds should be set (see above) in the data bus. No initialization is necessary for the Flow Conductor bus.

3.0 Expansion Bus (EBUS)

The CX28560 provides an access to a local bus interface called the Expansion Bus (EBUS), which provides a host processor to access any address in the peripheral memory space on the EBUS. Although EBUS use is optional, the most notable applications for the EBUS are the connections to peripheral devices, e.g., Bt8370/ Bt8398 T1/E1 framers, CN8398 (Octal DS1/E1 framers), and CX29503/M29513 BAM3/BAM3+ and CX29610 OptiPhy that are local to the CX28560's serial port.

Similarly to the CN28500, but unlike previous generations of HDLC controllers (CN8478/CN8474/CN8472), the CX28560 provides access to the EBUS through an interface similar to a mailbox interface. This interface provides all the EBUS read and write accesses to be carried over the PCI bus, allowing PCI bursts. This mechanism improves the PCI use when multiple EBUS accesses are necessary for accessing the configuration of peripheral devices. The PCI Function 1 is disabled. Therefore, the CX28560's EBUS service requests are capable of accepting or generating burst on the PCI bus. However, the EBUS interface signals are not capable of performing burst. Also, the CX28560 does not interface with EINT* signal of the EBUS. This signal should be tied directly or via external logic to the PCI interrupt INTB*.

Figures 3-1 and 3-2 illustrate block diagrams of the EBUS interface with and without local microprocessor (MPU).

Figure 3-1. EBUS Functional Block Diagram with Local MPU

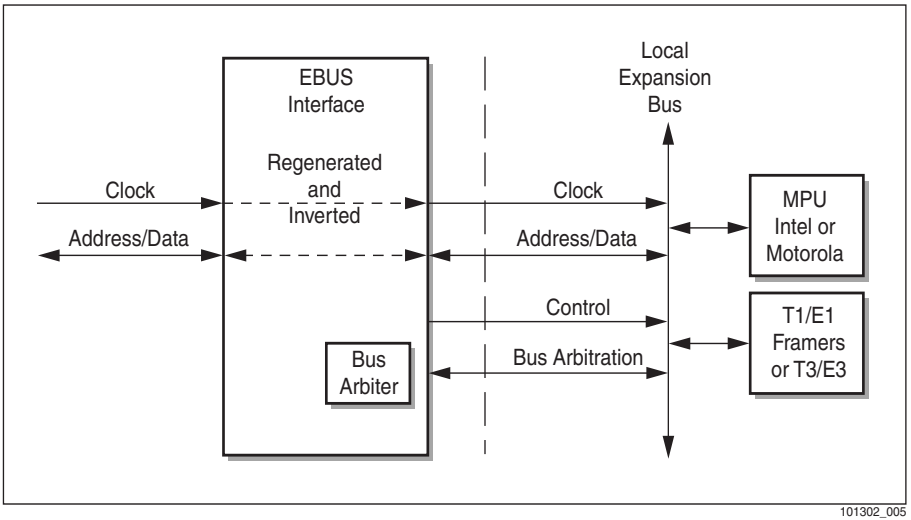
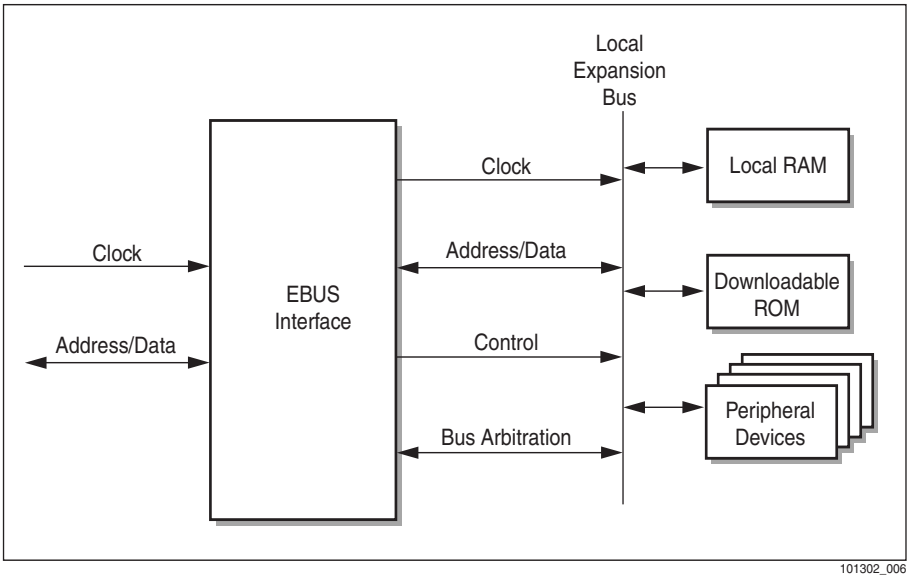


Figure 3-2. EBUS Functional Block Diagram without Local MPU



3.1 EBUS—Operational Mode

3.1.1 Initialization

After reset and after the PCI configuration is completed, the CX28560 provides the host the ability to read and write peripheral devices located on the EBUS (see Table 3-1). The Host Service Request mechanism allows the host to instruct the CX28560 to perform specific EBUS operations. The CX28560 can perform bulk service request commands. The Service Request Acknowledge (SACK) can be generated either after each service request command or at the end of each bulk service request, depending on the value of SACKIEN bit field set in the service request configuration descriptor (see Table 3-2). The CX28560 processes an SRQ by reading the Table 5-4, *Service Request Pointer Register* which contains the address of the first entry in the Host Descriptor table. Once configured and enabled, the host can configure local devices connected to the EBUS by issuing the EBUS Access Service Request (EBUS_WR or EBUS_RD). The command is a three dword memory location that contains the following dword fields:

- Access Control Field
- Shared Memory Pointer (Buffer Address) representing the starting address of the buffer location where the device structure resides
- EBUS Base Address Offset (the address of the first EBUS transaction)

Table 3-1. EBUS Service Request Descriptor

Dword Number	Bit 31					Bit 0
dword 0	OPCODE[31:27]	SACKIEN[26]	Reserved[25:19]	FIFO_BURST[18]	EBUS Byte Enable [17:14]	Length[13:0]
dword 1	Shared Memory Pointer[31:2](2)					
dword 2	EBUS Base Address Offset					
dword 3	Reserved(1)					

FOOTNOTE:
(1) All reserved bits must be written with 0s for forward compatibility.
(2) The two LSBs must be equal to zero for dword alignment.

Table 3-2. EBUS Service Request Field Descriptions

Dword Number	Descriptor Field	Size (Bits)	Value	Description
dword 0	OPCODE	5	6	EBUS Write command (EBUS_WR)
			7	EBUS Read command (EBUS_RD)
	SACKIEN	1	—	Enable (1) or disable (0) acknowledge via interrupt in the end of the command execution
	Reserved	7	0	Reserved bits should be written with 0s.
	FIFO_BURST	1	0	Do increment EBUS address (address on the target device) by one after each EBUS access. This is used to access a continuous segment or block of memory on the target device that is connected to the EBUS.
			1	Do not increment EBUS address for this access. On some devices, memory accesses are carried out the writing/reading of one memory location. By setting FIFO_BURST to one, CX28500 does not increment the EBUS address after an access. Hence, the address stays the same for the next EBUS access.
	EBUS Byte Enable (EBE)	4	—	The value driven over EBE[3:0]*. Each bit controls a corresponding byte access on the EBUS. For example, an EBE[3:0] value of 0001 means that Host data passes to the device attached to the EBUS on byte 0, the least significant byte, of the EBUS while the other three bytes are inaccessible.
	Length	14	—	Number of EBUS transactions.
dword 1	Shared Memory Pointer	32	—	The Shared Memory Pointer (Buffer Address) is a dword–aligned address of the first buffer to or from which data needs to be transferred from or to the EBUS. The two LSB's must be equal to zero for dword alignment.
dword 2	EBUS Base Address Offset	32	—	The EBUS Base Address Offset is the address for the first EBUS transaction.

The Shared Memory Pointer (Buffer Address) is a dword–aligned address of the first buffer to or from which data needs to be transferred from or to the EBUS. The EBUS Base Address Offset is the address for the first EBUS transaction. In the Access Control Field, the LENGTH bit field contains the information of the number of bytes transferred over the PCI. The maximum PCI burst read or write of EBUS transactions is 32 dwords.

When an EBUS_RD is issued, the CX28560 executes a PCI-bursted write of EBUS transactions and will store the data (EAD[31:0]) in an internal buffer.

When the EBUS transaction ends, the CX28560 bursts the data over the PCI to the location specified by Shared Memory Pointer (Buffer Address). The EBE[3:0]* drives the programmed Byte Enabled (BE) value set in the Access Control Field dword. If EBE[3:0]* is different from 0000, the Host must determine which bytes are valid.

If an EBUS Write command is enabled, the CX28560 transfers—via a PCI burst read—the data from the host memory into an internal buffer. The data is transferred over the EBUS in a series of write transactions. The EBE[3:0]* drives the programmed value Byte Enabled (BE) value set in the Access Control Field dword. If EBE[3:0]* is different from 0000, the host must insert the valid bytes into the appropriate location.

3.1.2 Clock

The ECLK, Expansion Bus Clock, is an inverted version of the PCI clock. The signal is output on the ECLK signal line. Whether or not a device on the EBUS requires a synchronous interface, the ECLK signal is available all the time the PCI clock is available (PCLK). The EBUS clock output can be disabled by appropriately setting the ECKEN bit field in EBUS Configuration register. If ECLK is disabled, the ECLK output is three-stated.

After PCI reset, the ECLK output pin is three-stated and the ECKEN field in EBUS Configuration register is cleared.

3.1.3 Interrupt

Similar to the CN28500, but unlike previous HDLC controllers (CN8478/CN8474/CN8472), the CX28560 is not connected to the EINT* pin of the EBUS. The EBUS interrupt line should be connected to PCI interrupt INTB* directly, if it is needed.

3.1.4 Address Duration

The CX28560 can extend the duration that the address bits are valid for any given EBUS address phase. This is accomplished by specifying a value from 0–3 in the ALAPSE bit field in EBUS Configuration register. The value specifies the additional ECLK periods the address bits remain asserted. That is, a value of 0 specifies the address remains asserted for one ECLK period, and a value of 3 specifies the address remains asserted for four ECLK periods. Disabling the ECLK signal output does not affect the delay mechanism.

Both pre- and post-address cycles are always present during the address phase of an EBUS cycle. The pre-address cycle is one ECLK period long and provides the CX28560 time to transition between the address phase and the following data phase. The pre- and post-cycles are not included in the address duration.

3.1.5 Data Duration

The CX28560 can extend the duration that the data bits are valid for any given EBUS data phase. This is accomplished by specifying a value from 0–7 in the ELAPSE bit field in EBUS Configuration register. The value specifies the additional ECLK periods the data bits remain asserted. That is, a value of 0 specifies the data remains asserted for one ECLK period, and a value of 7 specifies the data remains asserted for eight ECLK periods. Disabling the ECLK signal output does not affect the delay mechanism.

A pre- and post-data cycle is always present during the data phase of an EBUS cycle. The pre-data cycle is one ECLK period long and provides the CX28560 sufficient setup and hold time for the data signals. The post-data cycle is one ECLK period long and provides CX28560 sufficient time to transition between the data phase and the following bus cycle termination. The pre- and post-cycles are not included in the data duration.

3.1.6 Bus Access Interval

The CX28560 can be configured to wait a specified amount of time after it releases the EBUS and before it requests the EBUS. This is accomplished by specifying a value from 0–7 in the BLAPSE bit field in EBUS configuration register. The value specifies the additional ECLK periods the CX28560 waits immediately after releasing the bus. That is, a value of 0 specifies a wait of one ECLK period, and a value of 5 specifies six ECLK periods. Disabling the ECLK signal output does not affect this wait mechanism. The bus grant signal (HLDA/BG*) is deasserted by the bus arbiter only after the bus request signal (HOLD/BR*) is deasserted by the CX28560. As the amount of time between bus request deassertion and bus grant deassertion can vary from system to system, it is possible for a misinterpretation of the old bus grant signal as an approval to access the EBUS. The CX28560 provides the flexibility—through the bus access interval feature—to wait a specific number of ECLK periods between subsequent bus requests.

Refer to EBUS timing diagrams—Figure 9-8, *EBUS Write/Read Cycle, Intel-Style* (Intel) and Figure 9-9, *EBUS Write/Read Cycle, Motorola-Style* (Motorola).

3.1.7 PCI to EBUS Interaction

The CX28560 provides an identical EBUS interface to the CX28500 that is a significant improvement compared to previous HDLC devices (CN8478/CN8474/CN8472). PCI utilization is dramatically improved by enabling the EBUS accesses, reads and writes, to be burst over the PCI bus—when EBUS is extensively used to access EBUS peripheral during normal operation.

3.1.8 Microprocessor Interface

The MPUSEL bit field in EBUS Configuration register specifies the type of microprocessor interface to use for the EBUS.

If Intel-style protocol is selected, the following signals are effective:

- ALE*—Address Latch Enable, asserted low by the CX28560 to indicate that the address lines contain a valid address. This signal remains asserted for the duration of the access cycle.
- RD*—Read, strobed low by the CX28560 to enable data reads out of the device and is held high during writes.
- WR*—Write, strobed low by the CX28560 to enable data writes into the device and is held high during reads.
- HOLD—Hold Request, asserted high by the CX28560 when it requests the EBUS from a bus arbiter.
- HLDA—Hold Acknowledge, asserted high by bus arbiter in response to HOLD signal assertion. Remains asserted until after the HOLD signal is deasserted. If the EBUS is connected and there are no bus arbiters on the EBUS, this signal must be asserted high at all times.

If Motorola-style protocol, the following signals are effective:

- AS*—Address Strobe, driven low by the CX28560 to indicate that the address lines contain a valid address. This signal remains asserted for the duration of the access cycle.
- DS*—Data Strobe, strobed low by the CX28560 to enable data reads or data writes for the addressed device.
- R/WR*—Read/Write, held high throughout read operation and held low throughout write operation by the CX28560. This signal determines the meaning (read or write) of DS*.
- BR*—Bus Request, asserted low by the CX28560 when it requests the EBUS from a bus arbiter.
- BG*—Hold Acknowledge, asserted low by bus arbiter in response to BR* signal assertion. Remains asserted until after the BR* signal is deasserted. If the EBUS is connected and there are no bus arbiters on the EBUS, this signal must be asserted low at all times.
- BGACK*—Bus Grant Acknowledge, asserted low by the CX28560 when it detects BGACK* currently deasserted. As this signal is asserted, the CX28560 begins the EBUS access cycle. After the cycle is finished, this signal is deasserted indicating to the bus arbiter that the CX28560 has released the EBUS.

3.1.9 Arbitration

The HOLD and HLDA (Intel) or BR* and BG* (Motorola) signal lines are used by the CX28560 to arbitrate for the EBUS.

For Intel-style interfaces, the arbitration protocol is as follows (see Figure 9-8, *EBUS Write/Read Cycle, Intel-Style*).

1. The CX28560 three-states EAD[31:0], EBE*[3:0]. WR*, RD*, and ALE*.
2. The CX28560 requires EBUS access and asserts HOLD.
3. The CX28560 checks for HLDA assertion by bus arbiter.
4. If HLDA is found to be deasserted, the CX28560 waits for the HLDA signal to become asserted before continuing the EBUS operation.
5. If HLDA is found to be asserted, the CX28560 continues with the EBUS access as it has control of the EBUS.
6. The CX28560 drives EAD[31:0], EBE*[3:0], WR*, RD*, and ALE*.
7. The CX28560 completes EBUS access and deasserts HOLD.
8. Bus arbiter deasserts HLDA shortly thereafter.
9. The CX28560 three-states EAD[31:0], EBE*[3:0]. WR*, RD*, and ALE*.

For Motorola-style interfaces, the arbitration protocol is as follows (refer to Figure 9-9, *EBUS Write/Read Cycle, Motorola-Style*).

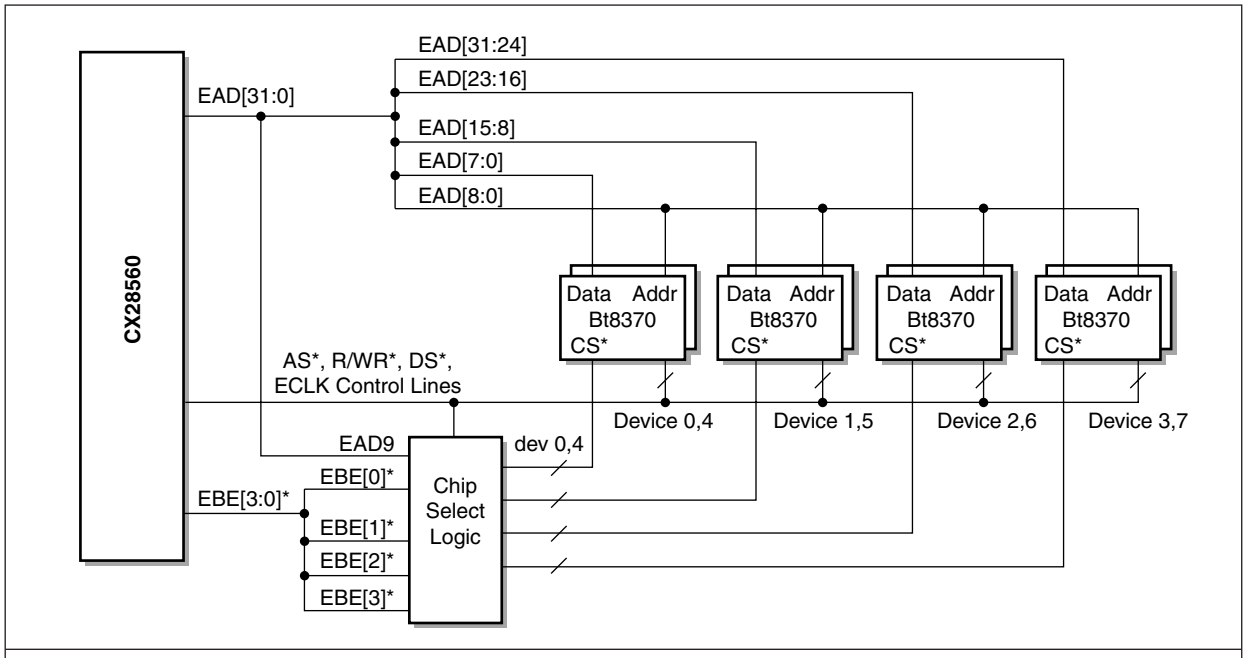
1. The CX28560 three-states EAD[31:0], EBE*[3:0]. R/WR*, DS*, and AS*.
2. The CX28560 requires EBUS access and asserts BR*.
3. The CX28560 checks for BG* assertion by bus arbiter.
4. If BG* is found to be deasserted, the CX28560 waits for the BG* signal to become asserted before continuing the EBUS operation.
5. If BG* is found to be asserted, the CX28560 continues with the EBUS access as it has control of the EBUS.
6. If BGACK* is not asserted, the CX28560 assumes control of the EBUS by asserting BGACK*.
7. The CX28560 drives EAD[31:0], EBE*[3:0], R/WR*, DS*, AS*.
8. Shortly after the EBUS cycle is started, the CX28560 deasserts BR*.
9. Bus arbiter deasserts BG* shortly thereafter.
10. The CX28560 completes EBUS cycle.
11. The CX28560 deasserts BGACK*.
12. The CX28560 three-states EAD[31:0], EBE*[3:0]. R/WR*, DS*, and AS*.

3.1.10 Connection

Using the EBUS address lines, EAD[17:0], and the byte enable lines, EBE[3:0]*, the EBUS can be connected in either a multiplexed or non-multiplexed address and data mode.

Figures 3-3 and 3-4 illustrate two examples of non-multiplexed address and data modes. Figures 3-5 and 3-6 illustrate four and eight separate byte-wide framer devices connected to the EBUS with each byte enable line used as the chip select for separate devices, which allows a full dword data transfer over the EBUS.

Figure 3-3. EBUS Connection, Non-Multiplexed Address/Data, 8 Framers, No Local MPU



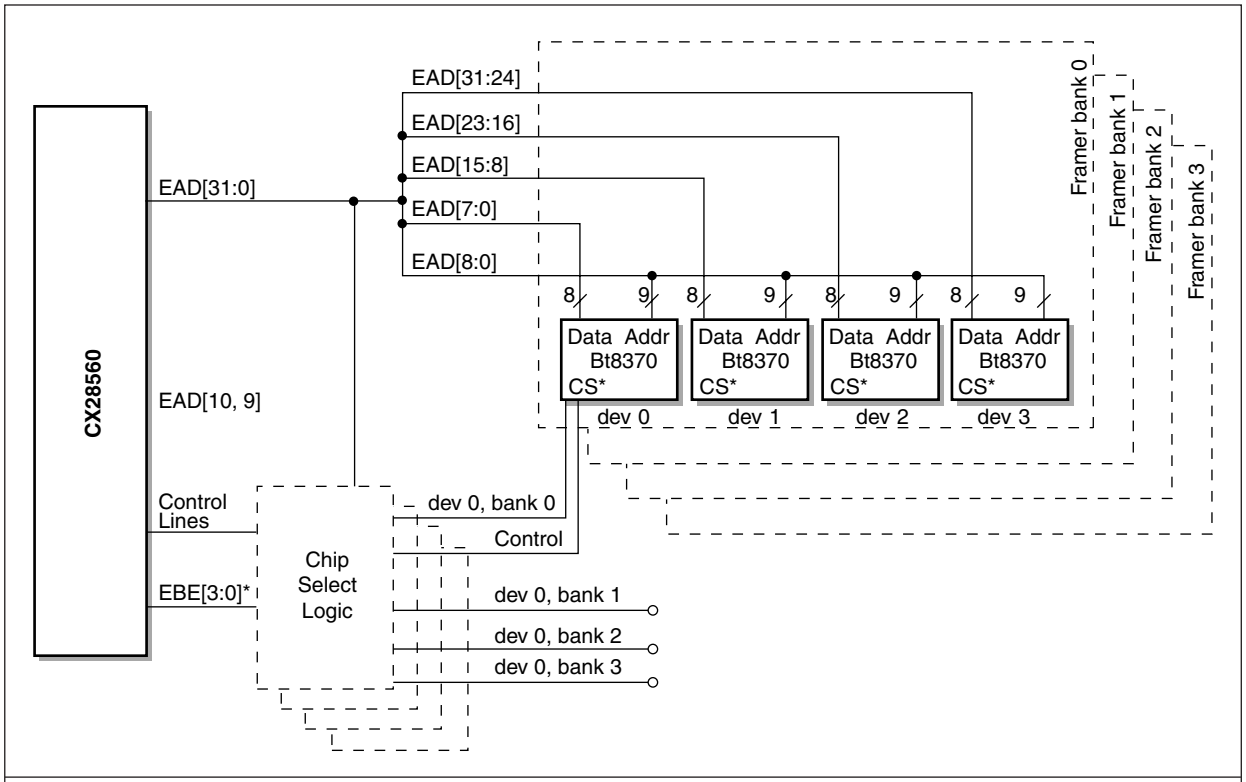
NOTE(S):

1. EBEx[3:0]* selects device x in each framer block.
2. EAD[31:0], AS* are supplied to each framer block.
3. EBEx*, AS* are supplied to each chip select block.

101302_007

The framers configuration in shared memory is that only the Least Significant Byte (LSB) contains the information of one frame configuration; the others are unused.

Figure 3-4. EBUS Connection, Non-Multiplexed Address/Data, 16 Framers, No Local MPU



NOTE(S):

1. EBEx[3:0]* selects device x in each framer block.
2. EAD[31:0], AS* are supplied to each framer block.
3. EBEx*, AS* are supplied to each chip select block.

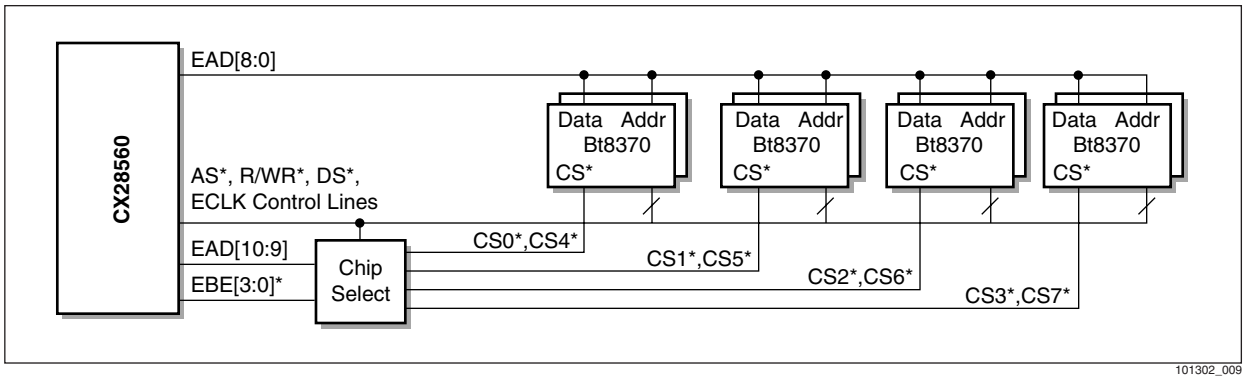
101302_008

In the multiplexed address and data mode, four byte-wide peripheral devices are connected to the EBUS. In this mode, 8 bits of the 32-bit EBUS transfer data to and from each device individually.

NOTE: The multiplexed address and data mode example does not allow for 4-byte data transfers.

Figure 3-5 illustrates the EBUS connection, multiplexed address/data, 8 framers, no local MPU.

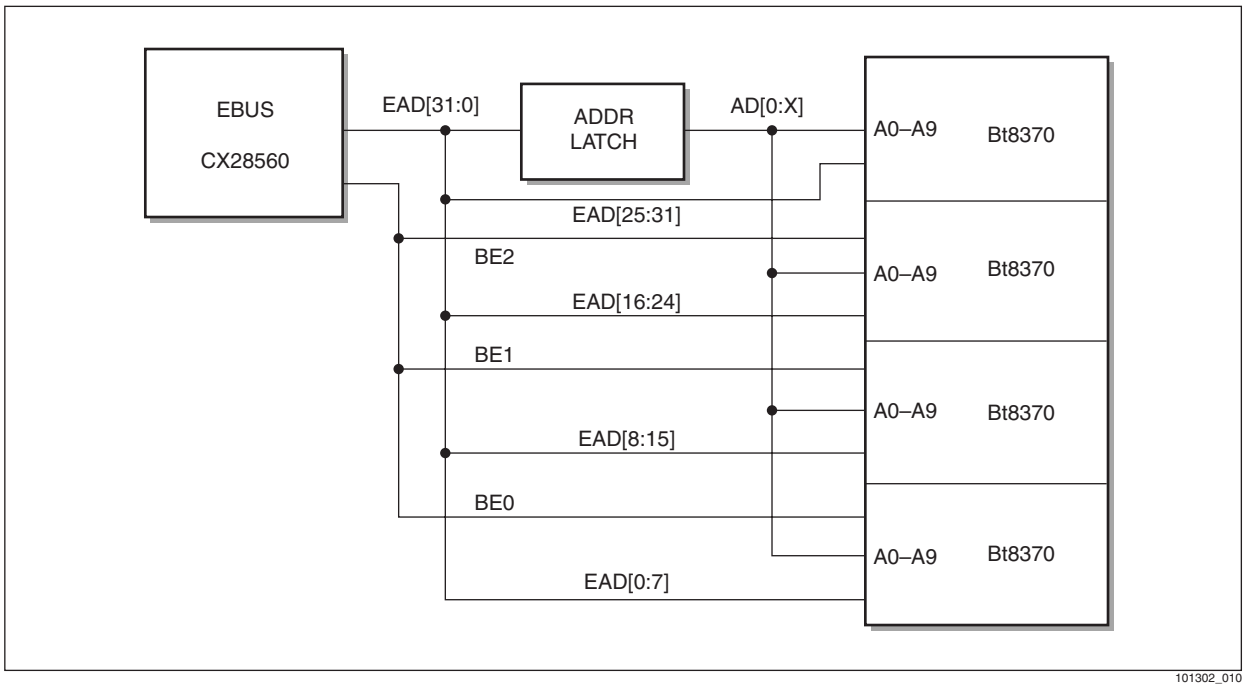
Figure 3-5. EBUS Connection, Multiplexed Address/Data, 8 Framers, No Local MPU



3.1.10.1 Multiplexing Address

Figure 3-6 illustrates the EBUS connections of four 8-bit peripheral devices. The four devices are multiplexing the address in shared memory. The framer's configuration software must read the whole block of framers configuration before it starts demultiplexing data per device.

Figure 3-6. EBUS Connection, Multiplexed Address/Data, 4 Framers, No Local MPU



4.0 CX28560 Serial Interface

4.1 Functional Description

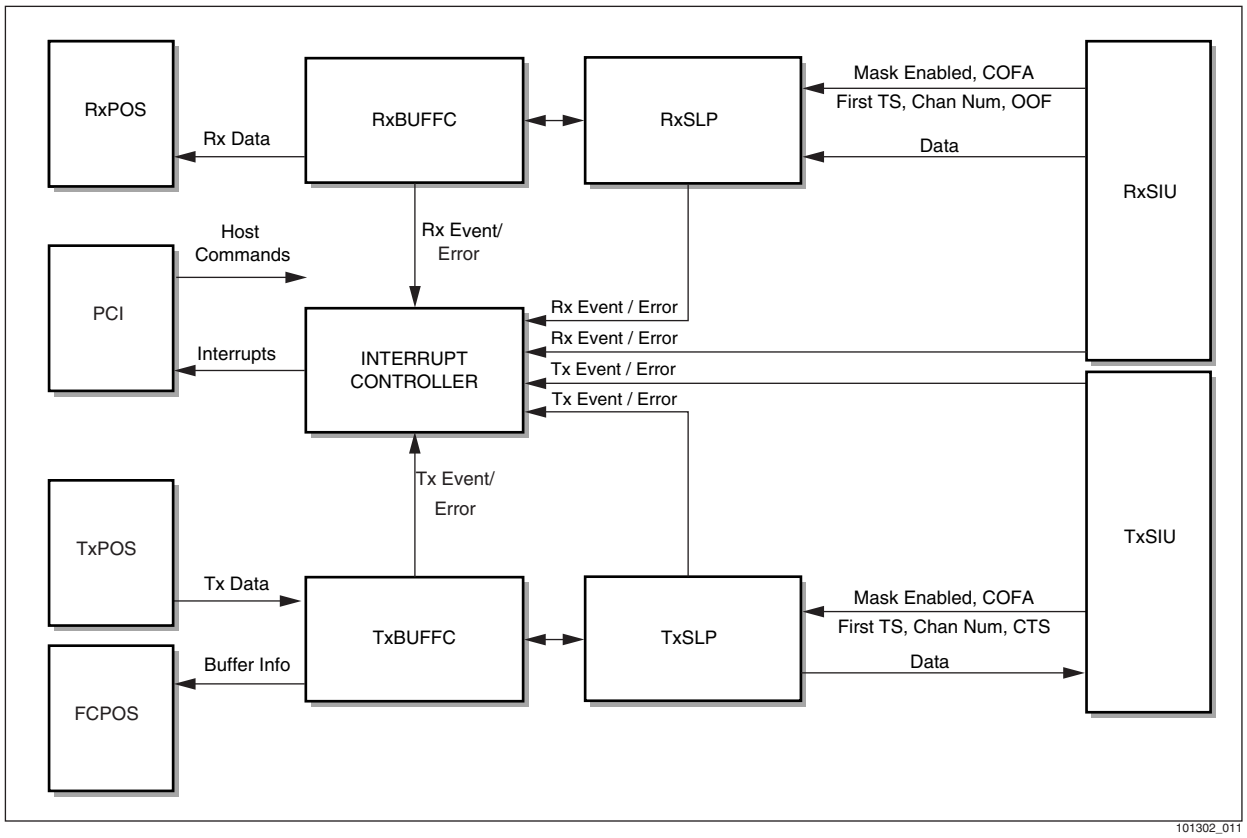
The serial interface consists of the following:

- Serial Interface Unit (SIU), TSIU, and RSIU for the receive and transmit directions
- Serial Line Processing (SLP), TSLP, and RSLP for the receive and transmit directions
- Buffer Controller (BUFFC), RBUFFC, and TBUFFC for the receive and transmit directions
- Interrupt Controller (IC)

A separate set of SIU, SLP, and BUFFC blocks services receive and transmit channels independently. A single Interrupt Controller is shared by the receive and transmit BUFFC, SLP, and SIU blocks.

Figure 4-1 illustrates the different signal connection between SIU and the host interface while it is configured to operate in conventional or with DS0 extraction mode.

Figure 4-1. Serial Interface Functional Block Diagram



4.2 Serial Interface Unit (SIU)

The SIU is the module that logically connects the 32 serial ports (line interface unit) with serial line processing by performing the serial-to-parallel conversion for the receive side, and parallel-to-serial for the transmit side. The SIU contains two main blocks, RSIU for the receive path and TSIU for the transmit path. The RSIU main function is multiplexing 32 serial ports into one logical port for the receive serial line processing block. The TSIU main function is demultiplexing one logical port from the transmit serial line processing block to 32 serial ports.

The SIU main functions:

- Multiplexing/demultiplexing 32 serial ports to 1 port for the receive path, and 1 port to 32 serial ports for the transmit path.
- Performs frame integrity check while operating in channelized mode. SIU verifies the length of incoming/outgoing frames according to the configured number of time slots. In case of error, a Change Of Frame Alignment (COFA) is reported (see Section 4.6.2 and Section 4.7.3).
- Translates the time slot to logical channel number using the configured receive and transmit time slot map.
- In TSBUS mode, in the receive direction, discards stuffed time slots, and in the transmit direction, stuffs time slots as required.
- Generates the following interrupts:
  - RxOOF, when ROOF signal is asserted
  - RxFREC, when ROOF signal is deasserted
  - RxCOFA, when RSYNC signal is asserted at an unexpected time
  - RxCREC, when COFA condition ends on a receive port
  - TxCOFA, when TSYNC signal is asserted at an unexpected time
  - TxCREC, when COFA condition ends on a transmit port.

4.3 Serial Line Processor (SLP)

The serial line processors (RSLP and TSLP) service the bytes in the receive and transmit path. The SLP coordinates all byte-level transactions between SIU and BUFFC. The SLP also interacts with the Interrupt Controller (IC) to notify the host of events and errors during the serial line processing.

The RSLP main features are as follows:

- HDLC mode handling
  - Message delineation—search for opening and closing flag (7Eh)
  - Abort detection (7Fh)
  - Check max/min message length
  - Verify byte alignment
  - Check FCS
  - Detect change of pad-fill
  - Zero deletion
- Transparent mode
  - Start to receive data from the first time slot assigned to the logical channel
- Handle channel activation/deactivation
- Handle OOF/COFA
- Invert incoming data
- Handle subchanneling
- Interrupts
  – BUFF—Channel-specific buffer error (underrun)
  – CHIC—Change to Idle Code—denotes a change of the inter-message pad-fill from an abort sequence (all 1s) to flags (7Eh)
  – CHABT—Change to Abort Code—denotes a change of the inter-message pad-fill from flags (7Eh) to an abort sequence (all 1s).

The TSLP main features are as follows:

- HDLC mode handling
  - Generate opening/closing /shared flag (7Eh)
  - Aborting of packets (generation of abort signal—all 1s)
  - Zero insertion after five consecutive 1s
  - Generate FCS depending upon the protocol
  - Handle CTS
  - Generate pad fill between frames
- Transparent mode
  - Start to transmit data from the first time slot assigned to the logical channel
  - Generate pad fill between messages
- Handle channel activation/deactivation
- Handle COFA
- Invert outgoing data
- Handle subchanneling
- Interrupts
  - BUFF—Channel specific buffer error (underrun)
  - EOM—End of Message

4.4 Buffer Controller

The buffer controllers (RxBUFFC and TxBUFFC) manage all memory operations between the SLPs and the host interface. The BUFFC receives requests from the SLPs either to fill or to flush internal FIFO buffers, provides/receives data for the SLPs, controls the Flexiframe timing scheduler, and transfers data to/from the POS-PHY data interface (through the host interface). In addition, the TxBUFFC communicates with the system across the POS-PHY FlowConductor Interface by sending report packets containing information regarding the amount of space freed in channels' buffers.

The BUFFC main features are as follows:

- Handles up to 2047 logical channels
- Static internal buffer allocation
- User has full control of internal buffer characteristics:
  - standard buffer length can be increased to support longer fragments
  - user-programmable thresholds in the transmit direction
  - FIFO flushing capability (after soft chip reset, channel activation, and channel deactivation service request)
- Message/Fragment handling
  - Addition/interpretation of message and fragment headers
- Automatic Tx Abort Command generation from TERR pin
- Flexiframe characteristics:
  - Time division scheduling scheme
  - Maximum 21,504 K slots per frame
  - Enables dynamic channels reconfiguration
  - Configurable gap between services
  - In the Transmit direction, controls FlowConductor requests
- Receive Performance Monitoring counters:
  - Octets
  - Packets
  - Packets with alignment errors
  - Packets with too short errors
  - Packets with too long errors
  - Packets with FCS errors
  - Packets terminating in an abort
- Transmit Performance Monitoring counters:
  - Octets
  - Packets
  - Packets transmitted terminating in an abort signal
- Receive Interrupts
  - Rx EOM—End Of Message without an error
  - Rx EOM—End Of Message with error (Overflow, OOF, COFA, FCS, ALIGN, ABT, LNG)
  - SHT—Too short. Also used as a general errored message interrupt when data has not yet been passed on for a message. (e.g., a 9-bit message)
- Transmit Interrupts
  - TxBOVFLW—BUFFC channel buffer overflow
  - TxPOSERR—An error occurred on the POS-PHY FlowConductor POS-PHY buffer overflow, Data POS-PHY parity error, Data POS-PHY TxERR pin asserted, or Data POS-PHY internal buffer overflow.

- General Interrupts (generated for both receive and transmit)
  - End of Channel Command (Activation, Deactivation) Execution interrupt
  - NFFRAMEI— Change to the new Flexiframe is complete

4.5 Interrupt Controller

The Interrupt Controller takes receive and transmit events/errors from RxSIU, RxSLP, RxBUFFC, and TxSIU, TxSLP, and TxBUFFC respectively. The Interrupt Controller coordinates the transfer of internally queued descriptors to an interrupt queue in shared memory, and coordinates notification of pending interrupts to the host.

4.6 Serial Port Interface Definition in Conventional Mode

A Receive Serial Port Interface (RSIU) connects to four input signals: RCLK, RDAT, RSYNC, and ROOF. A Transmit Serial Port Interface (TSIU) connects to three input signals and one output signal: TCLK, TSYNC, TCTS, and TDAT, respectively. The SIU receives and transmits data bytes to the Transmit Serial Line Processor (TSLP) and the Receive Serial Line Processor (RSLP). The receive and transmit data and synchronization signals are synchronous to the receive and transmit line clocks, respectively.

The CX28560 can be configured to sample in and latch out data signals, and sample in status and synchronization signals on either the rising or falling edges of the respective line clock, namely RCLK and TCLK. This configuration is accomplished by setting the ROOF_EDGE, RSYNC_EDGE, RDAT_EDGE, TSYNC_EDGE, and TDAT_EDGE bit fields. The default, after reset, is to sample in and latch out data synchronization and status on the falling edges of the respective line clock.

The port mode is configured by programming the RPORT_TYPE and TPORT_TYPE bit fields. When configured to operate in conventional mode, the receive and transmit directions are not related to each other, so each direction can be programmed independently of the other.

4.6.1 Frame Synchronization Flywheel

To maintain a time-base, in Conventional mode, the CX28560 uses the TSYNC and RSYNC signals. These signals keep track of the active bit in the current time slot. The mechanism is referred to as the frame synchronization flywheel. The flywheel counts the number of bits per frame and automatically rolls over the bit count according to the programmed mode. The TSYNC or RSYNC input marks the first bit in the frame. The mode specified in the RPORT_TYPE bit field and TPORT_TYPE bit and the start and end address of time slot pointer determine the number of bits in the frame. A flywheel exists for both the transmit and the receive functions for every port.

The flywheel is synchronized when the CX28560 detects TSYNC = 1 or RSYNC = 1, for transmit or receive functions, respectively. Once synchronized, the flywheel maintains synchronization without further assertion of the synchronization signal.

A time slot counter within each port is reset at the beginning of each frame and tracks the current time slot being serviced.

NOTE: In unchannelized mode, the CX28560 ignores the synchronizing signals and the frame synchronization flywheel mechanism is ignored.

4.6.2 Change Of Frame Alignment (COFA)

A Change Of Frame Alignment (COFA) condition is defined as a frame synchronization event detected when it was not expected, and also includes the detection of the first occurrence of frame synchronization in the receive direction. In unchannelized mode, there are no COFA conditions because the TSYNC and RSYNC signals are ignored in this mode.

When the serial interface detects a COFA condition, an internal COFA signal is asserted until the COFA condition is declared off. A COFA condition is declared off when there was a complete frame without an unexpected SYNC pulse. Thus, an internal COFA signal is asserted for at least two frame periods. During the frame period that the internal COFA is asserted, the CX28560's serial line processor (SLP) terminates all messages found to be active during the COFA condition relevant to that port.

Assertion of COFA condition generates a COFA interrupt encoded in the Interrupt Status Descriptor (ISD) toward the host if this interrupt is unmasked (see *RCOFA_EN* or/and *TCOFA_EN bit field*s). If a synchronization signal (SYNC) is received (low to high transition on TSYNC or RSYNC) while the internal COFA is asserted, an interrupt descriptor with the COFA interrupt encoding is generated immediately if this interrupt is not masked. When the internal COFA is deasserted, the CX28560 generates an interrupt descriptor with CREC event encoding if the interrupt is unmasked—this includes the COFA caused by the first sync received in the receive direction.

On assertion of the internal COFA, in the receive direction an end of message status is prepared with the error encoding set to COFA and passed to the system. The receive serial bit stream processing resumes when the COFA condition is declared off. If channels are configured in HDLC mode, channels resume immediately after the COFA condition is declared off. When configured to transparent mode, channels start operating in the first time slot assigned to the logical channel. Thus, after an RxCOFA, no channel recovery action is required because the channel recovers automatically.

In the transmit direction, the TSLP aborts the messages, immediately deactivates the relevant channels, and reports the deactivation to the TBUFFC. The TBUFFC flushes the channels buffer and waits for an activation command. As a recovery channel action, the host must re-activate the channel upon termination of the COFA condition. COFA detection is not applicable in unchannelized mode. When COFA condition occurs, the transmit output is three-stated. If operating in T1 mode, the F-bit may not be three-stated after a COFA condition.

4.6.3 Out Of Frame (OOF)/Frame Recovery (FREC)

The Receiver Out-Of-Frame (ROOF) signal is asserted by the serial interface sourcing the channelized data to the CX28560. This signal indicates that the interface device has lost frame synchronization.

In the case of multiplexed E1 lines (2xE1, 4xE1), any given port ROOF signal may be asserted and deasserted as the time slots are received from an Out-Of-Frame (OOF) E1 followed by an in-frame E1. ROOF assertion is detected by the Receiver Serial Interface (RSIU). If ROOF is asserted (transitions from low to high) and OOFIEN bit field in the RSIU Port Configuration Descriptor is set, an OOF interrupt is generated toward the host.

For each receive HDLC message that encountered an OOF condition, an end of message status is prepared with the error encoding set to OOF and passed to the system. For transparent mode channels, the OOF causes the data that is being transferred to the host to be replaced by an all 1s sequence. No special actions are taken in this case, and the host must rely on the OOF interrupt to learn about the OOF.

One to three time slots after ROOF is asserted, the CX28560 generates an interrupt descriptor with the OOF error encoded in the Interrupt Status Descriptor. While ROOF is asserted, if OOFABT bit field in the RSIU Port Configuration Descriptor is set, the receive process is disabled. Thus, the CX28560 terminates any active messages for all active channels operating over the port; otherwise, the receive process is enabled.

Notice that the OOF signal is examined on a per-time slot basis. Therefore, OOF assertion affects only those logical channels mapped to time slots where OOF is asserted. The remaining time slots on the same serial port are not affected by the OOF assertion on a specific time slot.

As ROOF is deasserted, the CX28560 immediately restarts normal processing on all active channels. One to three time slots after deassertion of ROOF is detected, the CX28560 generates an interrupt descriptor with the FREC (Frame Recovery) interrupt encoding if the interrupt is not masked (OOFIEN = 1, RSIU Port Configuration Descriptor).

4.6.4 General Serial Port Interrupt

ROOF signal can be used as a general serial port interrupt (SPORT). If OOFABT is zero, OOFIEN is set and ROOF signal deasserts, SPORT interrupt is generated, and the data stream processing is not affected. When ROOF transitions from low to high, the SPORT interrupt is cleared.

4.6.5 Channel Clear To Send (CTS)

The CX28560's transmit path can be configured to obey a Channel Clear To Send (CTS) external signal on a per-port basis by enabling the CTS_ENB bit in the TSIU Port Configuration register. CTS is sampled on the specified active edge of TCLK depending on CTS_EDGE.

If CTS is deasserted (low), the channel assigned to the time slot sends continuous idle characters after the current message has been completely transmitted. If CTS is asserted (high), message transmission continues. When configured to operate in CTS mode, the channels of this specific port will not start a new message transmission if the CTS is a logical 0. The channel response time to react to changes in the channel CTS signal is 32 bits.

4.6.6 Frame Alignment

To maintain a time-base, in conventional mode, the CX28560 uses the TSYNC and RSYNC signals. These signals keep track of the active bit in the current time slot. The mechanism is referred to as the frame synchronization flywheel. The flywheel counts the number of bits per frame and automatically rolls over the bit count according to the programmed mode. The TSYNC or RSYNC input marks the first bit in the frame. The mode specified in the RPORT_TYPE bit field and TPORT_TYPE bit field in, and the start and end address of time slot pointer determine the number of bits in the frame. A flywheel exists for both the transmit and the receive functions for every port.

The flywheel is synchronized when the CX28560 detects TSYNC = 1 or RSYNC = 1, for transmit or receive functions, respectively. Once synchronized, the flywheel maintains synchronization without further assertion of the synchronization signal.

The serial data stream that the CX28560 can manage consists of either packetized data or unpacketized data. The CX28560 supports two types of data-stream modes: HDLC and Transparent.

In transparent mode, message processing for every channel begins in the first time slot marked as the first time slot in the channel's frame structure. A user must configure the first time slot in the RSIU Time Slot Configuration Descriptor and TSIU Time Slot Configuration Descriptor.

For a channel configured in HDLC mode—either transmit or receive directions—the channel waits for a synchronization signal from the internal frame synchronization flywheel before starting processing a new message after channel activation.

A frame synchronization signal must be provided once, after that, the CX28560 keeps track of subsequent frame bit location with its flywheel mechanism. The frame alignment is not relevant when the port is configured in unchannelized mode, although in unchannelized mode each time slot is treated as the first time slot. By configuring more than one time slot in unchannelized mode, (i.e., using TTS_ENDAD /RTS_ENDAD and TTS_STARTAD/RTS_STARTAD mechanism to define one frame).

4.7 Serial Port Interface Definition TSBUS Mode

A port operation mode is configured by programming the TPORT_TYPE and RPORT_TYPE bit fields in RSIU and TSIU Port Configuration registers. When configured to operate in TSBUS mode, the receive and transmit directions are tied to each other. The same TPORT_TYPE/RPORT_TYPE must have the same number of time slots configured for each TSBUS port, and TSTB must be programmed the same for both directions, receive and transmit.

4.7.1 TSBUS Frame Synchronization Flywheel

The CX28560 uses the TSTB signal to maintain a time-base that keeps track of the active bit in the current time slot. The mechanism is referred to as the frame synchronization flywheel. The flywheel counts the number of bits per frame and automatically rolls over the bit count according to the programmed mode. The TSTB input marks the first bit in the frame. A flywheel exists for both transmit and the receive directions for each port. The TSTB assertion works the first bit of time slot in the TSBUS frame. The flywheel is synchronized when the CX28560 detects TSTB = 1. Once synchronized, the flywheel maintains synchronization without further assertion of the synchronization signal. A time slot counter within each port is reset at the beginning of each frame and tracks the current time slot being serviced.

4.7.2 TSBUS Group Synchronization Flywheel

In twelve of the CX28560's serial ports, group extraction is supported. This mode will normally be used to extract DS0 signals from a higher level of signal multiplexing, though is fully configurable for any system. The group extraction synchronization uses two extra signals, TGSYNC and RGSYNC, that are found only in the first twelve ports in order to maintain a time-base that keeps track of the active bit in the current time slot within a group. The mechanism is referred to as the group synchronization flywheel.

The mechanism is used when the present time slot as pointed to in the frame synchronization flywheel is configured to be a group time slot. In this case, the group number is retrieved and the group time slot map is referred to. The flywheel is synchronized when the CX28560 generates TGSYNC = 1 or detects RGSYNC = 1. Once synchronized, the flywheel maintains synchronization without further assertion of the group synchronization signal. The flywheel counts the number of time slots per group and automatically rolls over the count. The TGSYNC and the RGSYNC input marks the first time slot of a group. Flywheels exist for both transmit and receive directions for each group.

4.7.3 TSBUS Change Of Frame Alignment (COFA)

There is no COFA detection in TSBUS mode. If a SYNC signal is detected, the flywheel mechanism returns the current time slot pointer to the start of the port's allocation. It is therefore recommended to set the COFAIEN (see Table 5-39, *RSIU Port Configuration Register* and Table 5-53, *TSIU Port Configuration Register*) to 0 for TSBUS ports to avoid receiving a COFA interrupt on the first sync signal.

4.7.4 TSBUS Out Of Frame (OOF)/Frame Recovery (FREC)

There is no Out Of Frame (OOF) condition while operating in TSBUS mode. The ROOF signal is used as a TSTB input pin. For reference see Figure D-1, *CX28560 Time Slot Interface Pins*.

4.7.5 TSBUS Frame Alignment

The serial data stream that the CX28560 can manage consists of either packetized or unpacketized data. The CX28560 supports two types of data-stream modes: HDLC and Transparent.

In transparent mode, message processing for every channel begins in the time slot marked as the first time slot in the channel's structure. Regardless of the channel protocol, the user must configure the first time slot for both receive and transmit directions.

For a channel configured for HDLC mode, either transmit or receive direction, the channel waits for a synchronization signal from the internal frame synchronization flywheel before starting processing new messages after channel activation.

A Frame Synchronization Signal (TSTB) must be provided one time; after that, the CX28560 keeps track of subsequent frame bit location within the flywheel mechanism.

4.7.6 TSBUS Channel Clear To Send

While operating in TSBUS mode, there is no CTS signal because the related input pin is defined to be TSTB (for reference see Figure D-1, *CX28560 Time Slot Interface Pins* and Table 1-6, *Serial Interface (General)*.

4.7.7 TSBUS Interface

The TSBUS is a time slot interface. The digital communication data paths and overhead channels consist of payload data and overhead data derived from either SONET or SDH data streams, and payload and overhead data derived from either electrical DS3 or E3 data streams. One of the overhead channels may consist of HDSL messages generated and received by the Command Status Processor (CSP). The messages are provided by the local processor that is connected to access and configure local device registers. The TSBUS interface is capable of full-duplex (bi-directional) transmission of data between one device and the CX28560 device. The interface consists of two, 1-bit wide serial interfaces: a bi-directional payload TSBUS and bi-directional overhead TSBUS.

A TSBUS frame structure is defined as an integer multiplication of bytes. A TSBUS port can be either DS0 extraction (group extraction and synchronization is performed) or non-DS0 extraction (group extraction and synchronization is not performed).

When a port is defined as TSBUS non-DS0 extraction, its interface is defined by seven signals. When a port is defined as TSBUS DS0 extraction, its interface is defined by nine signals—the standard seven signals from the non-DS0 extraction mode, and an extra two group synchronization signals.

In the TSBUS mode, the Rx and Tx are synchronous (i.e., the first bit of Tx and Rx frame is sampled on the falling or rising edge of RCLK/TCLK on TDAT/RDAT). TSTB defines the frame synchronization, which marks the first bit of the Rx/Tx frame. TSTUFF acts as a flow control signal that indicates “stuff” (update) to be sent on the following time slot mapped to the logical channel. TSTUFF is sampled in the first two bits of the channel’s time slot.

In the TSBUS transmit direction, the CX28560 requires the stuff status for each time slot to be presented at its TSTUFF input exactly eight time slots in advance of the actual time slot for which the stuff status is applied. The amount of the TSTUFF advance is fixed at eight time slots even though the number of time slots within a frame might vary. In DS0 extraction mode, an extra signal (TGSYNC) performs group synchronization.

For the receive direction, CX28560 requires the stuff status for each time slot to be presented at its RSTUFF input on the current time slot for which the stuff is applied. In DS0 extraction mode, an extra signal (RGSYNC) performs group synchronization.

4.7.7.1 Payload TSBUS

Examples of the payload TSBUS operates at a data rate of 51.84 Mbps. It carries the data path signals derived from either SONET, SDH, electrical DS3 or the electrical E3 signals. The data on the payload TSBUS is framed and consists of 84 time slots.

Payload data paths are as follows:

- SONET/SDH Payload
- Electrical DS3 to DS1—28 x DS1 (672 time slots; F-bits not mapped with DS1 signals)
- Electrical E3 to E1—16 x E1 framers (496 time slots; time slot 0 not mapped)
- STS-1 to DS3/E3 to 28 x DS1 (672 time slots)/21 x E1 (651 time slots)
- 28 x DS1—672 time slots
- 21 x E1—651 time slots
- 16 x E1—496 time slots
- VT1.5—672 time slots
- VT2.0—651 time slots
- VT1.5 to DS1—672 time slots
- VT2.0 to E1—651 time slots
- TUG-2 to DS1—672 time slots
- TUG-2 to E1—651 time slots

4.7.7.2 Overhead TSBUS

Examples of the overhead TSBUS operates at a data rate of 12.96 Mbps. It carries PDH or SDH overhead communication channels and carries the data monitoring and data configuration for the device that communicates through TSBUS interface with the CX28560. The data on the overhead TSBUS is framed and consists of 84 time slots.

The sources and destinations of overhead data transferred to and from the overhead TSBUS are as follows:

- SONET/SDH
  - Section DCCR—2 time slots
  - Line DCCM—4 time slots
  - SPE Path F2 User Data—1 time slot
  - SPE Path F3 User Data—1 time slot
  - SPE N1 Tandem Connection—1 time slot
- DS3/E3 TDL Overhead—1 time slot
- DS1 F-bits—1 time slot
- E1 Si bits—1 time slot
- Command Status Processor (CSP)—13 time slots

TSBUS References

For a detailed description of the TSBUS interface, see *CX29503 Broadband Access Multiplexer* (document #100702A), section 2.10.

5.0 The CX28560 Memory Organization

The CX28560 interfaces with a system host by the transfer of data as fragments of packets over a dedicated data bus. The CX28560 also contains a set of internal registers that the host can configure over a PCI bus, which control the CX28560. In addition, a unidirectional flow control bus is used to monitor the amount of data in the CX28560's internal transmit buffers. This section describes the various data headers, flow control packets and the layout of individual registers that are required for the operation of the CX28560.

5.1 Memory Architecture

The CX28560 transfers data as fragments of packets prefixed with a fragment header. The fragments are transferred to the host over a dedicated data bus.

Configuration commands and monitoring information are stored in a shared memory from which both the host and the CX28560 write and read. This assumes a system topology in which a host and the CX28560 both have access to shared memory for data control. The host allocates and de-allocates the required memory space.

5.1.1 Register Map and Shared Memory Access

During the CX28560's PCI initialization, the system controller allocates a dedicated 1 MB memory range to the CX28560. The memory range allocated to the CX28560 must not map to any other physical or shared memory. Instead, the system configuration manager allocates a logical memory address range and notifies the system or bus controllers that any access to these ranges must result in a PCI access cycle. The CX28560 is assigned these address ranges through the PCI configuration cycle. Once configured, the CX28560 becomes a functional PCI device on the bus.

As the host accesses the CX28560's allocated address ranges, the host initiates the access cycles on the PCI bus. It is up to individual the CX28560 devices on the bus to claim the access cycle. As the CX28560's address ranges are accessed, it behaves as a PCI slave device while data is being read or written by the host. The CX28560 responds to all access cycles where the upper 12 bits of a PCI address match the upper 12 bits of the CX28560's Base Address register (see Chapter 2.0, PCI Register 4, Address 10h).

For the CX28560, a 1 MB-memory space is assigned to the CX28560 Base Address register, which is written into PCI configuration space Address 10h, register 4 in PCI Configuration registers. Once a base address is assigned, a register map is used to access individual device resident registers. The CX28560 cannot respond to an access cycle that the CX28560 itself initiates as the bus master. The register map provides the byte offset from the Base Address register where registers reside. The register map

layout is given in Table 5-1. It should be noted that there are two address spaces. The first one includes the registers that are directly accessed by the host through the PCI (direct access) and the second includes the registers that are accessed through the Service Request Mechanism (indirect access).

The only registers that can be directly accessed by the host as slave reads or writes are the Rx Port Alive, the Tx Port Alive, the Interrupt Status Descriptor, the Interrupt Queue Pointer, the Interrupt Queue Length, the Service Request Length, the Service Request Pointer, and the Soft Reset registers. These are specified in Table 5-1. When the host writes directly into a corresponding register, the CX28560 behaves as a PCI slave while this write is performed.

All other registers need to be accessed through the Service Request Mechanism. After the PCI reset, when the CX28560 is ready for configuration, these registers are updated with the appropriate shared memory values through a Configuration Write Service Request. After the host has configured the shared memory image of the CX28560's registers, and the CX28560 has finished its local configuration (i.e., SRQ_LEN bit field in Service Request Length is reset to zero by the CX28560), the host issues a service request by writing directly into the Service Request Length register. Writing to this location the actual value of the Service Request Descriptor Table Length from shared memory causes the CX28560 to start performing the Service Request Descriptor Table.

PCI Configuration Register 4

CX28560 **Base Address Register (BAR)**

Table 5-1. PCI Register Map (Direct Access)

Register	Access Type	Byte Offset	Number of Instances	Reset Value
Receive Port Alive Register	RO	00000h	(bit) Per Port	0
Transmit Port Alive Register	RO	00004h	(bit) Per Port	0
Interrupt Status Register	R/W	00008h	Per Chip	0
Interrupt Queue Pointer	R/W	0000Ch	Per Chip	0
Interrupt Queue Length	R/W	00010h	Per Chip	0
Service Request Length Register	R/W	00014h	Per Chip	0
Service Request Pointer Register	R/W	00018h	Per Chip	0
Soft Chip Reset Register	WO	00020h	Per Chip	0

NOTE(S):

1. There are two address spaces: The first address space includes registers that are directly accessed by host through the PCI. The second address space (shown in Table 5-2) represents the CX28560's register map accessible to the Service Request Mechanism. Therefore, all the registers shown in this table can be directly read or write by the host.
2. Although the post reset value of the Service Request Length is 0, until the CX28560 has finished all initializations, the value shown in the SRQ_LEN field of this register will be all 1s.

Table 5-2. Indirect Register Map Address Accessible via Service Request Mechanism (1 of 2)

Register	Access Type	Descriptor Address (22 bits)	Number of Instances	Reset Value
RBUFFC Flexiframe Memory	RW	000000–0053FF	21K Per Chip	X
RBUFFC Counter Memory	RO	008000–00BFFF	8 Per Channel	X
RBUFFC Channel Configuration Register	RW	00C000–00C7FF	1 Per Channel	X
RBUFFC DATA FIFO Size Register	RW	00FFFB	1 Per Chip	X
RBUFFC Flexiframe Control Register	RW	00FFFC	1 Per Chip	X
RBUFFC Fragment Size Register	RW	00FFFD	1 Per Chip	X
RBUFFC Flexiframe Slot Time Register	RW	00FFFE	1 Per Chip	X
RSLP Channel Status Register	RO	050800–050FFF	1 Per Channel	X
RSLP Channel Configuration Register	RW	051000–0517FF	1 Per Channel	X
RSLP Maximum Message Length Register 1	RW	053FFD	1 Per Chip	X
RSLP Maximum Message Length Register 2	RW	053FFE	1 Per Chip	X
RSLP Maximum Message Length Register 3	RW	053FFF	1 Per Chip	X
RSIU TS/Group Map	RW	094000–095FFF	8K Per Chip	X
RSIU Group Map	RW	096000–097FFF	8K Per Chip	X
RSIU Group Map Pointer Allocation Register	RW	098000–0981FF	1 Per Group (512)	X
RSIU Group State Register	RW	098200–0983FF	1 Per Group (512)	X

Table 5-2. Indirect Register Map Address Accessible via Service Request Mechanism (2 of 2)

Register	Access Type	Descriptor Address (22 bits)	Number of Instances	Reset Value
RSIU Time Slot/Group Map Pointer Allocation Register	RW	09BFC0–09BFDF	1 Per Port	X
RSIU Port Configuration Register	RW	09BFE0–09BFFF	1 Per Port	X
TBUFFC Counter Memory	RO	0DC000–0DDFFF	4 Per Channel	X
TBUFFC Channel Configuration Register	RW	0DF000–0DFFFF	2 Per Channel	X
TBUFFC Flexiframe Memory	RW	0E0000–0E53FF	21K Per Chip	X
TBUFFC Data FIFO Size Register	RW	0E7FFC	1 Per Chip	X
TBUFFC Flexiframe Control Register	RW	0E7FFD	1 Per Chip	X
TBUFFC Flexiframe Slot Time Register	RW	0E7FFE	1 Per Chip	X
TSLP Channel Status Register	RO	128800–128FFF	1 Per Channel	X
TSLP Channel Configuration Register	RW	129000–1297FF	1 Per Channel	X
TSIU TS/Group Map	RW	16C000–16DFFF	8K Per Chip	X
TSIU Group Map	RW	16E000–16FFFF	8K Per Chip	X
TSIU Group Map Pointers Register	RW	170000–1701FF	1 Per Group (512)	X
TSIU Group State Register	RW	170200–1703FF	1 Per Group (512)	X
TSIU Time Slot/Group Map Pointer Allocation Register	RW	173FC0–173FDF	1 Per Port	X
TSIU Port Configuration Register	RW	173FE0–173FFF	1 Per Port	X
Transmit POS-PHY Thresholds Register	RW	0E7FF9	1 Per Chip	X
Transmit POS-PHY Control Register	RW	0E7FFF	1 Per Chip	X
Receive POS-PHY Control Register	RW	00FFFF	1 Per Chip	X
EBUS Configuration Register	RW	1B4000	1 Per Chip	X
Global Configuration Register	RW	1B4001	1 Per Chip	X
These registers need to be accessed through the Service Request Mechanism.				

It is critically important that upon channel activation internal registers must be initialized. The CX28560 assumes the information is valid once a channel is activated.

5.2 Global Registers

5.2.1 Service Request Mechanism

The registers that need to be configured and checked to enable the activity of the service request mechanism are as follows:

- Service Request Length register (see Table 5-1)
- Service Request Pointer register (see Table 5-1)

The availability of the device is an example of information provided by querying the Service Request Length register. After the PCI reset, the CX28560 sets the SRQ_LEN bit field in Service Request register to all ones until it performs all the internal initialization. When the CX28560 is finished with the internal initialization, it clears this field to 0. The cleared SRQ_LEN provides to the host the information of the CX28560's readiness. From this point, the host is able to directly write this bit field with the actual number of service requests that the CX28560 needs to perform to configure its registers. The number of SRQs written by the host is stored in the SRQ_LEN bit field. While processing the service request commands, the SRQ_LEN field indicates how many commands are yet to be processed by the CX28560 before a new command can be issued. Host slave writes to this register trigger the execution of the service request list.

NOTE: Host slave writes to SRQ_LEN bit, while the previous list of service requests has not been processed (i.e., SRQ_LEN is reset) implies unpredictable behavior.

Table 5-3. Service Request Length Register

Bit	Field Name	Value	Description
31:10	RSVD	0	Reserved.
9:0	SRQ_LEN[9:0]	—	Service Request Length. After a PCI reset, host reads the SRQ_LEN bit field through PCI slave access. While the SRQ_LEN value equals all 1s, the CX28560 is not ready to start the configuration of the device. If the CX28560 resets this value, the device is ready to be configured. Host directly writes at this location the number of Service Request Descriptors (SRD) which were allocated in Service Request Descriptor Table (SRDT) i.e., shared memory. The SRDs used to be previously initialized and configured in SRDT. This value represents the number of service request commands queued by host (i.e., the SRDT), that are waiting to be performed. Real-time reads from SRQ_LEN provides the number of service request commands that are waiting to be served.

The Service Request Pointer register provides the address of the Service Request Descriptor Table (see Table 5-4). Host needs to allocate and initialize this table in shared memory.

Table 5-4. Service Request Pointer Register

Bit	Field Name	Value	Description
31:3	SRQ_PTR[31:3]	—	Service Request Pointer. These 29 bits are appended with 000b to form a 32-bit address Quadword aligned. This address points to the first entry on the Service Request Descriptor Table allocated in shared memory.
2:0	SRQ_PTR[2:0]	0	To ensure Quadword alignment.

5.2.1.1 Service Request Descriptors

A Service Request Descriptor (SRD) is a 4-dword location in shared memory. Actually, one represents an entry in the SRD table. The SRD is defined as a union type in C, which allows different commands to be configured in a 4-dword space. The SRD can handle three different configurations: Device Configuration Descriptor (DCD), EBUS Configuration Descriptor (ECD), and Channel Configuration Descriptor (CCD).

A list of service request commands is defined as a sequence of SRDs. The following instructions, referred to in the document as OPCODE, are supported:

- Configure a port/channel
- Read the CX28560 register or counters
- Expansion Bus (EBUS) read command
- EBUS write command
- Activate a channel
- Deactivate a channel
- No-operation command

Table 5-5 defines the Service Request Descriptor OPCODE.

A service request is issued to a specific channel, or per whole device. On completion of each service request command, an acknowledgment interrupt is generated (Service Acknowledge - SACK) and sent to the host. This interrupt may be disabled per service request by setting the SACKIEN bit of the SRD to 0. It is possible for the host to issue multiple service requests successively without expecting or receiving acknowledgments from each request if the SACKIEN bit was not set accordingly in the SRD.

One mode of operation is for the host to set the SACKIEN bit in only the last SRD so if a SACK Service Request Acknowledge interrupt is received, it will validate the whole list of service request commands.

Activate and Deactivate commands could take a long time before they are actually executed by the CX28560. The CX28560 returns the SACK (if SACKIEN bit is set) immediately after it started the command execution. Therefore, the host may not assume the command was actually executed just by detecting the SACK was returned. Another interrupt, End Of Command Execution (EOCE), is defined for each of these commands. The host may assume the command was actually executed only after receiving the appropriate EOCE.

A similar situation arises when performing a change of Flexiframe. The SACK interrupt (if enabled) will be returned once the new Flexiframe has been read into the CX28560's internal memory. However the system can only assume that the actual move to use the new Flexiframe has been made once the RNFFRAME or TNFFRAME bit (see Table 5-29 or Table 5-44) has been set to zero or the NFFRAMEI interrupt has been received.

Table 5-5. Service Request Descriptor—OPCODE Description

Command	Value	Description
NOP	0h	No Operation. This service request performs no action other than to facilitate a host Service Acknowledge Interrupt (SACK). This would be used as a UNIX ping-like operation to detect the presence of the CX28560.
CONFIG_WR	1h	Configuration Write. This is a request to copy from shared memory data into the CX28560's internal registers. This service request can be issued for one or more consecutive registers, depending on the value of LENGTH bit field set in Service Request Descriptor. Note: The Service Request Descriptor used for this command is Device Configuration Descriptor. The LENGTH bit value in this descriptor is up to 16 K. Assuming that the host configures an 16 K register structure in shared memory and the LENGTH bit field will be set accordingly. Note that over the PCI the configuration will be in bursts of 32 dwords (i.e., the maximum allowed PCI burst).
CONFIG_RD	2h	Configuration Read. This is a request to copy the configuration of the CX28560's internal register(s) into shared memory. The configuration located at the address specified by the CX28560 register Map Base Address Offset is read and copied to the address specified by the shared memory address. The number of Dwords copied is specified in the LENGTH bit field. The user needs to instruct the CX28560 to perform the correct number of reads so that when data is written in shared memory, no data overlapping occurs. The Service Request Descriptor used for this command is Device Configuration Descriptor.
CH_ACT	3h	Channel Activation. This is a request to activate a single channel. The CX28560 assumes that the channel was already configured. If the channel is currently active, this command results in a destructive termination of the current message being processed, as well as flushing any other messages residing in the channel's FIFO. The Service Request Descriptor used for this command is Channel Configuration Descriptor.
CH_DEACT	4h	Channel Deactivation. This is a request to deactivate a channel. This command results in a destructive termination of the current message being processed, as well as flushing of any other messages residing in the channel's FIFO. The SRD used for this command is Channel Configuration Descriptor.
RSVD	5h	Reserved.
EBUS_WR	6h	EBUS Write. This is a request to execute write transaction(s) over the EBUS. Data is copied from host memory to the EBUS.
EBUS_RD	7h	EBUS Read. This is a request to execute read transaction(s) over the EBUS. Data is copied from the EBUS Address specified in the 3rd dword of EBUS Configuration Descriptor to the shared memory location specified in the 2nd dword of EBUS Configuration Descriptor. The data length copied from one location to another location is specified by LENGTH bit field in EBUS Configuration Descriptor. Note: The EBUS_RD and EBUS_WR Service Request mechanism allow a maximum of 16 K dwords transfer to/from the EBUS. The transaction is split to bursts of 32 dwords over the PCI.
RSVD	8h–1Fh	Reserved.

5.2.1.2 Service Request Descriptors

Each SRD is 4 DWords. The SRDs used by the CX28560 are as follows:

- Device Configuration Descriptor (DCD)
- EBUS Configuration Descriptor (ECD)
- Channel Configuration Descriptor (CCD)

Device Configuration Descriptor (DCD)

Table 5-6 presents the structure of DCD.

Table 5-6. Device Configuration Descriptor

Dword Number	Bit 31				Bit 0
dword 0	OPCODE[31:27]	SACKIEN[26]	Reserved[25:14]	LENGTH[13:0]	
dword 1	Shared Memory Pointer				00
dword 2	Reserved[31:24]		Indirect Register Map Address[23:2]		00
dword 3	Reserved				

Table 5-7 describes these fields.

Table 5-7. DCD Field Descriptions

Descriptor Field	Size	Description
OPCODE	5	Command requested by the host. (CONFIG_WR, CONFIG_RD)
SACKIEN	1	0 = SACK interrupt disabled. 1 = SACK interrupt enabled. An appropriate interrupt is generated after the command is completed.
LENGTH	14	Number of double words in the memory transaction request. If ‘0’ the number of transfers is 16 K. Therefore, it allows for any number of dwords from 1–16384.
Shared Memory Pointer	30+2	Shared memory base address for a memory transaction request. The pointer is dword aligned by concatenating two zeros to the LSB and making it a 32b pointer.
Indirect Register Map Address	22	The register address for the configuation read or write request.

EBUS Configuration Descriptor (ECD)

Table 5-8 presents the EBUS Configuration Service Request Descriptor.

Table 5-8. EBUS Configuration Service Request Descriptor

Dword Number	Bit 31					Bit 0
dword 0	OPCODE[31:27]	SACKIEN[26]	Reserved[25:19]	INCDIS[18]	BYTE ENABLE[17:14]	LENGTH[13:0]
dword 1	Shared Memory Pointer					0 0
dword 2	EBUS Base					
dword 3	Reserved					

Table 5-9 describes these fields.

Table 5-9. ECD Field Descriptions

Descriptor Field	Size	Description
OPCODE	5	Command requested by the host. (EBUS_WR, EBUS_RD)
SACKIEN	1	0 = SACK interrupt disabled. 1 = SACK interrupt enabled.
LENGTH	14	Number of double words in the memory transaction request..
BYTE ENABLE	4	Determines which byte lanes carry meaningful data. BE [0] applies to byte 0 (LSB). BE[3] Applies to byte 3 (MSB). These bits are active high, i.e. '1' indicates enable, '0' indicates disable
INCDIS	1	Disable EBUS address incrementing for FIFO access. When this bit is set, the CX28560 will access the same address LENGTH times (FIFO access). When this bit is zero, the CX28560 will automatically increment the address transmitted by one for each access performed (i.e., final address will be EBUS Base Address + Length – 1).
Shared Memory Pointer	30 + 2	The address of shared memory EBUS base address, where the configuration of local devices exists. The pointer is Dword aligned (last 2 bits should be set to zero).
EBUS Base Address	32	EBUS base (byte aligned) address for an EBUS transaction.

Channel Configuration Descriptor (CCD)

Table 5-10 presents the structure of CCD.

Table 5-10. Channel Configuration Service Request Descriptor

Dword Number	Bit 31				Bit 0
dword 0	OPCODE[31:27]	SACKIEN[26]	Reserved[25:12]	Rx/Tx[11]	CHANNEL[10:0]
dword 1	Reserved				
dword 2	Reserved				
dword 3	Reserved				

Table 5-11 describes these fields.

Table 5-11. CCD Field Descriptions

Descriptor Field	Size	Description
OPCODE	5	Command requested by the host. (CH_ACT, CH_DEACT, NOP,)
SACKIEN	1	0 = SACK interrupt disabled. 1= SACK interrupt enabled – after completion of the command, a service acknowledge (SACK) interrupt will be generated.
CHANNEL	11	Channel Number. This field is interpreted as a channel number for the CH_ACT and CH_DEACT commands. The field is interpreted as reserved for the NOP command.
Tx/Rx	1	0 = the command is for a receive channel. 1 = the command is for a transmit channel.
RSVD	—	Reserved.

5.2.2 Port Alive Registers

The Receive and Transmit Port Alive registers are read-only registers. These registers can only be accessed via direct PCI transaction. Each bit of the Receive and Transmit Port Alive register represents the device port number. Refer to Table 5-12 and Table 5-13 for these registers.

Table 5-12. Receive Port Alive Register

Bit	Field Name	Value	Type	Description
31:0	RPA[31:0]	—	RO	This register controls the access to the Receive Port Configuration register. If one of the 32 bits is set, then the Receive Port Configuration for that specific port is allowed.

Table 5-13. Transmit Port Alive Register

Bit	Field Name	Value	Type	Description
31:0	TPA[31:0]	—	RO	This register controls the access to the Transmit Port Configuration register. If one of the 32 bits is set, then the Transmit Port Configuration for that specific port is allowed.

These registers operate as a gate which enables or disables the access to the Port Configuration register. If the corresponding bit of the Receive and Transmit Port Alive register is set, a new port configuration for the specified port is allowed.

After a PCI reset or Software Chip reset, all 32 bits of the Receive and Transmit Port Alive register are cleared (set to 0). Each bit is automatically set to 1 after 24–32 serial clock cycles occur on that specific port. After the corresponding bit is set to 1, the host can write to the Port Configuration register. The host cannot program a new port configuration until the corresponding bit/port is set to 1 in the Port Alive register depending upon the direction of receive or transmit.

A proper configuration sequence for accessing the Port Configuration register is as follows:

1. Host polls the Port Alive register for the specific port/direction and waits (24–32 serial clock cycles) until the corresponding bit in the Port Alive register is set (the polled bit is one).
2. Host issues a Service Request (SRQ) Port Configuration command and waits for a Service Request Acknowledge (SACK).
3. Host gets the SACK.

NOTE: Writing to the Port Configuration register causes the corresponding bit from the Port Alive register to be cleared. This bit is automatically set to 1 after 24–32 serial clock cycles occur; therefore, a new port configuration will be allowed.

4. Host checks if a new port configuration is allowed by checking the corresponding bit in the Port Alive register. Go to 1.

5.2.3 Soft Chip Reset Register

This register contains 1 bit. Any write of any value to a Soft Chip Reset (SCR) generates a soft reset for the CX28560. An SCR affects the CX28560 exactly as PCI Reset, except that the PCI block is not reset. No PCI configuration is performed after a SCR.

5.3 Interrupt Level Descriptors

The CX28560 generates interrupts for a variety of reasons. Interrupts are events or errors detected by the CX28560 during processing of the incoming serial data streams. Interrupts are generated by the CX28560 and forwarded to the host for servicing.

The CX28560 gathers the many events and errors (generated by all units such as RBUFFC and TBUFFC, RSLP and TSLP, and SIU) and notifies the host over the PCI. Interrupt Descriptors are generated by the CX28560 and forwarded to the host for servicing. Individual types of interrupts may be masked from being generated by setting the appropriate interrupt mask or interrupt disable bit fields in various descriptors. The interrupt mechanism, each individual interrupt, and interrupt controlling mechanisms are discussed in this section.

5.3.1 Interrupt Queue Register

The CX28560 employs a single Interrupt Queue Register to communicate interrupt information to the host. This register is stored within the CX28560. This register stores the location and the size of an interrupt queue (user configurable) in allocated shared memory where the interrupt descriptors will be directly placed by the CX28560 while acting as a PCI bus master. The CX28560 requires this information to transfer interrupt descriptors to shared memory. All the interrupts are processed by the host, in an Interrupt Service Routine (ISR). The CX28560's PCI interface must be configured to allow bus mastering.

The Interrupt Queue Register (i.e., Interrupt Queue Pointer and Interrupt Queue Length) is initialized by the host via a direct PCI write transaction. After a PCI Reset or Software Chip Reset (SCR), the Interrupt Queue Pointer is the first register that needs to be initialized. A typical initialization procedure is as follows:

1. The host writes in the Interrupt Queue Pointer register allocated by performing a direct write to the address of the Interrupt Queue in shared memory.
2. The host writes in the Interrupt Queue Length by performing a direct write to this location, the value of the interrupt queue length allocated in shared memory.

NOTE: The user can change, at any time, the length of the Interrupt Queue (IQLEN field in the Interrupt Queue Length register) or the pointer value of the Interrupt Queue Pointer (IQPTR field in the Interrupt Queue Pointer register). However, writing to these registers while the chip is operating may result in flushing the interrupts held in the internal FIFO.

Table 5-14. Interrupt Queue Pointer

Bit	Field Name	Value	Description
31:3	IQPTR[31:3]	—	Shared Memory Interrupt Queue Pointer These 29 bits are appended with 000b to form a 64-bit aligned address. This address points to the first entry (Quad-word) of the Interrupt Queue buffer. The host can change this field while the chip is operating. However, this results in flushing all interrupts residing in the internal interrupts FIFO.
2:0	IQPTR[2:0]	0	Ensures 64 bit alignment

Table 5-15. Interrupt Queue Length

Bit	Field Name	Value	Description
31:15	RSVD	0	Reserved
14:0	IQLEN[14:0]	—	Shared Memory Interrupt Queue Length This 15-bit number specifies the length of the Interrupt Queue buffer in Quad-words (i.e., the number of descriptors in the queue). ***NOTE(S):*** 1. The host may change this field while the chip is operating. However, this results in flushing all interrupts residing in the internal interrupts FIFO. After reset, IQLEN is set to 0. This has the effect of blocking all the interrupt processing by the CX28560.

5.3.1.1 Interrupt Descriptors

The interrupt descriptor describes the format of data transferred into the queue. There are two different types of the interrupt descriptor. The first type is used to represent BUFFC's block related interrupts and the second type is used to represent other interrupts. Both types are 64-bit fields. Generically, the interrupt descriptor includes fields for:

- Identifying the source of interrupt from within the CX28560 channel causing the interrupt (1-2047) and direction (receive or transmit)
- Events assisting the host in synchronization channel, port and independent activities
- Errors and unexpected conditions resulting in lost data, discontinued message processing, or prevented successful completion of a service request

All the interrupts are associated with a channel or direction with the following four exceptions:

1. When an OOF or COFA condition is detected on a serial port, only one interrupt is generated for the port until the condition is cleared and the condition reoccurs.

2. The ILOST interrupt bit indicates that an interrupt has been lost internally when the CX28560 generates more interrupt descriptors than can be stored in the Interrupt Queue in shared memory. The latency of host processing of the Interrupt Queue (handling the interrupts in the IRS) can be a factor in this, as can the length of the actual queue. This condition is conveyed by the CX28560 overwriting the ILOST bit field in the last interrupt descriptor in the internal queue prior to being transferred out to shared memory. The bit field is not specific to or associated with the Interrupt Descriptor being overwritten. Only one bit is overwritten and the integrity of the original descriptor is maintained.
3. The PERR interrupt bit indicates that a parity error was detected by the CX28560 during a PCI access cycle. This condition is conveyed by the CX28560 overwriting the PERR bit field in the last interrupt descriptor in the internal queue prior to being transferred out to shared memory. The bit field is not specific to or associated with the interrupt descriptor being overwritten. Only one bit is overwritten and the integrity of the original descriptor is maintained.
4. The POSERR interrupt indicates that either a POS-PHY buffering error occurred (underrun or overflow), or that a parity error was detected on the data in bus (CX28560's Transmit data POS-PHY bus).

The CX28560 has two types of interrupt descriptor. One is the BUFFC Interrupt Descriptor, the other is the Non-BUFFC Interrupt Descriptor.

The following items describe the errors/events reported in the BUFFC Interrupt Descriptor:

- RxEOM (Receive End of Message). This interrupt is accompanied by a message status - RxERR (Errored Message Coding). The message status included in an interrupt can be one of the following:
  - RxNOERR – No errors in the message
  - RxFCS – Frame Check Sequence Error
  - RxBUFF – Overflow
  - RxCOFA – Change Of Frame Alignment
  - RxOOF – Out Of Frame
  - RxABT – Abort Frame
  - RxLNG – Long Message
  - RxALIGN – Byte Alignment Error
- RxEOC/TxEOC (Receive/Transmit End of Command Execution). The RBUFFC or TBUFFC has completed the activation/deactivation of a channel.
- RxNFFRAMEI/ TxNFFRAMEI (Receive/Transmit Change to New Flexiframe Indication). The CX28560 receive/transmit BUFFC has completed the transition to the new Flexiframe.
- RxSHRT (Receive Too Short Message).
- RxPOSERR/TxPOSERR (Receive/Transmit Error).
- ILOST (Interrupt Lost).

The following items describe the errors/events reported in the Non-BUFFC Interrupt Descriptor.

- RxBUFF/TxBUFF (Receive and Transmit Buffer errors underrun and overflow)
- TxEOM (Transmit End Of Message)
- RxCHABT (Receive Change to Abort)
- RxCHIC (Receive Change to Idle)
- RxCOFA/TxCOFA (Receive and Transmit Change Of Frame Alignment)
- RxCREC/TxCREC (Receive and Transmit COFA recovery)
- RxOOF (Receive Out Of Frame)
- SACKERR (Service Acknowledge Error, an attempt to access an illegal address within the CX28560)
- RxFREC/RxSPORT (Receive Frame Recovery, Receive Serial Port Interrupt)

BUFFC Interrupt Descriptor Format

The BUFFC interrupt descriptor is 64 bits wide, and the detailed description of its fields is provided in Table 5-16. The most significant bit in the BUFFC interrupt descriptor is always read as 0.

Table 5-16. BUFFC Interrupt Descriptors Format (1 of 3)

Bit Field	Name	Value	Description
63	TYP	0	Interrupt descriptor—type 0.
62:58	RVSD	—	Reserved.
57:47	CH[10:0]	—	Channel number causing the interrupt.
46:43	RVSD	—	Reserved.
42	DIR	0	Direction—RX.
		1	Direction—TX.
41	RVSD		Reserved.
40	RXEOM/TXBOVFLW	0	No Receive End of Message occurred. No transmit channel internal overflow occurred.
		1	If DIR = 0 (receive)—an End of Message occurred, even if errors were detected (RXEOM) and the RXERR field is relevant. If DIR = 1 (transmit)—a channel's internal buffer overflowed.
39	RXEOC/TXEOC	0	End Of Command execution interrupt was not generated.
		1	The RXEOCT/TXEOCT field is relevant. If DIR = 0 (RX)—End Of Command execution interrupt occurs (RXEOC). If DIR = 1 (TX)—End Of Command execution interrupt occurs (TXEOC).
38	RXEOCT/TXEOCT	0	End of Command Type—Deactivate. This bit is only relevant if RXEOC/TXEOC is set.
		1	End of Command Type—Activate. This bit is only relevant if RXEOC/TXEOC is set.

Table 5-16. BUFFC Interrupt Descriptors Format (2 of 3)

Bit Field	Name	Value	Description
37	RXNFFRAMEI/TXNFFRAMEI	0	Transition to the New Flexiframe has not been completed.
		1	Transition to the New Flexiframe has been completed. If DIR = 0 then this is a RNFFRAMEI, otherwise if DIR = 1 then this indicates a TNFFRAMEI. The channel number field is not valid.
36:35	RVSD	—	Reserved
34:32	RXERR[2:0]		End Of Message Status Decoding. ***NOTE(S):*** This field is only valid if DIR = 0 and RXEOM = 1.
		0	Receiver message error (decoded) - no error.
		1	RXBUFF = Overflow
		2	RXCOFA = Change Of Frame Alignment.
		3	RXOOF = Out Of Frame.
		4	RXABT = Abort Termination. Generated when received message is terminated with an abort sequence (at least seven sequential ones).
		5	RXLNG = Long Message. Generated when received message length (after zero extraction) is greater than selected maximum message size (depended on RSLP Maximum message length registers). Message reception is terminated and further transfer of data to the host is not performed.
		6	RXALIGN = Byte Alignment Error. Generated when message payload size, after zero extraction, is not a multiple of 8 bits. This generally occurs with a FCS error. This interrupt also implies a FCS error. The FCS interrupt will not be generated if the ALIGN interrupt is issued.
		7	RXFCS = Frame Check Sequence Error. Generated when received HDLC frame is terminated with byte aligned 7Eh flag but computed FCS does not match received FCS.
31:27	RVSD	—	Reserved.
26	RXSHRT/TXRSVD	0	A receive too short message interrupt was not generated.
		1	If DIR = 0 (RX)—Rx Short Message occurs. If DIR = 1 (TX)—Reserved

Table 5-16. BUFFC Interrupt Descriptors Format (3 of 3)

Bit Field	Name	Value	Description
25:24	RXPOSERR/TXPOSERR	00	No POS-PHY Error (Receive & Transmit).
		01	Receive: POS PHY error (from URX) ------ Transmit: Flow Conductor POS PHY buffer overflow. The channel number field is not valid.
		10	Receive: Reserved ------- Transmit: Data POS PHY error (due to assertion of error line)
		11	Receive: Reserved ------- Transmit – Data POS PHY Fatal Error. Caused by the detection of either a parity error or the overflow of a POS-PHY internal buffer. The channel number field is not valid.
23:1	RSVD	0	Reserved
0	ILOST	0	No interrupts have been lost.
		1	Interrupt Lost. Generated when internal interrupt queue is full and more interrupt conditions are detected. As the CX28560 has no way to store the newest interrupt descriptors, it discards the new interrupts and overwrites this bit in the last interrupt in an internal queue prior to that interrupt being transferred out to shared memory. The integrity of the descriptor being overwritten is maintained completely.

Non-BUFFC Interrupt Descriptor Format

The Non-BUFFC Interrupt Descriptor is 64 bits wide, and the detailed description of its fields is provided in Table 5-17, *Non-BUFFC Interrupt Descriptors Format*. The most significant bit in the Non-BUFFC Interrupt Descriptor is always read as 1.

Table 5-17. Non-BUFFC Interrupt Descriptors Format (1 of 2)

Bit Field	Name	Value	Description
63	TYP	1	Interrupt descriptor—type 1.
62:58	RSVD	—	Reserved
57:47	CH [10:0]	—	Channel number causing the interrupt
46:43	RSVD	0	Reserved
42	CHDIR	0	Receive Channel Interrupt.
		1	Transmit Channel Interrupt.
41	RXBUFF/TXBUFF	—	Buffer Error. Data is lost. The CX28560 has no place to read or write data internally. If from transmitter, then internal buffer underrun. If from receiver, internal buffer overflows.
40:38	RSVD	0	Reserved
37	RXRSVD/ TXEOM	—	Receive: Reserved Transmit: end of message.
36:34	RSVD	0	Reserved
33	RXCHABT/TXRSVD	—	Receive: change to abort code Set to one when the received pad fill code changes from 7Eh to all ones ---------- Transmit: reserved.
32:30	RSVD	0	Reserved
29	RXCHIC/TXRSVD	—	Receive: change to idle code Set to one when a received pad fill code changes from all ones to 7Eh ---------- Transmit: reserved.
28:25	RSVD	0	Reserved
24:20	PRT [4:0]	—	Port number causing the interrupt.
19:18	RSVD	0	Reserved
17	PRTDIR	0	Receive Port Interrupt.
		1	Transmit Port interrupt.
16	RXCOFA/TXCOFA	—	Change Of Frame Alignment. Set to one when a COFA condition is detected.
15	RXOOF/TXRSVD	—	Receive: Out-Of-Frame. Set to one when serial port is configured in channelized mode and receiver-out-of-frame (ROOF) input signal assertion is detected. ---------- Transmit: reserved.
14	RXFREC/TXRSVD	—	Receive: frame recovery. Set to one when serial port transitions from Out-Of-Frame (OOF) back to in-frame. ---------- Transmit: Reserved.

Table 5-17. Non-BUFFC Interrupt Descriptors Format (2 of 2)

Bit Field	Name	Value	Description
13	RXCREC/TXCREC	—	Receive: COFA recovery. ---------- Transmit: COFA recovery. Set to one when serial port transitions from COFA back to in-frame.
12:4	RSVD	0	Reserved
3	SACK STATUS		SACK status bit. Only valid if the SACK bit is asserted.
		0	No error on exit.
		1	Service Acknowledge Error occurred. An attempt was made to access an illegal address. An illegal address is one that is not defined in any of the memory map registers.
2	SACK	—	BUFFC service acknowledge. Set to one at conclusion of host service, which was processed successfully. In case of an error being executed as a result of a host service, other interrupts may be generated – PERR, for example.
1	PERR	0	No PCI parity errors have been detected.
		1	PCI Bus Parity Error. Generated when the CX28560 detects a parity error on data being transferred into the CX28560 either from another PCI agent writing into the CX28560 or from the CX28560 reading from shared memory. This error is specific to the data phase (non-address cycle) of a PCI transfer while the CX28560. PCI system error signal, SERR*, is ignored by the CX28560. To mask the PERR interrupt, the CX28560's PCI Configuration Space, Function 0, Register 1, Parity Error Response field must be set to 0.
0	ILOST	0	No interrupts have been lost.
		1	Interrupt Lost. Generated when internal interrupt queue is full and more interrupt conditions are detected. As the CX28560 has no way to store the newest interrupt descriptors, it discards the new interrupts and overwrites this bit in the last interrupt in an internal queue prior to that interrupt being transferred out to shared memory. The integrity of the descriptor being overwritten is maintained completely.

5.3.1.2 Interrupt Status Register

The interrupt status register is located in a fixed position in the CX28560's internal register. The CX28560 updates this register after each transfer of interrupt descriptors from its internal queue to the Interrupt Queue in shared memory. The host is required to read this register from the CX28560 before it processes any interrupts. The contents of the interrupt status register are reset on hardware reset or soft chip reset or whenever any field in the Interrupt Queue Register is modified.

NOTE: This internal register is directly accessed by the host.

Table 5-18. Interrupt Status Register

Bit	Field Name	Host Access	Value	Description
31	MSTRABT	R	—	Master Abort. When the CX28560 encounters a PCI abort while operating as a PCI master, it does not attempt to recover from this error. In this case the CX28560 asserts the SERR* signal, and the MSTRABT bit and waits for the host to reset (i.e., PCI reset or Soft reset). This bit is asserted when the target does not assert DEVSEL within a specific PCLK cycles or when the target terminates a transaction in which the CX28560 is the master, with an abort (i.e., assertion of STOP# with a deassertion of DEVSEL) sequence.
30:16	WRPTR 14:0]	R	—	Write Interrupt Pointer. 15-bit Quadword index from start of Interrupt Queue up to where the CX28560 is going to insert the next Interrupt Descriptors. The host may read this value to get the location of the last descriptor, which was not served yet, in the queue. As the queue is circular, care must be taken to ensure roll over at beginning and end of queue. Only the CX28560 updates this value. The WRPTR is a read only bit field.
15	INTFULL	R	0 1	Interrupt Queue Not Full—shared memory. Interrupt Queue Full—shared memory. The host writing ANY value to the RDPTR clears the INTFULL status bit.
14:0	RDPTR[14:0]	R/W	—	Read Interrupt Pointer. 15-bit Quadword index from start of Interrupt Queue up to where the host first unread Interrupt Descriptor resides. The host may read this value to get the location of the first descriptor, which was not served yet, in the queue. As the queue is circular, care must be taken to ensure roll over at beginning and end of queue. Only the host updates this value. The RDPTR is a read/write bit field. ***NOTE(S):*** Writing the value of the RDPTR automatically resets the INTFULL status bit. Therefore, if the value written into RDPTR is the same value as was read from this field, it is assumed that the host has read all the interrupt descriptors.

5.3.2 Interrupt Handling

5.3.2.1 Initialization

Interrupt management resources are automatically reset upon the following:

- Hardware reset
- Soft reset
- Write to Interrupt Queue Pointer by a direct PCI write
- Write to Interrupt Queue Length by a direct PCI write

The CX28560 uses two interrupt queues. One is internal to CX28560 and is controlled exclusively by the DMA block. The other is the Interrupt Queue in shared memory, which is allocated and administered by the host, and written to (filled) by the CX28560.

Upon initialization, the data in the status descriptor is reset to all 0s, indicating the first location for next descriptor, the queue is not full, and no descriptors are currently in the queue. Any existing descriptors in the internal queue are discarded.

The host must allocate sufficient shared memory space for the Interrupt Queue. Up to 64 K dwords of queue space are accessible by the CX28560, setting the upper limit for the queue size. The CX28560 requires a minimum of two quadwords of queue space. This sets the lower limit for the queue size.

The host must store the pointer to the queue and the length in quadwords of the queue in the CX28560 within the Interrupt Queue Descriptor registers. Issuing the appropriate Host service to the CX28560 can do this. As the CX28560 takes in the new values, it automatically resets the controller logic as indicated above. This mechanism can also be used to switch interrupt queues while the CX28560 is in full operation.

5.3.2.2 Interrupt Descriptor Generation

Interrupt conditions are detected in both error and non-error cases. CX28560 makes a determination based on channel and device configuration whether reporting of the condition is to be masked or whether an Interrupt Descriptor is to be sent to the Host. If the interrupt is not masked, CX28560 generates a descriptor and stores it internally prior to transferring it to the Interrupt Queue in shared memory.

The internal queue is capable of holding 512 descriptors while CX28560 arbitrates to master the PCI bus and transfer the descriptors into the Interrupt Queue in shared memory.

As the PCI bus is mastered and after descriptors are transferred out to the shared memory, CX28560 updates the Interrupt Status Descriptor. When CX28560 updates the WRPTR field in the Interrupt Status Descriptor, it asserts the PCI INTA# signal line.

If during the transfer of descriptors, the Interrupt Queue in shared memory becomes full, CX28560 stops transferring descriptors until the Host indicates more descriptors can be written out. CX28560 indicates that it cannot transfer more descriptors into shared memory by setting the bit field INTFULL in the Interrupt Status Descriptor.

In cases where the internal queue is full (either because the Host queue is full or there was not enough PCI bandwidth) and new descriptors are generated, the new descriptors are discarded. CX28560 indicates it has lost interrupts internally by overwriting the bit field ILOST in the last Interrupt Descriptor in the internal queue. The ILOST indication represents one or more lost descriptors.

5.3.2.3 INTA# Signal Line

The Host must monitor the INTA# signal line at all times. An assertion of this line signifies the updating of the WRPTR field in the Interrupt Status Descriptor, indicating that Interrupt Descriptors have been transferred to the Interrupt Queue in shared memory from the internal interrupt queue.

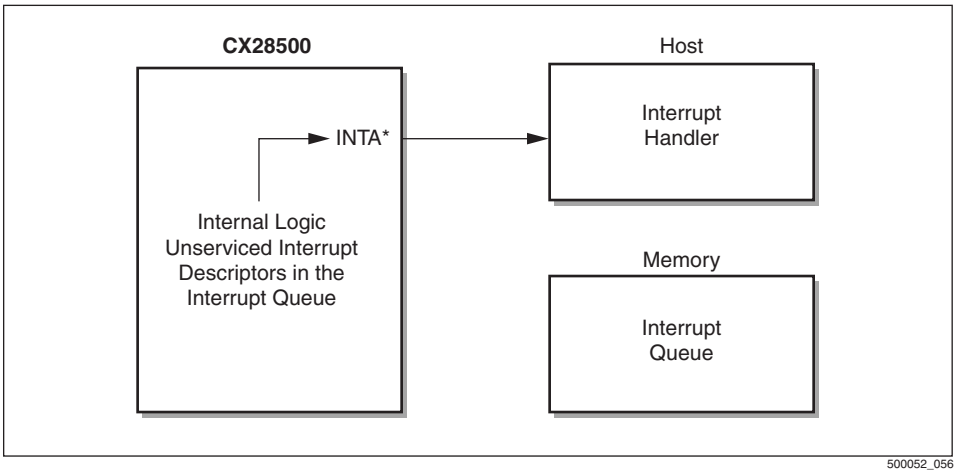
Upon detection of the INTA# assertion, the Host must perform a direct read of the Interrupt Status Descriptor from within CX28560. This descriptor provides the offset to the location of the first descriptor in the Host queue that has not been served, the offset to the location of the last descriptor serviced by the Host, and the determination if the queue is full. The INTA* signal is deasserted on each read of the Interrupt Status Descriptor.

The Host applies its interrupt service routines to service each of the descriptors. As the Host finishes servicing a number of descriptors, it must write the offset to the location of the last serviced descriptor back into the RDPTR field of the Interrupt Status Descriptor. A write to this field indicates to CX28560 that the descriptor locations, which were waiting to be serviced, have been serviced and new descriptors can be written.

NOTE: CX28560 continues to write to available space regardless of whether the Host updates the RDPTR field. The difference between the two interrupt queue pointers RDPTR and WRPTR indicates the number of interrupts still need to be serviced. When calculating the number of outstanding interrupts, please make sure to take care of offsets, or pointers, wraparound.

Figure 5-1 illustrates the operation of INTA*.

Figure 5-1. Interrupt Notification to Host



5.4 Global Configuration Register

The Global Configuration register specifies configuration information applying to the entire device. This register must be programmed before any channel is activated. The only field in this register that can be changed while the chip is operating (i.e., not immediately after reset) is the PCI_EN field.

The components and their descriptors are given in Table 5-19.

Table 5-19. Global Configuration Register

Bit	Field Name	Value	Description
31:14	RSVD	0	Reserved
13	POS-PHY_REG	0	POS-PHY Non-Registered Mode (normal mode). The RxENB/FRENB signal will be sampled according to the POS-PHY standard.
		1	POS-PHY Registered Mode. The RxENB/FRENB signal will be sampled one clock cycle later than defined in the POS-PHY standard.
12	RSVD	0	Reserved
11	PCI_TARGET_FBTB	0	Use the fast back-to-back feature as configured in the PCI configuration settings.
		1	The CX28560 as PCI master attempts to fast-back-to-back the PCI transaction to other targets regardless of PCI configuration settings. This bit is defined to force the CX28560's fast back-to-back capability regardless of the PCI configuration. The PCI specification states that if there is a single device in the system that does not support a fast back-to-back transaction as a target, the fast back-to-back mode is disabled. Setting this bit to 1 instructs the CX28560 to ignore the PCI configuration settings and execute fast back-to-back transactions when appropriate according to the PCI Specification. The host can set this bit only if the CX28560 is always accessing the same target which is capable of fast back to back transactions. This is not a violation of the PCI specification, rather it is an implementation of an allowed behavior.
10	PCI_BR	0	Little-Endian Storage Convention (Intel-style). The least significant byte to be stored in and retrieved from the lowest memory address.
		1	Big-Endian Storage Convention (Motorola-style). An example of little-big Endian byte ordering is shown in Appendix E.
9:1	RSVD	0	Reserved
0	PCI_EN	0	PCI Interrupt disabled—global interrupt mask.
		1	PCI Interrupt enabled

NOTE(S):
1. After reset, the value of Global Configuration register is 0.

5.5 EBUS Configuration Register

The EBUS Configuration Descriptor, defined in Table 5-20, specifies the configuration parameters for EBUS transactions. The host must configure this register before any attempt to access the EBUS.

Table 5-20. EBUS Configuration Register

Bit	Field Name	Value	Description
31:13	RSVD	0	Reserved.
12	MPUSEL	0	Expansion Bus Microprocessor Selection Motorola-style. Expansion bus supports the Motorola-style microprocessor interface and uses Motorola signals: Bus Request (BR*), Bus Grant (BG*), Address Strobe (AS*), Read/Write (R/WR*), and Data Strobe (DS*).
		1	Expansion Bus Microprocessor Selection– Intel-style. Expansion bus supports the Intel-style microprocessor interface and uses Intel signals: Hold Request (HOLD), Hold Acknowledge (HLDA), Address Latch Enable (ALE*), Write Strobe (WR*), and Read Strobe (RD*).
11	ECKEN	0	Expansion Bus Clock Disabled. ECLK output is three-stated.
		1	Expansion Bus Clock Enabled. The CX28560 re-drives and inverts PCLK input onto ECLK output pin.
10:8	ALAPSE[2:0]	—	Expansion Bus Address Duration. The CX28560 extends the duration of valid address bits during an EBUS address phase to ALAPSE+1 number of ECLK periods. The control lines ALE* (Intel) or AS* (Motorola) indicate that the address bits have had the desired set-up time.
7:4	BLAPSE[3:0]	—	Expansion Bus Access Interval. The CX28560 waits BLAPSE number of ECLK periods immediately after relinquishing the bus. This wait ensures that all the bus grant signals driven by the bus arbiter have sufficient time to be de-asserted as a result of bus request signals being de-asserted by the CX28560.
3:0	ELAPSE[3:0]	—	Expansion Bus Data Duration. The CX28560 extends the duration of valid data bits during an EBUS data phase to ELAPSE + 1 number of ECLK periods. The control lines RD* and WR* (Intel) or DS* and R/WR* (Motorola) indicate the data bits have had the desired setup time.

NOTE(S):
(1) After reset, the value of EBUS Configuration register is 0.

5.6 POS-PHY Control Registers

5.6.1 Transmit POS-PHY Thresholds Register

The Transmit POS-PHY Control register provides the necessary parameters for flow control on the POS-PHY interface.

Table 5-21. Transmit POS-PHY Thresholds Register

Bit	Field Name	Value	Description
31	DISBLPAR	0 1	Data arriving on the Transmit POS-PHY will be checked for correct parity. Parity checking is disabled.
31:25	RSVD	0	Transmit POS-PHY thresholds.
24:16	TPTPAHITH	—	PTPA High Threshold. Above this number of dwords (4 bytes) in the buffer, the bus request is deasserted.
15:9	RSVD	0	Reserved
8:0	TPTPALOWTH	—	PTPA Low Threshold. Below this number of dwords (4 bytes) in the buffer, the bus request is asserted.

5.6.2 Transmit POS-PHY Control Register

This register controls the parameter necessary to make the POS-PHY work.

Table 5-22. Transmit POS-PHY Control Register

Bit	Field Name	Value	Description
31:3	RSVD	0	Reserved.
2	TPOSBUFFFULLIEN	0	POS-PHY Buffer Full Interrupt Disabled.
		1	POS-PHY Buffer Full Interrupt Enabled. On encountering full POS-PHY buffers, an interrupt will be generated.
1	TPPARERRIEN	0	POS-PHY Parity Error Interrupt Disabled.
		1	POS-PHY Parity Error Interrupt Enabled. On detection of a parity bit error, a parity error interrupt will be generated.
0	TPERRIEN	0	POS-PHY Error Interrupt Disabled.
		1	POS-PHY Error Interrupt Enabled. When the POS-PHY Error pin is asserted an interrupt will be generated.

5.6.3 Receive POS-PHY Control Register

This register controls the parameter necessary to make the POS-PHY work. It determines whether an interrupt will be generated when the RBUFFC encounters the situation that it tries to send data to the POS-PHY, but there is no room in the POS-PHY buffer. This register is set once for the chip.

Table 5-23. Receive POS-PHY Control Register

Bit	Field Name	Value	Description
31:1	RSVD	0	Reserved.
0	RPOSBUFFFULLIEN	0	POS-PHY Buffer Full Interrupt Disabled.
		1	POS-PHY Buffer Full Interrupt Enabled. On encountering full POS-PHY buffers, an interrupt will be generated.

5.7 Receive Path Registers

Receive path registers contain the information necessary to configure the receive direction. This configuration includes registers that are related to the BUFFC block, host interface, registers that control the RSLP, and RSIU.

5.7.1 RSLP Channel Status Register

The RSLP Channel Status register is a Read Only (RO) register. It provides information from RSLP block regarding the channel state. There is one RSLP Channel status register for each of the CX28560's channels (i.e., 2047 registers).

Table 5-24. RSLP Channel Status Register

Bit	Field Name	Host	Value	Description
31:1	RSVD	R	—	Reserved.
0	RACTIVE	R	0	Channel Inactive. The channel has been deactivated due to either a service request channel deactivation or Reset (PCI Reset or Soft Chip Reset).
			1	Channel Active. The channel has been activated by service request channel activation

5.7.2 RSLP Channel Configuration Register

The Receive Channel Configuration register contains configuration bits applying to the logical channels within the CX28560. There are 2047 such registers, one for each channel. The RSLP Channel Configuration Register configures aspects of the channel common to all messages passing through the channel. One descriptor exists for each logical channel direction. Table 5-25 lists the values and descriptions of each channel configuration descriptor. For each channel to be used in the CX28560, this register must be configured before activation (no default values exist).

Table 5-25. RSLP Channel Configuration Register (1 of 2)

Bit	Field Name	Value	Description
31:30	RPROTCOL[1:0]	0	TRANSPARENT.
		1	HDLC with no FCS. Used in RSLP for full packet forwarding and/or channel monitoring application. For this mode the short message detection is disabled. Any number of bytes can be transmitted and received within any single message including messages of only one byte.
		2	HDLC with FCS16 (FCS—2 bytes).
		3	HDLC with FCS32 (FCS—4 bytes).
29	RINV	0	Data Inversion disabled.
		1	Data Inversion enabled. Message is received from SIU with polarity change (the inversion is done to all bits received).

Table 5-25. RSLP Channel Configuration Register (2 of 2)

Bit	Field Name	Value	Description
28:21	RMASK_SB[7:0]	—	Data Mask. Only bits with a value of 1 contain relevant data (e.g., Mask = 10000001, then only bits 0 and 7 contain channel's data). Enables the sub-channeling feature. Note 0h is an invalid value.
20:5	RSVD	0	Reserved.
4:3	RMAXSEL[1:0]	0	Message Length Check Disabled.
		1	Message Length Check Enabled. Use MAXFRM1 bit field in the message length descriptor for maximum receive message length limit.
		2	Message Length Check Enabled. Use MAXFRM2 bit field in the message length descriptors maximum receive message length limit.
		3	Message Length Check Enabled. Use MAXFRM3 bit field in the message length descriptor for maximum receive message length limit.
2	RFCSTRANS	0	FCS Transfer Normal. Do not transfer received FCS to the host along with data message.
		1	Non-FCS Mode. Transfer received FCS to the host along with data message. In Non-FCS Mode short message detection is disabled.
1	RBUFFIEN	0	Overflow Interrupt disabled.
		1	Overflow Interrupt enabled.
0	RIDLEIEN	0	CHABT, CHIC, SHT Interrupt disabled.
		1	CHABT, CHIC, SHT Interrupt enabled. When the RSLP detects a change to abort or a change to idle code, the relevant interrupt is generated. Setting this bit to 1 is also necessary if the Too Short counter in the RBUFFC is to be used.

5.7.3 RSLP Maximum Message Length Register

The RSLP Maximum Message Length register, defined in Table 5-26, can have three separate values for maximum message length: MAXFRM1, MAXFRM2, and MAXFRM 3. Their structure is shown in RSLP Channel Configuration register. The minimum message length is either 1, 3, or 5 depending on protocol mode: no FCS, 16-bit FCS, or 32-bit FCS, respectively. In the case of a short message, data is not transferred to the host but instead is discarded. In addition, an interrupt descriptor is generated toward the host indicating the short error condition. Note, a bit stream that contains messages of length less than 40 Bytes requires the CX28560 to be configured with large buffers. Although the CX28560 can work with small messages, the buffer calculations and bandwidth calculations have to be re-examined.

Each receive channel either selects one of these message length values or disables message length checking altogether.

The MAXSEL bit field (see Table 5-25) selects which (if any) register is used for received message length checking. If the CX28560 receives a message exceeding the allowed maximum, the current message processing is discontinued and terminates further transfer of data to the host. In addition, a Receive Message Header, corresponding to the partially received message, indicates a Long Message error condition, and an interrupt descriptor is generated toward the host indicating the same error condition.

Table 5-26. Maximum Message Length Register

Bit	Field Name	Value	Description
31:14	RSVD	0	Reserved.
13:0	RMAXFRM[13:0]	—	Defines a limit for the maximum number of bytes allowed in a received HDLC message. Valid values for the register range from 0 to 16 K – 1. The formula to set MAXFRM is: MAXFRM = Max Allowed Message Length (bytes) + FCS (bytes) – 2. Where: FCS = 0 for Non-FCS Mode FCS= 2 byte for HDLC-16 Mode FCS = 4 byte for HDLC-32 Mode. A Too Long Message interrupt is generated when the number of bytes in the processed message exceeds Max Allowed Message Length.
NOTE(S): The host may change the value of Maximum Message Length register only if the channel that uses its value (according to MAXSEL-bit in the configuration memory) is inactive.			

5.7.4 RBUFFC Channel Configuration Register

This register controls the operation mode for a channel. It contains parameters necessary for the division of the internal memory to channel FIFOs. There is one Channel Configuration Register for each logical channel (i.e., 2047).

The CX28560's internal Rx memory is a 320 KB dual RAM, which may be split to 2047 parts, one part for each channel. The allocation granularity is 8 bytes.

In the CX28560, regardless of its bit rate, each channel receives an identical allocation of memory. The difference in bit rates is accounted for by extra servicing of faster channels according to the Flexiframe algorithm. Hence the length of a channel's buffer is set once (see Table 5-30). However, for each active channel it is required to specify the start address of the internal data buffer. (see Appendix E:*Buffer Controller FIFO Size Calculation*)

NOTE: The host must set the buffers so there is no overlap between buffers belonging to different channels. Each receive channel must be allocated buffer space before the channel can be activated.

In addition the end address of each channel must be higher than the start address – no roll-over at the end of the data FIFO is permitted.

Table 5-27. RBUFFC Channel Configuration Register

Bit	Field Name	Value	Description
31:22	RSVD	0	Reserved.
21	REOMIEN	0	End Of Message (without errors) Interrupt Disabled.
		1	End Of Message (without errors) Interrupt Enabled. Any error-free message received will cause this interrupt to be generated.
20	RERRIEN	0	End Of Errored Message Interrupt Disabled.
		1	End Of Errored Message Interrupt Enabled. Any message received containing any error (other than too short) will cause this interrupt to be generated.
19	RTOOSHIEN	0	End Of Message with Too Short Error Interrupt Disabled.
		1	End Of Message with Too Short Error Interrupt Enabled. Any message containing an error for which data has not been passed to the RBUFFC will cause a too short error interrupt to be generated.
18	RCMDCIEN	0	End of Channel Command Execution Interrupt Disabled.
		1	End of Channel Command Execution Interrupt Enabled. On completion of command (activation or deactivation) an interrupt will be generated.
17:16	RSVD	0	Reserved.
15:0	RSTARTADD	—	Channel DATA FIFO Start Pointer—in units of Qwords.

5.7.5 RBUFFC Flexiframe Memory

The RBUFFC Flexiframe Memory provides the RBUFFC with the order in which to service the channels – a timing scheduler. The RBUFFC runs through the Flexiframe Memory line by line, servicing the channel number as in the Flexiframe memory. The Flexiframe holds a maximum of 21504 entries and a minimum of 12. The number of entries contained in the Flexiframe is stored in the Flexiframe Control Register, and should be an exact multiple of 4. The value 0 in the channel number represents an empty cycle and will be treated as a NOP by the RBUFFC. Because of the Flexiframe memory organization in lines of four registers each, access to registers must be in multiples of four registers.

Table 5-28. RBUFFC Flexiframe Memory

Bit	Field Name	Value	Description
31:11	RSVD	0	Reserved.
10:0	RCHANNEL	—	Logical Channel Number assigned to slot in Flexiframe.

5.7.6 RBUFFC Flexiframe Control Register

This register contains the characteristics of the Flexiframe being programmed. When moving to a new Flexiframe this register is vital for the smooth transition. In order to swap to a new Flexiframe, the host should write the new Flexiframe to the Table 5-28, then write the Flexiframe Control register with the new frame size, the RNFFRAMEI interrupt enable set to 1 or 0, and the RNFFRAME field set to 1. The host knows that the transition to the new Flexiframe has been made either when a NFFRAMEI interrupt is generated (if the RNFFRAMEIEN was set to 1) or by polling the RNFFRAME bit for a 0 value. An additional change of Flexiframe before some acknowledgement has been recorded may produce undefined behavior.

Table 5-29. RBUFFC Flexiframe Control Register

Bit	Field Name	Value	Description
31:26	RSVD	0	Reserved.
25	RNFFRAMEIEN	0	Change of Flexiframe Complete Interrupt Disabled.
		1	Change of Flexiframe Complete Interrupt Enabled. Once the RBUFFC has completed the switch to the new Flexiframe a Change of Flexiframe Complete interrupt will be generated.
24	RNFFRAME	—	New Flexiframe Indication. This bit serves as an indication to the RBUFFC to switch to the new Flexiframe. When the RBUFFC completes the switch to the new Flexiframe, it resets this indication to 0. It is illegal for the system to set this bit to 0 as this will produce undefined behavior.
23:15	RSVD	0	Reserved.
14:0	RFFRAMESIZE[14:0] RFFRAMESIZE[1:0]	— 3	Flexiframe Size. This field provides the RBUFFC the actual number of entries in the Flexiframe minus one. Because the number of entries in the Flexiframe must be a multiple of four, the last two bits of this field will be set to 11b. The value of this field may range from 11 to 21503 (indicating Flexiframe sizes of 12 to 21504 respectively).

5.7.7 RBUFFC DATA FIFO Size Register

This register defines the size of each channel's data FIFO in 8-byte granularity. This size is fixed once for the receive direction since all the channels are allocated the same amount of buffer memory regardless of their bit rate. The size of the buffer should be allocated as a multiple of 8, minimum 160 bytes per channel and maximum 32 KB (see Appendix E).

Table 5-30. RBUFFC Data FIFO Size Register

Bit	Field Name	Value	Description
31:14	RSVD	0	Reserved.
13:0	RDFIFOSIZE	— 0	Data FIFO Size (per channel) in DWords. The value in this register applies to all the channels. The value in this field must be even.

5.7.8 RBUFFC Fragment Size Register

This fixes the maximum number of words of payload (i.e., packet data, not fragment header) that will be transferred to the system over the POS-PHY data interface in the interval fixed by Table 5-32. For the calculation to determine the relevant Fragment Size (see Appendix E).

Table 5-31. RBUFFC Fragment Size Register

Bit	Field Name	Value	Description
31:8	RSVD	0	Reserved.
7:0	RNUMWORDSFRAG	—	Maximum number of words of data allocated to a fragment. The minimum programmable value is 8 Dwords, and the maximum 64 Dwords. The register is based on a one-based count. The length of the fragment is fixed once for the receive direction.

5.7.9 RBUFFC Flexiframe Slot Time Register

Number of Cycles per Slot – determines the number of cycles per slot and as consequence the timing of the write transaction of a fragment towards the POS-PHY.

Table 5-32. RBUFFC Flexiframe Slot Time Register

Bit	Field Name	Value	Description
31:8	RSVD	0	Reserved.
7:0	RNUMCYCLESLOT	—	Minimum number of cycles allocated per Flexiframe slot. This count is zero based, all values are supported. If this is larger than three plus the number of Dwords ready to be sent to the system, a gap will be created between fragments. The aim of this is to allow the system to fix the amount of time it needs to perform regular (and irregular) activities. When configured to 0, the RBUFFC will work "as fast as possible"—the minimum number of cycles possible (4) will be spent servicing empty slots.

5.7.10 RBUFFC Counter Memory

There are 2047 counters of each kind, one for each channel. The counters for each channel can be read by giving the base address of the channels counters and length long enough to encompass them all (counters can be read on a per channel basis or for all channels).

For a full description of the counters and their use, see Appendix A.

Table 5-33. RBUFFC Counter Memory

Length	Counter Name	Reset Value	Description
24	ROCTETCTR	0	Octet Counter. The number of data bytes/octets received per channel.
24	RMSGCTR	0	Message Counter. The number of non-errored messages received per channel.
24	RMALIGNERRCTR	0	Message Alignment Error Counter. The number of messages with message alignment errors received per channel.
24	RFCSERRCTR	0	FCS Error Counter. The number of messages with FCS errors received per channel.
24	RABRCERRCNT	0	Abort Condition Error Counter. The number of messages with abort condition errors received per channel.
24	RLONGMSGCNT	0	Too Long Message Error Counter. The number of messages with too long message errors received per channel.
24	RTSHORTMSGCNT	0	Too Short Message Error Counter. The number of messages with too short message errors received. To use this counter, the SHT interrupt must be enabled (see Table 5-25, *RSLP Channel Configuration Register*).

5.7.11 RSIU Time Slot Configuration Register

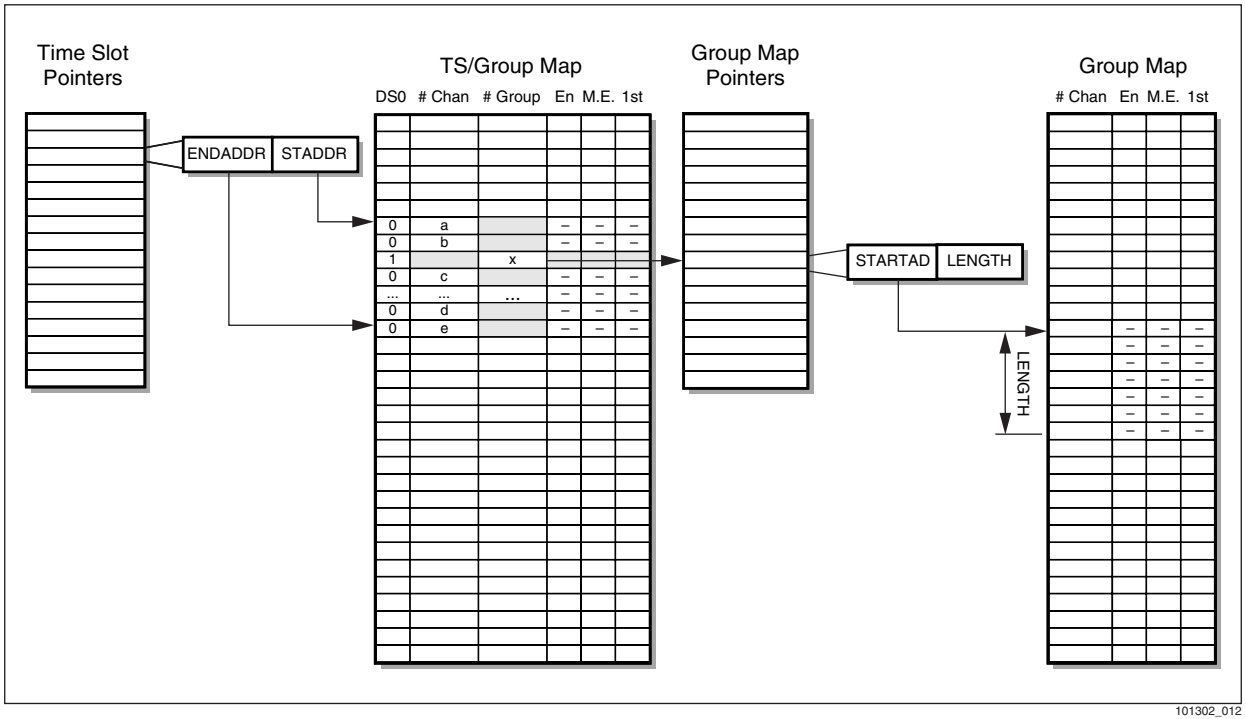
5.7.11.1 Receive Time Slot Map

The Receive Time Slot Map comprises two 8192 entry memories containing slot to group/channel mapping, and two 512 entry memories of pointers per port or per group. One set of maps is provided per direction. Each port is assigned a start and end address within the time slot/group map, and runs on the slots between these addresses. Each slot may be either a direct mapping to a channel number and relevant parameters, or a pointer to a group. If the slot contains a pointer to a group, this implies DS0 extraction is to be performed. The relevant address within the group map pointers will be accessed to retrieve start, length and current pointers, and the channel number and relevant parameters will be retrieved from the Group Map (see Figure 5-2).

NOTE: The group map and DS0 bit extraction are only to be used in ports that are configured as TSBUS mode in the Table 5-39; for other ports, a time slot mapped with the DS0 extraction bit set causes undefined behavior and so is illegal.

A channel may be mapped to more than one time slot within a port (hyper-channeling), but mapping of one channel to more than one port is illegal and will cause undefined ordering of data. Hence numerous mappings of time slots are possible, multiple time slots can be mapped to a single channel or in the case of TSBUS DS0 extraction mode mode, to a single group. For each serial port one time slot map is required (per direction), and when in TSBUS DS0 extraction mode, group maps should be provided per direction. Each map is configured independently. In the receive direction the registers described in Table 5-34, Table 5-35, Table 5-36 and Table 5-37 are used for configuration of the time slot map.

Figure 5-2. Receive Time Slot Map Pointers



5.7.11.2 RSIU Time Slot Configuration Descriptor

For each time slot in the Time Slot Map, there is an RSIU Time Slot Configuration Descriptor. There are 8192 entries in memory that set the translation between time slots and logical channels or groups for each of the CX28560's 32 ports. The actual mapping of these time slot descriptors to the 32 ports is done by 32 sets of pointer pairs (receive and transmit), one pair set for each port, which indicates the start and the end address of the memory location that belongs to the configured port. Time Slot pointer allocation is described in RSIU Time Slot Pointer Allocation. The bit fields of RSIU Time Slot Configuration Descriptor include information:

- This time slot should be referred to a group map for a higher level of extraction
- Time slot is enabled or disabled
- Time slot is a full DS0, or sub-channeling enabled so that only a part of 64 Kbps transports information
- Indicates if it is the first time slot assigned to the logical channel
- Logical channel number (max 2047).

Table 5-34 specify the content of each receive time slot configuration descriptor. The type of entry in the specific row of the TS/Group Map is determined by the DS0 bit.

Table 5-34. RSIU TS/Group Map

Bit	Field Name	Value	Description
31:18	RSVD	0	Reserved
17	RDS0	0	DS0 extraction mode is disabled
		1	DS0 mode is enabled. This bit must only be set if the port to which this time slot is connected is configured as TSBUS Mode see)
16:14	RSVD	0	Reserved
13:3	RCHANNEL[10:0]	—	If DS0 extraction mode is disabled for this time slot, this field represents the logical channel number assigned to the time slot.
	RDS0_GROUP	—	If DS0 extraction mode is enabled for this time slot, the lower 9 bits of this field represent the logical group number assigned to the time slot.
2	RTS_ENABLE	0	Time Slot Disabled or DS0 extraction mode is enabled.
		1	Time Slot Enabled. This bit is only valid if DS0 extraction mode is disabled (RDS0 = 0)
1	RMASKEN_SB	0	The RMASK_SB bit field () is ignored. All the 8 bits of the time slot are processed. This value is also possible if DS0 extraction mode is enabled.
		1	Allow data mask for time slot. Only the bits specified by the RMASK_SB bit field () are processed. This bit is only valid if DS0 extraction mode is disabled (RDS0 = 0)
0	RFIRST_TS	0	This bit field indicates that the specified time slot is not the first time slot of the logical channel or that DS0 extraction mode is enabled.
		1	This bit field indicates that the specified time slot is the first time slot of the logical channel. This bit is only valid if DS0 extraction mode is disabled (RDS0 = 0) ***NOTE(S):*** If a serial port is configured to transparent mode, each channel defined to operate over the serial port must have one time slot assigned to that logical channel as the first time slot for that channel.

When DS0 extraction mode is enabled, the receive group map for that group is referred to in order to attain relevant information regarding the channel number, slot enabled, mask enabled and first time slot bits. The format of an entry in the Group is shown in Table 5-35.

Table 5-35. RSIU Group Map

Bit	Field Name	Value	Description
31:14	RSVD	0	Reserved.
13:3	RCHANNEL[10:0]	—	Logical channel number assigned to the time slot.
2	RTS_ENABLE	0	Time Slot Disabled.
		1	Time Slot Enabled.
1	RMASKEN_SB	0	The RMASK_SB bit field (RSLP Channel Configuration Descriptor) is ignored. All the 8 bits of the time slot are processed.
		1	Allow data mask for the specified time slot. The bits specified by RMASK_SB bit field (RSLP Channel Configuration Descriptor) are processed.
0	RFIRST_TS	0	This bit field indicates that the specified time slot is not the first time slot of the logical channel.
		1	This bit field indicates that the specified time slot is the first time slot of the logical channel. ***NOTE(S):*** If a serial port is configured to operate in channelized mode, each channel defined to operate over the serial port must have one time slot assigned to that logical channel that is defined as the first time slot for that channel.

5.7.12 RSIU Time Slot Pointer Allocation Register

There is one RSIU Time Slot Pointer Allocation Descriptor for each of the CX28560's 32 serial ports. This register sets the start and end time slot address for the specific configured port. The difference between the configured end and start address specifies the number of time slots allocated for the specified serial port.

5.7.12.1 Time Slot Allocation Rules

1. When the serial port is configured to one of the conventional modes, and both pointers point to the same location, this port should be configured to operate in unchannelized mode. This is done by setting the RPORTTYP field in Table 5-39 to 0.
2. When the serial port is configured to one of the conventional modes, if there are two or more time slots, the RPORTTYP field in RSIU Port Configuration register must be set to either 5 for non-T1 framing, or 1 to enable T1 framing.
3. When the serial port is configured to TSBUS mode, RSIU Time Slot Pointer Allocation Descriptor is to be configured to support more than eight time slots and the RPORT_TYPE bit field in RSIU Port Configuration register must be set to TSBUS mode.

In the case of unchannelized mode (i.e., the RPORTTYP field in RSIU Port Configuration register is programmed to 0), the CX28560 assumes that only one entry (the one pointed to by STARTAD) is used for this port. This frees the ENDAD pointer to point to any location in the RSIU time slot memory.

Table 5-36 describes the bit fields in RSIU Time Slot Pointer Allocation Descriptor.

Table 5-36. RSIU Time Slot/Group Map Pointer Allocation Register

Bit	Field Name	Value	Description
31:29	RSVD	0	Reserved.
28:16	RENDAD_TS[12:0]	—	Ending location in the Receive Time Slot Map of the last time slot assigned to this port.
15:13	RSVD	0	Reserved.
12:0	RSTARTAD_TS[12:0]	—	Starting location in the Receive Time Slot Map of the first time slot assigned to this port.

5.7.13 RSIU Group Map Pointer Allocation Register

Splits Table 5-35 into group sections.

Table 5-37. RSIU Group Map Pointer Allocation Register

Bit	Field Name	Value	Description
31:19	RSVD	0	Reserved
18:6	RSTARTAD	—	Starting location in the TS Map of the first Group time slot.
5:0	RLENGTH	—	The number of time slots allocated to the group (zero-based count).

5.7.14 RSIU Group State Register

This memory is used internally by the RSIU. The state field of groups belonging to one port must be set to zero (i.e., disable state) before the port is enabled. The relevant group register is found at an offset of the group number from the base address. There is one register per group.

Table 5-38. RSIU Group State Register

Bit Field	Name	Value	Description
31:2	RSVD	0	Reserved
1:0	GROUP_STATE	0	Disable state, where Group is disabled
		1	Enable state, where Group is enabled
		2	Polling state – polling handling
		3	RSVD

5.7.15 RSIU Port Configuration Register

There is a Receive Port Configuration register for each serial port. It defines how the CX28560 interprets and synchronizes the received bit streams associated with the serial port.

Table 5-39 describes the bit fields in RSIU Port Configuration register.

Table 5-39. RSIU Port Configuration Register (1 of 2)

Bit	Field Name	Value	Description
31:15	RSVD	0	Reserved.
14	RGSYNC_EDGE	0	Receiver GSYNC—Falling Edge
		1	Receiver GSYNC—Rising Edge
13	RXENBL	0	Receive Port Disabled. Logically resets the time slot, regardless of RTS_ENABLE bit field in RSIU Time Slot Configuration Descriptor. This does not affect the bit values in any time slot descriptor.
		1	Receive Port Enabled. This bit field acts as a logical AND between RTS_ENABLE bit field in RSIU Time Slot Configuration Descriptor and time slot. Logically, if RTS_ENABLE bit field in RSIU Time Slot Configuration Descriptor is enabled, it allows all channels with time slot enable bits set to start processing data. This does not affect the bit values in any time slot descriptor.
12	RSVD	0	Reserved.

Table 5-39. RSIU Port Configuration Register (2 of 2)

Bit	Field Name	Value	Description
11:9	RPORT_TYPE [2:0]	0	Unchannelized Mode. It is the user's responsibility to configure the time slot map to contain one time slot.
		1	T1 mode. This mode implies 24 time slots and T1 signaling. It is the user's responsibility to configure the time slot map to contain exactly 24 time slots.
		2	Nx64 mode = 2 Time Slots It is the user's responsibility to configure the time slot map to contain exactly two time slots.
		3	Nx64 mode = 3 Time Slots It is the user's responsibility to configure the time slot map to contain exactly three time slots.
		4	Nx64 mode = 4 Time Slots It is the user's responsibility to configure the time slot map to contain exactly four time slots.
		5	Nx64 mode It is the user's responsibility to configure the time slot map to contain more than four time slots.
		6	TSBUS Mode It is the user's responsibility to configure the time slot map to contain at least eight time slots. For the first twelve ports this mode can also be used for DS0 extraction. This is performed by the use of the DS0 bit in the Time Slot/Group Map. This mode is considered to be DS0 extraction mode.
		7	Reserved
8:6	RSVD	0	Reserved
5	RSYNC_EDGE/ RSTUFF_ EDGE	0	Receiver Frame Synchronization/receive stuff indication—Falling Edge. RSYNC/RSTUFF input sampled in on falling edge of RCLK.
		1	Receiver Frame Synchronization/receive stuff indication—Rising Edge.
4	RDAT_EDGE	0	Receiver Data – Falling Edge. RDAT input sampled in on falling edge of RCLK.
		1	Receiver Data – Rising Edge.
3	ROOF_EDGE/ RTSTB_EDGE	0	Receiver Out Of Frame/TSBus Strobe—Falling Edge. ROOF/ TSTB input sampled in on falling edge of RCLK.
		1	Receiver Out Of Frame—Rising Edge.
2	ROOFABT	0	OOF Message Processing Enabled. When OOF condition is detected, continue processing incoming data. SIU should not report about the OOF.
		1	OOF Message Processing Disabled.
1	ROOFIEN	0	Out Of Frame/Frame Recovery Interrupt/ General INT Disabled
		1	Out Of Frame/Frame Recovery/General Interrupt Enabled.
0	RCOFAIEN	0	Change Of Frame Alignment Interrupt Disabled.
		1	Change Of Frame Alignment Interrupt Enabled. If COFA is detected, generate Interrupt indicating COFA.

5.8 Transmit Path Registers

Transmit path registers contain the information necessary to configure the receive direction. This configuration includes registers that are related to the BUFFC block, host interface, registers that control the TSLP, and TSIU.

5.8.1 TSLP Channel Status Register

The TSLP Channel Status register is a Read Only (RO) register. It provides information from TSLP block regarding the channel state. There is one TSLP Channel status register for each of the CX28560's channels (i.e., 2047 registers).

Table 5-40. TSLP Channel Status Register

Bit	Field Name	Host	Default Value	Value	Description
31:4	RSVD	R	X	0	Reserved.
3:1	RSVD	R	X	0	Reserved.
0	TACTIVE	R	X	0	Channel Inactive. The channel has been deactivated due to either a Service Request Channel Deactivation, Reset (PCI Reset or Soft Chip Reset), or one of the following transmit errors: TxBUFF, TxCOFA.
				1	Channel Active. The channel has been activated by service request channel activation.

5.8.2 TSLP Channel Configuration Register

The Transmit Channel Configuration register contains configuration bits applying to the logical channels within the CX28560. There are 2047 such registers, one for each channel. The TSLP Channel Configuration Register configures aspects of the channel common to all messages passing through the channel. One descriptor exists for each logical channel direction. Table 5-41 lists the values and descriptions of each channel configuration descriptor. For each channel to be used in the CX28560, this register must be configured before activation (no default values exist).

Table 5-41. TSLP Channel Configuration Register

Bit	Field Name	Value	Description
31:30	TPROTCOL[1:0]	0	TRANSPARENT.
		1	HDLC with no FCS. Used in TSLP for full packet forwarding and/or channel monitoring application.
		2	HDLC with FCS16 (FCS— 2 bytes).
		3	HDLC with FCS32 (FCS— 4 bytes).
29	TINV	0	Data Inversion Disabled.
		1	Data Inversion Enabled. Message is transferred to the SIU with polarity change (the inversion is done to all bits passed).
28:21	TMASK_SB[7:0]	—	Data Mask. Actual data is only transmitted on bits with a value of 1 (e.g., Mask = 10000001, then only bits 0 and 7 contain channel's data). The other bits are padded with non-data. Enables the sub-channeling feature. Note 0h is an invalid value.
20:3	RSVD	0	Reserved.
2	TPADJ	0	Pad Count Adjustment Disabled
		1	Pad Count Adjustment Enabled. The TSLP counts the number of zero insertions performed in a message, and reduces the number of inter-message idle codes transmitted accordingly. The reduction of the number of idle code bytes is calculated by dividing the number of zero insertions by 8 and rounding down. This feature allows the host approximate control over the bit rate on the line.
1	TBUFFIEN	0	Underrun Interrupt Disabled.
		1	Underrun Interrupt Enabled.
0	TEOMIEN	0	Transmit EOM Interrupt Disabled.
		1	Transmit EOM Interrupt Enabled. An interrupt is generated when an end of message is transmitted by the CX28560.

5.8.3 TBUFFC Channel Configuration Register

This register controls the operation mode for a channel. It contains parameters necessary for the division of the internal memory to channel FIFOs. There is one Channel Configuration Register for each logical channel (i.e., 2047).

The CX28560's internal Tx memory is a 384 KB dual RAM, which may be split to 2047 parts, one part for each channel. The allocation granularity is one Dword (4 bytes).

In the CX28560, regardless of its bit rate, each channel receives an identical allocation of memory. The difference in bit rates is accounted for by extra servicing of faster channels according to the Flexiframe algorithm. Hence the length of a channel's buffer is set once (see Table 5-45). However, for each active channel it is required to specify the start address of the internal data buffer.

Since this register is wider than 32 bits, it spreads over 2 consecutive addresses. When writing to this register, the first 32 least significant bits are written to the first address and the upper bits are written to the lowest possible bits in the second address.

NOTE: The host must set the buffers so there is no overlap between buffers belonging to different channels. Each receive channel must be allocated buffer space before the channel can be activated.

Table 5-42. TBUFFC Channel Configuration Register

Len	Field Name	Value	Description
33	TCMDCIEN	0	End of Channel Command Execution Interrupt Disabled.
		1	End of Channel Command Execution Interrupt Enabled. On completion of command (activation or deactivation) an interrupt will be generated.
32	TBOVFLWIEN	0	TBUFFC Channel Buffer Overflow Interrupt Disabled.
		1	TBUFFC Channel Buffer Overflow Interrupt Enabled.
31:30	TPROTOCOL	0	FCS Protocol. This should be the same as the corresponding TSLP configuration. TRANSPARENT
		1	HDLC with no FCS
		2	HDLC with 16 bit FCS
		3	HDLC with 32 bit FCS
29:13	TSTARTADD	—	Channel DATA FIFO Start Pointer—0 based. The addresses are allocated in Dword (4-byte) granularity.
12:0	TTHRESHOLD	—	Channel Buffer Threshold Level. The CX28560 will not start to transmit a new message until THRESHOLD number of Dwords (4 bytes) are stored in the channels internal buffer. If the message to be transmitted is less than the threshold, the CX28560 will start to transmit the message when the end of message is detected (threshold is a zero based count).

5.8.4 TBUFFC Flexiframe Memory

The TBUFFC Flexiframe Memory provides the TBUFFC with the order in which to service the channels – a timing scheduler. Once a channel is chosen by the Flexiframe algorithm, if necessary a transmission report is sent to the host over the Flow Conductor POS-PHY interface. The TBUFFC runs through the Flexiframe Memory line by line, servicing the channel number as in the Flexiframe memory. The Flexiframe holds a maximum of 21504 entries and a minimum of 12. The number of entries contained in the Flexiframe is stored in the Flexiframe Control Register, and should be an exact multiple of 4. The value 0 in the channel number represents an empty cycle and will be treated as a NOP by the TBUFFC.

Because of the Flexiframe memory organization in lines of four registers each, access to registers must be in multiples of four registers.

Table 5-43. TBUFFC Flexiframe Memory

Bit	Field Name	Value	Description
31:11	RSVD	0	Reserved.
10:0	TCHANNEL	—	Logical Channel Number assigned to slot in Flexiframe.

5.8.5 TBUFFC Flexiframe Control Register

This register contains the characteristics of the Flexiframe being programmed. When moving to a new Flexiframe this register is vital for the smooth transition. In order to swap to a new Flexiframe, the host should write the new Flexiframe to the Table 5-43, then write the Flexiframe Control register with the new frame size, the TNFFRAMEI interrupt enable set to 1 or 0, and the TNFFRAME field set to 1. The host knows that the transition to the new Flexiframe has been made either when a TNFFRAMEI interrupt is generated (if the TNFFRAMEIEN was set to 1) or by polling the TNFFRAME bit for a 0 value. An additional change of Flexiframe before some acknowledgement has been recorded may produce undefined behavior.

Table 5-44. TBUFFC Flexiframe Control Register

Bit	Field Name	Value	Description
31:26	RSVD	0	Reserved.
25	TNFFRAMEIEN	0	New Flexiframe Interrupt Disabled.
		1	New Flexiframe Interrupt Enabled. Once the TBUFFC has completed the switch to the new Flexiframe a New Flexiframe interrupt will be generated.
24	TNFFrame	—	New Flexiframe Indication. This bit serves as an indication to the TBUFFC to switch to the new Flexiframe. When the TBUFFC completes the switch to the new Flexiframe, it resets this indication to 0. It is illegal for the system to set this bit to 0 as this will produce undefined behavior.
23:15	RSVD	0	Reserved.
14:2 1:0	TFFrameSize[14:2] TFFrameSize[1:0]	— 3	Flexiframe Size. This field provides the RBUFFC the actual number of entries in the Flexiframe minus one. Since the number of entries in the Flexiframe must be a multiple of four, the last two bits of this field will be set to 11b. The value of this field may range from 11 to 21503 (indicating Flexiframe sizes of 12 to 21504 respectively).

5.8.6 TBUFFC DATA FIFO Size Register

This register defines the size of each channel's data FIFO in Dwords (4 bytes). This size is fixed once for the transmit direction since all the channels are allocated the same amount of buffer memory regardless of their bit rate. The size of the buffer should be allocated as a multiple of 4, minimum 80 bytes per channel and maximum 32 KB (see Appendix E).

Table 5-45. TBUFFC Data FIFO Size Register

Bit	Field Name	Value	Description
31:13	RSVD	0	Reserved.
12:0	TDfifoSize	—	Size of Data FIFO per channel in Dwords. The value in this register applies to all channels.

5.8.7 TBUFFC Flexiframe Slot Time Register

Number of Cycles per Slot—a number in clock cycles that indicates the minimum slot time. Per slot time, one fragment is received and one transmission report. Range from 6–255 clock cycles.

Table 5-46. TBUFFC Flexiframe Slot Time Register

Bit	Field Name	Value	Description
31:8	RSVD	0	Reserved.
7:0	TNumCycleSlot	—	Minimum number of cycles allocated per Flexiframe slot. This count is zero based, and has a minimum of 0 and a maximum of 255. If this is larger than three plus the number of Dwords ready to be sent to the system, a gap will be created between fragments. The aim of this is to allow the system to fix the amount of time it needs to perform regular (and irregular activities). When configured to 0, the TBUFFC will work in "fastest possible" mode, i.e., each slot will take a minimum of 6 cycles.

5.8.8 TBUFFC Counter Memory

There are 2047 counters of each kind, one for each channel. Each is at an offset of it's channels number from the base address. The counters for each channel can be read by giving the base address of the channels counters and length long enough to encompass them all.

For a full description of the counters and their use, see Appendix A.

Table 5-47. TBUFFC Counter Memory

Length	Counter Name	Reset Value	Description
24	TMsgCtr	0	Message Counter. The number of messages transmitted per channel
24	TOctetCtr	0	Octet Counter. The number of data octets transmitted per channel
24	TABRTMSG	0	Aborted Message Counter. The number of messages with message aborted during their transmission.

5.8.9 TSIU Time Slot Configuration Register

5.8.9.1 Transmit Time Slot Map

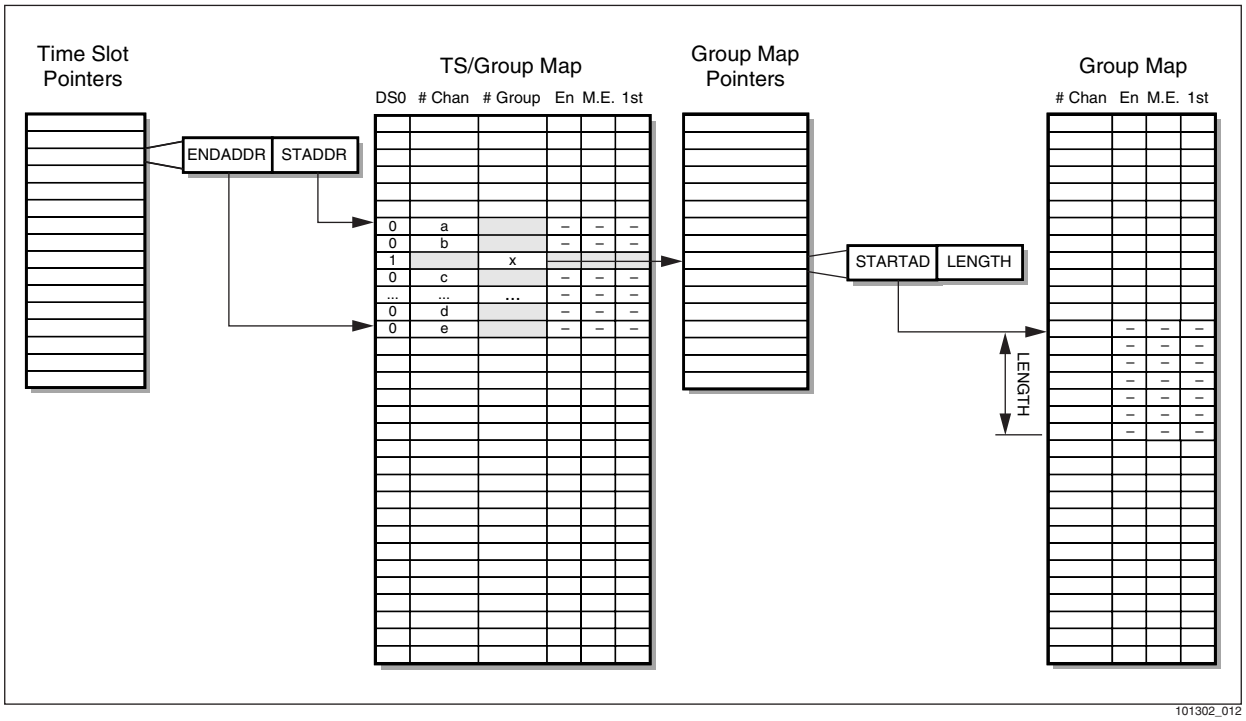
The Transmit Time Slot Map comprises two 8192 entry memories containing slot to group/channel mapping, and two 512 entry memories of pointers per port or per group. One set of maps is provided per direction. Each port is assigned a start and end address within the time slot/group map, and runs on the slots between these addresses. Each slot may be either a direct mapping to a channel number and relevant parameters, or a pointer to a group. If the slot contains a pointer to a group, this implies DS0 extraction is to be performed. The relevant address within the group map pointers will be accessed to retrieve start, length and current pointers, and the channel number and relevant parameters will be retrieved from the Group Map (See Figure 5-3).

NOTE: The group map and DS0 bit extraction are only to be used in ports that are configured as TSBUS mode in Table 5-53; for other ports, a time slot mapped with the DS0 extraction bit set causes undefined behavior and so is illegal.

A channel may be mapped to more than one time slot within a port (hyperchanneling), but mapping of one channel to more than one port is illegal and will cause undefined ordering of data. Hence numerous mappings of time slots are possible, multiple time slots can be mapped to a single channel or in the case of DS0 extraction mode, to a single group. For each serial port one time slot map is required (per direction), and when in TSBUS DS0 extraction mode, group maps should be provided per direction. Each map is configured independently.

In the transmit direction the registers described in Table 5-48, Table 5-49, Table 5-50, and Table 5-51 are used for configuration of the time slot map.

Figure 5-3. Transmit Time Slot Map Pointers



5.8.9.2 TSIU Time Slot Configuration Descriptor

For each time slot in the Time Slot Map, there is a TSIU Time Slot Configuration Descriptor. There are 8192 entries in memory that set the translation between time slots and logical channels or groups for each of the CX28560's 32 ports. The actual mapping of these time slot descriptors to the 32 ports is done by 32 sets of pointer pairs (receive and transmit), one pair set for each port, which indicates the start and the end address of the memory location that belongs to the configured port. Time Slot pointer allocation is described in TSIU Time Slot Pointer Allocation. The bit fields of TSIU Time Slot Configuration Descriptor include information:

- This time slot should be referred to a group map for a higher level of extraction
- Time slot is enabled or disabled
- Time slot is a full DS0, or sub-channeling enabled so that only a part of 64 Kbps transports information
- Indicates if it is the first time slot assigned to the logical channel
- Logical channel number (max 2047).

Table 5-48 specifies the content of each receive time slot configuration descriptor. The type of entry in the specific row of the TS/Group Map is determined by the DS0 bit.

Table 5-48. TSIU TS/Group Map

Bit	Field Name	Value	Description
31:18	RSVD	0	Reserved
17	TDS0	0	DS0 extraction mode is disabled
		1	DS0 mode is enabled. This bit must only be set if the port to which this time slot is connected is configured as TSBUS Mode see)
16:14	RSVD	0	Reserved
13:3	TCHANNEL[10:0]	—	If DS0 extraction mode is disabled for this time slot, this field represents the logical channel number assigned to the time slot.
	TDS0_GROUP	—	If DS0 extraction mode is enabled for this time slot, the lower 9 bits of this field represent the logical channel number assigned to the time slot.
2	TTS_ENABLE	0	Time Slot Disabled or DS0 extraction mode is enabled.
		1	Time Slot Enabled. This bit is only valid if DS0 extraction mode is disabled (TDS0 = 0)
1	TMASKEN_SB	0	The TMASK_SB bit field () is ignored. All the 8 bits of the time slot are processed. This value is also possible if DS0 extraction mode is enabled.
		1	Allow data mask for time slot. Only the bits specified by the TMASK_SB bit field () are processed. This bit is only valid if DS0 extraction mode is disabled (TDS0 = 0)
0	TLAST_TS	0	This bit field indicates that the specified time slot is not the last time slot of the logical channel or that DS0 extraction mode is disabled (TDS0 = 0)
		1	This bit field indicates that the specified time slot is the last time slot of the logical channel ***NOTE(S):*** If a serial port is configured to transparent mode, each channel defined to operate over the serial port must have one time slot assigned to that logical channel as the last time slot for that channel.

When DS0 extraction mode is enabled, the receive group map for that group is referred to in order to attain relevant information regarding the channel number, slot enabled, mask enabled and first time slot bits. The format of an entry in the Group is shown in Table 5-49.

Table 5-49. TSIU Group Map

Bit	Field Name	Value	Description
31:13	RSVD	0	Reserved.
13:3	TCHANNEL[10:0]	—	Logical channel number assigned to the time slot.
2	TTS_ENABLE	0	Time Slot Disabled.
		1	Time Slot Enabled.
1	TMASKEN_SB	0	The TMASK_SB bit field (TSLP Channel Configuration Descriptor) is ignored. All the 8 bits of the time slot are processed.
		1	Allow data mask for the specified time slot. The bits specified by TMASK_SB bit field (TSLP Channel Configuration Descriptor) are processed.
0	TLAST_TS	0	This bit field indicates that the specified time slot is not the last time slot of the logical channel.
		1	This bit field indicates that the specified time slot is the last time slot of the logical channel. ***NOTE(S):*** If a serial port is configured to operate in channelized mode, each channel defined to operate over the serial port must have one time slot assigned to that logical channel that is defined as the first time slot for that channel.

5.8.10 TSIU Time Slot Pointer Allocation Register

There is one TSIU Time Slot Pointer Allocation Descriptor for each of the CX28560's 32 serial ports. This register sets the start and end time slot address for the specific configured port. The difference between the configured end and start address specifies the number of time slots allocated for the specified serial port.

5.8.10.1 Time Slot Allocation Rules

1. If both pointers point to the same location, this port should be configured to operate in unchannelized mode. This is done by setting the TPORTTYP field in Table 5-53 to 0.
2. If there are two, three, or four time slots, the TPORTTYP field in TSIU Port Configuration register must be set to 2, 3, or 4 respectively.
3. If there are more than four time slots, the TPORTTYP field in TSIU Port Configuration register must be set to either 5, if it is not T1 framing, or 1 if it is.
4. If serial port is configured to TSBUS mode, TSIU Time Slot Pointer Allocation Descriptor is configured to support more than eight time slots and the TPORT_TYPE bit field in TSIU Port Configuration register must be set to TSBUS mode.

In the case of unchannelized mode (i.e., the TPORTTYP field in TSIU Port Configuration register is programmed to 0), the CX28560 assumes that only one entry (the one pointed to by TSTARTAD) is used for this port. This frees the TENDAD pointer to point to any location in the TSIU time slot memory.

Table 5-36 describes the bit fields in TSIU Time Slot Pointer Allocation Descriptor.

Table 5-50. TSIU Time Slot/Group Map Pointer Allocation Register

Bit	Field Name	Value	Description
31:29	RSVD	0	Reserved.
28:16	TENDAD_TS[12:0]	—	Ending location in the Receive Time Slot Map of the last time slot assigned to this port.
15:13	RSVD	0	Reserved.
12:0	TSTARTAD_TS[12:0]	—	Starting location in the Receive Time Slot Map of the first time slot assigned to this port.

5.8.11 TSIU Group Time Slot Map Pointers Register

Splits Table 5-49 into group sections.

Table 5-51. TSIU Group Map Pointers Register

Bit	Field Name	Value	Description
31:19	RSVD	0	Reserved
18:6	TSTARTAD	—	Starting location in the TS Map of the first Group time slot
5:0	TLENGTH	—	Number of Time Slots allocated to the group (zero based count)

5.8.12 TSIU Group State Register

This memory is used internally by the TSIU. Before a port is enabled, the state field of groups to be enabled must be set to zero before the port is enabled. The relevant group register is found at an offset of the group number from the base address. This is typically done on reset of the chip, or immediately after a port is disabled. There is one register per group.

Table 5-52. TSIU Group State Register

Bit Field	Name	Value	Description
31:2	RSVD	0	Reserved
1:0	GROUP_STATE	0	Disable state, where Group is disabled
		1	Enable state, where Group is enabled
		2	Polling state—polling handling
		3	RSVD

5.8.13 TSIU Port Configuration Register

There is a Transmit Port Configuration register for each serial port. It defines how the CX28560 interprets and synchronizes the received bit streams associated with the serial port.

Table 5-53 describes the bit fields in TSIU Port Configuration register.

Table 5-53. TSIU Port Configuration Register (1 of 2)

Bit	Field Name	Value	Description
31:15	RSVD	0	Reserved.
14	TGSYNC__EDGE	0	Transmitter GSYNC—Falling Edge
		1	Transmitter GSYNC—Rising Edge
13	TXENBL	0	Transmit Port Disabled. Logically resets the time slot, regardless of TTS_ENABLE bit field in TSIU Time Slot Configuration Descriptor. This does not affect the bit values in any time slot descriptor.
		1	Transmit Port Enabled. This bit field acts as a logical AND between TTS_ENABLE bit field in TSIU Time Slot Configuration Descriptor and time slot. Logically, if TTS_ENABLE bit field in TSIU Time Slot Configuration Descriptor is enabled, it allows all channels with time slot enable bits set to start processing data. This does not affect the bit values in any time slot descriptor.
12	RSVD	0	Reserved.
11:9	TPORT_TYPE [2:0]	0	Unchannelized Mode. It is the user's responsibility to configure the time slot map to contain one time slot.
		1	T1 mode. This mode implies 24 time slots and T1 signaling. It is the user's responsibility to configure the time slot map to contain exactly 24 time slots.
		2	Nx64 mode = 2 Time Slots It is the user's responsibility to configure the time slot map to contain exactly two time slots.
		3	Nx64 mode = 3 Time Slots It is the user's responsibility to configure the time slot map to contain exactly three time slots.
		4	Nx64 mode = 4 Time Slots It is the user's responsibility to configure the time slot map to contain exactly four time slots.
		5	Nx64 mode It is the user's responsibility to configure the time slot map to contain more than four time slots.
		6	TSBUS Mode. It is the user's responsibility to configure the time slot map to contain at least eight time slots. For the first twelve ports this mode can also be used for DS0 extraction. This is performed by the use of the DS0 bit in the Time Slot/Group Map. This mode is considered to be DS0 extraction mode.
		7	Reserved
8:6	RSVD	—	Reserved

Table 5-53. TSIU Port Configuration Register (2 of 2)

Bit	Field Name	Value	Description
5	TSYNC_EDGE/ TSTUFF_ EDGE	0	Transmitter Frame Synchronization/Transmitter Stuff indication—Falling Edge. TSYNC/TSTUFF input sampled in on falling edge of TCLK.
		1	Transmitter Frame Synchronization/Transmitter stuff indication—Rising Edge.
4	TDAT_EDGE	0	Transmitter Data—Falling Edge. TDAT output will be sampled on falling edge of TCLK.
		1	Transmitter Data—Rising Edge.
3	TCTS_EDGE/ TTSTB_EDGE	0	Transmitter Clear To Send/TSBUS Strobe—Falling Edge. TCTS/TSTB input sampled in on falling edge of TCLK.
		1	Transmitter Clear To Send/TSBUS Strobe—Rising Edge.
2	TCTSENB	0	Clear To Send Disabled.
		1	Clear To Send Enabled.
1	TRITX	0	Transmit Three-State Disabled. When a port is enabled, but a time slot within the port is not mapped via the Time Slot Map, the transmitter outputs logic 1 on the output data signal
		1	Transmit Three-state Enabled. When a port is enabled, but a time slot within the port is not mapped via the Time Slot Map, the transmitter three-states the output data signal.
0	TCOFAIEN	0	Change Of Frame Alignment Interrupt Disabled.
		1	Change Of Frame Alignment Interrupt Enabled. If COFA is detected, generate Interrupt indicating COFA.

5.9 POS-PHY Transaction Headers and Packets

5.9.1 Receive POS-PHY Data Bus

Data is accumulated in channel buffers in the CX28560 until either an End Of Message is detected, or enough data has been collated to form a fragment (as configured by the user). Once one of these conditions has been met, a fragment header is prepared, and the data is sent (preceded by the header) to the system over a 32 bit POS-PHY bus. All fragments sent to the system will be of equal length (as configured) except for the last fragment which contains only the last bytes of the message and may contain as little as 0 bytes of actual data.

Table 5-54. CX28560 Receive Header Format

Bit	Field Name	Value	Description
31:27	RSVD	0	Reserved.
26:16	CHANNEL	—	Logical Channel Number.
15:12	MSG STATUS	0	Message Error Encoding.
		0	No Error Occurs.
		1	Overflow. An internal buffer overflow occurred while the message was being received.
		2	Change Of Frame Alignment (COFA) A COFA condition was detected while the message was being received.
		3	Out Of Frame (OOF) A OOF condition was detected while the message was being received.
		4	Abort Condition An abort pattern (at least seven consecutive ones) was detected at the end of the message.
		5	Too Long Message The message received length reached the maximum set by the relevant maximum length register.
		6	Message Alignment Error The number of bits received in the message was not a multiple of 8 – i.e., the message was not byte aligned.
		7	FCS Error The calculated FCS did not match that which was received with the message.
11	SOP	0	This fragment is not the first fragment of a packet.
		1	This fragment is the first fragment of a packet.
10	EOP	0	This fragment is not the last fragment of a packet. The STATUS bits are not valid.
		1	This fragment is the last fragment of a packet. The STATUS bits are valid.
9:0	LENGTH	—	Payload Length. The number of bytes of data that follow the fragment header. When the EOP bit is not set this will always be the maximum length of a fragment. When the EOP bit is set, this field should be consulted to determine the number of data bytes contained in the fragment.

5.9.2 Transmit POS-PHY Data Bus

The system provides data to the CX28560 over a 32 bit POS-PHY bus. When and how much data is to be provided to the CX28560 can be calculated using the information received by the system over the Flow Conductor Bus. All fragments sent to the CX28560 by the system should be of equal length (as configured) except for the last fragment which contains only the last bytes of the message and, therefore, may contain as little as 0 bytes of actual data.

Table 5-55. CX28560 Transmit Data Header Format

Bit	Field Name	Value	Description
31:27	RSVD	0	Reserved.
26:16	CHANNEL	—	Logical Channel Number.
15	COMVALID		Command Valid Bit This bit is set on the last fragment of a packet to indicate that the Idle Code and Pad Count fields are valid.
		0	Command Bits Not Valid.
		1	Command Bits Valid.
14:12	IC	0	Inter-message Idle Code Encoding.
		0	HDLC – FLAGS (0x7E) TRANSPARENT – ALL ONES (0xFF)
		1	HDLC – ALL ONES (0xFF) TRANSPARENT – FLAGS (0x7E)
		2	ALL ZEROS (0x00)
		3	Reserved.
11:4	PADCNT	—	The minimum number of inter-message idle code bytes to be transmitted. HDLC: PADCNT indicates the minimum number of idle codes to be inserted between the closing flags and the next opening flag (0x7E). PADCNT = 0, yields a shared flag between two successive messages. PADCNT = 1, yields the bit pattern: <message><0x7E><0x7E><message> PADCNT = 2, yields the bit pattern: <message><0x7E><IC><0x7E><message> TRANSPARENT: Indicates the (minimum number + 1) of idle codes to be inserted between successive messages. PADCNT = 0, one Idle Code byte will be transmitted between messages <message><IC><message> PADCNT = 1, two Idle Code bytes will be transmitted between messages <message><IC><IC><message> ====== ***NOTE(S):*** There is no indication if more than PADCNT number of idle codes are inserted.
3	ABORT	0	No Abort Finish the message in an orderly manner
—	—	1	Abort Signal Abort the present message by adding at least 7 ones. Continue with next message as usual (i.e., no deactivation).
2:0	RSVD	—	Reserved.

5.9.3 Transmit Flow Conductor Bus

The CX28560 sends transmission reports to the system of the number of Dwords (4 bytes) freed since the previous report. The reports are sent in the form of packets over an 8-bit, 100 MHz POS-PHY bus. The requests packets contain two fields—the channel number and the number of Dwords freed. From this information the system can maintain an array of counters that count the amount of space presently available in each of the CX28560's channels buffers.

Table 5-56. CX28560 Flow Conductor Packet Format

Bit	Field Name	Value	Description
31:27	RSVD	0	Reserved.
26:16	CHANNEL	—	Logical Channel Number
15:0	WSENT	—	Number of Dwords freed up for this specific channel since the previous report was sent over the Flow Conductor Bus.

6.0 Functional Description

6.1 Initialization

6.1.1 Reset

There are two levels of reset:

1. Hard PCI Reset
2. Soft Chip Reset

There are two ways to assert a reset:

1. Assert the PCI reset signal pin, PRST*.
2. Assert a service request through the host interface to perform the soft chip reset.

After reset, the host must configure the CX28560 for it to operate. This configuration includes several stages that should be performed in the following order:

1. PCI Configuration—must be performed only after Hard PCI Reset
2. Interrupt Queue Configuration
3. Global Configuration
4. POS-PHY Configuration
5. Channels and Ports Configuration

NOTE: The Interrupt Queue must be configured before other registers. If the Interrupt Queue is not configured with the correct value of Shared Memory Interrupt Queue Pointer and Interrupt Queue Length, it may result in writes to location 0, because the Service Request Acknowledge (SACK) is written to a zero address location.

6.1.1.1 Hard PCI Reset

The PCI reset is the most thorough level of reset in the CX28560. All subsystems enter into their initial states. PCI reset is accomplished by asserting the PCI signal, PRST*.

The PRST* signal is an asynchronous signal on the PCI bus. The reset signal can be activated in several ways. The system must always assert the reset signal on power-up. Also, a host bus to PCI bus bridging device should provide a way for software to assert the reset signal. Additionally, software-controlled circuitry can be included in the system design to specifically assert the reset signal on demand.

Asserting PRST* towards the CX28560 guarantees that data transfer operations and PCI device operations does not commence until after the CX28560 has been properly initialized for operation. Upon entering PCI reset state, the CX28560 outputs a three-stated signal on all output pins and halts activity on all subsystems including the host interface, serial interface, and expansion bus. The effects of a PCI reset signal within the CX28560 takes ten PCI clock cycles to complete. After this time, the host may communicate with the CX28560 using the PCI configuration cycles.

After the PCI configuration, the device is not ready to start communication with the host via the service request mechanism until the SRQ_LEN bit field in Service Request register is set to zero.

6.1.1.2 Soft Chip Reset

A soft chip reset is a device-wide reset without the host interface's PCI state being reset. Serial interface operations and EBUS operations are halted. The soft chip reset state is entered in one of two ways:

1. As a result of the PCI reset
2. As a result of a soft chip reset host service request

A soft chip reset causes the following:

- Transmit data signals, TDAT, to be three-stated
- EBUS address-data lines to be three-stated and read enable and write enable outputs to be deasserted, halting all memory operations on EBUS
- All active channels to enter the channel deactivated state
- Buffer controllers to be reset, halting all POS-PHY transactions
- All the bits in the Interrupt Status register to clear
- SRQ_LEN and bits in Global Configuration Descriptor to clear

The host acts as if this was a PCI reset, except that the PCI configuration does not need to be repeated (is kept unchanged).

The host can assume that the reset was completed by the CX28560 and can start configuration of registers when the field SRQ_LEN is zero.

6.1.2 Configuration

A sequence of hierarchical initialization must occur after resets. The levels of hierarchy are as follows:

- PCI Configuration—only after hardware reset
- Interrupt Queue Configuration
- Global Configuration
- POS-PHY Configuration
- Channel and Port Configuration

Channel and port configuration involves programming many registers and must be done to comply with its own hierarchy, as explained below.

6.1.2.1 PCI Configuration

After power-up or a PCI reset sequence, the CX28560 enters a holding pattern. It waits for PCI configuration cycles directed specifically for the CX28560. They are actually directed at the PCI bus and PCI slot where the CX28560 resides.

PCI configuration involves PCI read and write cycles. These cycles are initiated by the host and performed by a host-bus-to-PCI-bus bridge device. The cycles are executed at the hardware signal level by the bridge device. The bridge device polls all possible slots on the bus it controls for a PCI device, and then iteratively reads the configuration space for all supported functions on each device. All information from the basic configuration sequence is forwarded to the system controller or host processor controlling the bridge device. During PCI configuration, the host can perform the following configuration for the CX28560:

- Read PCI configuration space (Device Identification, Vendor Identification, Class Code, and Revision Identification)
- Allocate 1 MB system memory range and assign the Base Address register using this memory range
- Allow fast back-to-back transactions
- Enable PCI system error signal line, SERR*
- Allow response for PCI parity error detection
- Allow PCI bus-master mode
- Allow PCI bus-slave mode
- Assign latency
- Assign interrupt line routing

6.1.2.2 Service Request Mechanism

After PCI configuration is complete, a set of hierarchical configuration sequences must be executed to begin operation at the channel level. The Service Request mechanism is the main communication channel between the CX28560 and the host. It is used to configure the CX28560's registers, read status registers, execute transactions over the EBUS, and activate ports and channels. The mechanism is fully described in Section 5.2.1.

6.1.2.3 Global Configuration

Global configuration is initiated by the host issuing service requests. Global configuration specifies information used across the entire device including all ports, all channels, and the EBUS.

For more information, refer to:

- Table 5-19, *Global Configuration Register*.
- Table 5-20, *EBUS Configuration Register*.

NOTE: Device identification at the PCI Configuration Level must be used to identify the number of supported ports and channels in the CX28560, which in turn will affect the CX28560's configuration.

6.1.2.4 Interrupt Queue Configuration

Part of global configuration involves interrupt queue configuration. For more information, refer to Chapter 5.0, Interrupt Queue Descriptor.

6.1.2.5 POS-PHY Configuration

After global configuration has been completed, and the PCI bus set up, POS-PHY Configuration should be performed by the host issuing service requests.

For more information, see the following registers in Chapter 5.0:

- Receive POS PHY Control Register
- Transmit POS PHY Control Register
- Transmit Threshold Register

6.1.2.6 Chip-Level Configuration

There a several registers that require configuration once per chip. They are configured by the host issuing service requests. For further information, see Chapter 5.0, the following registers:

- Receive BUFFC Data FIFO Size Register
- Receive BUFFC Flexiframe Control Register
- Receive BUFFC Fragment Size Register
- Receive BUFFC Flexiframe Slot Time Register
- Receive SLP Maximum Message Length Register (x3)
- Transmit BUFFC Data FIFO Size Register
- Transmit BUFFC Flexiframe Control Register
- Transmit BUFFC Flexiframe Slot Time Register

6.1.2.7 Channel and Port Configuration

After general configuration, a specific channel and port configuration must be performed for each supported channel and port.

- Receive BUFFC Flexiframe Memory
- Transmit BUFFC Flexiframe Memory
- Receive BUFFC Channel Configuration Register
- Receive SLP Channel Configuration Register
- Receive SIU Time Slot/Group Map
- Receive SIU Group Map
- Receive SIU Group State Register
- Receive SIU Time Slot/Group Map Pointer Allocation Register
- Receive SIU Port Configuration Register
- Transmit BUFFC Channel Configuration Register
- Transmit SLP Channel Configuration Register
- Transmit SIU Time Slot/Group Map
- Transmit SIU Group Map
- Transmit SIU Group State Register
- Transmit SIU Time Slot/Group Map Pointer Allocation Register
- Transmit SIU Port Configuration Register

Channel operations service request commands are:

- CH_ACT: Channel Activate
- CH_DEACT: Channel Deactivate

6.1.2.8 Typical Initialization Procedure

This section depicts a typical initialization procedure.

1. PCI Reset or Soft Chip Reset (a Soft Chip Reset is performed by a direct write to the CX28560 register map—in the Soft Chip Reset register)

NOTE: After performing a Soft Chip Reset, it is not necessary to reconfigure the PCI.

2. PCI configuration.
3. Allocate areas in the shared memory for:
   a. Interrupt Queue
   b. Service Request Table
   c. The CX28560's configuration registers (global and local per channel/port/TS basis).
4. Loop and wait for the Service Request Length register to be ready. This step confirms that the CX28560 completed its internal initialization.
   a. Read the SRQ_LEN through the PCI slave access and check if it is 0.
   b. If true, go to the next step.
   c. Otherwise continue to check.
5. Initialize the Interrupt Queue Pointer register and Interrupt Length register by performing a direct write to the CX28560 registers with the address of the Interrupt Queue located in the shared memory and its length.
6. Check the port alive availability (i.e., TxPortAlive and RxPortAlive) register by performing direct reads. For each active port the correspondent bit in TxPortAlive and RxPortAlive registers must be set to 1.
   a. While port not alive (this is equivalent with the correspondent bit not set) wait 8–16 serial clocks.
   b. If port not alive, poll until port alive.
   c. Otherwise go to the next step.

NOTE: If the port is not alive in 16 system clocks then there are no serial clocks applied specific port.

7. Initialize the Service Request Pointer (SRP) and Service Request Length (SRL) registers by performing a direct write to the CX28560 Service Request Pointer and Service Request Length register and update the value with the address all the SRP table and its length in shared memory.
8. Perform a CONFIG_WR Service Request and wait for the SACK or EOC (End of Command) indication which copies the content of the register in shared memory to the relevant CX28560 internal register. The host can perform one CONFIG_WR Service request given that all the register have been initialized in the shared memory prior to the CONFIG_WR Service Request, or can perform CONFIG_WR Service Request for each register individually.

Configuration Write Request Procedure

A detailed typical configuration write request procedure is

1. Allocate the Service Request table in the shared memory.

NOTE: This allocation can be done in the very beginning (see step 3 or in the configuration write request procedure)

2. Initialize the content of the Service Request Table.
3. Initialize the Service Request Pointer (SRP) with the address of Service Request table by performing a direct write to the Service Request Pointer register.
4. Start the execution by writing the table length into to the Service Request Length register by performing a direct write.
5. If other Service Request table is required, the host must poll the Service Request Length register by performing a direct read and check the SRQ_LEN field. If this fields is not zero, the CX28560 did not complete the execution of the last Service Request Table. The number written in the SRQ_LEN indicates how many Configuration Write commands (i.e., table entries) are pending for execution. While processing these commands, the CX28560 generates a SACK interrupt for each command in which the SACKIEN bit was set. When SRQ_LEN becomes 0, the host may start from Step One in *Configuration Write Request Procedure*, whereas prior to a new execution either frees the memory which was allocated for the prior Service Request table or uses the same memory as a pool memory.

The registers initialized through the Service Request Mechanism are as follows:

a. Global Configuration [1] (one per chip)
b. EBUS configuration [1] (one per chip)
c. RSLP Channel Configuration [2047] (one for each channel which is going to be activated)
d. RSLP Max. Message Length [3] (three registers)
e. RBUFFC Configuration [2047] (one for each channel which is going to be activated)
f. RBUFFC Flexiframe Memory [1] (one per chip)
g. RBUFFC Data FIFO Size [1] (one per chip)
h. RBUFFC Fragment Size [1] (one per chip)
i. RBUFFC Slot Time [1] (one per chip)
j. RBUFFC Flexiframe Control [1] (one per chip)
k. RSIU Time Slot/Group Map [8192] (for each time slot that is going to be used)

l. RSIU Group Map [8192] (for each group time slot that is going to be used)
m. RSIU Time Slot/Group Map Pointer Allocation [32] (one per port)
n. RSIU Group Map Pointer Allocation [64] (one per group required)
o. RSIU Group State Register [512] (for each group, the relevant state register should be set to zero).
p. RSIU Port Configuration [32] (for each port that should operate, this command activates the port)
q. TSLP Channel Configuration [2047] (one for each channel that is going to be activated)
r. TBUFFC Configuration [2047] (one for each channel that is going to be activated)
s. TBUFFC Flexiframe Memory [1] (one per chip)
t. TBUFFC Data FIFO Size [1] (one per chip)
u. TBUFFC Fragment Size [1] (one per chip)
v. TBUFFC Slot Time [1] (one per chip)
w. TBUFFC Flexiframe Control [1] (one per chip)
x. TSIU Time Slot/Group Map [8192] (for each time slot that is going to be used)
y. TSIU Group Map [8192] (for each group time slot that is going to be used)
z. TSIU Time Slot/Group Map Pointer Allocation [32] (one per port)
aa. TSIU Group Map Pointer Allocation [64] (one per group required)
ab. TSIU Group State Register [512] (for each group, the relevant state register should be set to zero).
ac. TSIU Port Configuration [32] (for each port which should operate, this command activates the port)
ad. Transmit POS-PHY Thresholds register [1] (once per chip)
ae. Transmit POS-PHY Control register [1] (once per chip)
af. Receive POS-PHY Control register [1] (once per chip)

6. Perform a CH_ACT Service Request and wait for SACK when the SACKIEN bit is set.
7. For each channel that must be activated, the host prepares a CH_ACT Service Request and inserts it into The Service Request table. The host may decide if to activate all channels by writing the Service Request queries into one single Service Request table or by splitting the service request commands into one or more tables. For each CH_ACT Service Request the host follows the same steps as were specified at 1–5 in this section.

6.2 Channel Operations

6.2.1 Channel Activation

After the previous levels of configuration are completed, individual channels are ready to be activated. Service requests are used to activate channels. Each channel consists of a transmit and a receive direction. Each direction is independent of the other and maintains its own state machines, configuration registers, and internal resources. To activate both transmit and receive directions of a channel, two separate service requests are required, one directed to the transmit direction and one to the receive. The CX28560 responds to each service request with the SACK Interrupt Descriptor, which notifies the host that the task was initiated. Note that the SACK interrupt will only be generated if the SACKIEN bit is asserted in the Service Request Descriptor.

If the channel to be activated requires a new Flexiframe, the new Flexiframe should first be written into the CX28560 (via the service request mechanism). Once the system has detected that the new Flexiframe is in place and in use (either by receiving a NFFRAME interrupt, or detecting that the NFFRAME bit has been set to zero by the CX28560), a channel that has now been included in the Flexiframe can be activated. Not writing the new Flexiframe first may cause overflows in the receive direction.

Channel Activation should only be performed on a non-active channel. Attempting to reactivate a channel by sending an activate command to an already active channel will produce undefined behavior by the CX28560. A channel has not been successfully deactivated until the End Of Command (EOC) interrupt is received.

NOTE: The notification to the host that the channel activation was completed is an EOC interrupt. This acknowledges the host that the SRQ was completed. The SACK command signifies that the CX28560 is ready to receive the next command, but not that the activation was completed.

6.2.1.1 Transmit Channel Activation

The following describes what the CX28560 does when a transmit channel is activated:

1. The internal channel FIFO is flushed in preparation for new messages.
2. All counters connected to the channel being activated are zeroed in preparation for a new channel connection.
3. Abort codes (all 1s) are transmitted until new data arrives for transmission.
4. Once fragments start arriving for the newly activated channel, the CX28560 assumes that these fragments are the start of a new packet. The internal buffer threshold is used to ensure that enough data to start transmitting without causing an underrun. Once the threshold has been crossed, transmission of messages can begin.
5. If the channel is configured in HDLC mode, the CX28560 transmits the message as HDLC frames, otherwise, the data is transmitted as if starting from a first time slot in the Serial Port frame.
6. Once a complete message has been transmitted, if the EOM interrupt is enabled, an EOM interrupt is generated, and the CX28560 transmits inter-message idle codes according to the fragment header received.
7. Go to 3.

6.2.1.2 Receive Channel Activation

The following describes what the CX28560 does when a receive channel is activated:

1. The internal channel FIFO is flushed in preparation for new messages.
2. All counters connected to the channel being activated are zeroed in preparation for a new channel connection.
3. In the case of a channel configured for HDLC processing, data is discarded until an opening flag sequence is detected. In the case of a transparent channel, data is discarded until the first time slot of a frame.
4. For an HDLC channel, the data is processed according to the HDLC standard, or for transparent channels, the data is simply collected.
5. Once either enough data for a fragment has been collated in the channel's FIFO or an end of message is detected, a fragment header is attached to the fragment data, and the complete fragment is passed to the host over the POS-PHY interface.
6. Once a complete message has been received, if the EOM interrupt is enabled, an EOM interrupt is generated, and the CX28560 scans the idle codes received between messages.
7. If the idle code has been swapped since the previous message, and the CHIC/CHABT interrupt is enabled, a CHIC/CHABT interrupt is generated. CHIC is generated when the change was to HDLC flags, CHABT is generated when the change was to an all 1s intermessage fill.
8. Go to 3.

6.2.2 Channel Deactivation

After the channel has been activated, channel deactivation via a service request suspends activity on an individual channel direction. Each channel consists of a transmit and a receive direction. Each direction is independent of the other and maintains its own state machines and configuration registers. To deactivate both the transmit and receive directions of a channel, two separate service requests are required, one directed towards the transmit and one to the receive. The CX28560 may respond to each service request with the SACK Interrupt Descriptor, which notifies the host that the task was initiated. Note that the SACK interrupt will only be generated if the SACKIEN bit is asserted in the Service Request Descriptor.

NOTE: The notification to the host that the task was completed is an EOC interrupt. This acknowledges the host that the SRQ was completed. A Channel Deactivation is an asynchronous command from the host interface to a transmit or receive section of a channel to suspend processing and halt memory transfers to/from the host.

6.2.2.1 Transmit Channel Deactivation

The following describes what the CX28560 does when transmit channel is deactivated:

1. The current message processing is terminated destructively. That is, data can be lost and messages prematurely aborted. The CX28560 does not give any indication of a lost message directly to the host, but does increment the aborted messages counter.
2. Internal FIFOs are flushed and the data is lost.
3. The TSLP is responsible for handling outbound bits when the serial port is asynchronously disabled. The data output pin, TDAT, is held at logic 1. Any data received by the CX28560 while a channel is deactivated is discarded.
4. The transmit channel remains in the suspended state until the channel is activated. The current channel direction configuration is maintained.

NOTE: Counters are automatically zeroed on deactivation, activation and one-second pulses.

6.2.2.2 Receive Channel Deactivation

The following describes what the CX28560 does when receive channel is deactivated:

1. Current message processing is terminated destructively. That is, data can be lost and messages prematurely aborted. The CX28560 does not give any direct indication of the lost messages.
2. Internal FIFOs are flushed and all data is lost.
3. The RSLP is responsible for handling inbound bits when the serial port is asynchronously disabled. Data transfers to the host are halted.
4. The receive channel remains in the suspended state until the channel is activated. The current channel direction configuration is maintained.

NOTE: Counters are automatically zeroed on deactivation, activation and one-second pulses.

6.2.3 Channel Reactivation

Channel reactivation is not supported. To reset a channel, it must first be deactivated, and then, once the End Of Command (EOC) interrupt has been received, activated.

6.3 Port Operations

6.3.1 Unmapped Time Slots

The host can stop the CX28560 from processing certain time slots regardless of the channel activation/deactivation/ reactivation commands. This can be performed by programming time slots in RSIU Time Slot Configuration and TSIU Time Slot Configuration to indicate that the specific time slots are not mapped. (See RTS_ENABLE and TTS_ENABLE bit fields in Chapter 5.0, RSIU Time Slot Configuration register and TSIU Time Slot Configuration register, respectively).

NOTE: The TDAT signal is either set to logic 1 or three-state according to bit TRITx in Chapter 5.0, TSIU Time Slot Configuration register.

6.3.2 Enabling a Port

The procedure required for enabling a receive port and a transmit port is identical.

A port can be enabled by writing a 1 to the ENBL bit in the SIU Port Configuration register. Once a port has been enabled, changing the time slot map allocation to the port (i.e., the STARTAD_TS and ENDAD_TS fields) is not allowed; however, changing the mapping of the time slots to channels is allowed The new port configuration is written to the SIU Port Configuration register.

When a port is configured to work in TSBUS mode and DS0 extraction is configured within the port (i.e., that the groups of channels have been assigned to one or more time slots by a group number), a special procedure is required. Before enabling the port, the group state machines of the groups included in the port must be reset. Resetting the group state machines is done by writing the value 0 to all 32 bits in the Group State register (see Chapter 5.0, RSIU and TSIU) for each group in each port to be enabled.

6.3.3 Disabling a Port

The procedure required for disabling a receive port and a transmit port is identical.

A port can be disabled by writing a 0 to the ENBL bit in the SIU Port Configuration register. Once a port has been disabled, changing the time slot map allocation to the port (i.e., the STARTAD_TS and ENDAD_TS fields) is allowed.

7.0 Basic Operations

The two main channel protocols, HDLC and transparent mode, are described in subsequent sections of this chapter. HDLC and transparent mode operations perform protocol-specific processing of their respective input and output serial bit streams, and behave differently in their treatment of those bit streams during abnormal conditions.

7.1 Protocol-Independent Operations

From a functional viewpoint, many of the CX28560 operations are protocol-independent, though some behavior may differ between the transmitter and receiver. The protocol-independent operations described below apply to all event and error handling:

- During buffer controller, SLP channel protocol, and SIU serial port operations, an event or error may occur that indicates the status of the message transfer process or that affects the outcome of the overall message transfer process. Unless masked, all such events and errors cause the CX28560 to write an interrupt descriptor to the shared memory interrupt queue. Interrupt descriptors identify the error or event condition, the transmit or receive direction, and the affected channel or port number.
- If the CX28560 suspends a channel's operation or deactivates a channel, the host must perform a channel reactivation by issuing a channel activation service request. This is referred to as "requiring reactivation."

7.1.1 Transmit

The CX28560 initiates data transfer to the serial interface only if the following conditions are true:

- TxENBL bit set to 1 in Chapter 5.0, TSIU Port Configuration register.
- Transmit channel is mapped to time slot(s), which are enabled in the port's Chapter 5.0, TSIU Time Slot Configuration register.
- Transmit channel has been activated by a host service request.
- The channel number appears at least once in the active Flexiframe.

If TxENBL bit is set to 0 (transmit port disabled), the serial data output signal is placed in high-impedance three-state. If TxENBL = 1 (port enabled) and a time slot is disabled, the corresponding time slot's transmitter output is either a three-state or all 1s signal depending on the state of the TRITx bit field in the port's TSIU Port Configuration register (see Chapter 5.0).

NOTE: If TxENBL = 1 and the port is configured in any channelized mode (i.e., not unchannelized), until the first TSYNC/TSTB pulse is detected, that port outputs either a three-state signal or an all 1s signal, depending on the state of the TRITx bit field.

7.1.2 Receive

The receiver processes data from the serial interface only if all of the following conditions are true:

- RxENBL bit is set to 1 in the port's RSIU Port Configuration register.(see Chapter 5.0)
- Receive channel is mapped to time slot(s), that are enabled in the port's RSIU Time Slot Configuration register. (see Chapter 5.0)
- Receive channel has been activated by a host via service request.
- The channel number appears at least once in the active Flexiframe.

If any of the first three above conditions is not true, the receiver ignores the incoming data stream. If the last condition is not true, eventually an overflow will occur for the channel.

Data is transferred to the system in fragments over the POS-PHY interface, prefixed with a fragment header. The first data is sent for a channel after activation as soon as a complete fragment has been completed.

7.2 HDLC Mode

The CX28560 supports three HDLC modes. The modes are assigned on a per-channel and direction basis by setting the PROTOCOL bit field within the RSLP/TSLP Channel Configuration registers. The HDLC modes are as follows:

- HDLC-NOCRC: HDLC support, no CRC
- HDLC-16CRC: HDLC support, 16-bit CRC
- HDLC-32CRC: HDLC support, 32-bit CRC

HDLC protocol-specific support in the transmitter includes the following:

- Generate opening/closing/shared flags
- Zero-bit insertion after five consecutive 1s are transmitted
- Generate pad fill between frames and adjust for zero insertions
- Generate 0-, 16- or 32-bit CRC (i.e., FCS)
- Generate abort sequences upon FIFO underflow condition or as instructed on a per-message basis by asserting the error line on the POS-PHY or by setting the abort bit in the fragment header of the last fragment of the packet.
- Data inversion of all bits (including flags and pad fill characters)

HDLC protocol-specific support in the receiver includes the following:

- Detection and extraction of opening/closing/shared flags
- Detection of shared-0 between successive flags
- Zero bit extraction after five consecutive 1s are received
- Detect changes in pad fill idle codes

- Check and extract 0-, 16- or 32-bit FCS
- Check frame length
- Check for octet alignment
- Check for abort sequence reception
- After channel activation, check for the first flag character to be received and generate a CHIC interrupt

In the transmit direction, the fragment header of a message specifies intermessage bit-level operations. Specifically, when the EOM bit field is set to 1 within the fragment header, it signifies that the present fragment represents the last fragment for the current message being transmitted and the bit fields IC and PADCNT take effect. These bits are described in this chapter, Section 7.2.5.

7.2.1 Frame Check Sequence

The CX28560 is configured to calculate and insert either a 16- or 32-bit Frame Check Sequence (FCS) for HDLC packets, provided the packet length contains a minimum of 2 octets. The FCS is always calculated over the entire packet length.

For all HDLC modes that require FCS calculation, the polynomials used to calculate FCS are according to ITU-T Q.921 and ISO 3309-1984.

- CRC-16: $x^{16} + x^{12} + x^5 + 1$
- CRC-32: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

7.2.2 Opening/Closing Flags

For HDLC modes only, the CX28560 supports the use of opening and closing message flags. The 7Eh (01111110b) flag is the opening and closing flag. An HDLC message is always bounded by this flag at the beginning and the end of the message.

The CX28560 supports receiving a shared flag where the closing flag of one message can act as the opening of the next message. The CX28560 also supports receiving a shared-zero bit between two flags—that is, the last zero bit of one flag is used as the first zero bit of the next flag. Receiving a shared zero between the FCS and the closing flag is not supported.

The CX28560 can be configured to transmit a shared flag between successive messages by configuring the bit field PADCNT in each transmit fragment header (specifically the last fragment header of a message). The CX28560 does not transmit shared-zero bits between successive flags.

7.2.3 Abort Codes

Seven consecutive 1s constitute an abort code. Receiving the abort code causes the current frame processing to be aborted and further data transfer into shared memory for that message is terminated. After detecting the abort code, The CX28560 enters a scan mode, which searches for a new opening flag character.

Notification of this detected condition is provided in the last fragment header of the message and/or an interrupt descriptor indicating the error condition Abort Flag Termination.

In cases where received idle codes transition to an abort code, an interrupt descriptor is generated toward the host (if enabled in RSLP Channel Configuration register—see Chapter 5.0), indicating the informational event Change To Abort Code. All received abort codes are discarded.

NOTE: Seven 1s are the abort condition the CX28560 checks for while receiving a message, but the criteria for detection and generation of a Change To Abort Code interrupt is equal to 14 consecutive 1s.

7.2.4 Zero-Bit Insertion/Deletion

The CX28560 provides zero-bit insertion and deletion when it encounters five consecutive 1s within a frame. In the receiver, the zero-bit is removed (discarded). In the transmitter, the zero-bit is inserted after each sequence of five 1s.

7.2.5 Message Configuration Bits—HDLC Mode

The last fragment of a transmit message is prefixed by a fragment header that contains message configuration bits to specify what data pattern is transmitted after the end of a current message and its respective closing flag have been transmitted. The bits are specified as follows:

- Idle Code specification, IC
- Inter-message Pad Fill Count, PADCNT
- Send an Abort Sequence, ABRT or assert ERR line on POS-PHY

NOTE: Message configuration bits are also used in Transparent mode with slightly different meanings. For details, see Idle Code.

7.2.5.1 Idle Code

The Idle Code (IC) specification allows one of a set of idle codes to be chosen to be transmitted after the current message in case the next message is not available to be transmitted or intermessage pad fill is requested via PADCNT.

1. IC = 0: Flag pad fill
2. IC = 1: All ones pad fill
3. IC = 2: All zeroes pad fill

7.2.5.2 Intermessage Pad Fill

The pad count (PADCNT) specification allows pad fill octets (a sequence of one or more specified idle codes) to be transmitted between messages. PADCNT is the minimum number of fill octets to be transmitted between the closing flag of one message and the opening flag of the next message in the following manner:

1. PADCNT = 0: Shared open/close flag
2. PADCNT = 1: Separate open/close flags, no idle code
3. PADCNT = 2: Separate open/close flags, at least one idle code
4. Etc.

7.2.5.3 Ending a Message with an Abort or Sending an Abort Sequence

If the abort and the EOM indications are set in a fragment header, the CX28560 interprets it as a request to end an in-progress message with the abort sequence. If the previous fragment header contained an End-Of-Message (EOM) indication, the abort request is ignored. If the previous fragment header was not EOM (i.e., a transmit message was in-progress), an abort code sequence is transmitted to end that partially sent message. Transmission of an abort code sequence is defined as 16 consecutive 1s.

7.2.6 Transmit Events

Transmit events are informational in nature and do not require channel recovery actions.

7.2.6.1 End Of Message (EOM)

Reason:

- TSLP has transmitted (actually, transferred to the TSIU) the last bit of a data buffer (excluding the FCS and closing flag) and the Transmit Fragment Header signifies that the fragment contained an end of a message (EOM = 1).

Effects:

- TxEOM interrupt (if EOMIEN = 1 in Chapter 5.0, TSLP Channel Configuration register).
- TSLP and TBUFFC continue normal processing. If the TBUFFC does not receive more data from the system over the POS-PHY before the TSLP needs to output the next data bit, TSLP outputs another octet of flag or idle code.

7.2.6.2 Transmit COFA Recovery (TCREC)

Reason:

- TSIU terminates the internal COFA condition due to the arrival of a TSYNC/TSTB pulse followed by at least the assigned number of time slots for this port without another unexpected TSYNC/TSTB pulse. This interrupt will also be generated when a suitable number of time slots have passed after a COFA interrupt generated by the first sync pulse.

Effects:

- TCREC Interrupt (if COFAIEN = 1 in Chapter 5.0, TSIU Port Configuration register).

Channel-Level Recovery Actions:

- Transmit channel reactivation should be performed.

7.2.7 Receive Events

Receive events are informational in nature and do not require channel recovery actions.

7.2.7.1 End Of Message (EOM)

Reason:

- RSLP has detected the end of a message (closing flag or an error condition).

Error conditions include: Overflow, COFA, OOF, Abort, Too Long, Alignment and FCS error.

Effects:

- If there were no errors, RxEOM interrupt (if EOMIEN = 1 in Chapter 5.0, RBUFFC Channel Configuration register). If there were errors, RxEOM interrupt (if ERRIEN = 1 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- RBUFFC sets EOM = 1 in Receive Fragment Header.
- RBUFFC and RSLP continue normal processing.

7.2.7.2 Change to Abort Code (CHABT)

Reason:

- RSLP detected received data changed from flag (7Eh) octets to abort code (zero followed by 15 consecutive 1s).

Effects:

7.2.7.3 CHABT Interrupt (if IDLEIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).

- RSLP and RBUFFC continue normal processing.

7.2.7.4 Change to Idle Code (CHIC)

Reason:

- RSLP detects received data changed to flag (7Eh) octets. The CX28560 requires detection of three consecutive flags before a CHIC event is generated.

Effects:

- CHIC interrupt (if IDLEIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).
- RSLP and RBUFFC continue normal processing.

NOTE: After channel activation/reactivation, the first flags detected on the line generate a CHIC interrupt.

7.2.7.5 Frame Recovery (FREC) or Generic Serial PORT (SPORT) Interrupt

Reason:

- RSIU detects the serial interface ROOF signal transition from an out-of-frame (ROOF = 1) to an in-frame (ROOF = 0) condition. If the ROOF signal is programmed for use as an out-of-frame indicator, this frame recovery event (ROOF returning low) generates a FREC interrupt. If the ROOF signal is used as a general-purpose interrupt input, this event generates a SPORT (Serial PORT) interrupt.

Effects:

- FREC/SPORT Interrupt (if OOFIEN = 1 in Chapter 5.0, RSIU Port Configuration register).
- RSLP and RBUFFC continue normal processing.

7.2.7.6 Receive COFA Recovery (RCREC)

Reason:

- RSIU terminates the internal COFA condition due to the arrival of a RSYNC/TSTB pulse followed by at least the assigned number of time slots for this port without another unexpected RSYNC/TSTB pulse. This interrupt is also generated after the COFA caused by the first sync pulse received on a port.

Effects:

- RCREC Interrupt (if COFAIEN = 1 in Chapter 5.0, RSIU Port Configuration register).
- RSLP and RBUFFC continue normal processing.

7.2.8 Transmit Errors

Transmit errors are service-affecting and require a corrective action by a controlling device (i.e., the host) to resume normal channel processing.

7.2.8.1 Transmit Underrun (BUFF)

The CX28560 needs to send more data towards the TSIU for an in-progress transmit message, but the internal channel FIFO is empty.

Reasons:

- Degradation of the host subsystem or application software.
- Host applied back-pressure on the Flow Conductor POS-PHY bus causing reports of buffer levels not to reach the host.

Effects:

- TxBUFF Interrupt (if BUFFIEN = 1 in Chapter 5.0, TSLP Channel Configuration register).
- Transmit channel enters deactivate state where the TSLP transmits a repetitive abort sequence of 16 consecutive 1s.
- Transmit output is three-stated.

Channel-Level Recovery Actions:

- Transmit channel reactivation is required.

7.2.8.2 Transmit Change Of Frame Alignment (COFA)

TSYNC or TSTB input signal transitions from low to high, but at an unexpected time in comparison to the internal frame synchronization flywheel mechanism. COFA errors are only applicable to channelized ports (i.e., unchannelized ports ignore the TSYNC input). Frame synchronization indicates the expected location of the first bit of time slot 0 on the transmit serial data output. Lacking frame synchronization, the transmitter cannot map or align time slots. This error affects all active channels on the respective port. Note that a similar error in TSBUS mode within the group map will not cause an interrupt to be generated.

Reason:

- Signal failure, glitch or realignment caused by the physical interface sourcing the TSYNC/TSTB input signal.

Effects:

- Causes serial interface to enter COFA condition until a TSYNC/TSTB pulse arrives and is followed by at least the assigned number of time slots for this port, without another unexpected TSYNC/TSTB pulse.
- For every active channel on the respective port, TSLP places channels into the deactivate state. Wherein, TSLP sends a repetitive abort sequence of 16 consecutive 1s.
- Transmit COFA Interrupt (if COFAIEN = 1 in Chapter 5.0, TSIU Time Slot Configuration register).

NOTE: COFA interrupt is generated immediately. To synchronize the host's response to a COFA condition, a COFA Recovery interrupt is also provided.

- Transmit output is three-stated.

Channel-Level Recovery Action:

- Transmit channel reactivation is required on receiving the Transmit COFA Recovery (CREC) interrupt.

7.2.8.3 Buffer Controller Channel FIFO Overflow (BOVFLW)

Reason:

- Degradation of the host subsystem or application software.
- Incorrect calculation of the size of internal FIFO required.

Effects:

- Semi-deactivation of the channel. No further data will be transmitted on the channel, and, if the overflow occurred mid-message, the last message in the internal FIFO that was stored before the overflow occurred will be aborted.

Channel-Level Recovery Actions:

- The affected channel should be deactivated and reactivated if required.

7.2.9 Receive Errors

Receive errors are service-affecting, but do not require corrective action by the host to resume normal processing.

7.2.9.1 Receive Overflow (BUFF)

The RSLP receives a signal from the RSIU that more data bits are available to be stored, but the RSLP channel FIFO is already full.

Reasons:

- Degradation of host subsystem performance. This will be caused by host assertion of back-pressure on the Data POS-PHY, not allowing the RBUFFC to transmit the data to the host, thus filling the receive buffers.
- Size of RBUFFC internal FIFO insufficient.

Effects:

- RxBUFF Interrupt (if BUFFIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).
- If a receive message was in progress, that message is marked as errored with an overflow error code. The RSLP scans for the opening flag of the next HDLC message and any subsequent receive messages are discarded until the internal FIFO has room to accept more RSIU data. Notice the channel remains active and channel recovery is automatic.
- When the in-progress message reaches the top of the internal FIFO, the entire HDLC message (before the overflow occurred) is transmitted to the host. In the last fragment the status will be set as follows: EOM = 1, ERROR = BUFF.
- RxERR interrupt is generated, if ERRIEN is set in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register, indicating a RxBUFF error overflow.
- RBUFFC is not affected and continues to transfer data for this channel to the system.

Channel-Level Recovery Actions:

- If possible, increase internal FIFO size assigned to this channel. For this action, all channels must first be deactivated.

Notice that channel reactivation is not required.

7.2.9.2 Receive Change Of Frame Alignment (COFA)

RSYNC or TSTB input signal transitions from low to high, but at an unexpected time compared to the frame synchronization flywheel mechanism. COFA errors are only applicable to channelized ports (i.e., unchannelized ports ignore the RSYNC/TSTB input). Frame synchronization indicates the expected location of the first bit of time slot 0 on the receive serial data input. Lacking frame synchronization, the receiver cannot map or align time slots. This error affects all active channels on the respective port, but does not require a host recovery action. Note that a similar error in TSBUS mode within the group map will not cause an interrupt to be generated.

Reason:

- Signal failure, glitch, or realignment caused by the physical interface sourcing the RSYNC or TSTB input signal.
- First Sync to arrive at a port (this COFA interrupt should be treated as a report of an event rather than as an error).

Effects:

- Causes serial interface to enter COFA condition until the RSYNC/TSTB pulse is followed by at least the assigned number of time slots for this port, without another unexpected RSYNC/TSTB pulse.
- If a receive message was in-progress, that message is marked as errored. RSLP scans for the opening flag of the next HDLC message and any subsequent receive messages are discarded until the internal COFA condition has ended.
- When the in-progress message reaches the top of the internal FIFO, the entire HDLC message is copied to shared memory buffers and Receive Buffer Status Descriptors are written with ONR = HOST and ERROR = COFA (if

INHRBSD = 0 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- Receive COFA Interrupt is generated (if COFAIEN = 1 in Chapter 5.0, RSIU Port Configuration register). Note that a TSTB change of alignment causes both a receive and a transmit COFA interrupt, since TSTB applies to both transmit and receive directions simultaneously.
- Normal operations continue after the COFA condition ends.
- RBUFFC is not affected and continues shared memory buffer processing.

Channel-Level Recovery Actions:
- None required.

7.2.9.3 Out-Of-Frame (OOF)

Out-of-frame or loss-of-frame indicates the entire receive serial data stream is invalid and all data input from that port should be ignored.

Reason:
- ROOF input pin is asserted (high) because the attached physical layer device is unable to recover a valid, framed signal.

Effects:
- OOF Interrupt (if OOFIEN = 1 and OOFABT = 1 in Chapter 5.0, RSIU Port Configuration register).
- If bit field OOFABT = 0, RSLP and RBUFFC continue as if no errors and transfer received data to the host normally.
- If bit field OOFABT = 1 and a receive message is in-progress, the current message is ended with OOF status and RSLP scans for the opening flag of the next HDLC message. When the in-progress message reaches the top of the internal FIFO, the message is transferred to the host and the last Fragment Header of the message is written ERROR = OOF.
- RBUFFC is not affected and continues shared memory buffer processing.
- Receive channels recover automatically when the ROOF input pin is deasserted (low), indicating the OOF condition has ended.

Channel-Level Recovery Actions:
- None required.

7.2.9.4 Frame Check Sequence (FCS) Error

In this case, the Frame Check Sequence (FCS) which the CX28560 calculated for the received HDLC message does not match the FCS located within the message.

Reason:
- Bit errors during transmission.

Effects:
- EOM Interrupt with RxFCS error status, (if ERRIEN = 1 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- When the message reaches the top of the internal FIFO, the HDLC message is transferred to the host and the last Fragment Header is written with ERROR = FCS.

- The RSLP scans for the opening flag of the next HDLC message.
- RBUFFC is not affected and continues to transfer message data to the host.

Channel-Level Recovery Actions:

- None required.

7.2.9.5 Octet Alignment Error (ALIGN)

The HDLC message size after zero-bit extraction was not a multiple of 8 bits.

Reasons:

- Bit errors during transmission.
- Incorrect message transmission from distant end.

Effects:

- EOM Interrupt with RxALIGN error status, (if ERRIEN = 1 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- When the message reaches the top of the internal FIFO, the HDLC message is transferred to the host and the Fragment Header is written with ERROR = ALIGN.
- The RSLP scans for the opening flag of the next HDLC message.
- RBUFFC is not affected and continues to transfer message data to the host.

Channel-Level Recovery Actions:

- None required.

7.2.9.6 Abort Termination (ABT)

The receiver detects an abort sequence from the distant end. An abort sequence is defined as any zero followed by 15 consecutive 1s.

Reasons:

- Distant end failed to complete transmission of the HDLC message.
- Path conditioning has replaced the normal channel content with an all 1s pattern, due to a network alarm condition.

Effects:

- EOM Interrupt with RxABT error status, (if ERRIEN = 1 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- When the message reaches the top of the internal FIFO, the HDLC message is transferred to the host with the last Fragment Header of the message written as ERROR = ABT.
- The RSLP scans for the opening flag of the next HDLC message.
- RBUFFC is not affected and continues to transfer message data to the host.

Channel-Level Recovery Actions:

- None required.

7.2.9.7 Long Message (LNG)

The received HDLC message length is determined to be greater than the maximum allowable message size per the MAXSEL bit field in Chapter 5.0, RSLP Maximum Message Length register.

Reason:

- Incorrect message transmission from distant end.

Effects:

- EOM Interrupt with RxLNG error status (if ERRIEN = 1 in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register).
- When the message reaches the top of the internal FIFO, the HDLC message—up to the maximum legal length—is transferred to the host, and the last fragment header of the message is written with ERROR = LNG.
- The RSLP scans for the opening flag of the next HDLC message.
- RBUFFC is not affected and continues to transfer message data to the host.

Channel-Level Recovery Actions:

- None required.

7.2.9.8 Short Message (SHT)

The total received HDLC message size (between open/close flags) is determined to be less than the number of FCS bits specified for that channel plus one octet. For example, a channel configured for 16-bit FCS must receive a minimum of three octets—one octet of payload and two octets of FCS—to avoid a short message error. In this example, receiving only two octets is considered a short message.

NOTE: Any message that ends with an error (any error except an overflow) and for which the entire message (regardless of its length) still resides in the internal SLP buffer (meaning no data has yet been transferred to the internal channel FIFO), the CX28560 generates a SHT interrupt and does not transfer any of that message to a shared memory buffer. In this case, no other indication is given for the errored message.

NOTE: Because the RxSHT interrupt in this case is reported immediately, its interrupt descriptor can arrive in the shared memory interrupt queue before an earlier message that remains queued in the internal BUFFC channel FIFO. Hence, interrupts from these two messages may appear out of sequence with respect to their actual order of arrival.

Reasons:

- Bit errors during transmission.
- Incorrect message transmission from distant end.

Effects:

- RxSHT Interrupt (if IDLEIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).
- RSLP resumes scanning for opening flag of the next HDLC message.

- RBUFFC is not affected and continues to transfer message data to the host.

Channel-Level Recovery Actions:

- None required.

7.3 Transparent Mode

The CX28560 supports a completely transparent mode where no distinction is made between information and non-information bits in the channel bit stream. This mode is assigned on a per-channel and per-direction basis by the PROTOCOL bit field in Chapter 5.0, RSLP Channel Configuration register and Chapter 5.0, TSLP Channel Configuration register.

7.3.1 Message Configuration Bits—Transparent Mode

The Transmit Fragment Header contains a group of bits that specify the data to be transmitted after the end of a transparent mode message. The bits are specified as follows:

- Idle Code specification, IC
- Intermessage Pad Fill Count, PADCNT
- Send an Abort Sequence.

NOTE: Message configuration bits are also used in HDLC mode, but their meaning is slightly different. Refer above to *Message Configuration Bits– HDLC Mod*e.

7.3.1.1 Idle Code

Idle Code (IC) bit field selects one of a set of idle pad fill octets to be sent after the current message is transmitted in the event the next fragment has not been received or inter-message pad fill is requested via PADCNT.

1. IC = 0: all ones pad fill.
2. IC = 1: HDLC Flag pad fill.
3. IC = 2: all zeroes pad fill.

7.3.1.2 Intermessage Pad Fill

Pad Count (PADCNT) bit field specifies how many pad fill octets (selected by IC) are transmitted between messages. PADCNT specifies the minimum number of pad fill octets plus one, as follows:

1. PADCNT = 0: one IC
2. PADCNT = 1: two ICs
3. PADCNT = 2: three ICs
4. etc.

7.3.1.3 Ending a Message with an Abort or Sending an Abort Sequence

When the ERR line on the Data POS-PHY is asserted, the CX28560 interprets this as a request to end an in-progress message with the abort sequence. Abort sequence for Transparent mode is defined to be a sequence of all 1s. The abort sequence is terminated only when new data is received for the channel. In this case, the CX28560

resynchronizes the start of the next message transmission to the time slot marked as the first time slot on that channel.

If an abort is requested, and the previous message had been aborted and no new data had been received, the abort command is simply ignored and the CX28560 awaits new data.

The host may set EOM = 1 in any transmit fragment to separate this transparent mode "message" from the next message, according to the IC and PADCNT bit fields. Unlike HDLC mode, the number of pad fill octets transmitted equals PADCNT + 1, and no flag characters are inserted.

7.3.2 Transmit Events

Transmit events are informational in nature and require no recovery actions.

7.3.2.1 End Of Message (EOM)

Reason:

TSLP has transmitted (actually, transferred to the TSIU) the last bit of a data buffer and the Transmit Fragment header signified the end of a message with bit field EOM = 1.

Effects:

- TxEOM interrupt (if EOMIEN = 1 in Chapter 5.0, TSLP Channel Configuration register).
- TSLP and TBUFFC continue normal message processing. If the TBUFFC does not receive more data before the internal channel FIFO becomes empty and the TSLP needs to output another data bit, TSLP outputs pad fill octets until more data is available.

7.3.3 Receive Events

Receive events are informational in nature and require no recovery actions.

7.3.3.1 End Of Message (EOM)

Reason:

- RSLP must force an end of a message due to a receive error condition. Error conditions include Overflow, COFA, or OOF.

Effects:

- RxEOM interrupt (if ERRIEN = 1 in Chapter 5.0, RSIU Time Slot Configuration register) with the appropriate RxERR status.
- RBUFFC sets bit field EOM = 1 in Receive Buffer Status Descriptor (if INHRBSD = 0 in Chapter 5.0, RSIU Time Slot Configuration register).
- RSLP continues normal processing after the error condition has ended.
- RBUFFC is not affected and continues shared memory buffer processing.

7.3.3.2 Frame Recovery (FREC)

Reason:

- SIU detects the serial interface has transitioned from an out-of-frame to an in-frame condition. If the ROOF pin is used as an out-of-frame indication, a FREC interrupt is generated. If the ROOF pin is used as a general purpose interrupt input, a SPORT (Serial PORT) interrupt is generated.

Effects:

- FREC/SPORT Interrupt (if OOFIEN = 1 in Chapter 5.0, RSIU Port Configuration register).
- RSLP and RBUFFC continue normal processing.

7.3.3.3 Receive COFA Recovery (RCREC)

Reason:

- SIU terminates the internal COFA condition due to a RSYNC/TSTB pulse followed by at least the assigned number of time slots for this port without another unexpected RSYNC/TSTB pulse.

Effects:

- RCREC Interrupt (if COFAIEN = 1 in Chapter 5.0, RSIU Port Configuration register).
- RSLP and RBUFFC continue normal processing.

7.3.4 Transmit Errors

Transmit Errors are service-affecting and require a corrective action by the host to resume normal processing.

7.3.4.1 Transmit Underrun (BUFF)

Same as HDLC mode.

Reasons:

- Degradation of the host subsystem or application software.
- Host applied back-pressure on the Flow Conductor POS-PHY bus causing reports of buffer levels not to reach the host.

Effects:

- TxBUFF Interrupt (if BUFFIEN = 1 in Chapter 5.0, TSLP Channel Configuration register).
- Transmit channel enters deactivate state, wherein TSLP sends a repetitive all 1s sequence.

Channel-Level Recovery Actions:

- Transmit channel reactivation is required.

7.3.4.2 Transmit Change Of Frame Alignment (COFA)

Reason:

- Signal failure, glitch, or realignment caused by the physical interface sourcing the TSYNC/TSTB input signal.

Effects:

- Causes serial interface to enter COFA condition until a TSYNC/TSTB pulse arrives and is followed by at least the assigned number of time slots for this port, without another unexpected TSYNC/TSTB pulse.
- For every active channel on the respective port, TSLP places channels into the deactivate state, wherein TSLP sends a repetitive all 1s sequence.
- Transmit output is three-stated.

Channel-Level Recovery Actions:

- Transmit channel reactivation is required.

7.3.4.3 Buffer Controller Channel FIFO Overflow (BOVFLW)

Reason:

- Degradation of the host subsystem or application software.
- Incorrect calculation of the size of internal FIFO required.

Effects:

- Semi-deactivation of the channel. No further data will be transmitted on the channel, and, if the overflow occurred mid-message, the last message in the internal FIFO that was stored before the overflow occurred will be aborted.

Channel-Level Recovery Actions:

- The affected channel should be deactivated and reactivated if required.

7.3.5 Receive Errors

Receive errors are service-affecting and may require a corrective action by the host to resume normal processing.

7.3.5.1 Receive Overflow (BUFF)

Same as HDLC mode.

Reasons:

- Degradation of the host subsystem performance. This will be caused by host assertion of back-pressure on the Data POS-PHY, not allowing the RBUFFC to transmit the data to the host, thus filling the receive buffers.
- Size of RBUFFC internal FIFO insufficient.

Effects:

- RxBUFF Interrupt (if BUFFIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).
- Data received during an overflow condition is discarded.

- Data in the internal FIFO is transferred to the host, the Receive Fragment header of the last fragment is set as EOM = 1, ERROR = BUFF.
- If ERRIEN is set in Chapter 5.0, RBUFFC Configuration register and Chapter 5.0, TBUFFC Configuration register, an RxERR interrupt is generated, indicating an RxBUFF overflow.
- When the overflow condition ends (i.e., space becomes available in the channel FIFO), RSLP automatically restarts data processing. However, RSLP ignores all time slots until reaching the time slot marked “first.”
- RBUFFC is not affected and continues shared memory buffer processing.

Channel-Level Recovery Actions:

- If possible, increase internal FIFO size assigned to this channel. For this action, all channel must first be deactivated.
- Notice that channel reactivation is not required.

7.3.5.2 Receive Change Of Frame Alignment (COFA)

Same as HDLC mode.

Reason:

- Signal failure, glitch, or realignment caused by the physical interface sourcing the RSYNC or TSTB input signal.

Effects:

- Causes serial interface to enter COFA condition until the RSYNC/TSTB pulse is followed by at least the assigned number of time slots for this port, without another unexpected RSYNC/TSTB pulse.
- Current message processing is ended for every active channel on this port. All data received prior to the COFA condition is transferred to the host, and the Receive Fragment header is written with ERROR = COFA. The only exception to this description happens when the COFA condition is detected within the first few bytes after channel activation or after the channel suffered an overflow or another COFA, as described in this section, Short COFA (SHT COFA).
- Receive COFA Interrupt (if COFAIEN = 1 in Chapter 5.0, RSIU Port Configuration register). When the COFA condition ends, RSLP restarts data processing automatically, however all time slots are ignored until the time slot marked “first”.
- RBUFFC is not affected and continues to transfer data to the host.

Channel-Level Recovery Actions:

- None required.

7.3.5.3 Out Of Frame (OOF)

Same as HDLC mode.

Reason:

- ROOF input pin is asserted (high) because the attached physical layer device is unable to recover a valid, framed signal.

Effects:

- OOF Interrupt (if OOFIEN = 1 and OOFABT = 1 in Chapter 5.0, RSIU Port Configuration register).
- If bit field OOFABT = 0, RSLP and RBUFFC continue as if there are no errors and transfer received data to the host.
- If bit field OOFABT = 1, all incoming data is replaced by all 1s (0xFF) data sequence. Normal data processing resumes when the ROOF input pin is deasserted (low), indicating the OOF condition has ended.
- RBUFFC is not affected and continues to transfer data to the host.

Channel-Level Recovery Actions:

- None required.

7.3.5.4 Short COFA (SHT COFA)

A short COFA interrupt is generated for any transparent mode message whose reception is ended due to a COFA error and for which no data was transferred from RSLP to RBUFFC or to the host. In this case, no other indication is provided for this errored message.

NOTE: Only transparent mode COFA creates such a scenario. The exact scenario is as follows: a COFA condition happens within the next few bytes after an abnormal message termination (i.e., a prior COFA or overflow error) or after a channel activation.

Reason:

- Signal failure, glitch or realignment caused by the physical interface sourcing the RSYNC or TSTB input signal.

Effects:

- RxSHT Interrupt (if IDLEIEN = 1 in Chapter 5.0, RSLP Channel Configuration register).
- RSLP restarts channel operation as soon as the COFA condition is recovered and the channel reaches its first assigned time slot.
- RBUFFC is not affected and continues to transfer data to the host.

Channel-Level Recovery Actions:

- None required.

8.0 Electrical and Mechanical Specification

8.1 Electrical and Environmental Specifications

8.1.1 Absolute Maximum Ratings

Stressing the device parameters beyond absolute maximum ratings 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.

Table 8-1. Absolute Maximum Ratings

Parameter	Symbol	Value		Unit
		Minimum	Maximum	
Core power supply	V_{dd}	–0.5	2.5	V
I/O Power Supply	V_{ddo}	–0.5	4.6	V
Continuous Power Dissipation	P_d	—	—	mW
Constant Voltage on any Signal Pin	V_i	–1.0	V_{dd} + 0.5	—
Constant Current on any Signal Pin	I_i	–10	10	mA
Transient Current on any Signal Pin	Latchup (@25 °C)	–300	300	mA
Transient Current on any Signal Pin	Latchup (@125 °C)	–150	150	mA
Transient Voltage on any Pin	ESD (HBM)	–2500	2500	V
Transient Voltage on any Pin	ESD (CDM)	–700	700	V
Operating Junction Temperature	T_j	–40	125	°C
Storage Temperature	T_s	–55	125	°C

8.1.2 Recommended Operating Conditions

Table 8-2. Recommended 3.3 V Operating Conditions

Parameter	Symbol	Value		Unit
		Minimum	Maximum	
Power Supply	V_{dd}	1.7	1.9	V
I/O Power Supply	V_{ddo}	3.135	3.465	V
Ambient Operating Temperature KPF EPF	T_{ac}	 0 –40	 +70 +85	 °C °C
High-Level Input Voltage	V_{ih} ***(1)***	2.0	V_{ddo} +0.5	V
Low-Level Input Voltage	V_{il} ***(1)***	0	0.8	V
High-Level Output Current Source	I_{oh}	200	400	µA
Low Level Output Current Sink	I_{ol}	2	4	mA
Output Capacitive Loading PCI and Line Interfaces	C_{ld}	30	85	pF
Output Capacitive Loading POS-PHY Interface	C_{ldPOS}	10	30	pF

NOTE(S): Note(s):
(1) (1) Apply to all pins, except the PCI interface, which is defined in Table 8-4.

8.1.3 Electrical Characteristics

Table 8-3. DC Characteristics for 3.3 V Operation

Parameter	Symbol	Value	Units
High-Level Output Voltage	V_{oh}	2.4	V
Low-Level Output Voltage	V_{ol}	0.4	V
Input Leakage Current	I_{l}	–10 to 10	µA
Three-state Leakage Current	I_{oz}	–10 to 10	µA
Resistive Pullup Current	I_{pr}	20 to 100	µA
Supply Current	I_{dd} + I_{ddo}	1000 + 340	mA

8.2 Timing and Switching Specifications

8.2.1 Overview

This section defines the timing and switching characteristics of CX28560. The major subsystems include the Host interface, the expansion bus interface, the POS-PHY interface and the serial interface. The Host interface is Peripheral Component Interface (PCI) compliant. For other references to PCI, see the *PCI Local Bus Specification*, Revision 2.2, December 18, 1998. The POS-PHY interface is compliant to the Frame-based ATM Interface (Level 3). For other references see ATM Forum Technical Committee document AF-PHY-0143.000, March 2000. The expansion bus and serial bus interfaces are similar to the Host interface timing characteristics; the differences and specific characteristics common to either interface are further defined.

8.2.2 Host Interface (PCI) Timing and Switching Characteristics

Reference the *PCI Local Bus Specification, Revision 2.2*, December 18, 1998 for information the following:

- Indeterminate inputs and metastability
- Power requirements, sequencing, and decoupling
- PCI DC specifications
- PCI AC specifications
- PCI V/I curves
- Maximum AC ratings and device protection

Table 8-4. PCI Interface DC Specifications

Symbol	Parameter	Condition	Min	Max	Units
V_{cc}	Supply Voltage	—	3	3.6	V
V_{ih}	Input High Voltage	—	0.5 V_{ddo}	V_{ddo}+ 0.5	V
V_{il}	Input Low Voltage	—	–0.5	0.3 V_{ddo}	V
I_{il}	Input Leakage Current*(1)*	$0 < V_{in} < V_{cc}$	—	+/-10	µA
V_{oh}	Output High Voltage	I_{out} = –500 µA	0.9 V_{ddo}	—	V
V_{ol}	Output Low Voltage*(2)*	I_{out} = 1500 µA	—	0.1 V_{ddo}	V
$C_{out}/C_{in}/C_{io}$	Output, Input, and I/O Pin Capacitance	—	—	10	pF
C_{clk}	PCLK Pin Capacitance	—	5	12	pF
C_{idsel}	IDSEL Pin Capacitance*(3)*	—	—	8	pF
L_{pin}	Pin Inductance	—	—	20	nH

NOTE(S):
(1) Input leakage currents include hi-Z output leakage for all bidirectional buffers with three-state outputs.
(2) Signals without pullup resistors must have 3 mA low output current. Signals requiring pullup must have 6 mA; the latter include FRAME*, TRDY*, IRDY*, DEVSEL*, STOP*, SERR*, and PERR*.
(3) Lower capacitance on this input-only pin allows for non-resistive coupling to AD[xx].

Table 8-5. PCI Clock (PCLK) Waveform Parameters, 3.3 V Clock

Symbol	Parameter	Min 33 MHz	Max 33 MHz	Units
T_{cyc}	Clock Cycle Time*(1)*	30	Infinite	ns
T_{high}	Clock High Time	11	—	ns
T_{low}	Clock Low Time	11	—	ns
—	Clock Slew Rate*(2)*	1	4	V/ns
V_{ptp}	Peak-to-Peak Voltage	0.4 V_{cc}	—	V

NOTE(S):

(1) CX28560 works with any clock frequency between DC and 33 MHz, nominally. The clock frequency may be changed at any time during operation of the system as long as clock edges remain monotonic, and minimum cycle and high and low times are not violated. The clock may only be stopped in a low state.

(2) Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform.

Figure 8-1. PCI Clock (PCLK) Waveform, 3.3 V Clock

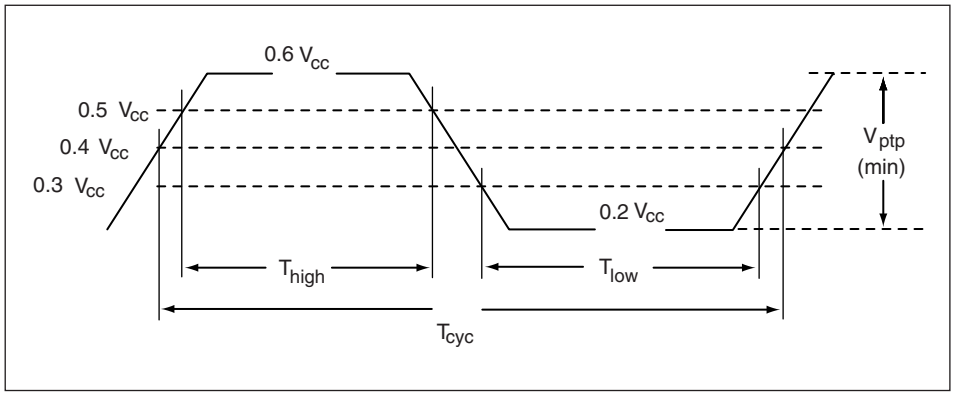


Table 8-6. PCI Reset Parameters

Symbol	Parameter	Min	Max	Units
T_{rst}	Reset Active Time after Power Stable	1	—	ms
T_{rst_clk}	Reset Active Time after Clock Stable	100	—	µs
$T_{rst\text{-}off}$	Reset Active to Float Delay	—	40	ns

Table 8-7. PCI Input/Output Timing Parameters

Symbol	Parameter	Min 33 MHz	Max 33 MHz	Units
T_{val}	PCLK to Signal Valid Delay–Bused Signal*(1, 2)*	1.6	11	ns
T_{val} (ptp)	PCLK to Signal Valid Delay–Point To Point*(1, 2)*	1.6	12	ns
T_{on}	Float to Active Delay*(3)*	2	—	ns
T_{off}	Active to Float Delay*(3)*	—	28	ns
T_{ds}	Input Setup Time to Clock–Bused Signal*(2)*	7	—	ns
T_{su} (ptp)	Input Setup Time to Clock–Point To Point*(2)*	10, 12	—	ns
T_{dh}	Input Hold Time from Clock	0	—	ns

NOTE(S):

(1) Minimum and maximum times are evaluated at 50 pF equivalent load. Actual test capacitance may vary, and results should be correlated to these specifications.

(2) REQ* and GNT* are the only point-to-point signals, and have different output valid delay and input setup times than do bused signals. GNT* has a setup of 10 ns; REQ* has a setup of 12 ns for 33 MHz.

(3) For purposes of active/float timing measurements, the hi-Z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification at 50 pF equivalent load.

Table 8-8. PCI I/O Measure Conditions

Symbol	Parameter	Value	Unit
V_{th}	Voltage Threshold High*(1)*	0.6 V_{ddo}	V
V_{tl}	Voltage Threshold Low*(1)*	0.2 V_{ddo}	V
V_{trise}	Voltage Rise Point	0.285 V_{ddo}	V
V_{tfall}	Voltage Fall Point	0.615 V_{ddo}	V
V_{test}	Voltage Test Point	0.4 V_{ddo}	V
V_{max}	Maximum Peak-to-Peak*(2)*	0.4 V_{ddo}	V
—	Input Signal Edge Rate	1	V/ns

NOTE(S):

(1) The input test is done with 0.1 V_{dd} of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. Production testing can use different voltage values, but must correlate results back to these parameters.

(2) V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing can use different voltage values, but must correlate results back to these parameters.

Figure 8-2. PCI Output Timing Waveform

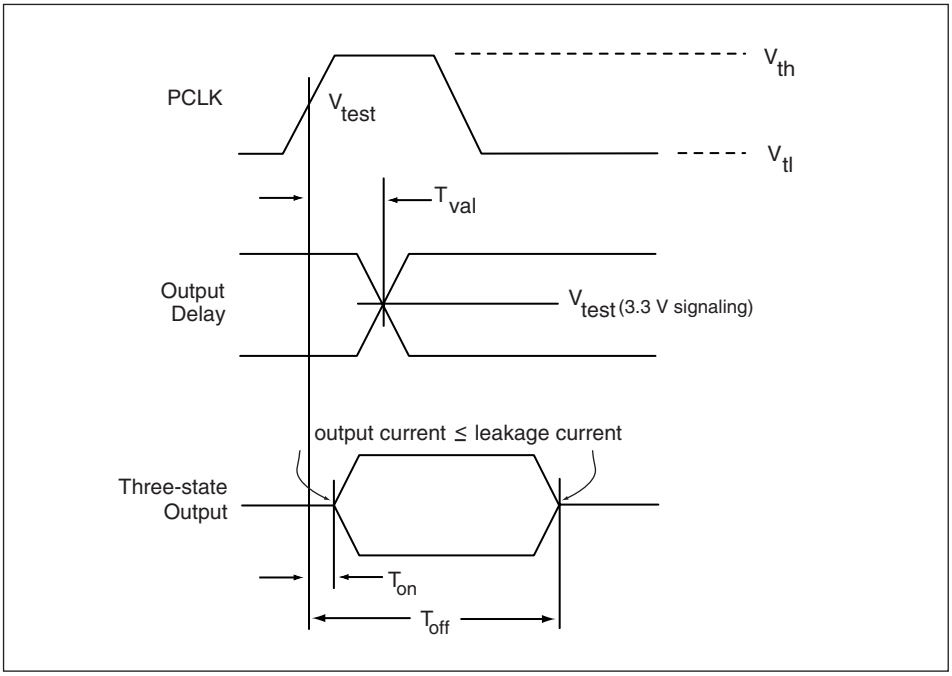
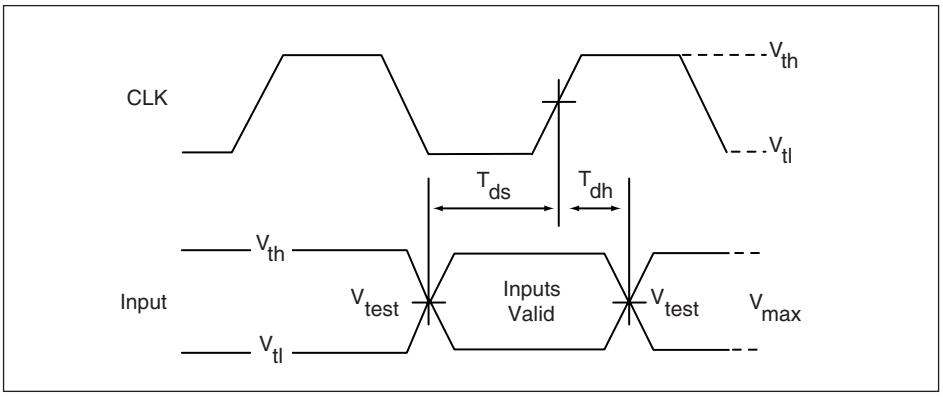


Figure 8-3. PCI Input Timing Waveform



8.2.3 Data Interface (POS-PHY) Timing and Switching Characteristics

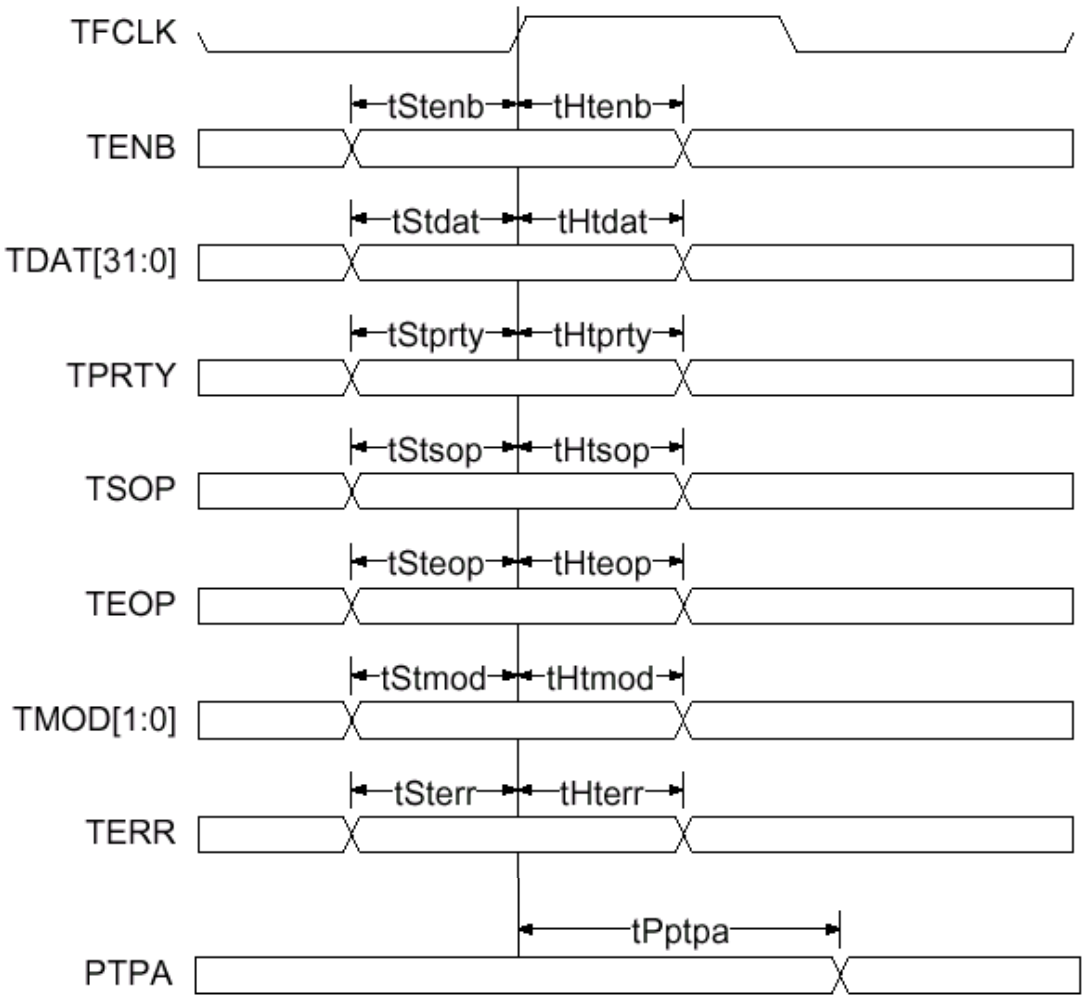
All AC timing is from the perspective of the CX28560.

Table 8-9. Transmit Interface Timing

Symbol	Description	Min	Max	Units
—	TFCLK Frequency*(1)*	—	104	MHz
—	TFCLK Duty Cycle	40	60	%
tStenb	TENB Setup time to TFCLK	2	—	ns
tHtenb	TENB Hold time to TFCLK	0.5	—	ns
tStdat	TDAT[15:0] Setup time to TFCLK	2	—	ns
tHtdat	TDAT[15:0] Hold time to TFCLK	0.5	—	ns
tStprty	TPRTY Setup time to TFCLK	2	—	ns
tHtprty	TPRTY Hold time to TFCLK	0.5	—	ns
tStsop	TSOP Setup time to TFCLK	2	—	ns
tHtsop	TSOP Hold time to TFCLK	0.5	—	ns
tSteop	TEOP Setup time to TFCLK	2	—	ns
tHteop	TEOP Hold time to TFCLK	0.5	—	ns
tStmod	TMOD Setup time to TFCLK	2	—	ns
tHtmod	TMOD Hold time to TFCLK	0.5	—	ns
tSterr	TERR Setup time to TFCLK	2	—	ns
tHterr	TERR Hold time to TFCLK	0.5	—	ns
tPptpa	TFCLK High to PTPA Valid	1.5	6	ns

NOTE(S):
(1) Recommended: 100 MHz

Figure 8-4. Transmit Physical Timing



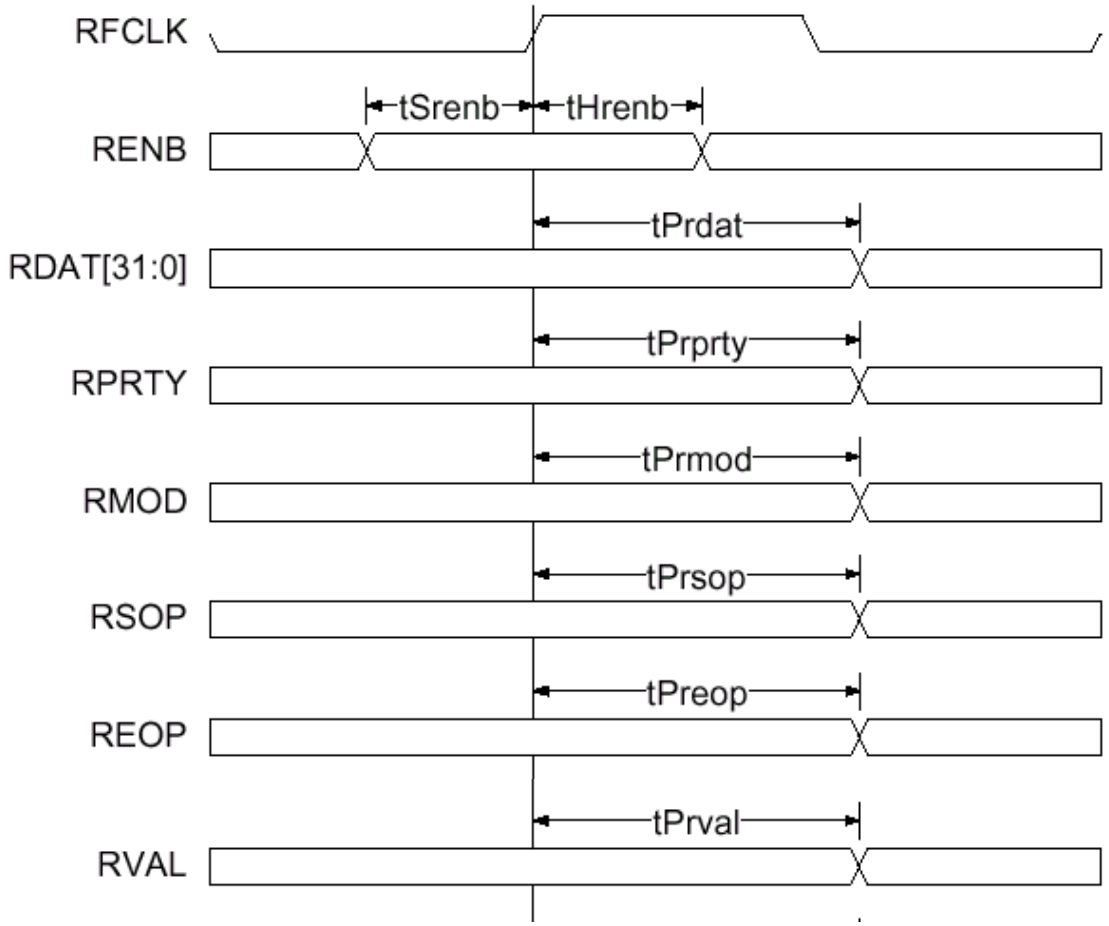
NOTE(S):

1. When a set-up time is specified between an input and a clock, the set-up time is the time in nanoseconds from the 1.4 Volt point of the input to the 1.4 Volt point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4 Volt point of the clock to the 1.4 Volt point of the input.
3. Output propagation delay time is the time in nanoseconds from the 1.4 Volt point of the reference signal to the 1.4 Volt point of the output.
4. Maximum output propagation delays are measured with a 30 pF load on the outputs.

Table 8-10. Receive Interface Timing

Symbol	Description	Min	Max	Units
—	RFCLK/FRFCLK Frequency	—	104	MHz
—	RFCLK/FRFCLK Duty Cycle	40	60	%
tSrenb	RENB/FRENB Set-up time to RFCLK/FRFCLK	2	—	ns
tHrenb	RENB/FRENB Hold time to RFCLK/FRFCLK	0.5	—	ns
tPrdat	RFCLK/FRFCLK High to RDAT/FRDAT Valid	1.5	6	ns
tPrprty	RFCLK/FRFCLK High to RPRTY/FRPRTY Valid	1.5	6	ns
tPrsop	RFCLK/FRFCLK High to RSOP/FRSOP Valid	1.5	6	ns
tPreop	RFCLK/FRFCLK High to REOP/FREOP Valid	1.5	6	ns
tPrmod	RFCLK High to RMOD Valid	1.5	6	ns
tPrval	RFCLK/FRFCLK High to RVAL/FRVAL Valid	1.5	6	ns
(1) Recommended: 100 MHz				

Figure 8-5. Receive Physical Timing



NOTE(S):

1. When a set-up time is specified between an input and a clock, the set-up time is the time in nanoseconds from the 1.4 Volt point of the input to the 1.4 Volt point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4 Volt point of the clock to the 1.4 Volt point of the input.
3. Output propagation delay time is the time in nanoseconds from the 1.4 Volt point of the reference signal to the 1.4 Volt point of the output.
4. Maximum output propagation delays are measured with a 30 pF load on the outputs.

8.2.4 Expansion Bus (EBUS) Timing and Switching Characteristics

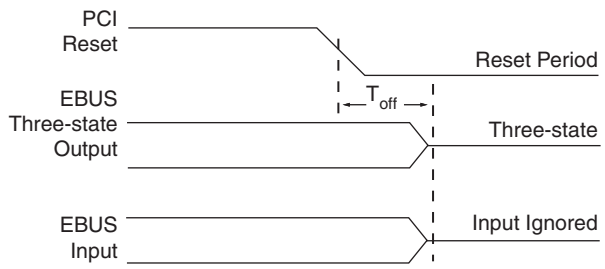
The EBUS timing is derived directly from the PCI clock (PCLK) input into CX28560. The EBUS clock can have the same frequency as the PCI clock, or it can have half the frequency of the PCI clock.

Table 8-11. EBUS Reset Parameters

Symbol	Parameter	Min	Max	Units
T_{off}	Active to Inactive Delay*(1)*	—	28	ns

NOTE(S):
(1) For purposes of active/float timing measurements, the hi-Z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification.

Figure 8-6. EBUS Reset Active to Inactive Delay



NOTE(S): The EBUS reset is dependent on the PRST* (PCI Reset) signal being asserted low.

Table 8-12. EBUS Input/Output Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{val}	ECLK to Signal Valid Delay*(1)*	–0.5	4.5	ns
T_{on}	Float to Active Delay*(2)*	2	—	ns
T_{off}	Active to Float Delay*(2)*	—	28	ns
T_{ds}	Input Setup Time to Clock	18	—	ns
T_{dh}	Input Hold Time from Clock	1	—	ns

NOTE(S):
(1) Minimum and maximum times are evaluated at 40 pF equivalent load. Actual test capacitance may vary, and results should be correlated to these specifications.
(2) For purposes of active/float timing measurements, the hi-Z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification at 40 pF equivalent load.

Table 8-13. EBUS Input/Output Measure Conditions

Symbol	Parameter	Value	Units
V_{th}	Voltage Threshold High*(1)*	0.6 V_{ddo}	V
V_{tl}	Voltage Threshold Low*(1)*	0.2 V_{ddo}	V
V_{test}	Voltage Test Point	0.4 V_{ddo}	V
V_{max}	Maximum Peak-to-Peak*(2)*	0.4 V_{ddo}	V
—	Input Signal Slew Rate	1.5	V/ns

NOTE(S):

(1) The input test for the 3.3 V environment is done with 0.1*V_{ddo} of overdrive. Timing parameters must be met with no more overdrive than this. Production testing may use different voltage values, but must correlate results back to these parameters.

(2) V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Figure 8-7. EBUS Output Timing Waveform

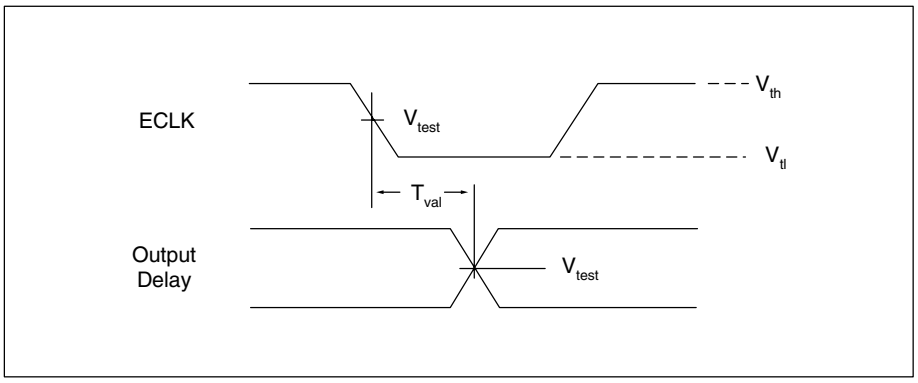
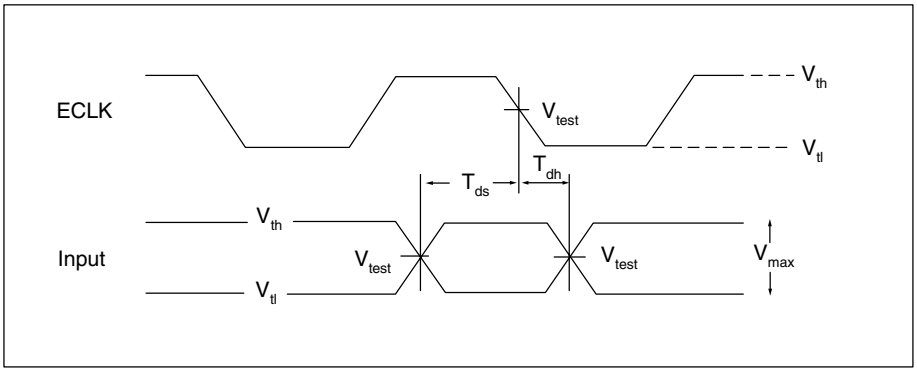
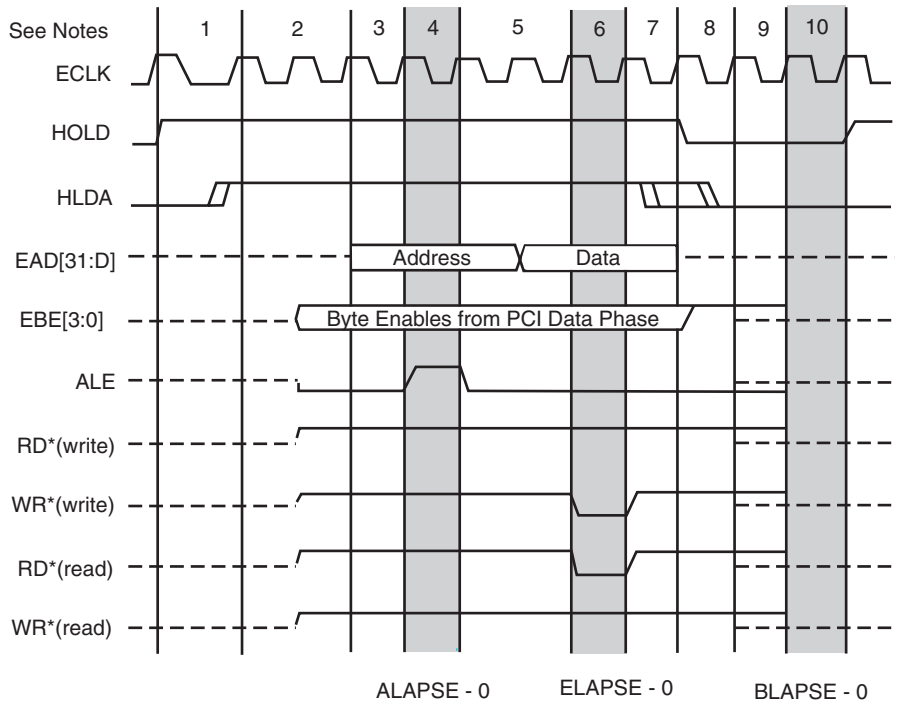


Figure 8-8. EBUS Input Timing Waveform



8.2.5 EBUS Arbitration Timing Specification

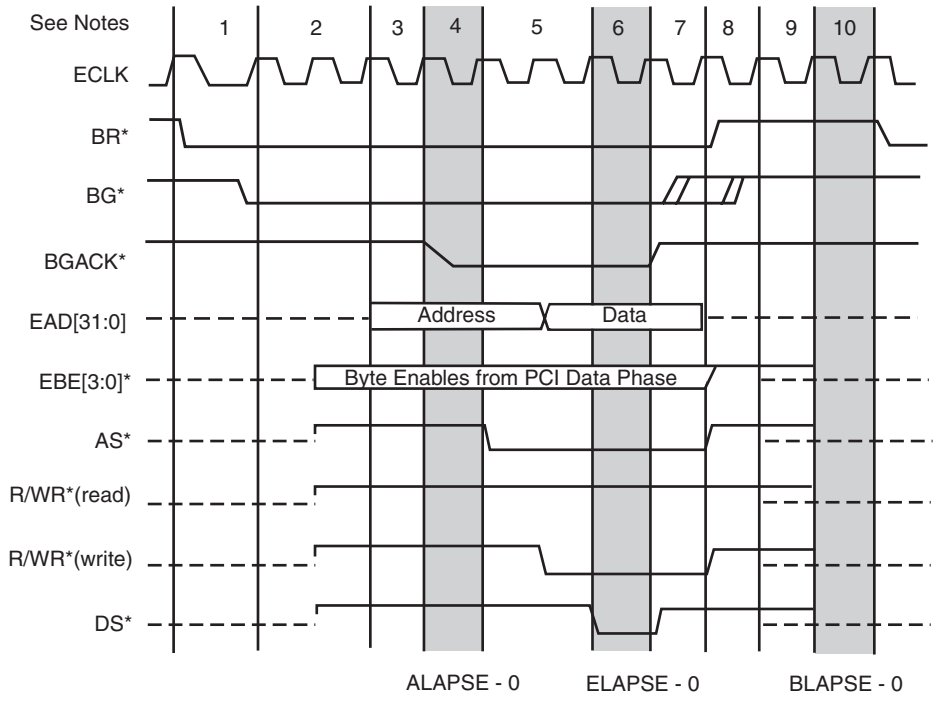
Figure 8-9. EBUS Write/Read Cycle, Intel-Style



NOTE(S):

1. HLDA assertion depends on the external bus arbiter. While HOLD and HLDA are both deasserted, CX28560 places shared EBUS signals in high impedance (three-state, shown as dashed lines).
2. One ECLK cycle after HLDA assertion, CX28560 outputs valid command bus signals: EBE, ALE, RD*, and WR*.
3. Two ECLK cycles after HLDA assertion, CX28560 outputs valid EAD address signals.
4. ALE assertion occurs 3 ECLK cycles after HOLD and HLDA are both asserted. ALAPSE inserts a variable number of ECLK cycles to extend ALE high pulse width and EAD address interval.
5. EAD address remains valid for one ECLK cycle after ALE falling edge. During a write transaction, CX28560 outputs valid EAD write data one ECLK prior to WR* assertion. During a read transaction, EAD data lines are inputs.
6. ELAPSE inserts a variable number of ECLK cycles to extend RD*/WR* low pulse width and EAD data intervals. Read data inputs are sampled on ECLK rising edge coincident with RD* deassertion.
7. EAD write data and EBE byte enables remain valid for one ECLK cycle after RD*/WR* deassertion.
8. One ECLK after RD* or WR* deassertion, HOLD is deasserted and the bus is parked (command bus deasserted, EAD three-state). The bus parked state ends when HLDA is deasserted.
9. Command bus is unparked (three-stated) one ECLK after HLDA deassertion; two different unpark phases are shown, indicating the dependence on HLDA deassertion. If HLDA remained asserted until the next bus request, then command bus remains parked until one ECLK cycle following the next HOLD assertion. Caution: Whenever HLDA is deasserted, all shared EBUS signals are forced to three-state after one ECLK cycle, regardless of whether the EBUS transaction was completed. CX28560 does not reissue or repeat such an aborted transaction.
10. BLAPSE inserts a variable number of ECLK cycles to extend HOLD deassertion interval until the next bus request.

Figure 8-10. EBUS Write/Read Cycle, Motorola-Style



NOTE(S):

1. BG* assertion depends on the external bus arbiter. While BG* and BR* are both deasserted, CX28560 places shared EBUS signals in high impedance (three-state, shown as dashed lines).
2. One ECLK cycle after BG* assertion, CX28560 outputs valid command bus signals: EBE, AS*, R/WR*, and DS*.
3. Two ECLK cycles after BG* assertion, CX28560 outputs valid EAD address signals. BGACK* assertion occurs three ECLK cycles after BG* and BR* are both asserted.
4. ALAPSE inserts a variable number of ECLK cycles to extend AS* high pulse width and EAD address interval.
5. EAD address remains valid for one ECLK cycle after AS* falling edge. During a write transaction, CX28560 asserts R/WR* and outputs valid EAD write data one ECLK prior to DS* assertion. During a read transaction, EAD data lines are inputs.
6. ELAPSE inserts a variable number of ECLK cycles to extend DS* low pulse width and EAD data interval. Read data inputs are sampled on ECLK rising edge coincident with DS* deassertion.
7. EAD write data, EBE, R/WR* and AS* signals remain valid for one ECLK cycle after BGACK* and DS* are deasserted.
8. One ECLK cycle after BGACK* deassertion, the BR* output is deasserted and the bus is parked (command bus deasserted, EAD three-state). The bus parked state ends when the external bus arbiter deasserts BG*.
9. Command bus is unparked (three-stated) one ECLK after BG* deassertion; two different unpark phases are shown, indicating the dependence on BG* deassertion. If BG* remained asserted until the next bus request, then command bus remains parked until one ECLK following the next BR* assertion. Caution: Whenever BG* is deasserted, all shared EBUS signals are forced to three-state after one ECLK cycle, regardless of whether the EBUS transaction was completed. CX28560 does not reissue or repeat such an aborted transaction.
10. BLAPSE inserts a variable number of ECLK cycles to extend BR* deassertion interval until the next bus request.

8.2.6 Serial Interface Timing and Switching Characteristics

Table 8-14. Serial Interface Clock (RCLK, TCLK) Parameters

Symbol	Parameter	Min	Max	Units
F_c	Clock Frequency	DC	52	MHz
T_r	Clock Rise Time for Fc ≤ 10 MHz	—	20	ns
T_r	Clock Rise Time for Fc >10 MHz	—	3	ns
T_f	Clock Fall Time for Fc ≤ 10 MHz	—	20	ns
—	Clock Fall Time for Fc >10 MHz	—	3	ns
—	Clock Duty Cycle	40	60	%

Figure 8-11. Serial Interface Clock (RCLK,TCLK) Waveform

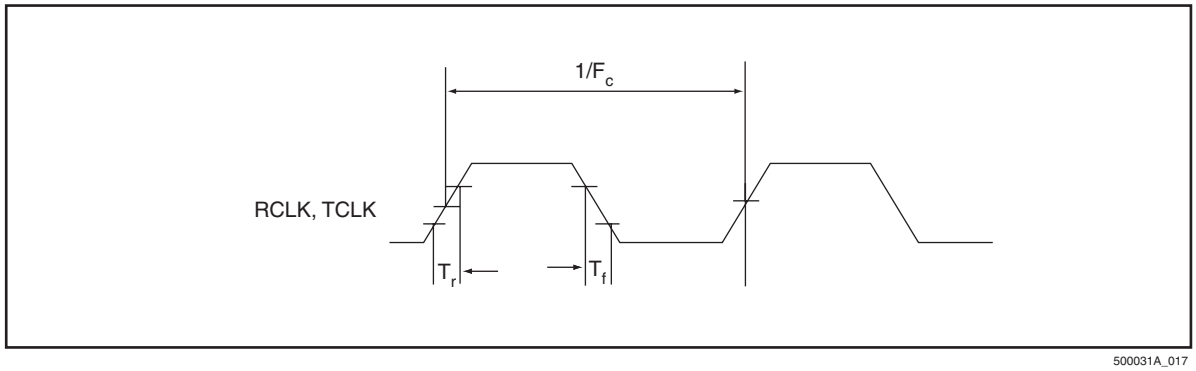


Table 8-15. Serial Interface Input/Output Timing Parameters

Symbol	Parameter	Min	Max	Units
T^{2}_{val}	Clock to Signal Valid Delay for Fc ≤ 10 MHz	2	30	ns
	Clock to Signal Valid Delay for Fc > 10 MHz	2	8	ns
T^{3}_{ds}	Data Setup Time for Fc ≤ 10 MHz	15	—	ns
	Data Setup Time for Fc >10 MHz	2	—	ns
T^{3}_{dh}	Data Hold Time for Fc ≤ 10 MHz	15	—	ns
	Data Hold Time for Fc >10 MHz	3	—	ns

NOTE(S):
1. Parameters were characterized with C load = 70 pF
2. Output Delay
3. Input Signals

Table 8-16. Serial Interface Input/Output Measure Conditions

Symbol	Parameter	Value	Units
V_{th}	Voltage Threshold High*(1)*	0.6 V_{ddo}	V
V_{tl}	Voltage Threshold Low*(1)*	0.2 V_{ddo}	V
V_{test}	Voltage Test Point	0.4 V_{ddo}	V
V_{max}	Maximum Peak-to-Peak*(2)*	0.4 V_{ddo}	V
—	Input Signal Slew Rate	1.5	V/ns
C_{ld}	Maximum Load capacitance–output and I/0	70	pF

NOTE(S):

(1) The input test for the 3.3 V environment is done with 0.1*V_{ddo} of overdrive. Timing parameters must be met with no more overdrive than this. Production testing may use different voltage values, but must correlate results back to these parameters.

(2) V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Figure 8-12. Serial Interface Data Input Waveform

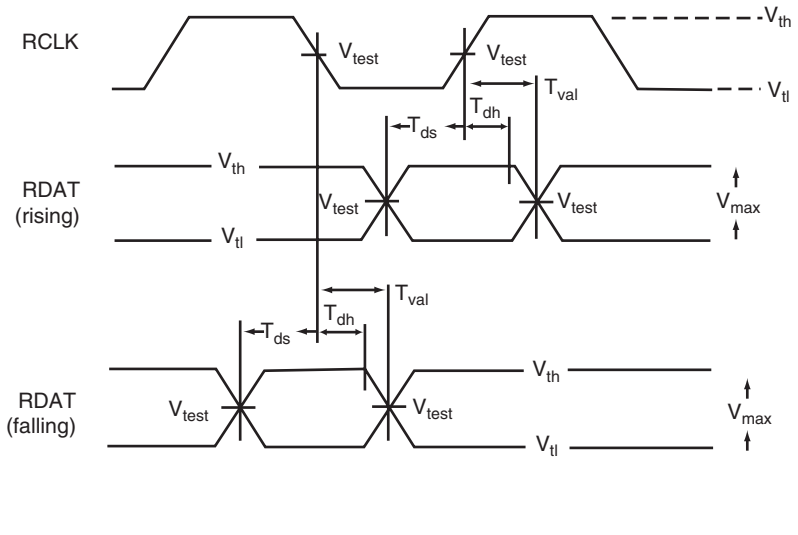


Figure 8-13. Serial Interface Data Delay Output Waveform

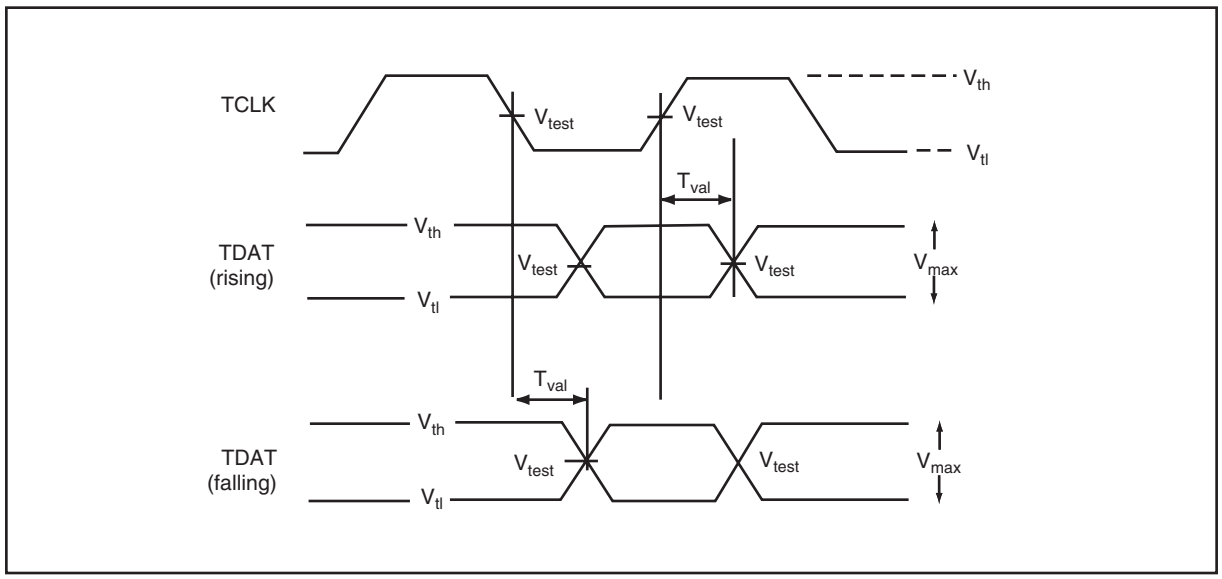
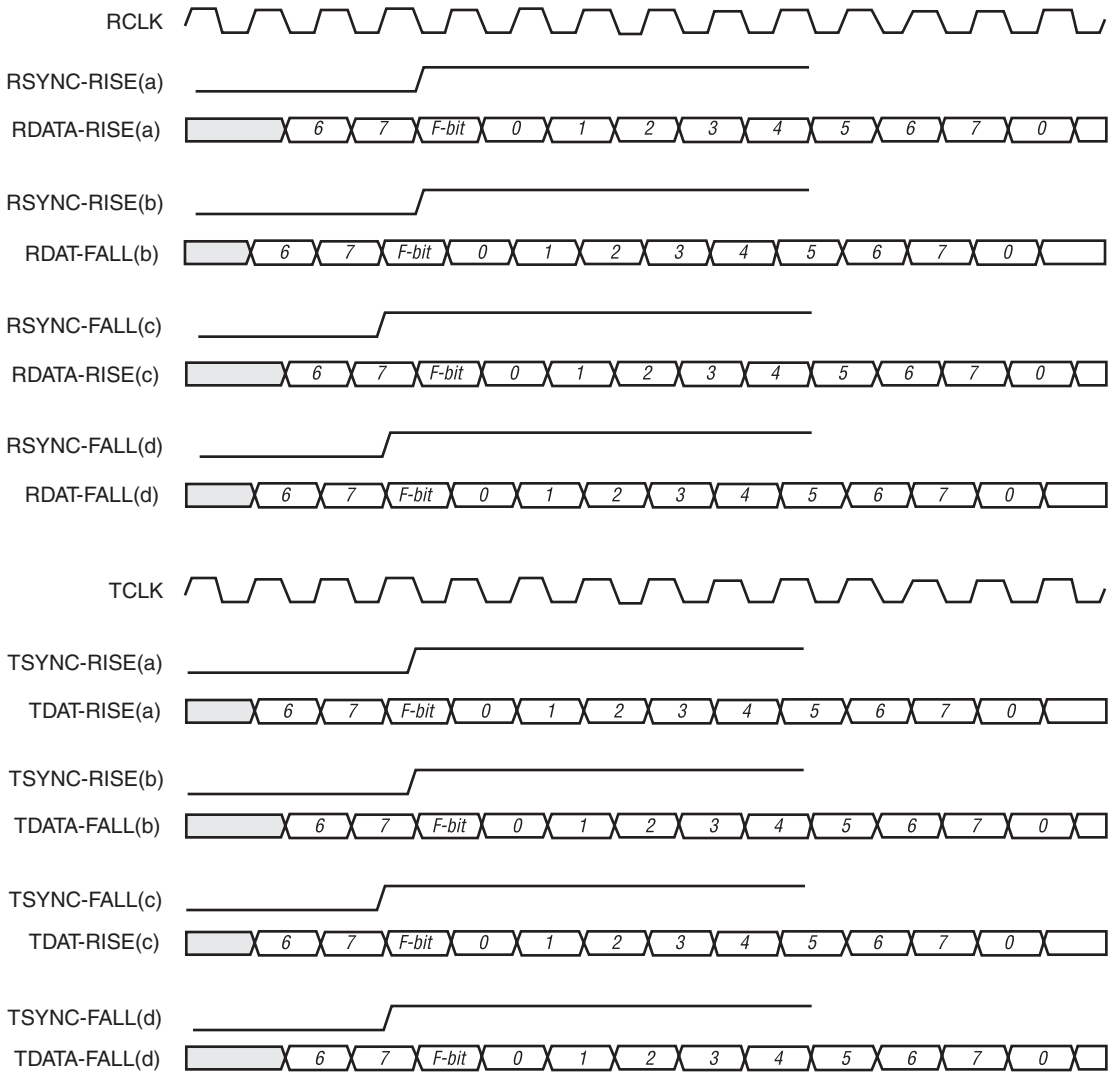


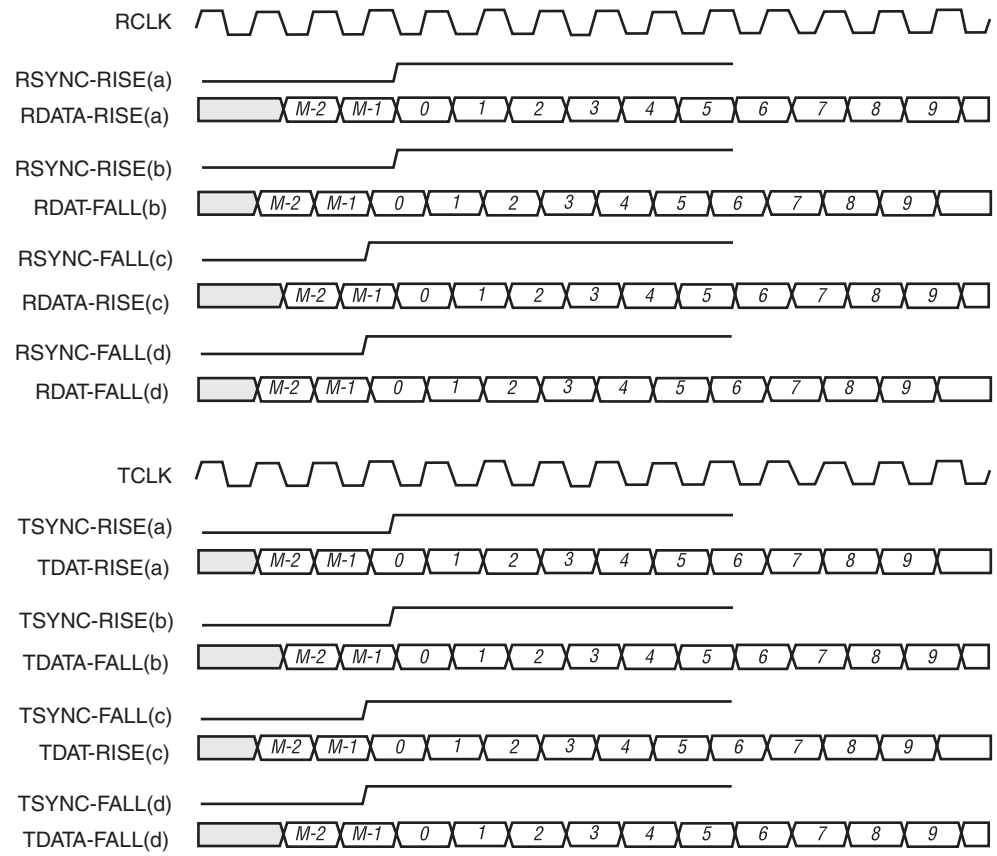
Figure 8-14. Transmit and Receive T1 Mode



NOTE(S):

1. T1 Mode employs 24 time slots (0–23) with 8 bits per time slot (0–7) and 1 Frame-bit every 193 clock periods. One frame of 193 bits occurs every 125 µs (1.544 MHz).
2. RSYNC and TSYNC must be asserted for a minimum of 1 CLK period.
3. CX28560 can be configured to sample RSYNC, TSYNC, RDAT, and TDAT on either a rising or falling clock edge independently of any other signal sampling configuration.
4. Relationships between the various configurations of active edges for the synchronization signal and the data signal are shown using a common clock signal for receive and transmit operations. Note the relationship between the frame bit (within RDAT, TDAT) and the frame synchronization signal (e.g., RSYNC, TSYNC).
5. All received signals (e.g., RSYNC, RDAT, TSYNC) are “sampled” in on the specified clock edge (e.g., RCLK, TCLK). All transmit data signals (TDAT) are latched on the specified clock edge.
6. In configuration (a), synchronization and data signals are sampled/latched on a rising clock edge.
7. In configuration (b), synchronization signal is sampled on a rising clock edge and the data signal is sampled/latched on a falling clock edge.
8. In configuration (c), synchronization signal is sampled on a falling clock edge and the data signal is sampled/latched on a rising clock edge.
9. In configuration (d), synchronization and data signals are sampled/latched on a falling clock edge.

Figure 8-15. Transmit and Receive Channelized Non-T1 (i.e., N x 64) Mode



LEGEND: M = N∞8 bits, where M = number of time slots.

NOTE(S):

1. E1 Mode employs 32 time slots (0–31) with 8 bits per time slot (0–7) and 256 bits per frame and one frame every 125 µs (2.048 MHz).
2. 2xE1 Mode employs 64 time slots (0–63) with 8 bits per time slot (0–7) and 512 bits per frame and one frame every 125 µs (4.096 MHz).
3. 4xE1 Mode employs 128 time slots (0–127) with 8 bits per time slot (0–7) and 1024 bits per frame and one frame every 125 µs (8.192 MHz).
4. RSYNC and TSYNC must be asserted for a minimum of 1 CLK period.
5. CX28560 can be configured to sample RSYNC, TSYNC, RDAT, and TDAT on either a rising or falling clock edge independently of any other signal sampling configuration.
6. Relationships between the various configurations of active edges for the synchronization signal and the data signal are shown using a common clock signal for receive and transmit operations. Note the relationship between the frame bit (within RDAT, TDAT) and the frame synchronization signal (e.g. RSYNC, TSYNC).
7. All received signals (e.g., RSYNC, RDAT, TSYNC) are sampled in on the specified clock edge (e.g. RCLK, TCLK). All transmit data signals (TDAT) are latched on the specified clock edge.
8. In configuration (a), synchronization and data signals are sampled/latched on a rising clock edge.
9. In configuration (b), synchronization signal is sampled on a rising clock edge and the data signal is sampled/latched on a falling clock edge.
10. In configuration (c), synchronization signal is sampled on a falling clock edge and the data signal is sampled/latched on a rising clock edge.
11. In configuration (d), synchronization and data signals are sampled/latched on a falling clock edge.
12. In TSBUS mode, the timing is identical to that in non-T1 mode. In order to convert the above diagram to TSBUS mode, the names of the signals should be replaced as described in Chapter 1.0, Pin Descriptions. The additional 2 pins in the first 12 ports (RGSYNC and TGSYNC) behave in an identical manner to RDAT and TDAT respectively.

8.2.7 Test and Diagnostic Interface Timing

Table 8-17. Test and Diagnostic Interface Timing Requirements

Symbol	Parameter	Minimum	Maximum	Units
1	TCK Pulse-Width High	80	—	ns
2	TCK Pulse-Width Low	80	—	ns
3	TMS, TDI Setup Prior to TCK Rising Edge(1)	15	—	ns
4	TMS, TDI Hold after TCK High(1)	20	—	ns

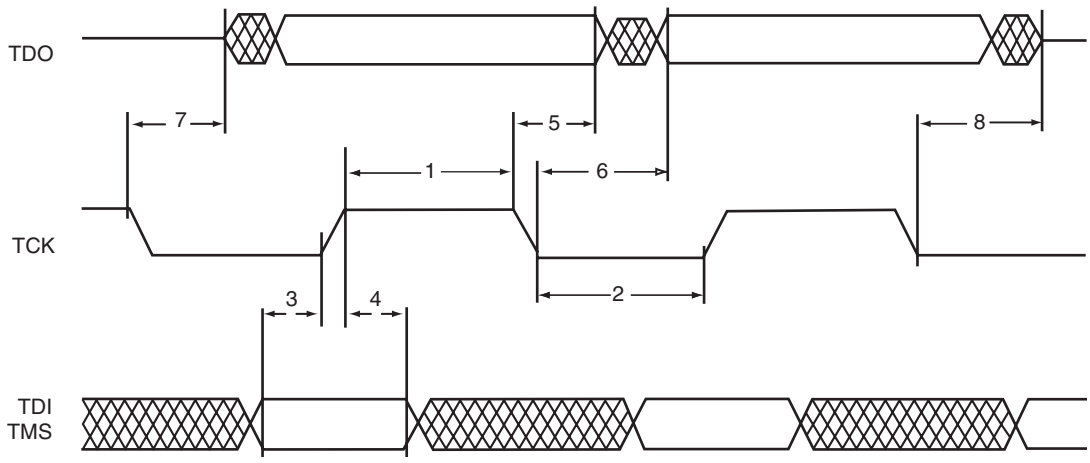
NOTE(S):
(1) Also applies to functional inputs for SAMPLE/PRELOAD and EXTEST instructions.

Table 8-18. Test and Diagnostic Interface Switching Characteristics

Symbol	Parameter	Minimum	Maximum	Units
5	TDO Hold after TCK Falling Edge	0	—	ns
6	TDO Delay after TCK Low	—	50	ns
7	TDO Enable (Low Z) after TCK Falling Edge	2	15	ns
8	TDO Disable (High Z) after TCK Low	—	25	ns

NOTE(S): Also applies to functional outputs for the EXTEST instruction.

Figure 8-16. JTAG Interface Timing



NOTE(S): Please refer to Tables 8-14 and 8-15 for numerical symbol reference.

8.3 Package Thermal Specification

Theta JA for:

0 lfpm: 10.0 °C/W

100 lfpm: 8.8 °C/W

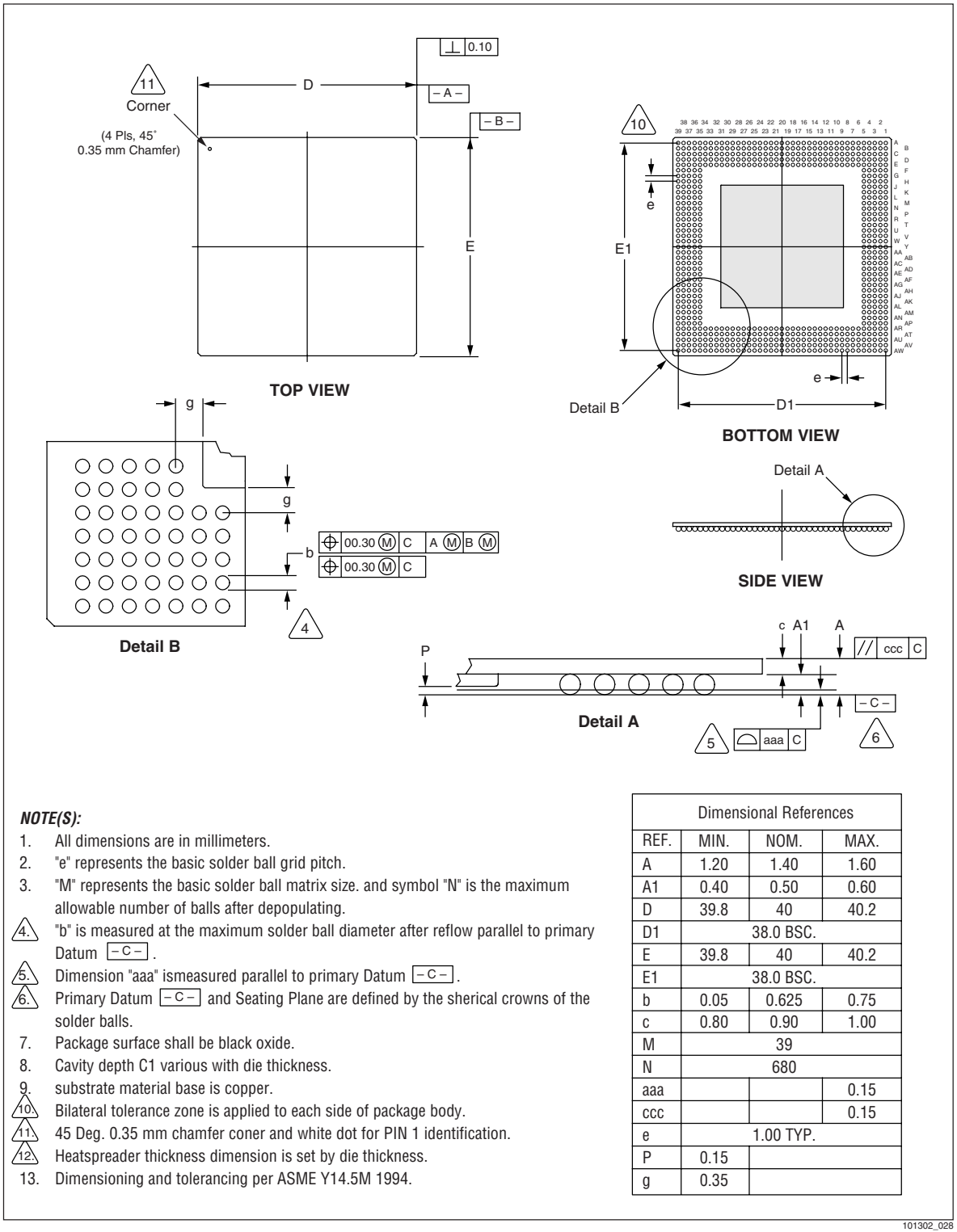
200 lfpm: 8.3 °C/W

400 lfpm: 7.7 °C/W

600 lfpm: 7.1 °C/W

8.4 Mechanical Specification

Figure 8-17. Package Diagram



NOTE(S):

1. All dimensions are in millimeters.
2. "e" represents the basic solder ball grid pitch.
3. "M" represents the basic solder ball matrix size. and symbol "N" is the maximum allowable number of balls after depopulating.
4. "b" is measured at the maximum solder ball diameter after reflow parallel to primary Datum – C – .
5. Dimension "aaa" ismeasured parallel to primary Datum – C – .
6. Primary Datum – C – and Seating Plane are defined by the sherical crowns of the solder balls.
7. Package surface shall be black oxide.
8. Cavity depth C1 various with die thickness.
9. substrate material base is copper.
10. Bilateral tolerance zone is applied to each side of package body.
11. 45 Deg. 0.35 mm chamfer coner and white dot for PIN 1 identification.
12. Heatspreader thickness dimension is set by die thickness.
13. Dimensioning and tolerancing per ASME Y14.5M 1994.

Dimensional References			
REF.	MIN.	NOM.	MAX.
A	1.20	1.40	1.60
A1	0.40	0.50	0.60
D	39.8	40	40.2
D1		38.0 BSC.	
E	39.8	40	40.2
E1		38.0 BSC.	
b	0.05	0.625	0.75
c	0.80	0.90	1.00
M		39	
N		680	
aaa			0.15
ccc			0.15
e		1.00 TYP.	
P	0.15		
g	0.35		

101302_028

9.0 Package Description

Table 9-1. Pin List for 28560 HDLC Controller—Alphabetic Order (1 of 2)

Ball	Reference
A1	Gnd
A2	Gnd
A3	Gnd
A4	Gnd
A36	Gnd
A37	Gnd
A38	Gnd
A39	Gnd
B1	Gnd
B2	Gnd
B3	Gnd
B4	Gnd
B10	Gnd
B16	Gnd
B22	Gnd
B28	Gnd
B34	Gnd
B36	Gnd
B37	Gnd
B38	Gnd
B39	Gnd
C1	Gnd
C2	Gnd
C3	Gnd
C4	Gnd
C36	Gnd
C37	Gnd
C38	Gnd
C39	Gnd
D1	Gnd
D2	Gnd
D3	Gnd
D4	Gnd
D36	Gnd
D37	Gnd
D38	Gnd

Ball	Reference
D39	Gnd
E1	Gnd
E5	Gnd
E8	Gnd
E10	Gnd
E12	Gnd
E14	Gnd
E16	Gnd
E18	Gnd
E20	Gnd
E22	Gnd
E24	Gnd
E26	Gnd
E28	Gnd
E30	Gnd
E32	Gnd
E34	Gnd
E39	Gnd
F3	Gnd
H2	Gnd
H5	Gnd
H35	Gnd
H38	Gnd
K5	Gnd
K35	Gnd
M2	Gnd
M5	Gnd
M35	Gnd
M38	Gnd
P5	Gnd
P35	Gnd
T2	Gnd
T5	Gnd
T35	Gnd
T38	Gnd
V5	Gnd

Ball	Reference
V35	Gnd
Y2	Gnd
Y5	Gnd
Y35	Gnd
Y38	Gnd
AB5	Gnd
AB35	Gnd
AD2	Gnd
AD5	Gnd
AD35	Gnd
AD38	Gnd
AF5	Gnd
AF35	Gnd
AH2	Gnd
AH5	Gnd
AH35	Gnd
AH38	Gnd
AK3	Gnd
AK5	Gnd
AK35	Gnd
AL2	Gnd
AM1	Gnd
AM3	Gnd
AM5	Gnd
AM35	Gnd
AM38	Gnd
AN4	Gnd
AP2	Gnd
AR2	Gnd
AR5	Gnd
AR8	Gnd
AR10	Gnd
AR12	Gnd
AR14	Gnd
AR16	Gnd
AR18	Gnd

Table 9-1. Pin List for 28560 HDLC Controller—Alphabetic Order (2 of 2)

Ball	Reference
AR20	Gnd
AR22	Gnd
AR24	Gnd
AR26	Gnd
AR28	Gnd
AR30	Gnd
AR32	Gnd
AR35	Gnd
AT1	Gnd
AT2	Gnd
AT3	Gnd
AT5	Gnd
AT8	Gnd
AT14	Gnd
AT35	Gnd
AT36	Gnd
AT37	Gnd
AT38	Gnd
AT39	Gnd
AU1	Gnd
AU2	Gnd
AU3	Gnd
AU4	Gnd
AU6	Gnd
AU9	Gnd
AU12	Gnd
AU36	Gnd
AU37	Gnd
AU38	Gnd
AU39	Gnd
AV1	Gnd
AV2	Gnd
AV3	Gnd
AV4	Gnd
AV7	Gnd

Ball	Reference
AV10	Gnd
AV15	Gnd
AV19	Gnd
AV23	Gnd
AV27	Gnd
AV31	Gnd
AV34	Gnd
AV36	Gnd
AV37	Gnd
AV38	Gnd
AV39	Gnd
AW1	Gnd
AW2	Gnd
AW3	Gnd
AW4	Gnd
AW5	Gnd
AW13	Gnd
AW34	Gnd
AW35	Gnd
AW36	Gnd
AW37	Gnd
AW38	Gnd
AW39	Gnd

Table 9-2. BGA Assignments for Power (Vddc, Vddo and Vgg)

Ball	Supply Type
E6	Vddc
E7	Vddc
E11	Vddc
E15	Vddc
E19	Vddc
E23	Vddc
E27	Vddc
E31	Vddc
F35	Vddc
G35	Vddc
J5	Vddc
L35	Vddc
N5	Vddc
R35	Vddc
U5	Vddc
W35	Vddc
AA5	Vddc
AC35	Vddc
AE5	Vddc
AG35	Vddc
AJ5	Vddc
AL35	Vddc
AN5	Vddc
AP5	Vddc
AR9	Vddc
AR13	Vddc
AR17	Vddc
AR21	Vddc
AR25	Vddc
AR29	Vddc
AR33	Vddc
AR34	Vddc
E9	Vddo
E13	Vddo
E17	Vddo

Ball	Supply Type
E21	Vddo
E25	Vddo
E29	Vddo
E33	Vddo
F5	Vddo
G5	Vddo
J35	Vddo
L5	Vddo
N35	Vddo
R5	Vddo
U35	Vddo
W5	Vddo
AA35	Vddo
AC5	Vddo
AE35	Vddo
AG5	Vddo
AJ35	Vddo
AL5	Vddo
AN35	Vddo
AP35	Vddo
AR6	Vddo
AR7	Vddo
AR11	Vddo
AR15	Vddo
AR19	Vddo
AR23	Vddo
AR27	Vddo
AR31	Vddo
AT11	Vddo
E35	Vgg
AU7	Vgg

Table 9-3. Signals (1 of 7)

Pad#	Name	Package Ball
3	TDAT[17]	F4
4	TSYNC[17]/TSTUFF[17]	E2
5	TCLK[17]	E3
6	RCLK[17]	E4
9	ROOF[17]/CTS[17]/TSTB[17]	A5
10	RSYNC[17]/RSTUFF[17]	B5
11	RDAT[17]	C5
12	ROOF[16]/CTS[16]/TSTB[16]	D5
13	TDAT[16]	A6
14	TSYNC[16]/TSTUFF[16]	B6
15	TCLK[16]	C6
16	RCLK[16]	D6
19	RSYNC[16]/RSTUFF[16]	A7
20	RDAT[16]	B7
21	ROOF[15]/CTS[15]/TSTB[15]	C7
22	TDAT[15]	D7
23	TSYNC[15]/TSTUFF[15]	A8
24	TCLK[15]	B8
27	RCLK[15]	C8
28	RSYNC[15]/RSTUFF[15]	D8
29	RDAT[15]	A9
30	ROOF[14]/CTS[14]/TSTB[14]	B9
31	TDAT[14]	C9
32	TSYNC[14]/TSTUFF[14]	D9
33	TCLK[14]	A10
36	RCLK[14]	C10
37	RSYNC[14]/RSTUFF[14]	D10
38	RDAT[14]	A11
39	ROOF[13]/CTS[13]/TSTB[13]	B11
40	TDAT[13]	C11
41	TSYNC[13]/TSTUFF[13]	D11
42	TCLK[13]	A12
43	RCLK[13]	B12
46	RSYNC[13]/RSTUFF[13]	C12
47	RDAT[13]	D12
48	ROOF[12]/CTS[12]/TSTB[12]	A13
49	TDAT[12]	B13
50	TSYNC[12]/TSTUFF[12]	C13
51	TCLK[12]	D13
52	RCLK[12]	A14
53	RSYNC[12]/RSTUFF[12]	B14
54	RDAT[12]	C14
57	ROOF[11]/CTS[11]/TSTB[11]	D14
58	TDAT[11]	A15
59	TSYNC[11]/TSTUFF[11]	B15
60	TCLK[11]	C15
61	RCLK[11]	D15
64	RDAT[11]	A16
65	TGSYNC[11]	C16
66	RGSYNC[11]	D16
67	RSYNC[11]/RSTUFF[11]	A17
68	ROOF[10]/CTS[10]/TSTB[10]	B17
69	TDAT[10]	C17
71	TSYNC[10]/TSTUFF[10]	D17
72	TCLK[10]	A18
73	TGSYNC[10]	B18
74	RGSYNC[10]	C18
77	RSYNC[10]/RSTUFF[10]	D18
78	RCLK[10]	A19
79	RDAT[10]	B19
80	ROOF[9]/CTS[9]/TSTB[9]	C19
81	TDAT[9]	D19
82	TSYNC[9]/TSTUFF[9]	A20
83	TGSYNC[9]	B20
84	RGSYNC[9]	C20
85	TCLK[9]	D20

Table 9-3. Signals (2 of 7)

Pad#	Name	Package Ball
86	RCLK[9]	A21
87	RSYNC[9]/RSTUFF[9]	B21
88	RDAT[9]	C21
89	ROOF[8]/CTS[8]/TSTB[8]	D21
90	TGSYNC[8]	A22
93	RGSYNC[8]	C22
94	TDAT[8]	D22
95	TSYNC[8]/TSTUFF[8]	A23
96	TCLK[8]	B23
99	RCLK[8]	C23
100	RSYNC[8]/RSTUFF[8]	D23
101	RDAT[8]	A24
102	TGSYNC[7]	B24
103	RGSYNC[7]	C24
104	ROOF[7]/CTS[7]/TSTB[7]	D24
105	TDAT[7]	A25
106	TSYNC[7]/TSTUFF[7]	B25
107	TCLK[7]	C25
110	RCLK[7]	D25
111	RSYNC[7]/RSTUFF[7]	A26
112	RDAT[7]	B26
113	ROOF[6]/CTS[6]/TSTB[6]	C26
115	TDAT[6]	D26
116	TSYNC[6]/TSTUFF[6]	A27
117	TGSYNC[6]	B27
118	RGSYNC[6]	C27
119	TCLK[6]	D27
122	RCLK[6]	A28
123	RSYNC[6]/RSTUFF[6]	C28
124	RDAT[6]	D28
125	ROOF[5]/CTS[5]/TSTB[5]	A29
126	TDAT[5]	B29
129	TSYNC[5]/TSTUFF[5]	C29

Pad#	Name	Package Ball
130	TCLK[5]	D29
131	TGSYNC[5]	A30
132	RGSYNC[5]	B30
133	RCLK[5]	C30
136	RSYNC[5]/RSTUFF[5]	D30
137	RDAT[5]	A31
138	ROOF[4]/CTS[4]/TSTB[4]	B31
139	TDAT[4]	C31
141	TSYNC[4]/TSTUFF[4]	D31
142	TCLK[4]	A32
143	TGSYNC[4]	B32
144	RGSYNC[4]	C32
147	RCLK[4]	D32
148	RSYNC[4]/RSTUFF[4]	A33
149	RDAT[4]	B33
150	ROOF[3]/CTS[3]/TSTB[3]	C33
151	TDAT[3]	D33
154	TSYNC[3]/TSTUFF[3]	A34
155	TCLK[3]	C34
156	TGSYNC[3]	D34
157	RGSYNC[3]	A35
160	RCLK[3]	B35
161	RSYNC[3]/RSTUFF[3]	C35
162	RDAT[3]	D35
163	ROOF[2]/CTS[2]/TSTB[2]	E38
164	TDAT[2]	E37
165	TSYNC[2]/TSTUFF[2]	E36
169	TCLK[2]	F39
170	TGSYNC[2]	F38
171	RGSYNC[2]	F37
173	RCLK[2]	F36
176	RSYNC[2]/RSTUFF[2]	G39
177	RDAT[2]	G38

Table 9-3. Signals (3 of 7)

Pad#	Name	Package Ball
178	ROOF[1]/CTS[1]/TSTB[1]	G37
181	TDAT[1]	G36
182	TSYNC[1]/TSTUFF[1]	H39
185	TCLK[1]	H37
186	TGSYNC[1]	H36
187	RGSYNC[1]	J39
188	RCLK[1]	J38
189	RSYNC[1]/RSTUFF[1]	J37
190	RDAT[1]	J36
193	ROOF[0]/CTS[0]/TSTB[0]	K39
194	TDAT[0]	K38
195	TSYNC[0]/TSTUFF[0]	K37
198	TCLK[0]	K36
199	TGSYNC[0]	L39
200	RGSYNC[0]	L38
201	RCLK[0]	L37
202	RSYNC[0]/RSTUFF[0]	L36
203	RDAT[0]	M39
204	ROOF[24]/CTS[24]/TSTB[24]	M37
205	TDAT[24]	M36
206	TSYNC[24]/TSTUFF[24]	N39
207	TCLK[24]	N38
210	RCLK[24]	N37
211	RSYNC[24]/RSTUFF[24]	N36
214	RDAT[24]	P39
215	ROOF[25]/CTS[25]/TSTB[25]	P38
216	TDAT[25]	P37
217	TSYNC[25]/TSTUFF[25]	P36
218	RCLK[25]	R39
221	TCLK[25]	R38
222	RSYNC[25]/RSTUFF[25]	R37
223	RDAT[25]	R36
224	ROOF[26]/CTS[26]/TSTB[26]	T39

Pad#	Name	Package Ball
225	TDAT[26]	T37
227	TCLK[26]	T36
228	TSYNC[26]/TSTUFF[26]	U39
229	RCLK[26]	U38
232	RSYNC[26]/RSTUFF[26]	U37
233	RDAT[26]	U36
234	ROOF[27]/CTS[27]/TSTB[27]	V39
235	TDAT[27]	V38
236	TSYNC[27]/TSTUFF[27]	V37
237	TCLK[27]	V36
238	RCLK[27]	W39
239	RSYNC[27]/RSTUFF[27]	W38
240	RDAT[27]	W37
243	ROOF[28]/CTS[28]/TSTB[28]	W36
244	TDAT[28]	Y39
245	TSYNC[28]/TSTUFF[28]	Y37
246	TCLK[28]	Y36
247	RCLK[28]	AA39
248	RSYNC[28]/RSTUFF[28]	AA38
249	RDAT[28]	AA37
250	ROOF[29]/CTS[29]/TSTB[29]	AA36
251	TDAT[29]	AB39
252	TSYNC[29]/TSTUFF[29]	AB38
255	TCLK[29]	AB37
256	RCLK[29]	AB36
257	RSYNC[29]/RSTUFF[29]	AC39
258	RDAT[29]	AC38
259	ROOF[30]/CTS[30]/TSTB[30]	AC37
260	TDAT[30]	AC36
263	TSYNC[30]/TSTUFF[30]	AD39
264	TCLK[30]	AD37
265	RCLK[30]	AD36
266	RSYNC[30]/RSTUFF[30]	AE39

Table 9-3. Signals (4 of 7)

Pad#	Name	Package Ball
267	RDAT[30]	AE38
268	ROOF[31]/CTS[31]/TSTB[31]	AE37
269	TDAT[31]	AE36
270	TSYNC[31]/TSTUFF[31]	AF39
271	TCLK[31]	AF38
272	RCLK[31]	AF37
273	RSYNC[31]/RSTUFF[31]	AF36
274	RDAT[31]	AG39
276	FRCLAV	AG38
277	FREOP	AG37
280	FRSOP	AG36
281	FRPRTY	AH39
282	FRDAT[0]	AH37
285	FRDAT[1]	AH36
286	FRDAT[2]	AJ39
287	FRDAT[3]	AJ38
290	FRDAT[4]	AJ37
291	FRDAT[5]	AJ36
294	FRDAT[6]	AK39
295	FRDAT[7]	AK38
296	FRENB	AK37
297	FRVAL	AK36
298	FRFCLK	AL39
301	TDATA[0]	AL38
302	TDATA[1]	AL37
303	TDATA[2]	AL36
304	TDATA[3]	AM39
305	TDATA[4]	AM37
306	TDATA[5]	AM36
307	TDATA[6]	AN39
308	TDATA[7]	AN38
309	TDATA[8]	AN37
312	TDATA[9]	AN36

Pad#	Name	Package Ball
313	TDATA[10]	AP39
314	TDATA[11]	AP38
315	TDATA[12]	AP37
316	TDATA[13]	AP36
317	TDATA[14]	AR39
318	TDATA[15]	AR38
319	TDATA[16]	AR37
322	TDATA[17]	AR36
323	TDATA[18]	AU35
324	TDATA[19]	AV35
325	TDATA[20]	AU34
326	TDATA[21]	AT34
327	TDATA[22]	AW33
328	TDATA[23]	AV33
332	TDATA[24]	AU33
333	TDATA[25]	AT33
334	TDATA[26]	AW32
335	TDATA[27]	AV32
336	TDATA[28]	AU32
337	TDATA[29]	AT32
338	TDATA[30]	AW31
339	TDATA[31]	AU31
342	PTPA	AT31
343	TERR	AW30
344	TEOP	AV30
345	TSOP	AU30
346	TPRTY	AT30
349	TMOD[0]	AW29
350	TMOD[1]	AV29
351	TENB	AU29
354	TFCLK	AT29
355	TM[0]	AW28
356	TM[1]	AV28

Table 9-3. Signals (5 of 7)

Pad#	Name	Package Ball
357	TM[2]	AU28
359	TM[3]	AT28
360	AD[0]	AW27
361	AD[1]	AU27
362	AD[2]	AT27
363	AD[3]	AW26
366	AD[4]	AV26
367	AD[5]	AU26
368	CBE[0]	AT26
372	AD[6]	AV25
369	AD[7]	AW25
373	AD[8]	AU25
375	AD[9]	AW24
374	AD[10]	AT25
376	AD[11]	AV24
377	AD[12]	AU24
378	AD[13]	AT24
379	AD[14]	AW23
382	AD[15]	AU23
383	CBE[1]	AT23
384	PAR	AW22
385	SERR	AV22
386	PERR	AU22
387	STOP	AT22
390	DEVSEL	AW21
391	TRDY	AV21
394	IRDY	AU21
395	FRAME	AT21
396	CBE[2]	AW20
397	AD[16]	AV20
398	AD[17]	AU20
399	AD[18]	AT20
400	AD[19]	AW19

Pad#	Name	Package Ball
403	AD[20]	AU19
404	AD[21]	AT19
405	AD[22]	AW18
406	AD[23]	AV18
409	IDSEL	AU18
410	CBE[3]	AT18
411	AD[24]	AW17
412	AD[25]	AV17
413	AD[26]	AU17
416	AD[27]	AT17
417	AD[28]	AW16
418	AD[29]	AV16
419	AD[30]	AU16
420	AD[31]	AT16
421	REQ	AW15
422	GNT	AU15
424	PCLK	AT15
427	PRST	AW14
428	INTA	AV14
431	ONESEC	AU14
434	RCLAV	AV13
435	REOP	AU13
436	RSOP	AT13
439	RPRTY	AW12
440	RMOD[0]	AV12
441	RMOD[1]	AT12
444	RDATA[0]	AW11
445	RDATA[1]	AV11
446	RDATA[2]	AU11
449	RDATA[3]	AW10
450	RDATA[4]	AU10
455	RDATA[5]	AT10
456	RDATA[6]	AW9

Table 9-3. Signals (6 of 7)

Pad#	Name	Package Ball
460	RDATA[7]	AV9
461	RDATA[8]	AT9
464	RDATA[9]	AW8
465	RDATA[10]	AV8
470	RDATA[11]	AU8
471	RDATA[12]	AW7
477	RDATA[13]	AT7
478	RDATA[14]	AW6
481	RDATA[15]	AV6
482	RDATA[16]	AT6
485	RDATA[17]	AV5
486	RDATA[18]	AU5
489	RENB	AT4
490	RVAL	AR4
493	RDATA[19]	AR3
494	RDATA[20]	AR1
496	RFCLK	AP4
498	RDATA[21]	AP3
499	RDATA[22]	AP1
502	RDATA[23]	AN3
503	RDATA[24]	AN2
506	RDATA[25]	AN1
507	RDATA[26]	AM2
508	RDATA[27]	AM4
511	RDATA[28]	AL1
512	RDATA[29]	AL3
515	RDATA[30]	AL4
516	RDATA[31]	AK1
519	EAD[0]	AK2
520	EAD[1]	AK4
521	EAD[2]	AJ1
522	EAD[3]	AJ2
525	EAD[4]	AJ3

Pad#	Name	Package Ball
526	EAD[5]	AJ4
527	EAD[6]	AH1
528	EAD[7]	AH3
531	EAD[8]	AH4
532	EAD[9]	AG1
533	EAD[10]	AG2
536	EAD[11]	AG3
537	EAD[12]	AG4
538	EAD[13]	AF1
539	EAD[14]	AF2
542	EAD[15]	AF3
543	EAD[16]	AF4
544	EAD[17]	AE1
545	EAD[18]	AE2
546	EAD[19]	AE3
549	EAD[20]	AE4
550	EAD[21]	AD1
551	EAD[22]	AD3
554	EAD[23]	AD4
555	EAD[24]	AC1
558	EAD[25]	AC2
559	EAD[26]	AC3
560	EAD[27]	AC4
561	EAD[28]	AB1
564	EAD[29]	AB2
565	EAD[30]	AB3
566	EAD[31]	AB4
567	WR (R/WR)	AA1
568	RD (DS)	AA2
571	ECLK	AA3
572	ALE (AS)	AA4
573	HOLD (BR)	Y1
574	HLDA (BG*)	Y3

Table 9-3. Signals (7 of 7)

Pad#	Name	Package Ball
575	BGACK	Y4
578	EBE[3]	W1
579	EBE[2]	W2
582	EBE[1]	W3
583	EBE[0]	W4
586	TDI	V1
587	TDO	V2
588	TMS	V3
589	TCK	V4
590	TRST	U1
591	ROOF[23]/CTS[23]/TSTB[23]	U2
592	TDAT[23]	U3
593	TSYNC[23]/TSTUFF[23]	U4
594	TCLK[23]	T1
597	RCLK[23]	T3
598	RSYNC[23]/RSTUFF[23]	T4
599	RDAT[23]	R1
600	ROOF[22]/CTS[22]/TSTB[22]	R2
603	TDAT[22]	R3
604	TSYNC[22]/TSTUFF[22]	R4
607	TCLK[22]	P1
608	RCLK[22]	P2
609	RSYNC[22]/RSTUFF[22]	P3
610	RDAT[22]	P4
611	ROOF[21]/CTS[21]/TSTB[21]	N1
612	TDAT[21]	N2
613	TSYNC[21]/TSTUFF[21]	N3
614	TCLK[21]	N4
615	RCLK[21]	M1
618	RSYNC[21]/RSTUFF[21]	M3
619	RDAT[21]	M4
620	ROOF[20]/CTS[20]/TSTB[20]	L1
623	TDAT[20]	L2
624	TSYNC[20]/TSTUFF[20]	L3
625	TCLK[20]	L4
626	RCLK[20]	K1
627	RSYNC[20]/RSTUFF[20]	K2
628	RDAT[20]	K3
629	ROOF[19]/CTS[19]/TSTB[19]	K4
630	TDAT[19]	J1
631	TSYNC[19]/TSTUFF[19]	J2
632	TCLK[19]	J3
635	RCLK[19]	J4
636	RSYNC[19]/RSTUFF[19]	H1
637	RDAT[19]	H3
638	ROOF[18]/CTS[18]/TSTB[18]	H4
639	TDAT[18]	G1
640	TSYNC[18]/TSTUFF[18]	G2
643	TCLK[18]	G3
644	RCLK[18]	G4
647	RSYNC[18]/RSTUFF[18]	F1
648	RDAT[18]	F2

NOTE(S:
1. 166 Ground (GND)pads
2. 32 Core supply pads (Vddc)-1.8V
3. 32 Output supply pads (Vddo)-3.3V
4. 2 Protection Circuit ESD (Vgg)

Figure 9-1. Pin Diagram

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	
A	GND	GND	GND	GND	9	13	19	23	29	33	38	42	48	52	58	64	67	72	78	82	86	90	95	101	105	111	116	122	125	131	137	142	148	154	157	GND	GND	GND	GND	A
B	GND	GND	GND	GND	10	14	20	24	30	GND	39	43	49	53	59	GND	68	73	79	83	87	GND	96	102	106	112	117	GND	126	132	138	143	149	GND	160	GND	GND	GND	GND	B
C	GND	GND	GND	GND	11	15	21	27	31	36	40	46	50	54	60	65	69	74	80	84	88	93	99	103	107	113	118	123	129	133	139	144	150	155	161	GND	GND	GND	GND	C
D	GND	GND	GND	GND	12	16	22	28	32	37	41	47	51	57	61	66	71	77	81	85	89	94	100	104	110	115	119	124	130	136	141	147	151	156	162	GND	GND	GND	GND	D
E	GND	4	5	6	GND	Vddc	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	168	165	164	163	GND	E
F	647	648	GND	3	Vddo																														Vddc	173	171	170	169	F
G	639	640	643	644	Vddo																														Vddc	181	178	177	176	G
H	636	GND	637	638	GND																														GND	186	185	GND	182	H
J	630	631	632	635	Vddc																														Vddo	190	189	188	187	J
K	626	627	628	629	GND																														GND	198	195	194	193	K
L	620	623	624	625	Vddo																														Vddc	202	201	200	199	L
M	615	GND	618	619	GND																														GND	205	204	GND	203	M
N	611	612	613	614	Vddc																														Vddo	211	210	207	206	N
P	607	608	609	610	GND																														GND	217	216	215	214	P
R	599	600	603	604	Vddo																														Vddc	223	222	221	218	R
T	594	GND	597	598	GND																														GND	227	225	GND	224	T
U	590	591	592	593	Vddc																														Vddo	233	232	229	228	U
V	586	587	588	589	GND																														GND	237	236	235	234	V
W	578	579	582	583	Vddo																														Vddc	243	240	239	238	W
Y	573	GND	574	575	GND																														GND	246	245	GND	244	Y
AA	567	568	571	572	Vddc																														Vddo	250	249	248	247	AA
AB	561	564	565	566	GND																														GND	256	255	252	251	AB
AC	555	558	559	560	Vddo																														Vddc	260	259	258	257	AC
AD	550	GND	551	554	GND																														GND	265	264	GND	263	AD
AE	544	545	546	549	Vddc																														Vdd0	269	268	267	266	AE
AF	538	539	542	543	GND																														GND	273	272	271	270	AF
AG	532	533	536	537	Vddo																														Vddc	280	277	276	274	AG
AH	527	GND	528	531	GND																														GND	285	282	GND	281	AH
AJ	521	522	525	526	Vddc																														Vddo	291	290	287	286	AJ
AK	516	519	GND	520	GND																														GND	297	296	295	294	AK
AL	511	GND	512	515	Vddo																														Vddc	303	302	301	298	AL
AM	GND	507	GND	508	GND																														GND	306	305	GND	304	AM
AN	506	503	502	GND	Vddc																														Vddo	312	309	308	307	AN
AP	499	GND	498	496	Vddc																														Vddo	316	315	314	313	AP
AR	494	GND	493	490	GND	Vddo	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	GND	Vddo	GND	Vddc	Vddc	GND	322	319	318	317	AR
AT	GND	GND	GND	489	GND	482	477	GND	461	455	Vddo	441	436	GND	424	420	416	410	404	399	395	387	383	378	374	368	362	359	354	346	342	337	333	326	GND	GND	GND	GND	GND	AT
AU	GND	GND	GND	GND	486	GND	474	470	GND	450	446	GND	435	431	422	419	413	409	403	398	394	386	382	377	373	367	361	357	351	345	339	336	332	325	323	GND	GND	GND	GND	AU
AV	GND	GND	GND	GND	485	481	GND	465	460	GND	445	440	434	428	GND	418	412	406	GND	397	391	385	GND	376	372	366	GND	356	350	344	GND	335	328	GND	324	GND	GND	GND	GND	AV
AW	GND	GND	GND	GND	GND	478	471	464	456	449	444	439	GND	427	421	417	411	405	400	396	390	384	379	375	369	363	360	355	349	343	338	334	327	GND	GND	GND	GND	GND	GND	AW
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	

101302_025

Appendix A: Counters

The CX28560 provides the system with a complete set of Management Information Base (MIB) counters per channel in both the receive and transmit directions. Each counter is 24 bits wide and saturates on reaching its maximum value.

A.1 One-Second Pin

The one-second pin is an input to the CX28560 that provides the boundaries of each latching period. The system can choose to send a pulse on this pin at any (not necessarily constant) interval. The maximum value a counter can take is 24'hFFFFFF. This is sufficient for a minimum of one second's worth of data on any legally configured channel. This may suffice for longer time periods for low bit-rate channels.

```
Amount of time counters will suffice = (maximum
value of counter) / (number of times the event
occurs per second)
```

Example 1, the octet counter for a T1 channel.

```
Amount of time counter will suffice =
(24'hFFFFFF)/44736000*8 = 3 seconds
```

Example 2, the octet counter for a 52 Mbps channel.

```
Amount of time counter will suffice =
(24'hFFFFFF)/6500000 = 2.58 seconds
```

A.2 Counter Latching

Counters are latched within a negligible delay of a pulse on the onesec pin. The latching of counters implies that the values are held in the background to be read by the system, and updates (during the next latching period) are made to an active set of counters. Note that the system does not need to keep track of which set of counters is the background because the CX28560 controls the internal addressing; externally, both sets of counters are at the same address. The values held in the background counters are overwritten when the next one-second pulse is received. The latching of all counters is simultaneous, and the latched values can be read by a service routine request. On activation and deactivation of a channel, all counters related to that channel are set to zero.

A.3 Counter Descriptions

A.3.1 Receive Counters

In the receive direction, the following counters are provided per channel:

- Octet counter:
  - A count of all octets received for this channel. This count does not include HDLC flags, abort sequences, or idle codes. The count does include FCS bytes and message data of all messages received including errored messages.
- Message counter:
  - A count of all non-errored messages received for a channel. An errored message would either fall into one of the categories below, or have been discarded mid-message due to local conditions (i.e., due to a COFA or OOF condition being detected or an internal FIFO overflow occurring). Errored messages that are not discarded due to local conditions are counted in the counters below.
- Alignment Error counter:
  - A count of all messages that arrive containing an alignment error. An message containing an alignment error is defined as a message that, after removal of HDLC flags and HDLC zero insertions, contains a number of bits not divisible by eight.
- FCS Error counter:
  - A count of all messages that arrive containing an FCS error.
- Abort Condition counter:
  - A count of all messages that arrive ending in an abort condition. In this case, an abort condition is considered to be seven consecutive ones.
- Too Long counter:
  - A count of all messages that arrive that are longer than the maximum length. The maximum length of a received message can be adjusted by selecting one of three 14-bit registers that define a limit for the maximum number of bytes allowed in the message.
- Too Short counter:
  - A count of all messages that are considered too short. A short message is a message with less than the minimum of an 8-bit payload between two flags (e.g., at least 3 message bytes must be received in 16-bit FCS). Too short also includes errors such as Abort, COFA, OOF, Alignment and FCS in the case that no data had yet been transferred to the Buffer Controller from the RSLP.

A.3.1.1 Multiple Errors on A Single Message

Each message is only counted once according to the following priority:

1. Abort condition
2. Too long message
3. Message alignment error
4. FCS error
5. No error occurs

That is to say, a message containing both an FCS error and alignment error is counted only once in the Alignment Error counter.

A.3.2 Transmit Counters

In the transmit direction, the following counters are provided per channel:

- Octet counter:
  - A count of all octets transmitted for this channel. This count does not include HDLC flags, abort sequences, or idle codes. The count does include FCS bytes and message data, including data of messages that were ultimately aborted.
- Message counter:
  - A count of all messages transmitted for a channel.
- Aborted message counter:
  - A count of all messages received from the system terminating in an abort command.

A.4 Reading Counters

The reading of the values of latched counters is performed via service routine requests over the PCI.

A.4.1 Receive Direction

In the receive direction, the channels are arranged in the CX28560's memory in groups of 8 register addresses (7 counters + 1 reserved). To read all counters for channel N, create a service request routine with the fields listed in Table A-1.

Table A-1. Service Request Routine Field for Counter Read (Receive)

Descriptor Field	Size	Description
OPCODE	5	CONFIG_RD
SACKIEN	1	0—SACK interrupt disabled. 1—SACK interrupt enabled. An appropriate interrupt is generated after the command is completed.
LENGTH	14	Number of double words in the memory transaction request. If 0, the number of transfers is 16 K. Therefore it allows for any number of dwords of 1–16384. To read all the receive counters for one channel, this should be set to 8. To read all the counters of all the channels, this should be set to 16384.
Shared Memory Pointer	30 + 2	Shared memory base address for a memory transaction request. The pointer is dword-aligned by concatenating two zeros to the lsb and making it a 32-bit pointer. This address is set according to the system's needs.
CX28560 BASE	22 + 2	The CX28560 base (dword-aligned) address for a memory transaction request. The CX28560 base addresses are specified in bytes but dword-aligned, i.e., with the 2 LSbs as 00. To read channel N's counters, this should be set to (suffixed by 00 for dword alignment): (COUNTER BASE ADDRESS = 22'h008000) + (8 * N) To read all the channels' counters, this should be set to (suffixed by 00 for dword alignment): (COUNTER BASE ADDRESS = 22'h008000)

The counters will be written to the shared memory as listed in Table A-2.

Table A-2. Receive Counters in Shared Memory

31:24	23:0
Reserved	Octet Counter
Reserved	Message Counter
Reserved	Alignment Error Counter
Reserved	FCS Error Counter
Reserved	Abort Condition Counter
Reserved	Too Long Message Counter
Reserved	Too Short Message Counter
Reserved	Reserved

A.4.2 Transmit Direction

In the transmit direction, the channels are arranged in the CX28560 memory in groups of 4 register addresses (3 counters + 1 reserved). In order to read all counters for channel N, a service request routine should be created with the fields listed in Table A-3.

Table A-3. Service Request Routine Field for Counter Read (Transmit)

Descriptor Field	Size	Description
OPCODE	5	CONFIG_RD
SACKIEN	1	0 = SACK interrupt disabled. 1 = SACK interrupt enabled. An appropriate interrupt is generated after the command is completed.
LENGTH	14	Number of double words in the memory transaction request. If 0 the number of transfers is 16K. Therefore it allows for any number of dwords of 1–16384. To read all transmit counters for one channel, this should be set to 4. To read all the counters of all channels, this should be set to 8192.
Shared Memory Pointer	30 + 2	Shared memory base address for a memory transaction request. The pointer is dword-aligned by concatenating two zeros to the lsb and making it a 32-bit pointer. This address is set according to the system's needs.
CX28560 BASE	22 + 2	The CX28560 base (dword-aligned) address for a memory transaction request. The CX28560 base addresses are specified in bytes but dword aligned i.e., with the 2 LSbs as 00. To read channel N's counters, this should be set to (suffixed by 00 for dword alignment): (COUNTER BASE ADDRESS=22'h008000) + (4 * N) To read all the channels' counters, this should be set to (suffixed by 00 for dword alignment): (COUNTER BASE ADDRESS = 22'h008000)

The counters will be written to the shared memory as listed in Table A-4.

Table A-4. Transmit Counters in Shared Memory

31:24	23:0
Reserved	Message Counter
Reserved	Octet Counter
Reserved	Abort Command Counter
Reserved	Reserved

Appendix B: Flexiframe Algorithm

B.1 Overview

The aim of the Flexiframe algorithm is to facilitate the static allocation of internal memory between channels, such that each channel, regardless of its bit rate, will require an equal amount of memory (see Appendix E: Calculation of Buffer Size). In order to do this, channels of a higher bit rate are serviced, in proportion to their bit rate, more often than lower bit rate channels. A full implementation of the Flexiframe algorithm is included in the CX28560 drivers (code can be provided on request).

NOTE: When all channels are of the same bit rate, the Flexiframe algorithm takes its most simple form—a list of the channels.

The following description applies to both the receive and transmit Flexiframes.

The Flexiframe algorithm provides a schedule according to which the CX28560 services channels. The Flexiframe is a list of the channels written to the CX28560 memory that, together with various user-configurable registers, fixes the buffer controller work mode. The Flexiframe is a simple list of channel numbers in slots. Each slot contains one channel number or NOP command (slot channel number = 0), and represents one service by the buffer controller.

In the receive direction, during each slot/service a maximum of one fragment of message data and fragment header will be sent over the POS-PHY to the System. During a service the buffer of the channel whose number was the next in the Flexiframe is examined. If the buffer contained either the end of a message or enough data to form a fragment, data ia sent to the system. The length of the fragment sent is fixed in a Receive Buffer Controller register (See Chapter 5.0).

In the transmit direction, during each slot/service the transmit buffer controller sends a report to the system over the Flow Conductor POS-PHY interface regarding the next channel in the Flexiframe to be served (see Appendix C, Flow Conductor).

The parameters that can be fixed in the CX28560 that control the Flexiframe are as follows:

- Fragment length (Receive only)

The maximum number of 256 bytes per fragment. According to this value, the receive buffer controller decides whether enough data has been collected to send a fragment. Enough data is defined to be either the number of 4-bytes as shown in the reference fragment length register (see Section 5.7.8 *RBUFFC Fragment Size Register*), or the existence of an end of message if one appears before this amount of data is reached.

- Slot time (receive and transmit)

The minimum number of system clock cycles (at 100 MHz) that each buffer controller will spend on a given slot in the Flexiframe. The number of clock cycles actually used will depend on the length of the fragment, the fact that 4 bytes are transmitted per clock, and that after each fragment there is a break of 4 cycles before the next fragment is started.

In the receive direction, if the fragment length register has a larger value than the slot time or is less than 4 lower than the slot time, fragments will be transmitted with a gap of 4 cycles between them. If the fragment length register is lower than the slot time by more than 4, a fragment will be transmitted every slot time.

In the transmit direction, the system transmits fragments over the POS-PHY data bus a maximum of once per slot time; if the fragment takes longer to transfer than the slot time, the slot time is extended. Once per slot time, a channel number is read from the Flexiframe, and an update report is sent to the system over the Flow Conductor POS-PHY bus.

- Flexiframe length (Receive and Transmit)

The maximum length of a Flexiframe is 21,504 entries. The actual length of the Flexiframe produced by the algorithm should be written to the relevant register (see Chapter 5.0).

B.2 New Flexiframe Required

A new Flexiframe is required when one of the following is necessary:

- A new channel is to be activated
- Reset of the chip

B.3 Algorithm

B.3.1 Splitting Channel Bit Rates into Groups

To provide a software efficient algorithm, channels are organized into groups/tables according to their bit rates and standard range definitions. This allows any channel bit rate to be considered as one of a standard number (9) of bit rates. The standard range definitions are based on a binary system, whereby each range limit is half the bit rate of the previous limit.

For example, the fastest channel is of bit rate 52 Mbps, so the limits of the top group are 52 Mbps and 26 Mbps. Any channel bit rate falling between these two limits will be treated as if it is a channel of bit rate 52 Mbps. Any channel falling into the next category (13 Mbps–26 Mbps) will be treated as a 26 Mbps channel, etc. The standard limits are:

```
#define LIMIT0_152
#define LIMIT1_226
#define LIMIT2_313
#define LIMIT3_46.5
#define LIMIT4_53.25
#define LIMIT5_61.625
#define LIMIT6_70.813
#define LIMIT7_80.406
#define LIMIT8_90.203
#define LIMIT9_100.101
```

To calculate the number of clocks between services necessary for the group (as calculated for the highest bit rate for the group), the following is considered.

The slots allocated per channel will be separated by a standard step size for each group. The standard step size is calculated according to the channel's bit rate and the minimum number of accumulated bytes of data that will require servicing (average).

B.3.2 Harmonic Bit Rates

The main inefficiency in the above division of bit rates is that a channel just above one of the limits is considered as a channel with double its actual bit rate. This can be avoided if harmonics are introduced. The harmonic method treats the mid-bit rate between the limits as a new boundary. Those channels that fall between the lower limit and the mid-range limit are assigned to the group below and the group below that (thus treating it as a channel of bit rate of the mid-range value). The channels within the groups are treated the same regardless of whether they were assigned to that group due to being in the upper half of the group's limits, or due to being in the lower half of the group above.

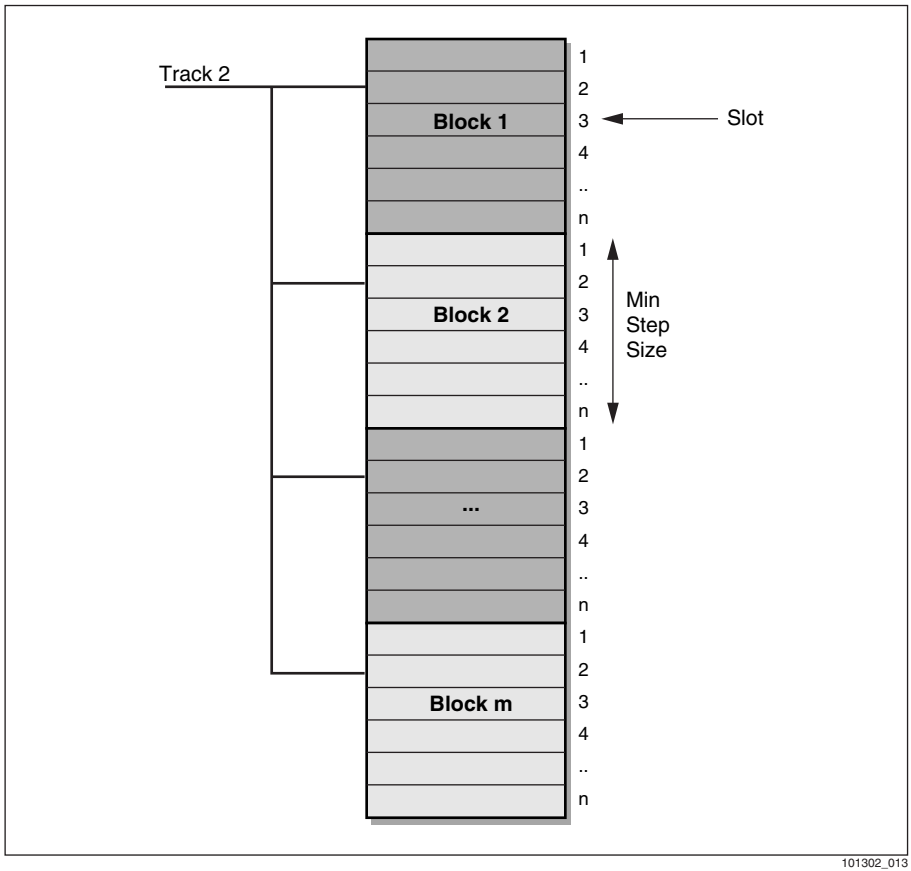
B.3.3 Calculating Step Size Per Group

Assume that packet data is transferred in fragments of length FRAGLEN, and that the receive direction the slot time register (see Chapter 5.0) is set to FRAGLEN + 4 words. The transfer of a packet of size (FRAGLEN + 1 words) will use the POS-PHY and Flexiframe bandwidth usually occupied by 2 complete fragments. Hence in order to withstand the bandwidth wastage caused in the worst case scenario of packets of length FRAGLEN + 1, a channel should be serviced at a frequency that allows the accumulation of 1/2 FRAGLEN worth of data.

If then this interval (STEPSIZE) is calculated for the first group (that which treats each of its channels as if it were a 52 Mbps channel), this will provide the minimum step size. Other step sizes are multiples of the minimum step size (due to binary allocation).

For example, the fragment length is set in the register (see Chapter 5.0) as 14, (each fragment is of length 32 bytes), and the slot time is set to be 20 system clock cycles (200 ns). The gap between each service of a 52 Mbps channel is calculated according to the number of slots it will take the channel to accumulate 28 B of data. This amount of time is 4307 ns, which is the equivalent of 430 clock cycles. Each clock cycle is 200 ns, hence the number of slots between services for a 52 Mbps channel is 21. Due to the binary allocation, it is then simple mathematics that a channel in the next group down requires servicing every 42 slots (2 * 21), etc.

Table B-1. The Flexiframe Structure



The Flexiframe is split into blocks, each block containing MINSTEPSIZE slots. Each block is split into MINSTEPSIZE tracks, where the first slot in each block is allocated to the first track, the second slot in each block to the second track, etc.

B.3.4 Assigning Channels to Slots/Tracks

In the assignment of the channels to the slots, these rules should be followed:

- The channels should be assigned on a group-by-group basis, assigning the channels from the fastest group first and then in decreasing bit-rate order;
- Each track should be filled completely before a new track is started. Note that a channel in the first group will occupy an entire track (slots assigned at an interval of MINSTEPSIZE), a channel in the second group will occupy half a track (slots assigned at an interval of 2 * MINSTEPSIZE), etc. Hence, a track can be fully occupied by 1 channel from group 1, 2 channels from group 2, 2^{n-1} channels from group n; or any suitable combination of the above (for example, 1 channel from group 2, 1 channel from group 3, and 2 channels from group 4).

B.4 Pseudo-Code

The input to the algorithm consists of two lists: chInput and bwInput, which must be of the same length. The output is the list of channel numbers: chOutput. In an interim step, input channels are classified into one of eleven groups.

B.4.1 Assigning Input Channels to Groups

```
For each (ch, bw) in (chInput, bwInput) do
   If (bw in [39 .. 52]) then
      Add ch to group[0]
   Else if (bw in [26 .. 39]) then
      Add ch to group[1]
      Add ch to group[2]
   Else if (bw in [19.5 .. 26]) then
      Add ch to group[1]
   Else if (bw in [13 .. 19.5]) then
      Add ch to group[2]
      Add ch to group[3]
   Else if (bw in [9.75 .. 13]) then
      Add ch to group[2]
   Else if (bw in [6.5 .. 9.75]) then
      Add ch to group[3]
      Add ch to group[4]
   Else if (bw in [4.875 .. 6.5]) then
      Add ch to group[3]
   Else if (bw in [3.25 .. 4.875]) then
```

```
            Add ch to group[4]
            Add ch to group[5]
         Else if (bw in [2.438 .. 3.25]) then
            Add ch to group[4]
         Else if (bw in [1.625 .. 2.438]) then
            Add ch to group[5]
            Add ch to group[6]
         Else if (bw in [1.219 .. 1.625]) then
            Add ch to group[5]
         Else if (bw in [0.813 .. 1.219]) then
            Add ch to group[6]
            Add ch to group[7]
         Else if (bw in [0.609 .. 0.813]) then
            Add ch to group[6]
         Else if (bw in [0.406 .. 0.609]) then
            Add ch to group[7]
            Add ch to group[8]
         Else if (bw in [0.304 .. 0.406]) then
            Add ch to group[7]
         Else if (bw in [0.203 .. 0.304]) then
            Add ch to group[8]
            Add ch to group[9]
         Else if (bw in [0.152 .. 0.203]) then
            Add ch to group[8]
         Else if (bw in [0.101 .. 0.152]) then
            Add ch to group[9]
            Add ch to group[10]
         Else if (bw in [0 .. 0.101]) then
            Add ch to group[9]
         End if
      End for
```

B.4.1.1 Computing the Number of Tracks to be Used

Each track is 1024 slots long, and up to 21 tracks are interleaved in one Flexiframe structure. This step computes the number of tracks needed, according to the channel group sizes that were computed in the previous step.

```
NumTracks = 0.0
For i in [0 .. 10] do
    NumTracks += group[i].size() / (2^i)
End for
NumTracks = Round to next integer (NumTracks)
If NumTracks > 21 then
    Return “error: bad combination of channel
    bandwidths.”
End if
```

B.4.2 Building the Output

Allocate an output list of length (1024 * NumTracks)

Fill the output list with empty slots (channel number 0)

```
                    CurTrack = 0
                    CurTrackUtilization = 0.0
                    CurFirst = 0
                    For i in [0 .. 10] do
                        CurSeparation = NumTracks * (2^i)
                        For each ch in group[i] do
                            If CurTrackUtilization < 1.0 then
                            // Find an empty slot in the current
                            track:
                            While output[CurFirst] <> Empty do
CurFirst += NumTracks
                            End while
                        Else
                            // Get the first slot in the next track:
                            CurTrack++
                            CurFirst = CurTrack
                            CurTrackUtilization = 0.0
                        End if
                            // Update track utilization
                            CurTrackUtilization += 1 / (2^i)
                            // Insert the channel number in the
                            output
                            For j in [0 .. 2^(10-i)-1] do
Output [CurFirst + CurSeparation*j] = ch
                            End for
                        End for
                    End for
```

B.5 Analysis

This analysis is based a number of fixed parameters. If these parameters are to be changed, the analysis should be performed with the new values. The fixed parameters are:

- Fragment length of 32 bytes (plus 4 header bytes)
- Slot time of 20 cycles

To arrive at the minimum frame size required for all configurations, the analysis was performed several times until a minimum was reached. This document only includes the proof that a 21,504 slot Flexiframe will suffice, and not that this is the minimum required.

Table B-2 forms the basis for the analysis. Contained in it are the bit-rate group limits, variable allocation to group ranges, and the number of slots to be allocated to each variable in a 21,504 slot Flexiframe.

Table B-2. Flexiframe Analysis Parameters

Channel MAX Bandwidth [Mbps]	Step Size	Number of Slots in a 21,504 Frame	Variable (Number of channels in the [Range] [Mbps])
52	21	1024	a [52 – 39]
39 (mid range)	42 + 84	512 + 256	b [39 – 26]
26	42	512	c [26 – 19.5]
19.5 (mid range)	84 + 168	256 + 128	d [19.5 – 13]
13	84	256	e [13 – 9.75]
9.75 (mid range)	168 + 336	128 + 64	f [9.75 – 6.5]
6.5	168	128	g [6.5 – 4.875]
4.875 (mid range)	336 + 672	64 + 32	h [4.875 – 3.25]
3.25	336	64	i [3.25 – 2.438]
2.438 (mid range)	672 + 1344	32 + 16	j [2.438 – 1.625]
1.625	672	32	k [1.625 – 1.219]
1.219 (mid range)	1344 + 2688	16 + 8	l [1.219 – 0.813]
0.813	1344	16	m [0.813 – 0.609]
0.609 (mid range)	2688 + 5376	8 + 4	n [0.609 – 0.406]
0.406	2688	8	o [0.406 – 0.304]
0.304 (mid range)	5376 + 10752	4 + 2	p [0.304 – 0.203]
0.203	5376	4	q [0.203 – 0.152]
0.152 (mid range)	10752 + 21504	2 + 1	r [0.152 – 0.101]
0.101	10752	2	s [0.101 – 0.064]
0.064 (min)	—	—	—

B.5.1 Equations for Analysis

To prove that a 21,504 Flexiframe is a sufficient number of slots for any legal configuration of the CX28560, it is necessary to state the rules of a legal configuration and to express them mathematically:

a. The number of channels is equal or less than 2047:
$a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s < 2048$

b. The maximum bandwidth that can enter the CX28560 is 700 Mbps:
$39 * a + 26 * b + 19.5 * c + 13 * d + 9.75 * e + 6.5 * f + 4.875 * g + 3.25 * h + 2.438 * i + 1.625 * j + 1.219 * k + 0.813 * l + 0.609 * m + 0.406 * n + 0.304 * o + 0.203 * p + 0.152 * q + 0.101 * r + 0.064 * s <= 700$

Note that for each range the number of channels in the range is multiplied by the lower limit of the range because this limit is the worst case.

The number of slots in the frame is equal or less then the frame size – 21504:

$1024 * a + 512 * (b + c) + 256 * (b + d + e) + 128 * (d + f + g) + 64 * (f + h + i) + 32 * (h + j + k) + 16 * (j + l + m) + 8 * (l + n + o) + 4 * (n + p + q) + 2 * (p + r + s) + 1 * r <= 21504$

B.5.2 Solution for Equations

Equations (a) and (b) were entered into linear-programming problem solving software together with a command to find the maximum value of equation (c) under the constraints of (a) and (b).

The maximum value of the last equation found was 20,899. Hence, a 21,504 is large enough to encompass the required slot assignment for any configuration of channel bit rates that conforms to the first 2 equations (less than 2028 channels, and aggregate bit rate less than or equal to 700 Mbps).

B.5.3 Building the Flexiframe

The following equations show that if the Flexiframe is built according to the guidance outlined above, building a Flexiframe for any legal configuration of the CX28560 is possible:

It has been shown above that in 21,504 slots the following is true:

1. A = 1024 a + 768 b + 512 c + 384 d + 256 e + 192 f + 128 g + 96 h + 64 i + 48 j + 32 k + 24 l + 16 m + 12 n + 8 o + 6 p + 4 q + 3 r + 2 s <= 21,504

Now considering the first 10,752 slots of the frame. If the frame is to be built, the following must be true:

2. B = 512 a + 384 b + 256 c + 192 d + 128 e + 96 f + 64 g + 48 h + 32 I + 24 j + 16 k + 12 l + 8 m + 6 n + 4 o + 3 p + 2 q + r + s <= 10752

Since (1) has been proven, and all of the following are true:

A = 2 B + r

A <= 21,504

r >= 0

By simple substitution the following holds:

2 B + r <= 21,504

B <= 10,752

Hence equation (2) holds.

The same method of substitution and comparison can be used to show that all the following equations are true:

3. 256 a + 192 b + 128 c + 96 d + 64 e + 48 f + 32 g + 24 h + 16 i + 12 j + 8 k + 6 l + 4 m + 3 n + 2 o + p + q <= 5,376
4. 128 a + 96 b + 64 c + 48 d + 32 e + 24 f + 16 g + 12 h + 8 i + 6 j + 4 k + 3 l + 2 m + n + o <= 2,688
5. 64 a + 48 b + 32 c + 24 d + 16 e + 12 f + 8 g + 6 h + 4 i + 3 j + 2 k + l + m <= 1344
6. 32 a + 24 b + 16 c + 12 d + 8 e + 6 f + 4 g + 3 h + 2 i + j + k <= 672
7. 16 a + 12 b + 8 c + 6 d + 4 e + 3 f + 2 g + h + i <= 336
8. 8 a + 6 b + 4 c + 3 d + 2 e + f + g <= 168
9. 4 a + 3 b + 2 c + d + e <= 84
10. 2 a + b + c <= 42
11. a <= 21

The above set of equations therefore show that if, when building a Flexiframe, the rules outlined are followed, it will always be possible to build a Flexiframe.

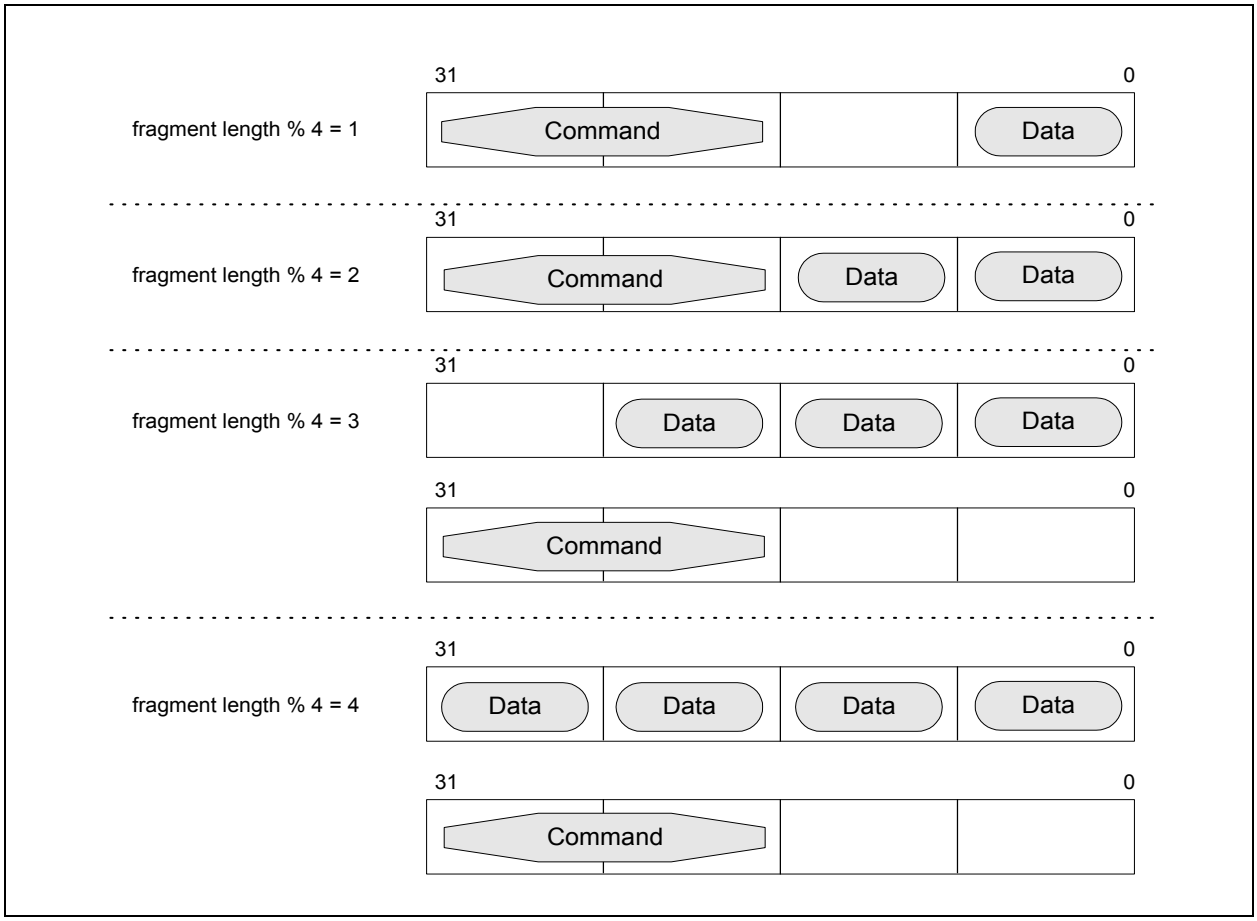
Appendix C: Flow Conductor Interface

C.1 Overview

The Flow Conductor interface provides the system with the information required to control the rate of the flow of data to the transmit buffer controller. This is necessary because the System has no other way to know the amount of data presently in the CX28560's buffers—the line bit rate of the channel does not provide this information because HDLC processing can cause a significant skew from the line rate.

Information is provided to the system in the form of reports of the number of words sent or removed from the buffer since the last report was sent together with the relevant channel number. Thus, the system can maintain a set of counters, one per channel, of the amount of space available in each channel's internal buffer. The counters are initialized to the size of the buffer on channel activation, then each report received from the CX28560 increments the counters, and each fragment sent by the system to the CX28560 causes the relevant counter to be decremented. The only other consideration is the storage of a message's last fragment header (that contains the message command bits). This header is stored in 2 bytes in the channel's buffer according to Figure C-1.

Figure C-1. Data and Command Storage in Internal Buffer



If the length of the payload in a fragment is FRAGLEN, the number of 4 bytes a fragment occupies can be calculated as follows:

If the fragment is not the last fragment in a packet, it will occupy (FRAGLEN / 4) 4 bytes in the internal buffer.

If the fragment is the last fragment in a packet, the number of 4 bytes it will occupy is as follows:

In case FRAGLEN % 4 = 1, num_4bytes = (FRAGLEN + 3) / 4;
In case FRAGLEN % 4 = 2, num_4bytes = (FRAGLEN + 2) / 4;
In case FRAGLEN % 4 = 3, num_4bytes = (FRAGLEN + 5) / 4;
In case FRAGLEN % 4 = 4, num_4bytes = (FRAGLEN + 4) / 4;

C.2 Example

Take, for example, a channel that has just been activated, so the internal buffer is empty. The free space counter for that channel should be set to the size of the buffer allocated in the Buffer Size register (Chapter 5.0, RBUFFC Data FIFO Size Register). FRAGLEN is the number of bytes in the fragment.

When a fragment of length FRAGLEN is sent to the CX28560, not containing an end of message, the free space counter should be decremented by FRAGLEN / 4 (this will be the whole number MAXFRAGLEN/4).

When a fragment of length FRAGLEN is sent to the CX28560, containing an end of message, the free space counter should be decremented by:

Switch (FRAGLEN % 4)

Case 1: (FRAGLEN + 3) / 4
Case 2: (FRAGLEN + 2) / 4
Case 3: (FRAGLEN + 5) / 4
Case 4: (FRAGLEN + 4) / 4

When a report is received, the counter should be incremented by the value received in the report of the number of words sent (WSENT). This value of WSENT takes into consideration the storage of command bytes in the internal buffer, so no further calculation is required.

Appendix D: TSBUS

D.1 Connection Between CX28560 and Other TSBUS Device

This section details the signals required to implement the TSBUS Interface. Figure D-1 illustrates the TSBUS connections between the other device and CX28560. The signals required are summarized in Tables D-1 and D-2. The TSBUS consists of the Payload and the Overhead bus. Each bus has a Transmit and Receive path. The receive path is defined from the other device to CX28560, and the transmit path is defined from CX28560 to the other device.

CX28560 can only generate the TSB_TSYNCI signal during non-stuffed transmit payload time slots. CX28560 must not generate the TSB_TSYNCI signal during stuffed transmit payload time slots. A stuffed transmit payload time slot is defined as the eighth TSBUS payload byte following the assertion of a payload transmit STUFF signal.

Figure D-1. CX28560 Time Slot Interface Pins

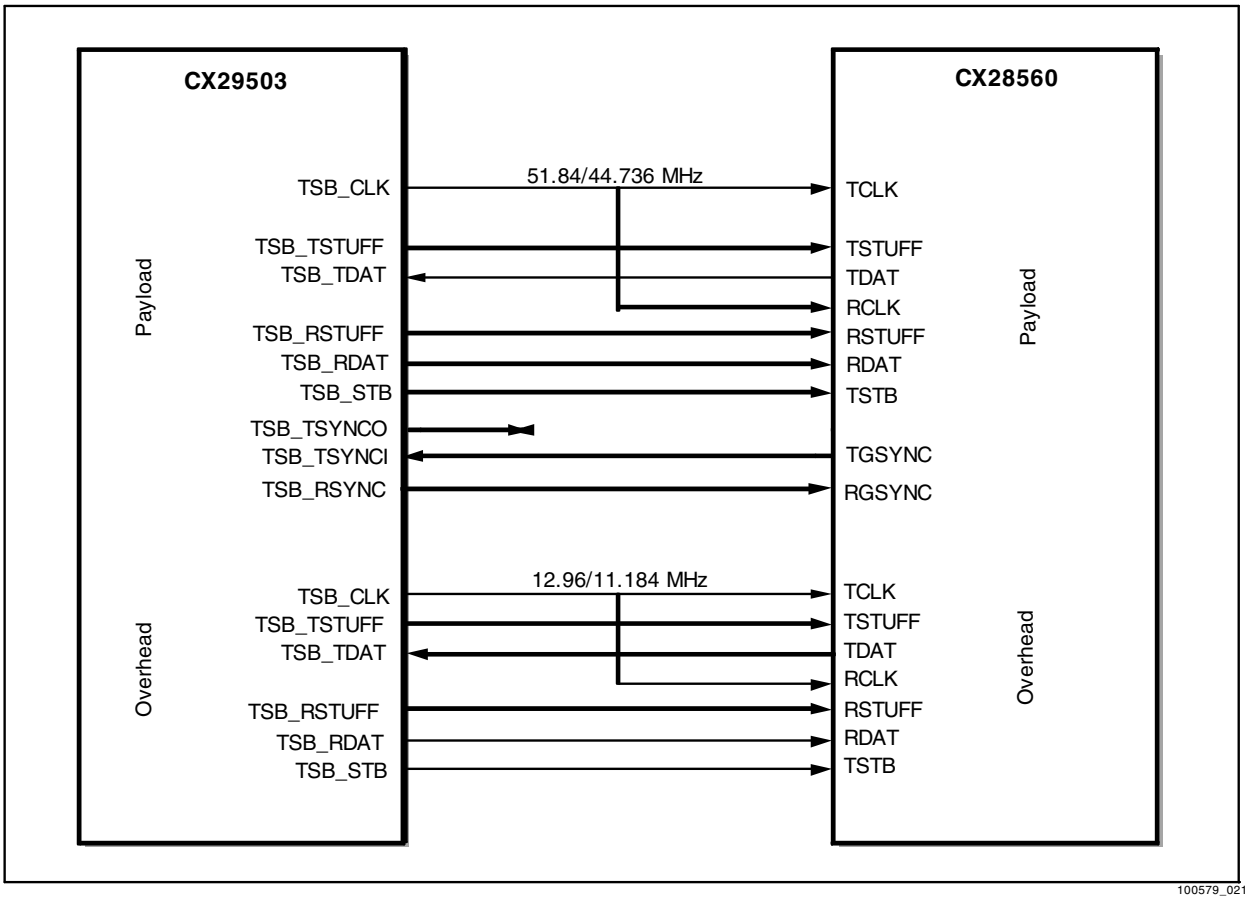


Table D-1. System Side Interface: Payload Time Slot Bus

Symbol	Reset Behavior	I/O	Definition
TSB_CLK	Low	OUT	Payload Time Slot Bus Clock: This clock is based on SIB_TXHSCLK. It is used for all timing on the Payload Time Slot Bus. Clock Rate is 51.84 Mbps (±20 ppm)
TSB_STB	Low	OUT	Payload Time Slot Bus Strobe: A strobe signal that indicates the start of a frame with 84 time slots carrying payload data. The Strobe indicates the beginning of each Payload time slot Frame.
TSB_TDAT	—	IN	Payload Time Slot Bus Transmit Data: This is the serial payload data to be received by TSBUS. This signal is sampled on the rising edge of TSB_CLK.
TSB_TSTUFF	High	OUT	Payload Time Slot Bus Transmit Stuff Indication: When high, indicates a stuff byte must be transmitted in place of the data byte arriving 8 time slots later.
TSB_RDAT	Low	OUT	Payload Time Slot Bus Receive Data: This is the received serial payload data. It is transmitted from TBUS on the rising edge of TSB_CLK.
TSB_RSTUFF	High	OUT	Payload Time Slot Bus Receive Stuff Indication: When high, indicates that data on TSB_RDAT is not valid data. TSB_RDAT is stuffed with all 1s.

Table D-2. System Side Interface: Overhead Time Slot Bus

Symbol	Reset Behavior	I/O	Definition
TSB_OCLK	Low	OUT	Payload Time Slot Bus Clock: This clock is based on SIB_TXHSCLK. It is used for all timing on the Payload Time Slot Bus. Clock Rate is 51.84 Mbps (±20 ppm)
TSB_OSTB	Low	OUT	Payload Time Slot Bus Strobe: A strobe signal that indicates the start of a frame with 84 time slots carrying payload data. The Strobe indicates the beginning of each Payload time slot Frame.
TSB_OTDAT	—	IN	Payload Time Slot Bus Transmit Data: This is the serial payload data to be received by TSBUS. This signal is sampled on the rising edge of TSB_CLK.
TSB_OTSTUFF	High	OUT	Payload Time Slot Bus Transmit Stuff Indication: When high, indicates a stuff byte must be transmitted in place of the data byte arriving 8 time slots later.
TSB_ORDAT	Low	OUT	Payload Time Slot Bus Receive Data: This is the received serial payload data. It is transmitted from TSBUS on the rising edge of TSB_CLK.
TSB_ORSTUFF	High	OUT	Payload Time Slot Bus Receive Stuff Indication: When high, indicates that data on TSB_RDAT is not valid data. TSB_RDAT is stuffed with all 1s.

Figure D-2. Source/Destination of TSBUS Block Line-Side Signals

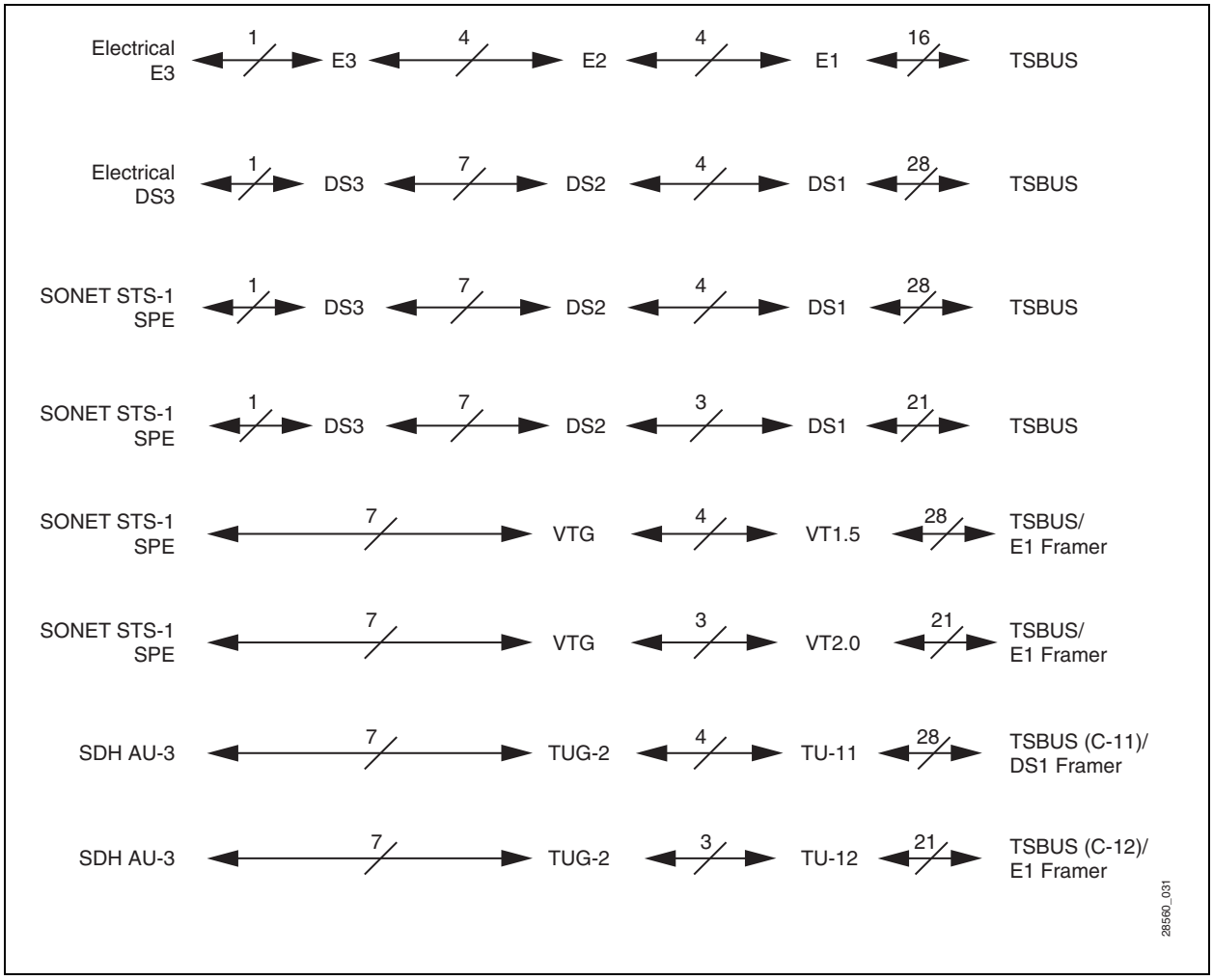


Table D-3. System Side Interface: Overhead Time Slot Bus Frame

TSBUS Source/Destination	Overhead Data Communication Channel Mapped to Virtual Serial Port (VSP)	
	Description	Data Rate
DS1/E1 Framer No. 1-28	F-bit Data Link/Sa4 Bit Data Link	112 Kbps
STS-12/STS-3/STM-1 Mapper	Regenerator Section Data Communication Channel (DCCR) Bytes 1–3	194 Kbps
STS-12/STS-3/STM-1 Mapper	Multiplex Section (Line) Data Communication Channel (DCCM) Bytes 1–9	583 Kbps
SONET/SDH SDS-1/AU-3 Mapper	Path User Channel: F2	64 Kbps
SONET/SDH STS-1/AU-3 Mapper	Path User Channel: F3	64 Kbps
SONET/SDH STS-1/AU-3 Mapper	SPE/AU Path Overhead Nibble N1 (4 LSBs) Path Data Channel/Bit Oriented or LAPD Tandem Connection	32 Kbps
Unused Communication Time Slots	Future Use	3.564 Mbps
Command Status Processor (CSP)	TBUS Register Management	6.48 Mbps

D.1.1 VSP Mapping of Intermixed Digital Level 2 Signals

The following Digital Level 2 signals can transport either DS1 or E1 signals: VTG, TUG-2, and DS2. SONET, SDH, and PDH transport their respective Level 2 signals in sets of seven Level 2 signals. This set of seven Level 2 signals can operate in mixed mode where a portion of the seven Level 2 multiplexed signals transport DS1 signals and the remainder transport E1 signals. Any given Level 2 signal in mixed mode can only transport DS1 signals or E1 signals. It cannot transport both signals.

Table D-4 defines the mapping of DS1 and E1 signals when they are extracted from a mixed set of seven VTGs, a mixed set of seven TUG-2s, or a mixed set of seven DS2s.

Each level 2 signal has a set of 3 or 4 related framers. All framers within a set must be configured for the same type of signal. This prevents framers for different data paths from multiplexing data into the same time slot.

There are four framers in a set for DS1, VT1.5, and VC-11 signals. There are three framers in a set for E1, VT2.0, and VC-12 signals.

The types of Level 2 signals that can be mixed together are limited to the following combinations:

1. DS2 signals containing DS1 signals and the DS2 signals containing E1 signals.
2. VTG signals containing VT1.5, which contain DS1 signals and VTG signals containing VT2.0, which contain E1 signals.
3. TUG-2 signals containing VC-11, which contain DS1 signals and VTG-2 signals containing VC-12, which contain E1 signals.

Table D-4. VSP Mapping of Intermixed Digital Level 2 Signals Containing Either DS1 or E1 Signals

Framer Set No.	Framer No.	Concatenated Time Slot Numbers		VSP No.
		Framer Configured to Extract DS1 Signal	Framer Configured to Extract E1 Signal	
1	1	1, 29, 57	1, 22, 43 64	1
2	2	2, 30, 58	2, 23, 44, 65	2
3	3	3, 31, 59	3, 24, 45, 66	3
4	4	4, 32, 60	4, 25, 46, 67	4
5	5	5, 33, 61	5, 26, 47, 68	5
6	6	6, 34, 62	6, 27, 48, 69	6
7	7	7, 35, 63	7, 28, 49, 70	7
1	8	8, 36, 64	8, 29, 50, 71	8
2	9	9, 37, 65	9, 30, 51, 72	9
3	10	10, 38, 66	10, 31, 52, 73	10
4	11	11, 39, 67	11, 32, 53, 74	11
5	12	12, 40, 68	12, 33, 54, 75	12
6	13	13, 41, 69	13, 34, 55, 76	13
7	14	14, 42, 70	14, 35, 56, 77	14
1	15	15, 43, 71	15, 37, 57, 78	15
2	16	16, 44, 72	16, 37, 58, 79	16
3	17	17, 45, 73	17, 38, 59, 80	17
4	18	18, 46, 74	18, 39, 60, 81	18
5	19	19, 47, 75	19, 40, 61, 82	19
6	20	20, 48, 76	20, 41, 62, 83	20
7	21	21, 49, 77	21, 42, 63, 84	21
1	22	22, 50, 78	NA	22
2	23	23, 51, 79	NA	23
3	24	24, 52, 80	NA	24
4	25	25, 53, 81	NA	25
5	26	26, 54, 82	NA	26
6	27	27, 55, 83	NA	27
7	28	28, 56, 84	NA	28

NOTE(S): Framers with the same Set Number must be configured for the same data signal (i.e., all framers within a set must be configured for DS1 or E1 signals but not both).

D.2 Timing Details

D.2.1 Payload Bus, AC Characteristics

The TSBUS device operates as the master of the transmit TSBUS and the CX28560 (HDLC Controller) device responds as slave. TSBUS generates TSBUS clocks and control signals and the CX28560 device responds by transmitting TSBUS data to or receiving TSBUS data from TSBUS.

TSBUS generates a TSBUS Frame Strobe (TSB_STB) on the rising edge of TSB_CLK as seen in Figure D-2. The Time Slot Bus frame strobe TSB_STB indicates the start of an N time slot Frame carrying payload data.

The Time Slot bus exchanges data over two I/O chip boundaries so care must be taken in ensuring that the data is exchanged on the right phase of the master TSBUS clock TSB_CLK. A possible solution for ensuring correct data exchange is for the Slave (CX28560) to transmit data on the Rising edge of TSB_CLK, and sample the Received data on the falling edge of TSB_CLK.

There is only one Time Slot Frame strobe used (TSB_STB) for transmit and receive direction. There is also only one clock (TSB_CLK) used in the definition of bit boundaries for transmit and receive. This results in the Time Slot Frame alignment of the receive and transmit payload (illustrated in Figure D-2). Each time slot in the Time Slot Bus consists of eight serial data bits. The MSB bit for each time slot is transmitted first.

D.2.2 Transmit Timing

The TSBUS device operates as the master of the Transmit TSBUS, and the CX28560 device responds as slave. The TSBUS generates clock, Frame sync signal, and Stuff signal. CX28560 will generate Transmit data (TSB_TDAT) or generate an all-1s Stuff pattern eight time slots after receiving an active Stuff signal. The TSBUS will generate a Frame sync Strobe (TSB_STB) output synchronously with the rising edge of TSB_CLK. Figure D-3 illustrates the timing requirements for the Transmit. Figure D-3 illustrates the Stuff signal. The target timing values are listed in Table D-4.

Figure D-3. Payload Time Slot Bus Transmit Data (TSB_TDAT)

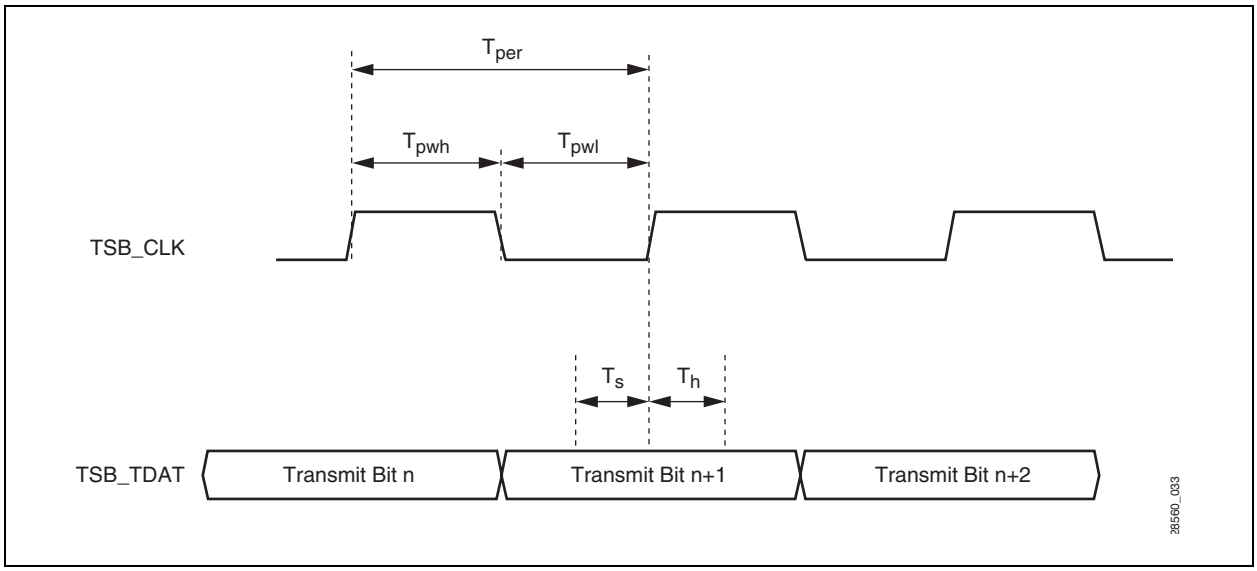
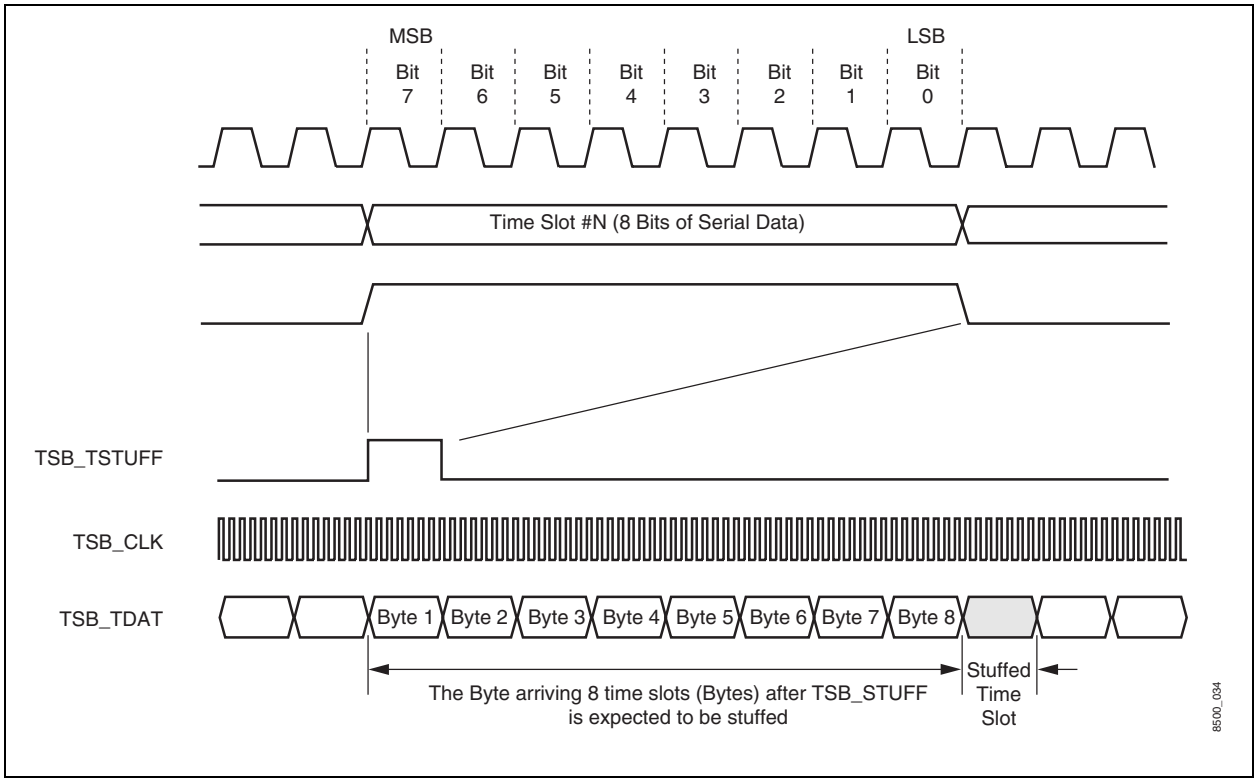


Figure D-4. Payload Time Slot Bus Transmit Stuff Indicator (TSB_TSTUFF)



D.2.3 Receive Timing

The TSBUS device operates as the master of the receive TSBUS and the CX28560 device responds as slave. The TSBUS generates clock, data, Frame sync signal, and the Stuff signal. The TSBUS generates an all ones stuff pattern in place of the payload data during the same time slot that the Stuff signal is active. The TSBUS generates control and data outputs synchronously with the rising edge of TSB_CLK. The nominal clock frequency is 51.84 Mbps. Figure D-5 shows the timing requirements for the receive interface. See Figure D-6 for the Stuff signal.

Figure D-5. Payload Time Slot Bus Receive Data (TSB_RDAT)

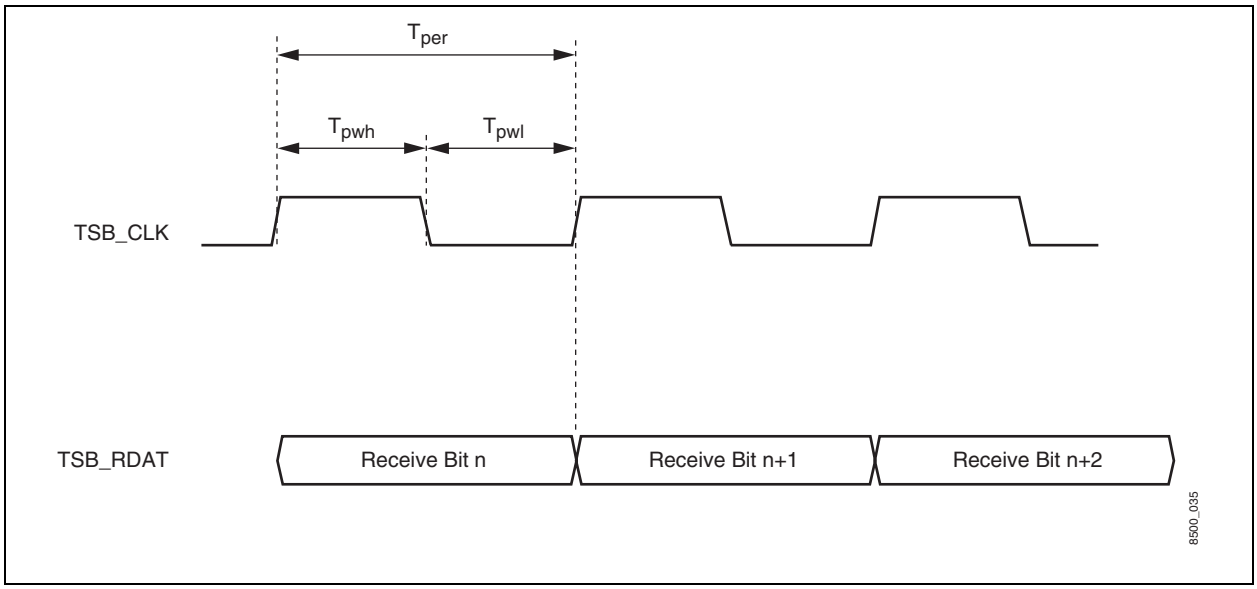
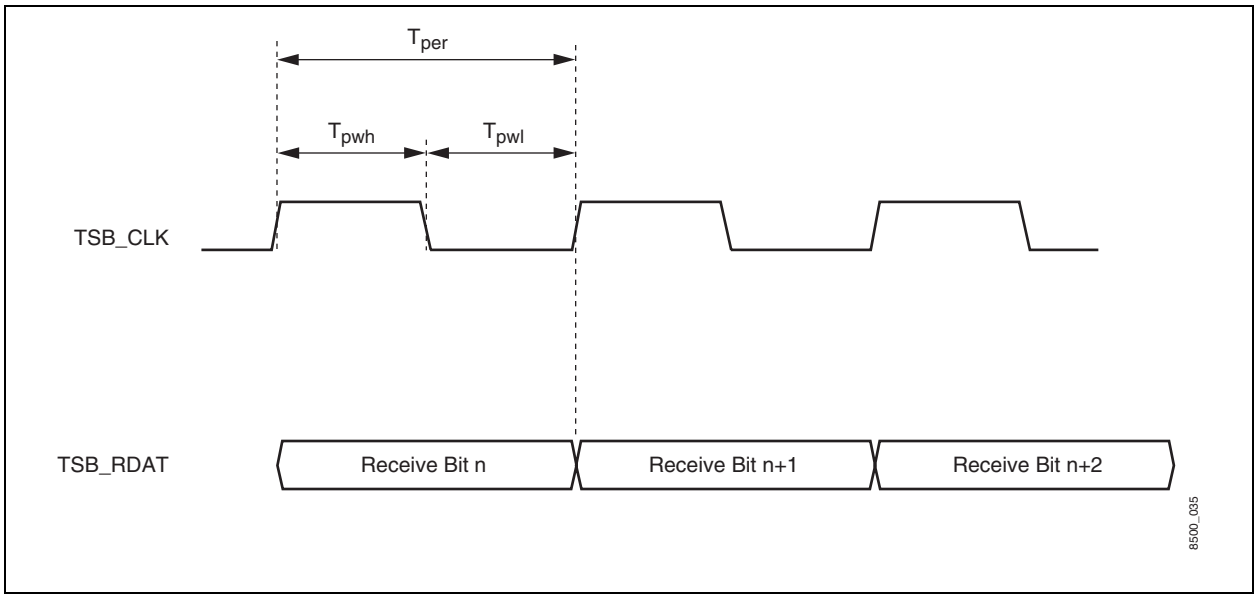


Figure D-6. Payload Time Slot Bus Receive Stuff Indicator (TSB_RSTUFF)



Figures D-7 through D-9 provide timing diagrams for TSB_TSYNCO, TSB_TSYNCI, and TSB_RSYNC. These diagrams show that TSB_TSYNCI, TSB_TSYNCO, and TSB_RSYNC are currently defined as bit-wide signals when asserted. The CX28560 only uses TSB_TSYNCI and TSB_RSYNC, not TSB_TSYNCO.

Figure D-7. TSBUS Interface to CX28560 Transmit SYNC Timing (TSB_TSYNCO)

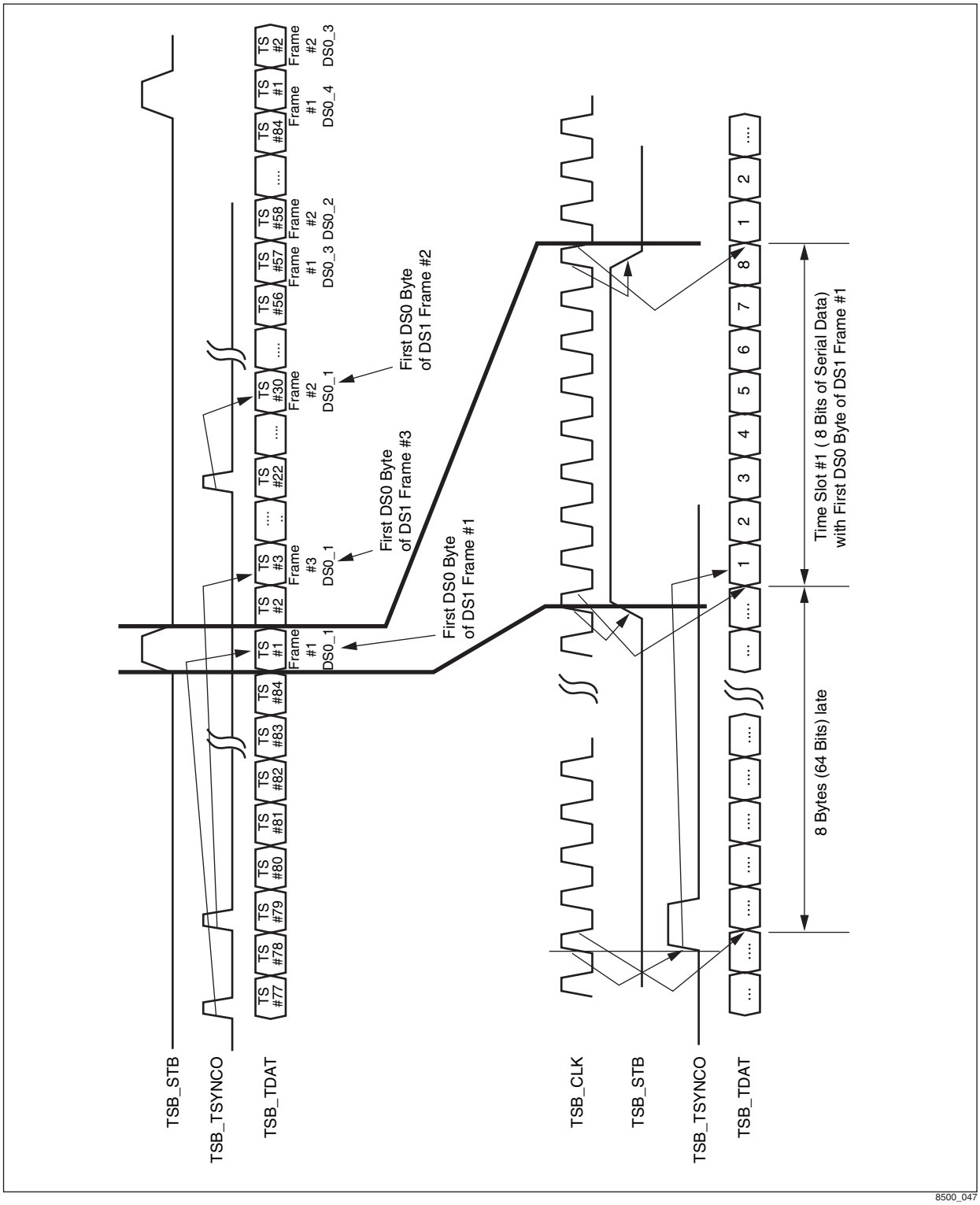


Figure D-8. TSBUS Interface to CX28560 Transmit SYNC Timing (TSB_TSYNCI)

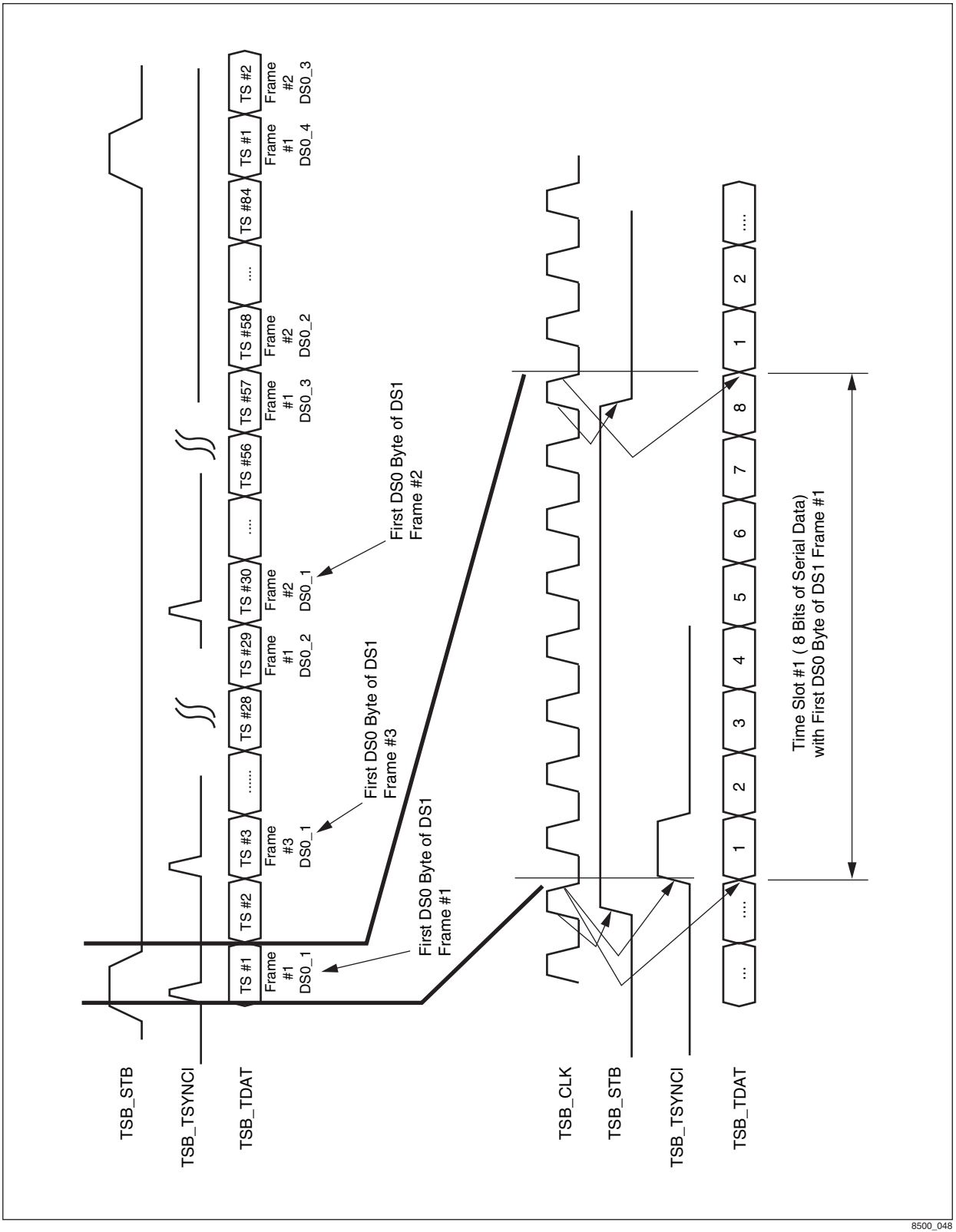
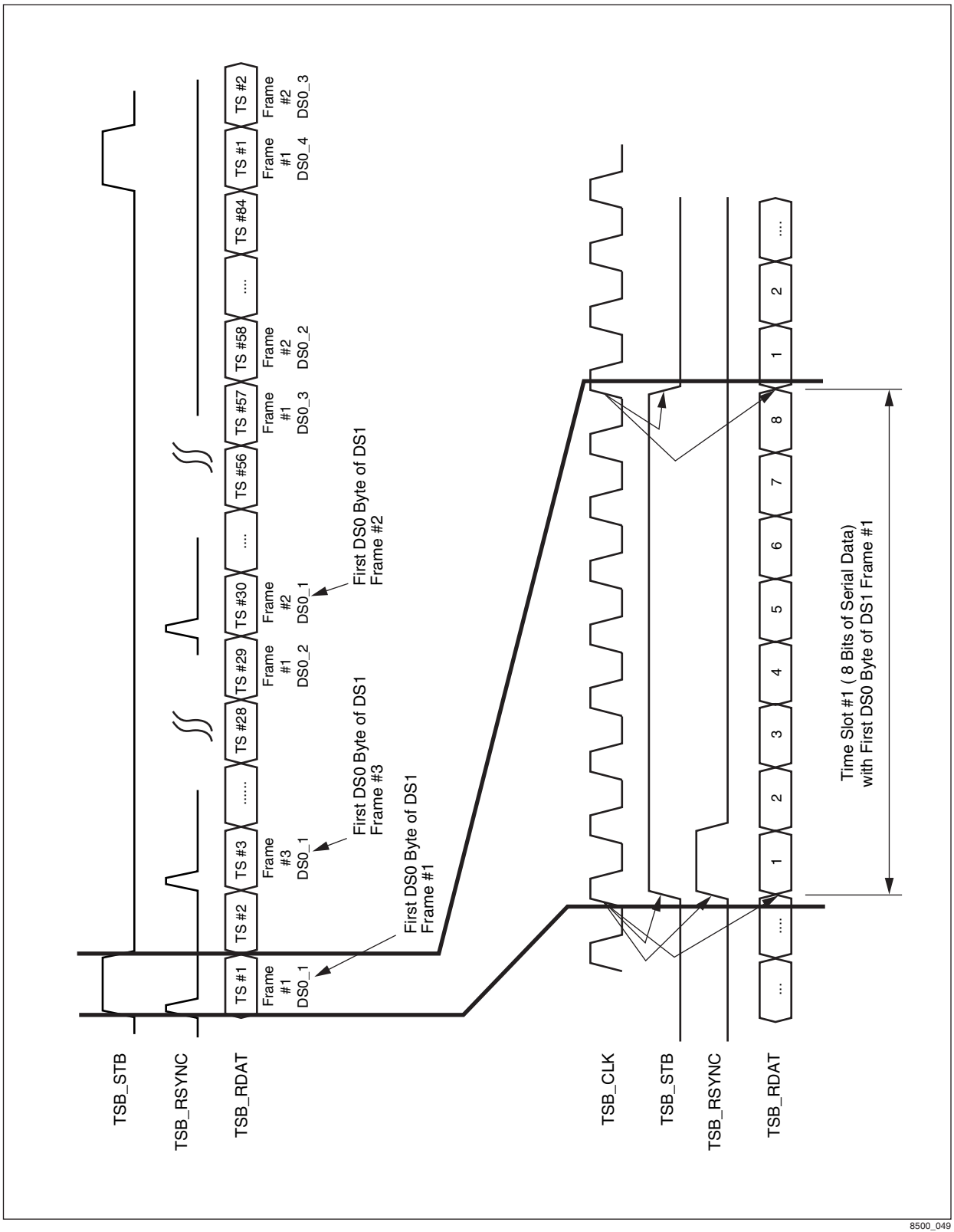


Figure D-9. TSBUS Interface to CX28560 Receive SYNC Timing (TSB_RSYNC)



D.3 Overhead Bus, AC Characteristics

Same operation as the Payload TSBUS, only difference is the TSB_OCLK rate of 12.96 Mbps compared to the Payload rate of 51.84 Mbps.

D.3.1 Transmit Timing

See Section D.2.2.

D.3.2 Receive Timing

See Section D.2.3.

Appendix E: Buffer Controller FIFO Size Calculation

E.1 Introduction

This appendix aims to prove that there exists a maximum buffer size required to contain the information in the buffer controller, and to calculate that maximum. The analysis is based on the Flexiframe algorithm and the CX28560 receive buffer controller design. Fuller explanations can be found in the relevant documentation. The proof applies to 56-byte fragments, and a minimum packet length of 40 bytes. Extrapolation to other values for these parameters is provided via example at the end of the analysis.

E.1.1 Terminology

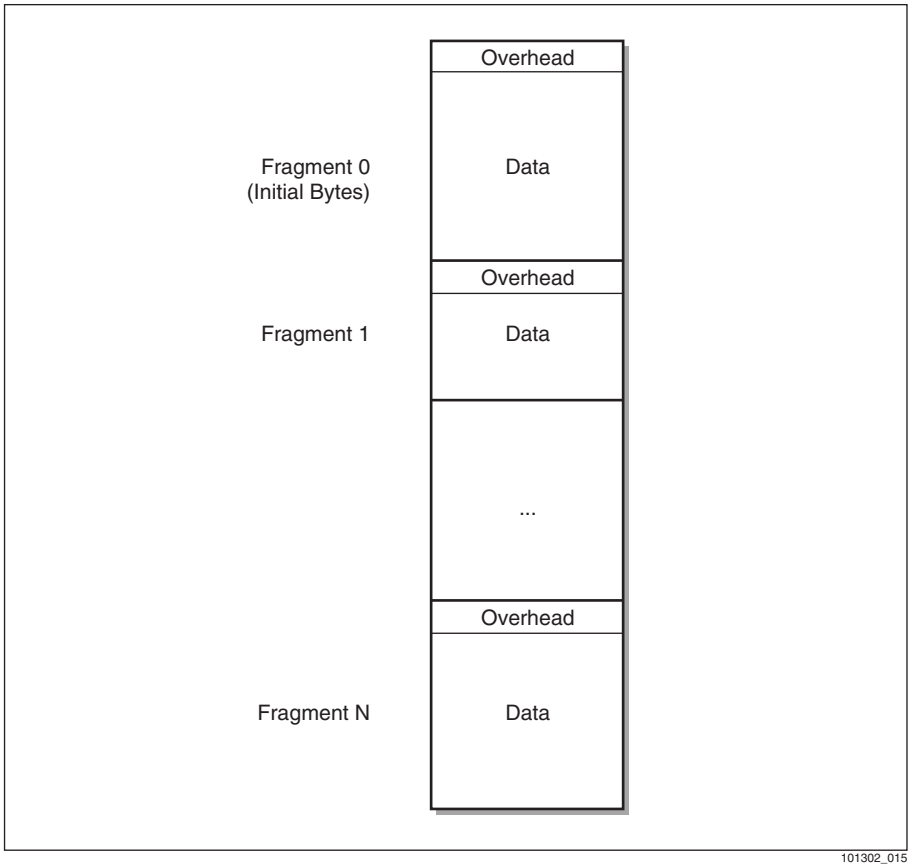
Flexiframe is the algorithm used to implement a channel service scheduler.

RSLP—Receive Serial Line Processor. Block that interfaces with the buffer controller providing a maximum of 32 bits of data and one 8-bit message status per system clock.

An overflow is said to have occurred when new data/status arrives from the RSLP and there is no further space available in the FIFO.

Figure E-1 illustrates the FIFO.

Figure E-1. BUFFC Internal FIFO



Overhead bytes contain space reserved for the fragment header, and the space reserved for last bytes when appropriate.

Y_t is the number of overhead bytes in fragment t.

Data Bytes contain the message data received from the RSLP.

X_t is the number of data bytes in fragment t.

N is the number of fragments in the FIFO ready for transmission to the system.

E.1.2 Assumptions

The minimum message size is 40 bytes. Messages that are shorter than this may cause the buffer to overflow under some conditions (continuous reception of short messages at line rate is sufficient). This assumption also applies to a stream of aborted messages.

E.1.3 Overkill

The analysis presented in the continuation of this document does not take into account any HDLC processing of data when calculating amounts of data received by the CX28560. The HDLC processing includes the following:

- Zero insertions—a maximum of an extra 1/6 of the data received by the RSLP is not passed to the buffer controller—though a minimum of 0— hence not overkill;
- At least one flag is received per message—negligible affect in long messages, but for the short messages used in the analysis they have a larger affect—overkill; therefore, for every message considered to have arrived, 1 input byte will be removed (the equivalent of removing one flag per message).
- CRC bytes—0, 2, or 4 bytes that may or may not be passed to the buffer controller. Since they may be passed to the buffer controller they are not overkill. However, due to the architecture, the RSLP passes to the BUFFC one of the data/status combinations listed in Table , .

Table E-1. Data/Status Combinations

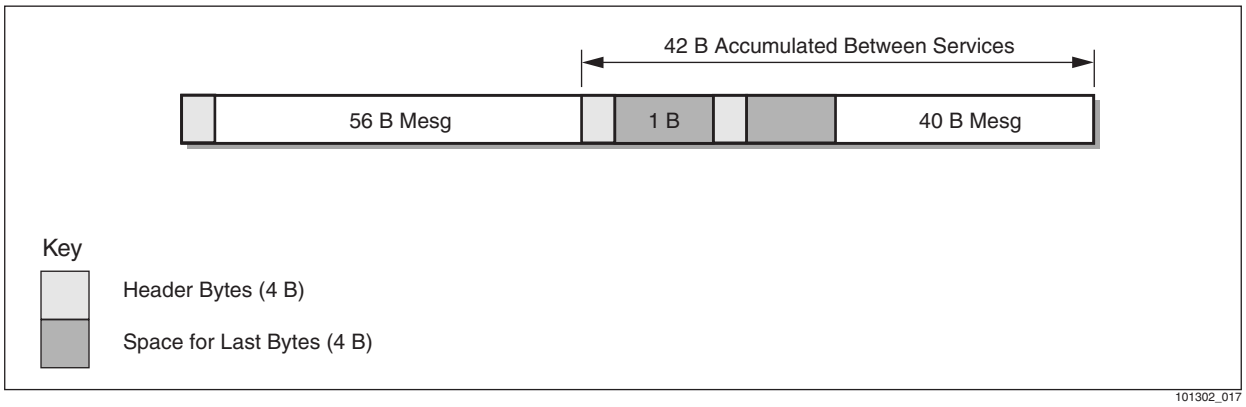
State	Data_Hi	Data_Low			
0	X	Status	X	X	X
1	X	Status	X	X	Data
2	X	Status	X	Data	Data
3	X	Status	Data	Data	Data
4	X	Data	Data	Data	Data
5	Status	Data	Data	Data	Data

The passing of the status later than the data can only be caused by the RSLP removing CRC bytes and detecting a flag. Because this would require the removal of 2 bytes from the possible data that could arrive, this situation is not included in the analysis.

E.2 Expanding Data In

Because the worst case scenario involves mid-range channels (to maximize amount of data between services), examples of data expansion have only been provided for mid-range channels.

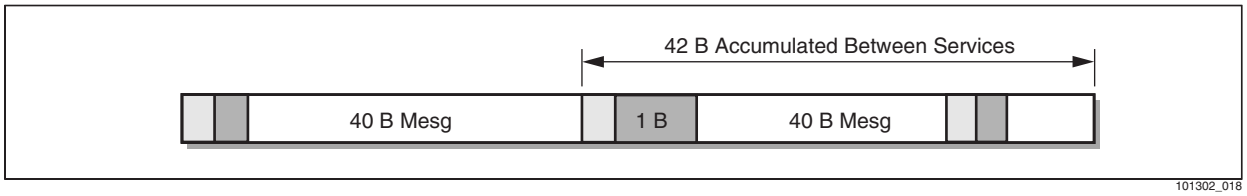
E.2.1 Ending a 57-Byte Message



This accumulation requires 32 bytes in the FIFO. Note the next message header and last bytes are not accumulated because 1 byte that entered the FIFO was a FLAG from the end of the 57-byte message. Hence, 41 data bytes entered (1 byte to finish 57-byte message plus 40 message bytes). This is not enough for the closing flag of the 40+ byte message.

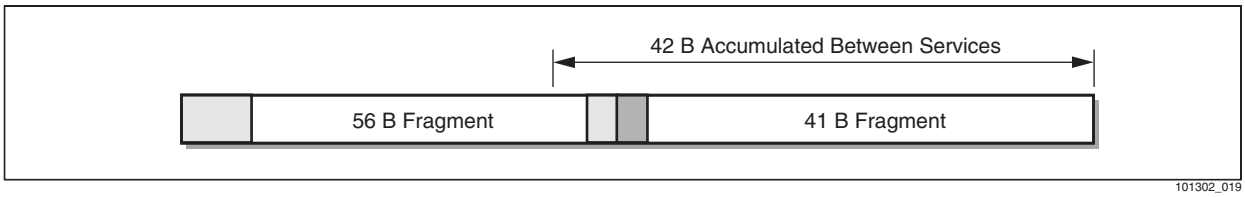
(BUFFC thinks it is still in the middle of storing the message).

E.2.2 Byte Message



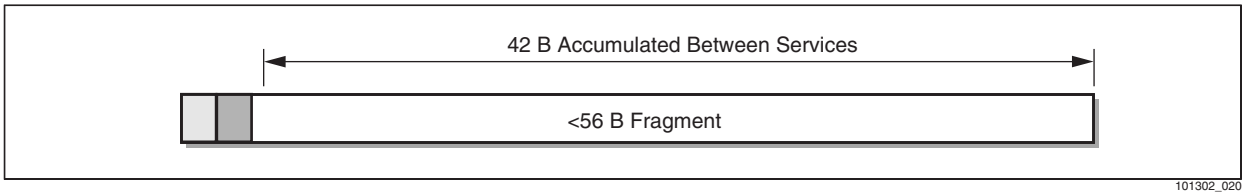
This accumulation requires 32 bytes in the FIFO: 8 bytes for header and last bytes, 40 bytes for message that arrived (plus one FLAG byte). This time, the place for header and last bytes is set aside because there are enough bytes for the closing FLAG of the 40-byte message to have arrived.

E.2.3 Ending a Fragment with No End of Message



This accumulation requires 52 bytes in the FIFO.

E.2.4 Not Ending a Fragment



This accumulation requires 44 bytes in the FIFO.

Hence, maximum number of FIFO bytes that a 42-byte in can require is 56 bytes.

E.3 General Buffer Wastage

Mesg Len	Bytes in FIFO	% Waste
45	52	13.5
41	48	14.6
56	60	6.7
57	68	16.2
61	72	15.3

Most wasteful in terms of bytes => 57-byte messages

Mesg Len	Bytes in Fifo	# Trans	Trans/Mesg Byte	Trans/FIFO Byte
45	52	1	0.022	0.019
41	48	1	0.024	0.021
56	60	1	0.018	0.017
57	68	2	0.035	0.029
61	72	2	0.033	0.028

Most wasteful in terms of transactions => 57-byte messages

Conclusion: If a buffer that has been filled with 57-byte messages can successfully converge to a solution, any other combination will similarly converge.

E.4 Overview of Analysis

E.4.1 Preliminary Calculations

- Calculate maximum amounts of data that can arrive between services.
- Calculate maximum number of services that can be missed due to algorithms used.

E.5 Receive Analysis

- Calculate buffer size required to contain all data that will arrive due to missed services.
- Compare amounts of data received between services and amount removed by service to check that it is not necessary to enlarge the buffer to take extra data received into account.
- Show that the sequence of servicing converges for 57-byte messages a check whether, at any point, the sequence requires a larger buffer than the minimum.
- If, at the end of 6 services, the amount of data in the FIFO and the amount of space used in the FIFO is less than the initial conditions, proof has worked.

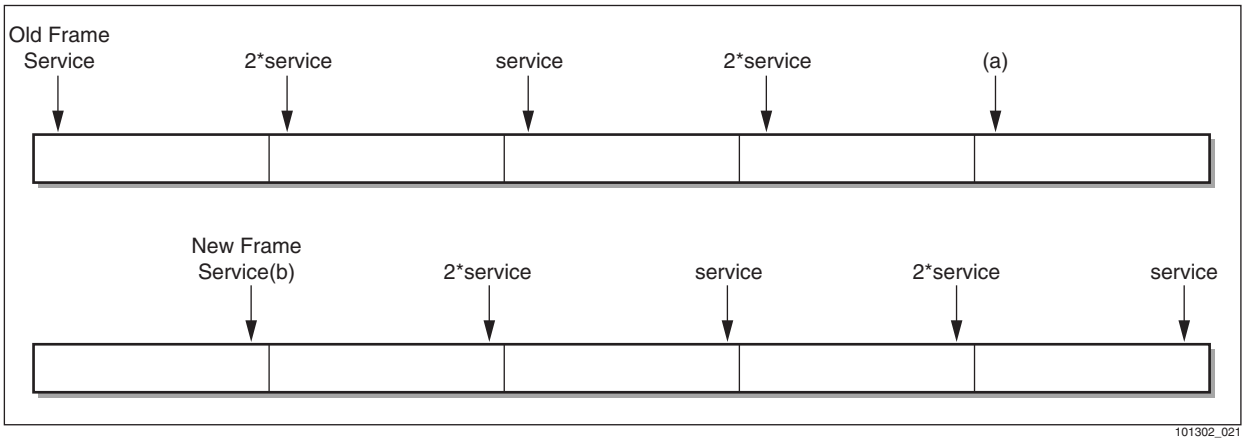
E.5.1 Preliminaries

E.5.1.1 Missed Services

The analysis assumes that the worst possible starting position is that a channel that has been allocated mid-range steps has just missed a service when it accumulates its next fragment for transmission. The scenario continues that the missed service was the last time that the channel was to be serviced in the frame, and the system has written a new frame to the memory.

Figure E-2 illustrates this specific worst case on a frame.

Figure E-2. Worst Case on a Frame



Between points (a) and (b), up to 84 bytes of data may accumulate (42 bytes between each service).

The 42 Bytes comes from the following:

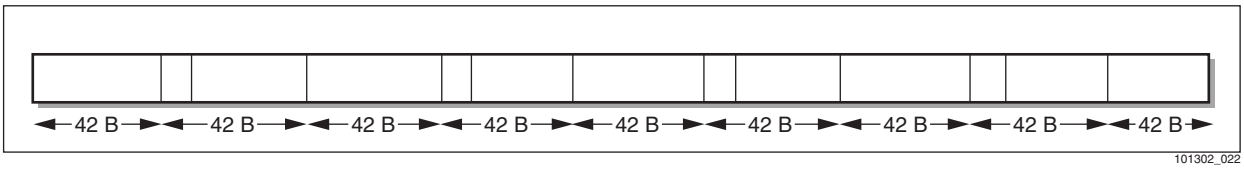
> Due to the mid-range, in the worst case, each channel that uses the mid range receives (x + x / 2) = 3x / 2 services in a frame according to its BW, (while x is the number of services for the lower closest full range). The number of bytes received per frame is 28 B * 3x / 2 = 42 B * x. X services are guaranteed to receive service with constant slots between them; thus, the worst distance between two services is 42 bytes.

At (a) the channel FIFO contains initial bytes amount of data.

The extra bytes accumulated for the frame swapping only needs to be included in the calculation once. This is because the next time the frame is swapped, the maximum amount of time between the last time a service could have taken place and the first service of the frame is one step size. Hence, the amount of data accumulated in the buffer is reduced, and the next time the frame is swapped, the extra accumulated data will be stored in the freed bytes from the previous over-allocation.

Figure E-3 illustrates servicing a mid-range channel.

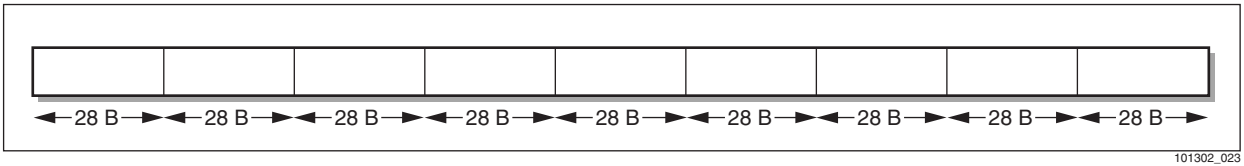
Figure E-3. Servicing a Normal Channel



Note the above is an absolute worst case because the first time the channel is serviced, it is serviced once and then must wait 42 bytes to be serviced twice in succession.

Figure E-4 illustrates servicing a normal channel.

Figure E-4. Worst Case Servicing of a Mid-range Channel



E.5.2 Calculation of Step Size Between Services

The slots per channel are given with gaps, according to the channel BW. The gap size represents the time at which the channel accumulates 28 bytes of data in its specific BW. The 28 bytes that may accumulate is the amount of data that, after expansion, will use the 32 bytes of data in a burst. The expansion is due to the fact that a message length is not optimal to the 56-byte bursts. It may take 2 bursts to send a 56-byte message. The 2 bursts BW is 112 bytes, so the ratio between the original message size to the actual BW allocated for it is 112 / 56 = 2. Example: the accumulation of 28 bytes at a 52 Mbps channel takes 4307 ns (431 clock cycles). A fast channel must be serviced every such period. It means that a fast channel is to be serviced every 431 / 20 = 21.55 slots -> 21 slots**.**

$$MinStepSize = \frac{SystemClock}{TimeperSlot} \times \frac{FragmentSize}{2} \times \frac{8}{bitrate}$$

E.5.2.1 Starting Position

Initial bytes + extra bytes due to missed service + extra bytes due to swapped frame

Initial bytes form one (or less fragment).

Extra bytes are 42 bytes each—a total of 84 extra bytes

Number of fragments available to be transferred (not including initial fragment) = N

Each fragment contains x bytes of data and y bytes of overhead (headers, last bytes) where:

$$0 \le x \le 56$$

$$4 \le y \le 8$$

$$x_1 + \ldots + x_N \le 84$$

$$x + y \le 60$$

x + y = initial bytes ≤ 60

In general, x ≥ 40.

This is only not true when a full fragment is followed by a shorter end of message fragment. In this case:

$$x_1 + x_2 \ge 56$$

There are 84 bytes entering the block. These can be divided as follows:

$$X_2 + \text{all singular xs} \ge x_2 + 3 * x_3 \ge N = 4$$

$$X_2 + \text{all doubles} \ge x_2 + (x_1 + x_2) + x_1 \ge N = 4$$

$$X_2 + \text{single/double mix} \ge x_2 + x_3 + x_1 \ge N = 3$$

(note in worst case $x_2 = 0$ at start of “other”)

Hence, max N = 4.

Sum (x_1: x_4) ≤ 84

Sum (y_1: y_4) ≤ 4 * max y = 32

Hence, initially buffer contains:

Initial bytes + Sum (x_1: x_4) + Sum (y_1: y_4) ≤ 60 + 84 + 32 = 176 bytes

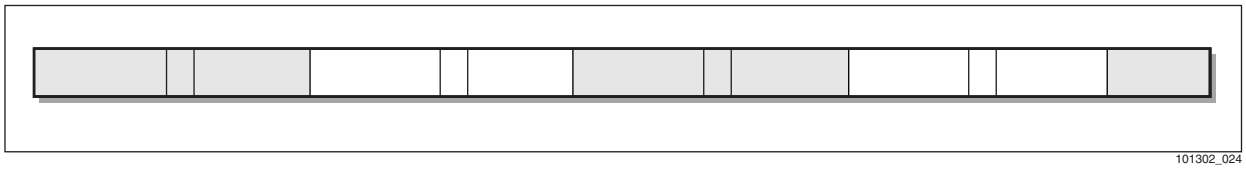
E.5.2.2 Servicing

A series of 56-byte messages may cause the amount of data in the channel's FIFO to reach a steady state of FULL. This is acceptable because nothing can be done to tip the steady state in the direction of overflow, and eventually the stream of 56-byte messages will either change to different size messages, or the channel will be deactivated.

So now assuming that either the messages are shorter (worst case 40-byte messages, or that they are longer—now worst case is 57-byte).

Figure E-5 illustrates worst case servicing of a mid-range channel (services maximum distance apart).

Figure E-5. Worst Case Servicing of a Mid-range Channel



Servicing of a mid-range channel can be seen as repetitions of the shaded grey area above; i.e., every 3 services plus 3 fillings the cycle is repeated.

Hence, if after 3 services and 3 fillings there is less data than at first, it could be on the way to a convergent solution (not least since bandwidth out > bandwidth in).

E.5.2.3 57-Byte Messages

Reach end of first white area—after that is repetitions.

Start position (takes into account flags):

	56, 1, 56, 1, 24		≥ 168, 140
a	32 bytes out	≥ 1, 56, 1, 24	≥ 108, 82
b	42 bytes in	≥ 1, 56, 1, 56, 1, 8	≥ 160, 123
c	1, 32 bytes out	≥ 1, 56, 1, 8	≥ 92, 66
d	42 bytes in	≥ 1, 56, 1, 50	≥ 136, 108
e	1 byte out	≥ 56, 1, 50	≥ 128, 107
f	42 bytes in	≥ 56, 1, 56, 1, 35	≥ 180, 149
g	56, 1 bytes out	≥ 56, 1, 35	≥ 112, 92
h	42 bytes in	≥ 56, 1, 56, 1, 20	≥ 164, 134

Since start with both less data and less FIFO bytes, can assume convergence.

Though minimum buffer required is 180 bytes to take into account state (f).

E.5.2.4 Last Bit (Byte)

Since during the service of a channel (20 cycles), a fast channel can accumulate a byte which in turn could take an extra row in the FIFO, an extra 4 bytes of space must be added to the maximum calculated above.

This gives a buffer space of 184 bytes per channel, or total of 368 KB.

E.5.3 Example

Because the minimum buffer size required per channel is dependent on a number of independent parameters (minimum packet size, existence of mid-range channels, likeness of channels configured), the next paragraph provides an example for the calculation of the minimum buffer size. It is recommended that the characteristics of the system be decided and the buffer size be calculated according to these characteristics.

E.5.3.1 Channels of Same Bit Rate, Large Minimum Packet Size

In this example, the bit-rate of the channel and mid-range considerations are irrelevant, because the Flexiframe will be a simple list of the channels. Hence the above calculation is overkill. For fragments of length FRAGLEN bytes, the time between two consecutive services is guaranteed to be less than the time a channel will take to accumulate 1/2 FRAGLEN bytes of data. Assuming that the minimum packet length (MINPKTLEN) is less than the fragment length, and that initially there is one fragment and 2 missed services, the sequence of servicing in Table E-2 will lead to the equation for the number of channels configurable for a specified FRAGLEN.

Table E-2. Servicing Sequence

Stage	Action	Buffer Content (Services)	Buffer Content (Bytes)
(A)	Start position	FRAGLEN, 1, FRAGLEN – 1	2 * FRAGLEN + 3 * 4
(B)	FRAGLEN out	1, FRAGLEN -1	FRAGLEN + 2 * 4
(C)	½FRAGLEN in	1, FRAGLEN, 1, (1/2 FRAGLEN –1)	(3 / 2)FRAGLEN + 4 * 4
(D)	1 out	FRAGLEN, 1, (1/2 FRAGLEN –2)	(3 / 2)FRAGLEN – 1 + 4 *4
(E)	½FRAGLEN in	FRAGLEN, 1, (FRAGLEN –2)	2 * FRAGLEN – 1 + 3 * 4

Since we start with both less data and less FIFO bytes, we can assume convergence.

Assuming that the minimum fragment size is 32 bytes, the minimum buffer size required is created in stage (A), i.e., that of 2 * FRAGLEN + 12.

NOTE: An additional 4 bytes should be added if, during the configured slot time, a channel can accumulate an extra byte of data.

From this equation and the fact that there is 384 KB of memory available for allocation between channels, the maximum number of channels configurable for a specific fragment size can be calculated:

```
Number of Channels Configurable = (384 * 1024)
/ (2 * FRAGLEN + 12 + 4)
```

NOTE: If the minimum packet size is known to be smaller than the fragment size, this may affect the size of buffer required for each channel and this should be taken into account.

E.6 Transmit FIFO Calculation

E.6.1 Service Request Scheme

Reports are sent to the system containing a count of the number of double words that were delivered to the TSLP since the last request (including CMND dwords). The CX28560 sends reports for a channel according to the Flexiframe algorithm, as long as the channel is activated and overflow did not occur. The system should maintain an empty dword counter for each channel, which will indicate how many empty dwords the CX28560's buffer has for that channel. For each request from the CX28560, the system first updates the empty byte count for this channel and decides if it can deliver another fragment of data or not, according to the next amount of data it has to deliver (see Appendix C: FlowConductor).

If a threshold (configurable per channel) has been passed or if a full message is in the buffer, the CX28560 starts transmitting packets. In case of all packets of 40 bytes or 57 bytes, at the beginning of the transmission the channel does not transmit at full speed until the buffer has been filled. This occurs each time the buffer becomes empty, and starts transmitting packets of 40 bytes or 57 bytes at full speed. The slow down in a channels transmission rate may also occur when the frame is changed at worst case conditions.

In normal operation there also may be a temporary slowdown in the channel's rate when the packet size changes from short to long (due to the time until threshold dwords are filled to start transmitting the new packet).

The transition from long packets to 57-byte packets (without taking into consideration the frame change effect) will cause no slowdown of the channel bit rate.

A slowdown in a channel's rate can also occur when the packets are shorter than 28 bytes (this is the minimum size of fragments that can hold a full rate under the Flexiframe scheme for this buffer size).

Buffer calculations:

- For a 52-Mbps channel:

  ```
  60 B (base) + 28 B (missed request) + 28 B
  (frame change) + 5 µs (latency)* 52 Mbps/8=

      60 B (base) + 28 B (missed request) + 28 B
      (frame change) + 33 B (latency) = 149 B ->
      152 B (to be divided by 4) (181 B with 10 µs
      latency)
  ```

- For a 39-Mbps channel: (mid range):

  ```
  60 B (base) + 42 B (missed request) + 42 B
  (frame change) + 5 µs (latency)*39 Mbps / 8 =

  6 B (base) + 42 B (missed request) + 42 B
  (frame change) + 25 B (latency) = 169 B->172 B
  (to be divided by 4) (193 B with 10 µs latency)
  ```

Explanation:

Base	the minimum size of buffer needed beyond the TRN threshold. It is needed because the system does not send a fragment of data even if it has only 1 byte to send unless there is enough place for a full fragment size (32 bytes for normal fragment + 4-byte CMND).
Missed request	if a request is missed because there was not enough space for a full fragment, until the next request opportunity the TxSLP will transmit more data at the mean time.
Frame change	when the frame is changed, a service opportunity of a certain channel can be moved, so the channel might miss a request opportunity to a frame change. The request opportunity may be maximum moved in one service opportunity distance, so the affect is the same as missed request.
Latency	the latency affect from the fragment request time until it is received. The latency specified here is the maximum time that passes from the request opportunity (slot time) until the whole fragment is received, stored at the DATA FIFO and all the Write memory information is updated internally. The actual time that passes since the request is put out on the PRX_OUT until the fragment is received on the PTX_IN should be less then that

- Other wastes:
  - Indication bits: CMND (command bit) -> 43(dwords per channel) * 2047 / 8 ≥ ~11 KB added.

If we use the same buffer size for all channels, for 2 K-1 channels we will need:

```
5 µs latency: 172-byte (data buffer size) *
2047 + 11 KB(CMND bit) = 354.84 KB (per channel
= 43 dwords of data +43 CMND bits))
```

NOTE: Notes: All slower channels will have the same value for frame change and will consume fewer buffers only due to a smaller latency affect.

Appendix F: Example of Little-Big Endian Byte Ordering

An example of Little-Big-Endian byte ordering is shown in the next table. For the example a 32-bit dword was used—76543210h:

Table F-1. Little Endian

Address	x+3	x+2	x+1	x
Data	76h	54h	32h	10h

Table F-2. Big Endian

Address	x+3	x+2	x+1	x
Data	10h	32h	54h	76h

NOTE: When Little-Big-Endian byte ordering is used, this only refers to the data portion of the interface to the Host, meaning that only data transfers are affected.

Appendix G: Example of an Arbitration for Fast and Non-Fast Back-to-Back Transactions

Figure G-1 illustrates, in a specific configuration, CX28560's PCI transactions while operating as a master, and fast back-to-back feature enabled. CX28560 performs as a master while operating at 32-bit address-data, a burst write of 2 dwords, which are transferred during the first cycle and a burst write of 3 dwords, which are transferred during the second cycle. Both transaction cycles require 4 PCLK cycles.

Figure G-1. PCI Burst Write: Two 32-bit Fast Back-to-Back Transactions to Same Target

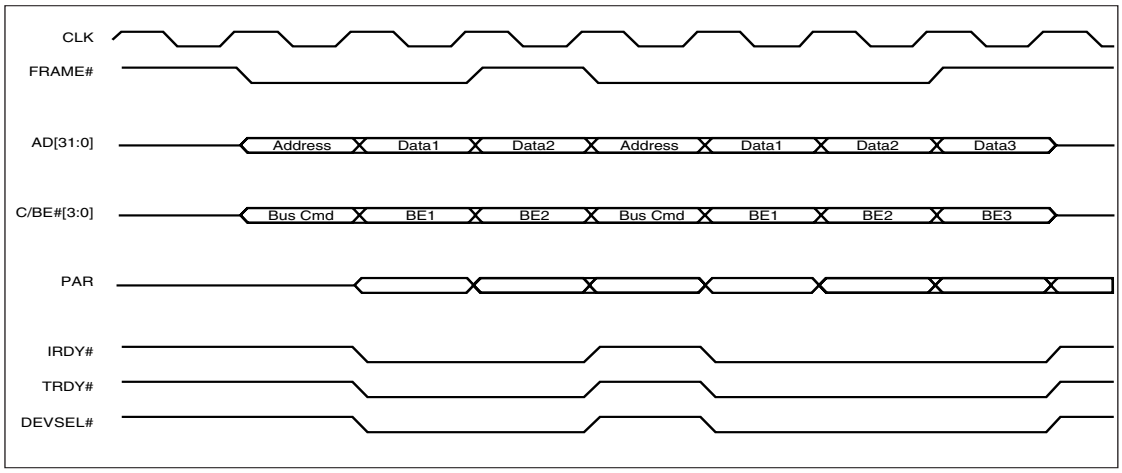
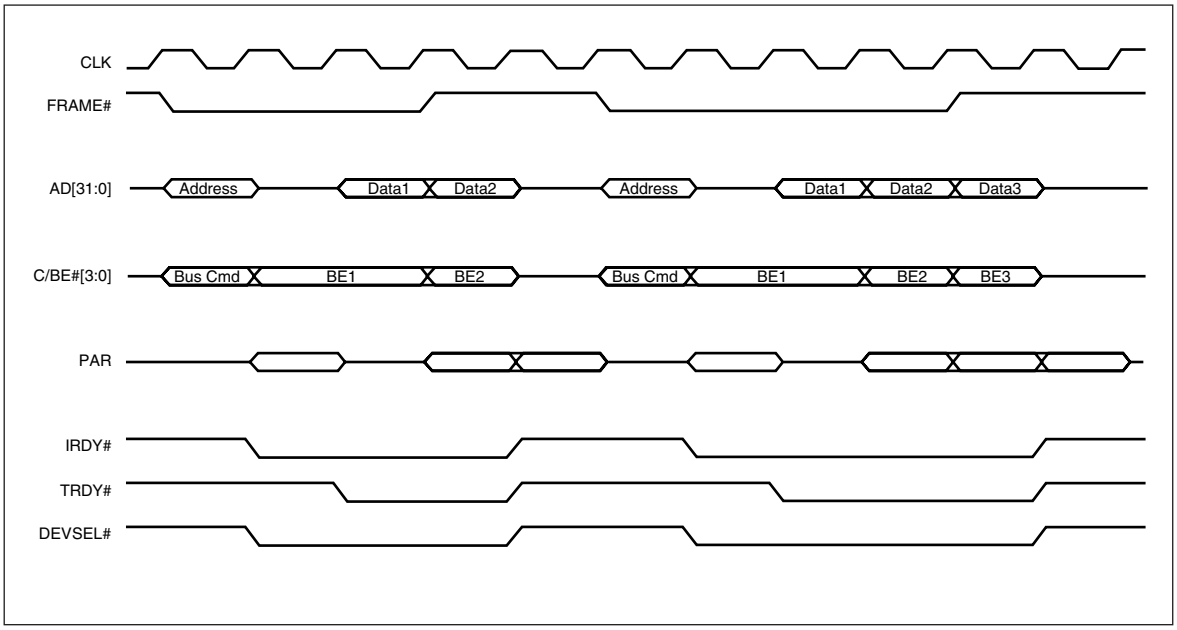


Figure G-2 illustrates how CX28560 operates at 32-bit address-data and performs a burst read of 2 dwords transfer during the first cycle and 3 dwords transfer during the second cycle. The fast back-to-back feature is disabled. It can be observed that the first cycle takes 5 PCLK cycles (with one PCLK post-data phase) and the second cycle of transferring 3 dwords requires 6 PCLK cycles.

Figure G-2. PCI Burst: Two 32-bit Transactions



Appendix H: PCI Utilization

H.1 Overview

This appendix will provide the system with an approximation of the utilization of one CX28560 on the 32 bit, 33 MHz PCI bus. The calculations can be extrapolated, or tailored to the actual needs of the system.

H.2 Analysis

It is assumed that the interrupts being written by the CX28560 to the shared memory has a negligible affect on the PCI utilization. In addition, the configuration of internal registers is also considered a negligible factor on the overall PCI Utilization. The major contributors to the CX28560 PCI activity are the writing of Flexiframes to the CX28560 (at 21 K entries each) and the reading of all the channels' counters once per second.

H.2.1 Internal Considerations

The time to perform a write or read of 32 bits is the total of:

- Internal wastage:
  - This includes time for processing commands, internal bus time, and reaction time by the blocks. This is not a maximum figure, because no absolute maximum exists, only a statistical maximum.
  - 70 clocks @ 100 MHz = 700 ns
- PCI bus time:
  - The calculation below allows each entry in the Flexiframe to be written, and each counter to be read in separate descriptors. In addition, it allows for >20 cycle latency.
  - 30 clocks @ 33 MHz = 900 ns

Hence the total time taken to perform one host service routine is 1600 ns (per 32-bit register).

H.2.2 Conditions

Assuming that, in the worst case, in one second the system will:

- Read 10 counters for 2047 channels (takes the time of reading 12) => 12 * 2047 * 1600
- Change the Flexiframe 4 times (2 times transmit, 2 times receive) => 4 * 21 * 1024 * 1600

The total time to perform these commands is 0.176 seconds. Hence the total utilization of one CX28560 is 0.176.

Appendix I: Maximum Number of Channels Calculation

The maximum number of channels configurable is dependant on a number of parameters. In the analysis included in this specification, it is proven that for fragments of 56 bytes and a slot time of 20 cycles, for any dynamic combination of 2047 channels, a Flexiframe can be built, and that sufficient buffering can be supplied.

The simplest form of the calculation of the maximum number of channels configurable from the fragment length is by simple extrapolation from the above example, as follows:

- A maximum of 2047 channels can be configured with a 56-byte fragment
- A maximum of 1024 channels can be configured with a 112-byte fragment
- A maximum of 512 channels can be configured with a 224-byte fragment

This analysis (and therefore the example above) provides for maximum flexibility in the allocation of channels bandwidths allowing the user to configure a channel with any bit rate.

However by introducing the channels' bandwidths into the equation used to calculate the maximum number of channels, the length of the fragment can sometimes be increased. It can be seen in Appendix E that the buffer calculation assumes half-rate allocations. For a known set of frequencies in the system a more efficient Flexiframe structure can be created. This structure causes less oversubscribing and a service will be provided when needed. It will not use the half-rate method, as a result the buffer size needed for each channel will be smaller. It will be 68 + 64 + 24 = 156 bytes. In this case, the CX28560 can support 2 K channels with 64-byte fragments. Longer fragments will force the max number of channels to be lower.

MINDSPEED™

www.mindspeed.com

Tel. (949) 579-3000
Headquarters Newport Beach
4000 MacArthur Blvd., East Tower
Newport Beach, CA. 92660